

ESTTA Tracking number: **ESTTA416230**

Filing date: **06/24/2011**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE TRADEMARK TRIAL AND APPEAL BOARD

Proceeding	91193335
Party	Defendant RStudio, Inc.
Correspondence Address	CHARLES E. WEINSTEIN FOLEY HOAG LLP 155 SEAPORT BLVD, STE 1600 BOSTON, MA 02210-2600 UNITED STATES ARufo@foleyhoag.com, JHuston@foleyhoag.com, USTRademark@foleyhoag.com, cweinstein@foleyhoag.com, jjarvis@foleyhoag.com
Submission	Testimony For Defendant
Filer's Name	Anthony E. Rufo
Filer's e-mail	arufo@foleyhoag.com
Signature	/Anthony E. Rufo/
Date	06/24/2011
Attachments	002 Exhibits 8.pdf (151 pages)(7688559 bytes)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE TRADEMARK TRIAL AND APPEAL BOARD

EMBARCADERO TECHNOLOGIES, INC.,

Opposer,

v.

RSTUDIO, INC.

Applicant.

Opposition No. 91193335

Applications S.N.

77/691980

77/691984

77/697987

APPLICANT'S NOTICE OF FILING OF TRIAL TESTIMONY

Please take notice that pursuant to Trademark Rule 2.125(c), Applicant RStudio, Inc. is hereby filing electronically via the ESTTA system the deposition transcript of the trial testimony of Applicant's witness Joseph Allaire, taken on April 15, 2011 together with all associated exhibits.

Respectfully submitted,

RSTUDIO, INC.,

/Anthony E. Rufo/

Julia Huston

Charles E. Weinstein

Joshua S. Jarvis

Anthony E. Rufo

Foley Hoag LLP

155 Seaport Boulevard

Boston, MA 02210

Tel. 617/832-1000

jhuston@foleyhoag.com

cweinstein@foleyhoag.com

jjarvis@foleyhoag.com

arufo@foleyhoag.com

Attorneys for Applicant

Dated: June 24, 2011

CERTIFICATE OF SERVICE

I hereby certify that I have this day served a true copy of the above-identified Notice of Filing of Trial Testimony upon Opposer's attorneys of record:

Martin R. Greenstein
Mariela P. Vidolova
TechMark A Law Corporation
4820 Harwood Road, 2nd Floor
San Jose, CA 95124-5273

via First-Class Mail and e-mail to MRG@TechMark.com and MPV@TechMark.com.

/Anthony E. Rufo/
Anthony E. Rufo

Examples of ER Used as Abbreviation for ENTITY RELATIONSHIP

1	ARIS - (Architecture of Integrated Information Systems)	They follow the existing ER model. "The information captured by the ARIS tool set is stored in a database following the ERM (Entity Relationship Model)"	http://www.pera.net/Methodologies/ARIS/ARIS.html
2	Classroom material	Notes: "There are three basic elements in ER models." Common use of ER model: "In 1976, Chen developed the Entity-Relationship (ER) model, a high-level data model..."	http://www.umsl.edu/~saunders/visu/visu/er.htm
3	Classroom material	Dr. Angela B. Shifflet of Winford College	http://www.offord-ics.org/DataAndVisualization/ermodel/material.htm
4	ConceptDraw PRO ER Diagram tool	Product title uses ER in place of entity relationship	http://www.conceptdraw.com/en/products/erdiap_er_diagram.php
5	DB-VA Designer's Guide	Title: "Designing Data Model by Entity Relationship Diagram" and refers in presentation as ERD	http://media.visual-paradigm.com/mediadocuments/dbva406g/pdf/dbva_designer_guide_chapter4.pdf
6	Definition for Entity Relationship Diagram	"Definition: An entity-relationship (ER) diagram is a specialized graphic..."	http://databases.about.com/cs/specificproducts/p/er.htm
7	Definition for Entity-relationship model	"definition - The entity-relationship model (or ER model) is a way of graphically representing..."	http://searchserver.technical.com/definition/entity-relationship-model
8	Definition for Entity-relationship model	Definition for Entity-relationship model: "... process are called entity-relationship diagrams, or ER diagrams for short."	http://www.workingpro.com/termwku/entity-relationship_model
9	Definition of entity relationship model	"A database model that describes the attributes of entities and the relationships among them. An entity is a file (table). Today, ER models are often compiled."	http://www.pcmag.com/encyclopedia_term/0,2542,1=entity-relationship+model&1=4266200_0_0
10	ER Diagram convention	Ziff Davis, Inc. Scribd	http://www.scribd.com/doc/5053989/ER-Diagram-convention
11	FAQ Page	DuChan, Inc.	http://www.duchan.com/FAQ/Designer%20FAQ%20page.htm
12	Instructional Guide	Professor Richard Holowczak at Zedlin School of Business	http://cisnet.bowdoin.edu/holowczak/classes/cis394/04/01/entityrelationship/
13	List of ER diagramming tools	Microsoft Published Article	http://www.download.com/0p1q1n1g1n1y+relationship+diagram.html
14	Microsoft Published Article	Topic: "When All You Want is an ER Diagram"	http://msdn.microsoft.com/en-us/library/aa224823%28SQL_80%29.aspx
15	MS Visio Classroom guide	Topic: "Creating ER Diagrams with MS Visio"	http://www.ahli.org/assess/0415/ahb_nouss/ahb_03.html
16	Online Archives	Online archives of an "introduction to data modeling using the Entity Relationship (ER) approach"	http://www.utexas.edu/archives/ark:/61902/3/databases/databases/er/index.html
17	Personal Website	Developed "Entity-Relationship Model (ER Model)"	http://bit.csc.lsu.edu/~chen/er.htm
18	Product description for Aqua Data Studio	Product description defines ER: "Aqua Data Studio offers an Entity Relationship (ER) Modeler for all major RDBMSs."	http://www.aquafold.com/er-modeler.html
19	Product Description for TRMS Data Modeler	Provided in product overview: "Entity relationship diagram editor (ER diagram)"	http://www.tmssoftware.com/erdiagram.asp
20	Product description for Toad Data Modeler 3 - database design tool	Defines ER: "Using the graphically very well arranged Entity Relationship Diagrams (ER Diagrams) you can..."	http://www.parasoft.com/en/products.aspx
21	Product description for SQL Maestro for MySQL	Uses ER in product description to mean Entity Relationship: "Database Designer allows you to create physical ER Diagram..."	http://www.sqlmaestro.com/products/mysqldbmaestro/duabaso_designer/
22	Product Documentation	Specifically defines ERD: "Diagrams created by this process are called Entity-Relationship Diagrams, ER Diagrams, or ERDs."	http://www.sparsystems.com/enterprise_architect_user_guide/modeling_guides/entity_relationship_diagrams_e.html
23	Product overview for Design for Databases	Defines ERD: "Entity relationship diagram (ERD) is used..."	http://www.visual-paradigm.com/products/visual-paradigm-er-diagramming.jsp
24	Published Article: "Entity Relationship Modeling with UML"	"Design for databases to create ER diagrams"	http://www.onitlog.com/tools/ermodel/design.htm
25	Published Article: "Entity Relationship Modeling"	Defines ER: "The goal of entity relationship (ER) modeling is to define a..."	http://www.information-management.com/info/mec/2003/12/36266-1.html
26	Question and Answer Forum	Defines ER in first sentence of article: "Entity Relationship Modeling (ER modeling) is by far..."	http://www.dovanticles.com/development-cycles/Entity-Relationship-Modeling/
27	Research Paper: "English Sentence Structure and Entity-Relationship Diagram"	Diagram helps in the following ways..."	http://en.alleexperts.com/g/Oracle-145/Entity-Relationship-Diagrammer.htm
28	Structure and Entity-Relationship Diagram	Defines ER in first sentence of paper: "The entity-relationship (ER) diagrammatic technique is a graphic way of displaying..."	http://bit.csc.lsu.edu/~chen/p2/english.pdf

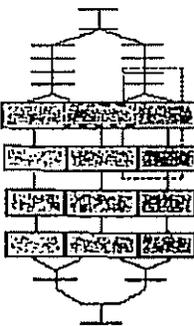


Examples of ER Used as Abbreviation for ENTITY RELATIONSHIP

29	Search for "ER" in Wikipedia leads to "Entity-relationship model" under technology section	Wikipedia	Has section heading labeled "ER diagramming tools"	http://en.wikipedia.org/wiki/Entity-relationship_model ER diagramming tools
30	Database Design - Entity Relationship Modeling Examples	Developer Shed	"... how to design a database and understand an Entity Relationship (ER) diagram."	http://www.codewalkers.com/codewalkers/Database-Code/Relationships-Entities-and-Databases-Design/
31	Book: Entity-Relationship Approach - ER 93: 12th International Conference on the Entity-Relationship Approach	Springer	Title: "Entity-Relationship Approach - ER 93: 12th International Conference on the Entity-Relationship Approach"	http://www.amazon.com/dp/0540582177
32	Book: Database Design Using Entity-Relationship Diagrams	Silke Bagul and Richard Eard	Summary: "Entity-relationship (E-R) diagrams are time-tested models..."	http://www.crcpress.com/product/isbn/97800849315480
33	Tutorials	SmartDraw	ER Defined: "Entity-Relationship Diagrams (ERDs) illustrate the logical structure of databases."	http://www.smartdraw.com/business/relationships/entity-relationship-diagram/
34	Classroom material: NCGIA Core Curriculum in Geographic Information Science	Thomas H. Meyer	Section 3: "The Entity-Relationship (ER) model is generally attributed to Chen (1976)."	http://www.noglia.ucsb.edu/giscc/units/045/045.html
35	Definition for Entity Relationship	TechDictionary	Search for "Entity Relationship" returns: "Entity-Relationship modeling: A discipline for examining and representing the components and interrelationships in a database system. Also known as ER modeling, this discipline factors a database system into entities, attributes, and relationships."	http://www.techdictionary.com/search.action?qs=0
36	Definition: Entity Relation Model	Toolbox.com	"We use an Entity Relation Model (ERM) to create a data model of a system..."	http://it.toolbox.com/wiki/index.php/Entity_Relationship_Diagram
37	Definition: Entity Relationship Diagram	Toolbox.com	"An Entity Relationship Diagram (ERD) is a snapshot of data structures."	http://it.toolbox.com/wiki/index.php/Entity_Relationship_Diagram
38	Definition: ER	computeruser.com	Search for "ER": first result returns: "(Entity Relationship Model) ER model is a conceptual data model that views..."	http://www.computeruser.com/dictionary/
39	Dictionary	dictionary.com	Compiling Dictionary Definition: "ER-Relationship: Entity-Relationship: (noun) The Entity Relationship Diagram (ERD)"	http://dictionary.reference.com/browse/ER
40	ERD Example	Visual Paradigm	Image description: "The Entity Relationship Diagram (ERD) illustrates the logical structure of the databases"	http://www.visual-paradigm.com/vp/Gallery/databasemodeling/EntityRelationshipDiagram.html
41	Journal	appears.com	Discussing ERD: "That is where an entity-relationship diagram (ERD) comes in."	http://www.appears.com/assets/Journal-Entity-Reverse-Entity-Relationship-Diagram%66308
42	Online notes	ComputingStudents.com	Product description: "Draw entity relationship diagrams (ER diagrams) easily with Edraw!"	http://www.edrawsoft.com/etehh-end.php
43	Product description: Chen EDR Software	EdrawSoft	Title: "Entities and Entity-Relationship (ER) Modeling"	http://www.edrawsoft.com/etehh-end.php
44	Product description: Design for Databases	Datamatic	Description: "The software uses entity relationship diagrams (ERDs) to graphically design databases..."	http://www.datamatic.com/dzq/index.html
45	Question and Answer Forum	TechTarget.com	Title: "Data modeling: Entity relationship (E-R) vs. dimensional data models"	http://search.techtarget.com/search/Data-modeling-Dimensional-vs-E-R
46	Web and XML Glossary	enr.net	Entity: "ER (Entity-Relationship Model)"	http://enr.net/glossary/
47	Published Document: Entity Relationship Modeling with UML	IBM Corporation	First paragraph: "One of the most misunderstood terms in the software industry is actually one we know very well: entity relationship (ER). That's because we often lack a common definition that is understood by all members of the development team. We assume that everyone on the team shares the same clear understanding of the methodology, syntax, and mechanics associated with ER and ER modeling."	http://www.ibm.com/devexp/networks/uml/library/content/03July250027852785_uml.pdf
48	Research Paper: "Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned"	Peter Chen	Defines ER in first sentence of paper: "Entity-Relationship (ER) modeling is..."	http://bit.csc.lsu.edu/~chen/pd/chen_Pioneers.pdf
49	Research Paper: "A Comparative Analysis of Entity-Relationship Diagrams"	Evers, Park, and Song	First sentence: "The purpose of this article is to collect widely used entity-relationship diagram (ERD) notations..."	http://www.ischool.utexas.edu/faculty/songp/publications/er_jece-er-uml.pdf
50	Tool description	Acquafold, Inc.	Tool description: "Tools - Entity Relationship The (ER) Diagram Generator"	http://www.acquafold.com/er-er-diagram.html

ARIS – (Architecture of Integrated Information Systems)

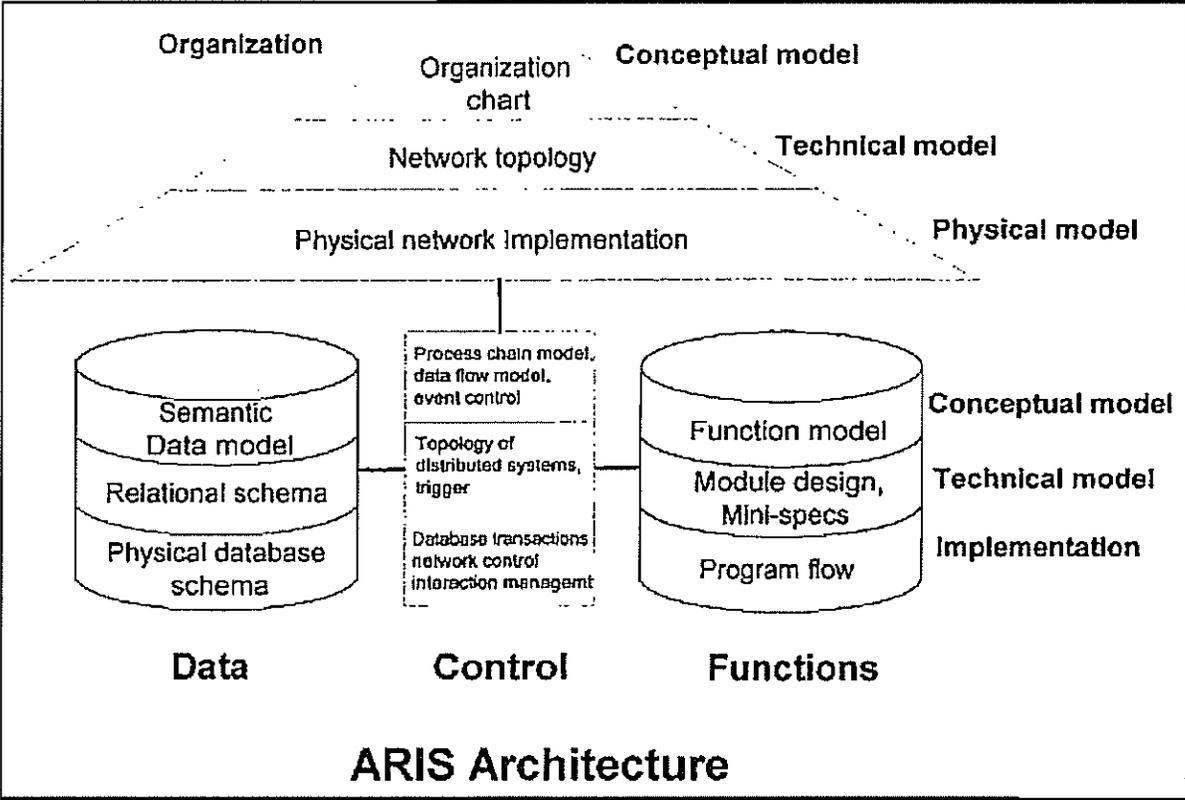
Category	Type 1 (Generic) Methodology and Architecture
Summary	<p>The ARIS-architecture distinguishes between Organization, Function, Information and Control views. It uses a graphic modelling system supported by software which models data movement and tasks.</p> <p>ARIS focuses on the analysis and requirements definition phase during the design of managerial information systems, not on the execution of business processes.</p>
Comments	<p>ARIS provides a generic and well-documented methodological framework. It is closely associated with SAP to which it can directly export models for incorporation in SAP systems.</p> <p>Software is complex to use and has relatively long learning curve. Current version (2000) does not handle multiple languages well.</p>
Relationship to PERA	<p>The following describes how this Methodology fits within the PERA Enterprise Framework (GERAM). The yellow background on the PERA diagram indicates where this methodology is most commonly applied. Click on the PERA diagram for more explanation.</p>

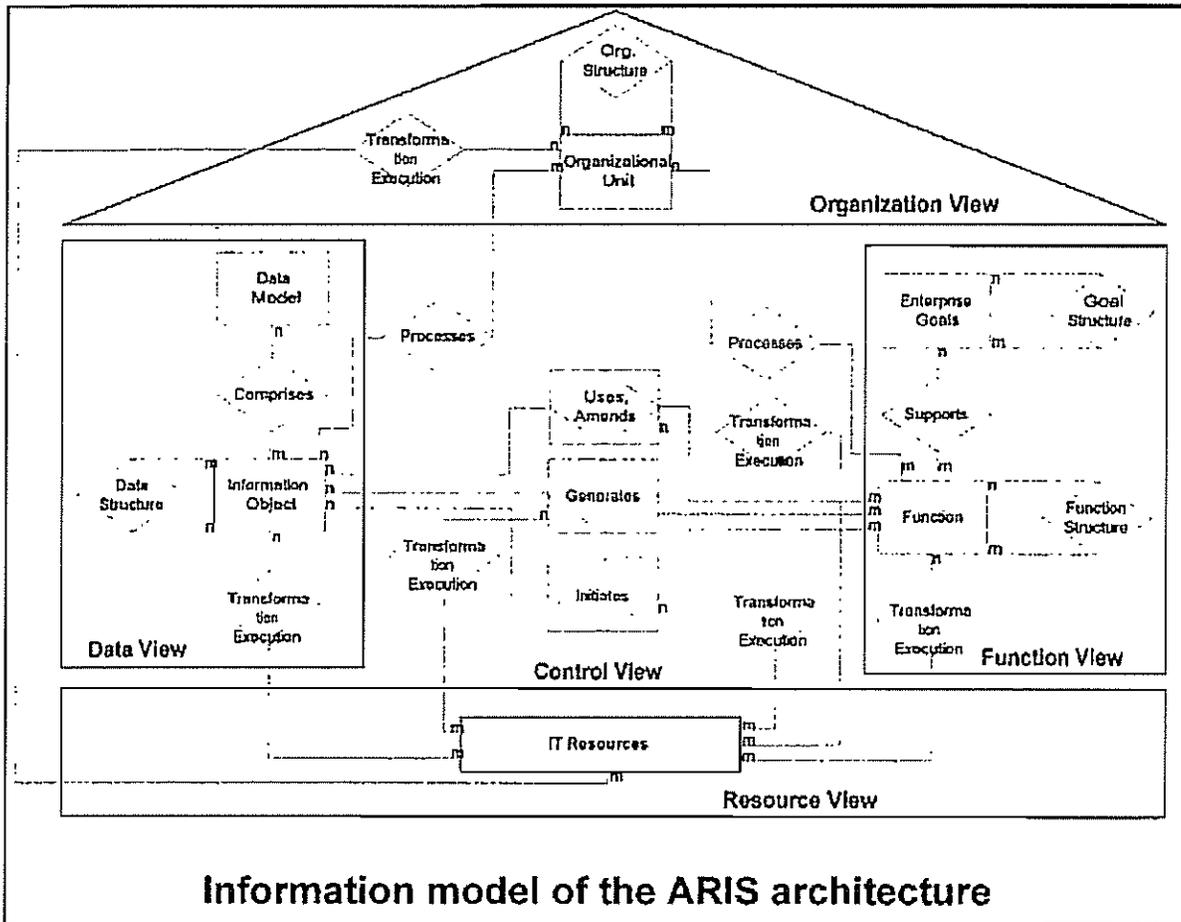
Industry	Widely applicable to large-scale industrial and business enterprises of all types including discrete and process manufacturing, service industries, government and military.	<p style="text-align: center;">PERA Framework</p>  <p style="text-align: center;">click for explanation</p>
Enterprise Phases:	<p>Preliminary Engineering: Used to perform workflow analysis, resulting in functional or procurement specifications. Modeling predicts "bottlenecks" and time delays.</p> <p>Detail Engineering: May be used for direct transfer of functional definitions to SAP. May also be used to maintain</p> <p>Construction Phase: May be used during Implementation to guide acceptance testing.</p> <p>Operations Phase: May be used to maintain workflows and model the effect of proposed modifications.</p>	
	Facilities: N/A	

RS681

Enterprise Elements:	<p>People: Relates to Human and Organizational tasks to the extent that these are automated. May be used to detail interfaces with automated information systems.</p> <p>Control & Info Systems: Used in Information Systems but not plant automation.</p>	<p>Technical Discussion</p> <p>In ARIS, business processes are described by</p>
-----------------------------	--	--

process chain diagrams. The modeling is done using a tool-set instead of a language. Several sub-tools are available, each displayed in its own window. The information captured by the ARIS tool-set is stored in a database following the ERM (Entity-Relationship-Model). In Scheer (1994) it is argued that a formal language imposes restrictions on the day-to-day usability by potential end users.





Information model of the ARIS architecture

The ARIS Model and its application in relationship to PERA, CIMOSA and other Type 2 Models is discussed in the paper "Workflow Management within the ARIS Framework"

Entity-Relationship Diagrams (ERD)

2

Data models are tools used in analysis to describe the data requirements and assumptions in the system from a top-down perspective. They also set the stage for the design of databases later on in the SDLC.

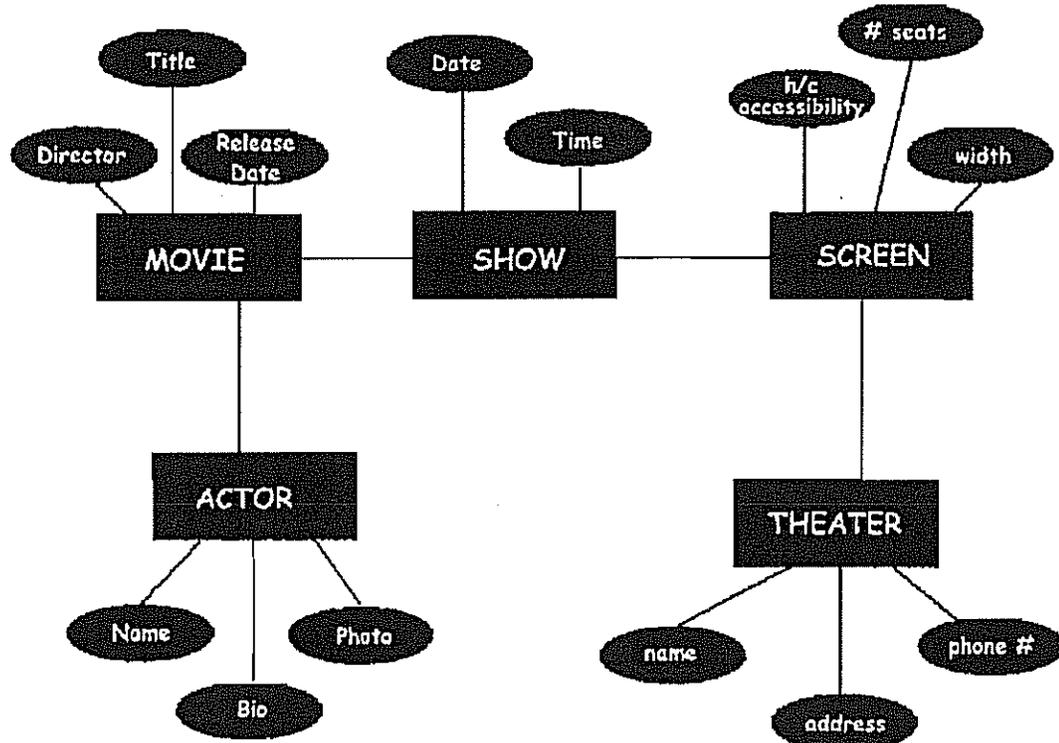
There are three basic elements in ER models:

Entities are the "things" about which we seek information.

Attributes are the data we collect about the entities.

Relationships provide the structure needed to draw information from multiple entities.

Generally, ERD's look like this:



adapted from another professor.

Developing an ERD

Developing an ERD requires an understanding of the system and its components. Before discussing the procedure, let's look at a narrative created by Professor Harman.

Consider a hospital:

Patients are treated in a single ward by the doctors assigned to them. Usually each patient will be assigned a single doctor, but in rare cases they will have two.

Healthcare assistants also attend to the patients, a number of these are associated with each ward.

Initially the system will be concerned solely with drug treatment. Each patient is required to take a variety of drugs a certain number of times per day and for varying lengths of time.

The system must record details concerning patient treatment and staff payment. Some staff are paid part time and doctors and care assistants work varying amounts of overtime at varying rates (subject to grade).

The system will also need to track what treatments are required for which patients and when and it should be capable of calculating the cost of treatment per week for each patient (though it is currently unclear to what use this information will be put).

How do we start an ERD?

1. Define Entities: these are usually nouns used in descriptions of the system, in the discussion of business rules, or in documentation; identified in the narrative (see highlighted items above).
2. Define Relationships: these are usually verbs used in descriptions of the system or in discussion of the business rules (entity _____ entity); identified in the narrative (see highlighted items above).
3. Add attributes to the relations; these are determined by the queries, and may also suggest new entities, e.g. grade; or they may suggest the need for keys or identifiers.

What questions can we ask?

- a. Which doctors work in which wards?
- b. How much will be spent in a ward in a given week?
- c. How much will a patient cost to treat?
- d. How much does a doctor cost per week?
- e. Which assistants can a patient expect to see?
- f. Which drugs are being used?

4. Add cardinality to the relations

Many-to-Many must be resolved to two one-to-manys with an additional entity

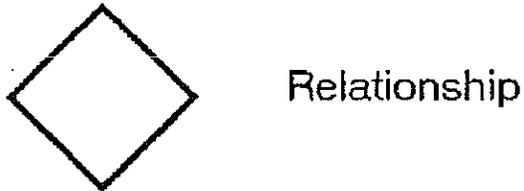
Usually automatically happens

Sometimes involves introduction of a link entity (which will be all foreign key) Examples:
Patient-Drug

5. This flexibility allows us to consider a variety of questions such as:

- a. Which beds are free?
- b. Which assistants work for Dr. X?
- c. What is the least expensive prescription?
- d. How many doctors are there in the hospital?
- e. Which patients are family related?

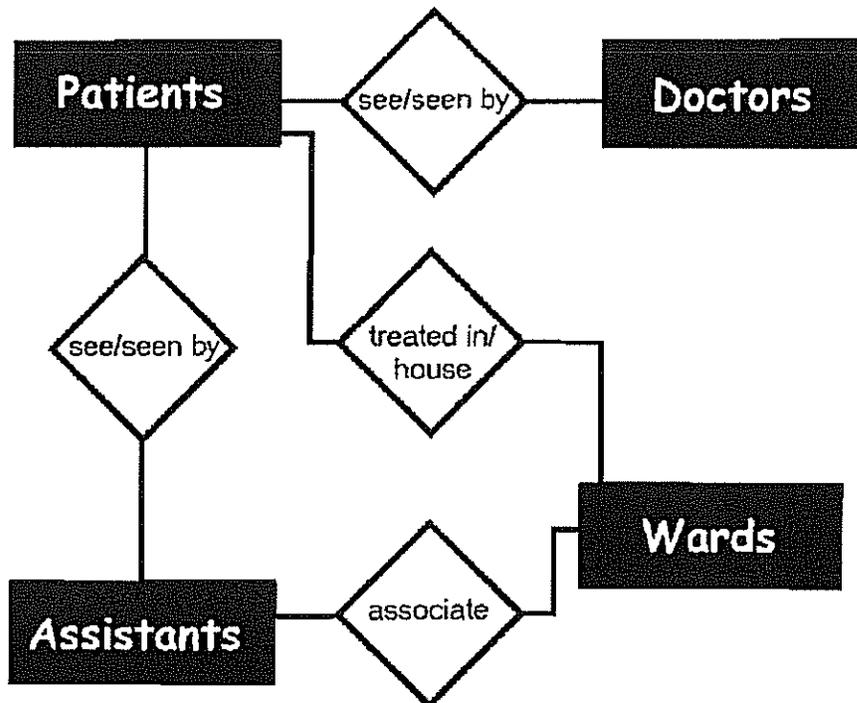
6. Represent that information with symbols. Generally E-R Diagrams require the use of the following symbols:



Reading an ERD

It takes some practice reading an ERD, but they can be used with clients to discuss business rules.

These allow us to represent the information from above such as the E-R Diagram below:



ERD brings out issues:

- Many-to-Manys
- Ambiguities
- Entities and their relationships
- What data needs to be stored
- The Degree of a relationship

Now, think about a university in terms of an ERD. What entities, relationships and attributes might you consider? Look at this simplified view. There is also an example of a simplified view of an airline on that page.

You can investigate more about ERDs by viewing these sources available on the Internet:

- E-R Diagrams, Tables and their Meaning
- Entity-Relationship Diagrams
- Entity-Relationship Diagrams
- Entity-Relationship Diagrams
- 4.1 Lecture: Entity Relationship Analysis
- Five Entities related to Agent
- Crosswalk Exercise - ALMRS Customer with the Standard Guidelines
- Domain Analysis
- ERDIAG

| [UM-St. Louis Home Page](#) | [College of Business Page](#) | [IS Home Page](#) | [Analysis Current Page](#) |

This page was last modified on: 08/13/2000 20:05:40

URL: http://www.umsl.edu/~sauterv/analysis/er/er_intro.html

Page Owner: [Professor Sauter \(Vicki.Sauter@umsl.edu\)](mailto:Vicki.Sauter@umsl.edu)

© Vicki L. Sauter. All rights Reserved.

Data and Visualization Entity-Relationship Model

Introduction

We have covered the concepts of relational databases in "Introduction to Databases," how to access such databases in "Accessing Databases with SQL," creation of web pages with forms in "Creating Web Pages" and "Web Forms for Database Queries," and CGI programming to interface between web pages and databases and to process data in "CGI Programs and Web Forms" and "CGI Programs in C++ Using the MySQL C API," "In Genomic Data," "Genomic Sequence Comparison," and "Searching Genomic Databases," we studied some of the algorithms to process genomic data and how to use these algorithms in conjunction with the above tasks. Until now, however, we have employed existing databases. The current module, "Relational Database Development," and "Creating and Changing Databases with SQL" discuss how we can design and produce databases. The ability to do so is important for development of databases for our own use or for larger computational science applications. Throughout this discussion, we consider the "College Physics Example" of the module "Computational Science and Web-Accessed Databases" as well as other applications.

Three-Level Architecture

We can consider the database on three levels of abstraction: external, conceptual, and internal. (See Figure 1.)

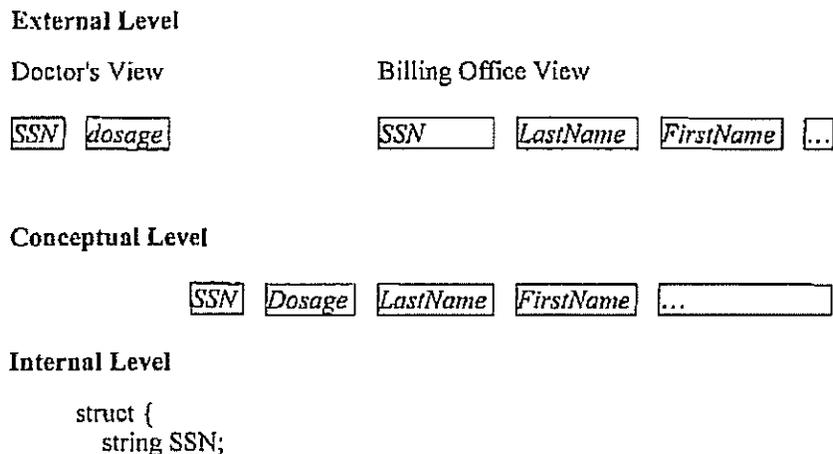
The **external level** has the users' views of the database. Depending on their needs, different users access different parts of the database. For example, a doctor performing drug tests should be able to access the patients' medical data but not their hospital bills. However, a billing clerk should have a very different view of the database.

The **conceptual level** describes the logical structure of an entire database, including descriptions of the data and relationships among the data. For example, at this level we would describe a row of the table *StudyA05* as containing the values for *SSN* and *placebo*. However, we would not give the details of the physical storage of the fields and records.

The **internal level** gives the details of the physical storage of the database on the computer. This level contains such details as the number of bytes for each data item, ordering of records, and data compression techniques. For example, at this level we would describe the attribute *SSN* as 10 bytes to store the nine-digit social security number.

Definition	The external level of a database has the users' views of the database. The conceptual level describes the logical structure of the entire database, including descriptions of the data and relationships among the data. The internal level gives the details of the physical storage of the database on the computer.
------------	--

Figure 1. Three-Level Architecture



```

double dosage;
string LastName;
string FirstName;
...
}

```

ER Model for Conceptual Design

In 1976, Chen developed the **Entity-Relationship (ER) model**, a high-level data model that is useful in developing a conceptual design for a database. Creation of an ER diagram, which is one of the first steps in designing a database, helps the designer(s) to understand and to specify the desired components of the database and the relationships among those components. An ER model is a diagram containing entities or "items", relationships among them, and attributes of the entities and the relationships.

Definition	The Entity-Relationship (ER) model, a high-level data model that is useful in developing a conceptual design for a database.
------------	--

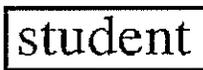
To make the description of the model more complete, we consider the example of a physics department at a college that maintains a database of experimental results. Throughout a laboratory, students collaborate and share their results and access data sets from other semesters on a computer system. For example, in the laboratory session on "Freely Falling Objects with Significant Drag," students determine the drag coefficient by dropping dust balls from different heights and measuring the times they take to fall. Each team enters its results into the distributed database, and the class analyzes the data. After a team enters data into the web-accessed database, all students can obtain the measurements simultaneously. To simplify the analysis, we assume that the database only stores results related to this experiment over a period of several years.

Entity

An **entity** is a real-world item or concept that exists on its own. In our example, a particular student (such as, "Emanuel Vagas"), team, lab section, or experiment is an entity. The set of all possible values for an entity, such as all possible students, is the **entity type**. In an ER model, we diagram an entity type as a rectangle containing the type name, such as *student* (see Figure 2).

Definition	An entity is a real-world item or concept that exists on its own. The set of all possible values for an entity is the entity type.
------------	--

Figure 2. ER diagram notation for entity *student*



Attribute

Each entity has **attributes**, or particular properties that describe the entity. For example, student Emanuel Vagas has properties of his own Student Identification number, name, and grade. A particular value of an attribute, such as 93 for the grade, is a **value** of the attribute. Most of the data in a database consists of values of attributes. The set of all possible values of an attribute, such as integers from 0 to 100 for a grade, is the **attribute domain**. In an ER model, an attribute name appears in an oval that has a line to the corresponding entity box, such as in Figure 3.

Definition	An attribute of an entity is a particular property that describes the entity. The set of all possible values of an attribute is the attribute domain.
------------	---



Toll free (US/Canada): +1 (877) 441-1150
Customer Ticket System

- Home
- Products
- Trials
- Buy
- Contact Us
- Support Services
- News
- Community

Home :: Products :: ConceptDraw PRO :: ER Diagram tool

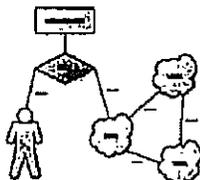
ConceptDraw PRO ER Diagram tool

Draw entity relationship diagrams (ER diagrams) easily with ConceptDraw PRO - ER Diagram tool!



Database diagram. Database modeling with ER Diagram tool.

ConceptDraw PRO serving as ERD software allows you to visually create Entity Relationship diagrams (ERD) for various database systems, it supports ODBC-compatible databases (Oracle, MS SQL, InterBase, MySQL, MS Access, Firebird and others). The built-in scripting language allows to connect to ODBC-compatible databases and automatically generate the structure of a database (reverse engineering of the database structure).



When creating an ERD our ERD software considers individual database options such as referential integrity, constraints, domains, triggers, stored procedures, functions, packages, views, sequences, synonyms etc.

Read a more detailed description of ConceptDraw PRO Pro for

database design

Don't forget to download the free evaluation version of ConceptDraw PRO - ERD software

ConceptDraw PRO Pro is a very easy-to-use and intuitive database design tool and ER Diagram tool which can save you hundreds hours of work.

ConceptDraw PRO Pro is not only an excellent tool for ER Diagram, but also the one that allows you to reverse engineer already existing database structures, create detailed HTML or PDF reports.

If you have any questions on how to use ConceptDraw PRO as ERD software please, contact us.

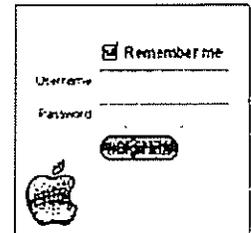
ConceptDraw PRO Pro is excellent choice of Database Designers for modeling, reverse and forward engineering of relational databases.

RS690

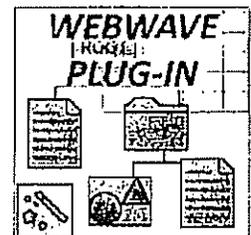
Choose your location:

Global [dropdown arrow]

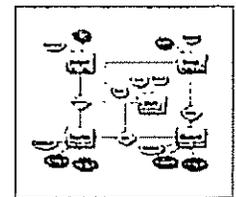
- ▶ Download Trial
- ▶ Buy Now
- ▶ Upgrade



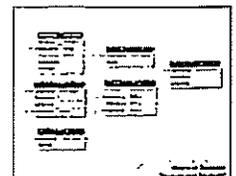
Web Layout Objects for Mac
CONCEPTDRAW

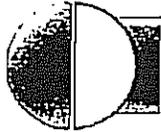


Web Site Prototype Builder



Chen ER Diagram





Download ConceptDraw PRO
Free 30-day trial, Windows, English



Operating Systems: Macintosh® (Mac OS X 10.4.10 or later)
Microsoft® Windows® 7/VISTA/XP

Languages: English

Price: cd_name currency_sing **BUY NOW**

Download FREE trial version - ERD diagram software

- Home
- Products
- Download
- Buy
- Resellers
- Contact Us
- Privacy Policy

Subscribe to CS Odessa Newsletter

4

Designing Data Model by Entity Relationship Diagram

Chapter 4 - Designing Data Model by Entity Relationship Diagram

DB Visual ARCHITECT (DB-VA) provides you a visual modeling environment for the object model of an application. This chapter shows you how to depict the object models by using Entity Relationship Diagram.

In this chapter:

- Introduction
- Creating Data Model by Entity Relationship Diagram
- Creating Array Table in Data Model
- Creating Partial Table in Data Model
- Copying SQL Statements

Introduction

An entity is an object in the business or system with well-defined characteristics which are represented by columns showing what information can be stored. In relational databases, an entity refers to a record structure, i.e. table.

A data model provides the lower-level detail of a relational database of an application. It shows the physical database models and their relationships in an application. An entity relationship diagram can be used to describe the entities inside a system and their relationships with each other; the entity relationship diagram is also known as a data model.

DB-VA supports visual modeling for data models, not only by creating a new data model, but also by transforming from an object model. As DB-VA automates object-relational mapping, DB-VA supports the generation of database, code and persistence layer for Java model API and .NET model API, which in turn streamlines the model-code-deploy software development process.

Creating Data Model by Entity Relationship Diagram

Entity relationship diagram is a graphical representation of a data model of an application. It acts as the basis for mapping the application to the relational database.

The following section describes how you can depict the data model using the entity relationship diagram.

DB-VA provides you with two ways to create a Class Diagram:

1. Drawing an Entity Relationship Diagram (ERD)
2. Synchronizing from Object Model to Data Model

Drawing an Entity Relationship Diagram

You can create a new ERD in one of the three ways:

- On the menu, click **File > New Diagram > Others > Entity Relationship Diagram**.

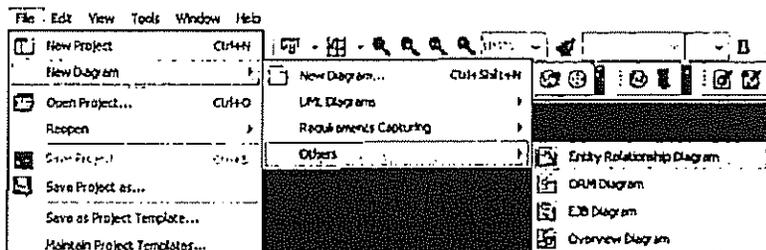


Figure 4.1 - Create an ERD by click on menu

- On the Diagram Navigator, right-click Entity Relationship Diagram > Create Entity Relationship Diagram.

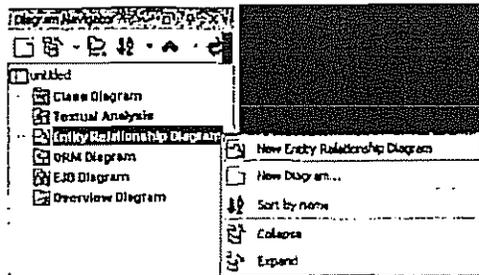


Figure 4.2 - Create ERD by click on Diagram Navigator

- On the toolbar, click the New Entity Relationship Diagram icon.

A new Entity Relationship Diagram pane is displayed.

Adding Entity

1. On the diagram toolbar, click the Entity shape icon.

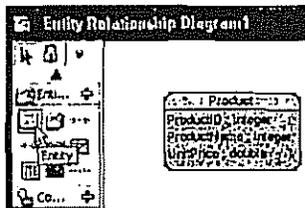


Figure 4.3 - Create an Entity by using the Entity icon

2. Click a location in the diagram pane.

DB-VA places an icon representing the entity element on the diagram.

3. Type a name for the Entity element.
 - You can edit the name by double-clicking the name or by pressing the *F2* button.

Modifying Entity Specification

An entity specification displays the entity properties and constraints.

You can display the Entity Specification in one of the two ways:

- Click on an entity class, click the Open Specification resource located at the top-right corner of the entity.

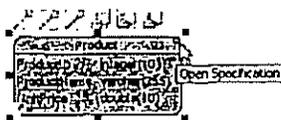


Figure 4.4 - "Open Specification" resource

- Right-click the entity, click **Open Specification...** from the pop-up menu.

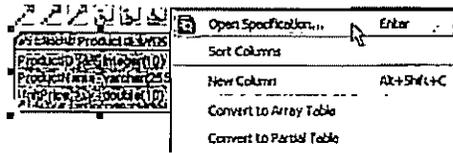


Figure 4.5 - Open specification by click on popup menu

Entity Specification dialog box is displayed, you can modify the entity properties and constraints.

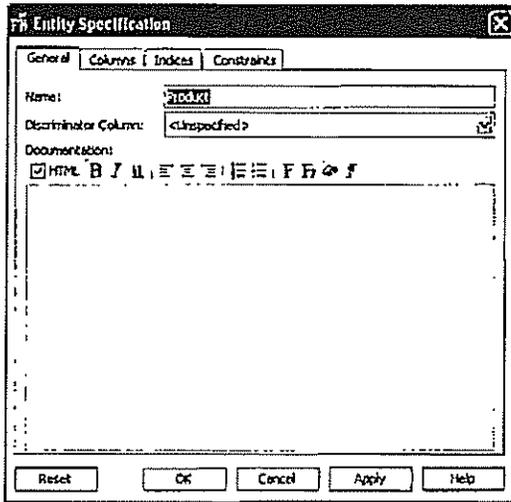


Figure 4.6 - Entity Specification dialog

Adding Column

You can add a new column to the entity in one of the three ways:

- Right-click on an entity, select **New Column**.

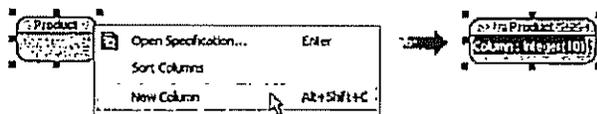


Figure 4.7 - Add column by click on popup menu

A new column is added, type the column name and type in the form of "column_name: type". You can also edit the column name by double-clicking the column name or by pressing the *F2* button.

- Click on an entity, press the keyboard shortcut - *Alt + Shift + C*.
- Add column in **Entity Specification** dialog:
 1. Right-click the entity element, click **Open Specification**.
 2. Click the **Columns** tab, then click **Add**.

Adding Relationship

Relationship shows how the entities are related to each other.

You can add a relationship to the entities in one of the two ways:

- Using Resource-Centric Interface
 1. Click on an entity, a group of valid editing resources is displayed around the shape.
 2. Mouse over the resource, select the desired resource, such as "One-to-One Relationship -> Entity".

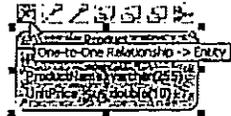


Figure 4.11 - "One-to-one Relationship -> Entity" resource

3. Drag the resource to the related entity.

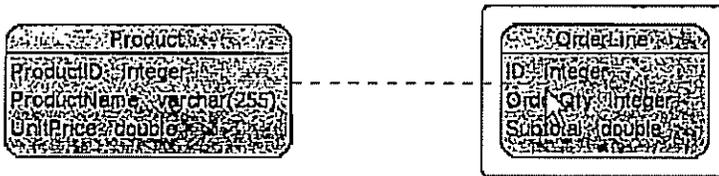


Figure 4.12 - Drag the resource to the related entity

- Using Toolbar icon
 1. On the diagram toolbar, click the Relationship icon.
 - One-to-One Relationship \leftrightarrow
 - One-to-Many Relationship \leftrightarrow
 - Many-to-Many Relationship \leftrightarrow
 2. Click on an entity and drag to the related entity.

A connector is added between the two entities.

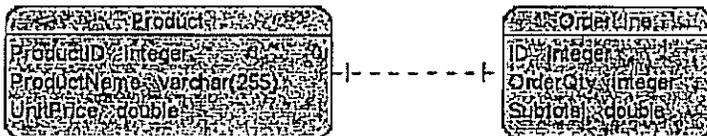


Figure 4.13 - an one-to-one relationship created

Modifying Relationship Specification

The relationship specification displays the relationship properties, such as name, phrase, and cardinality of the related entities. To open the relationship specification dialog box:

1. Right-click on the connection line, click **Open Specification...** from the pop-up menu.

Relationship Specification dialog box is displayed, you have to modify the relationship properties, Phrase and Cardinality.

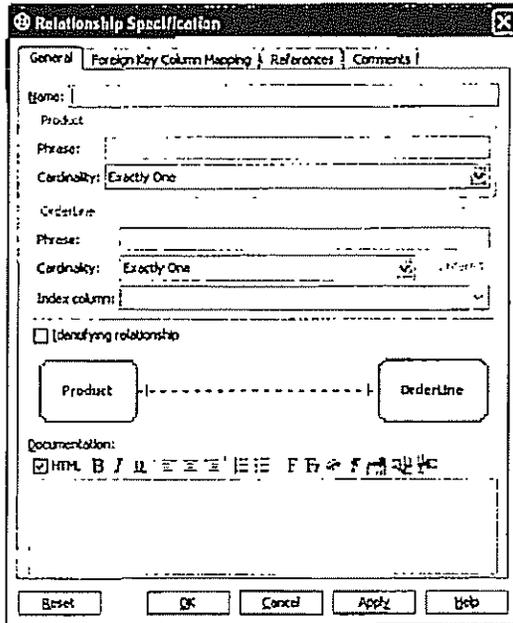


Figure 4.14 - Relationship Specification dialog

Synchronizing from Object Model to Data Model

DB-VA allows you to generate the ERD from a class diagram by synchronization as if there is a class diagram.

You can synchronize the Class Diagram to ERD in one of the three methods:

- On the menu, click **Tools > Object-Relational Mapping (ORM) > Synchronize to Entity Relationship Diagram.**

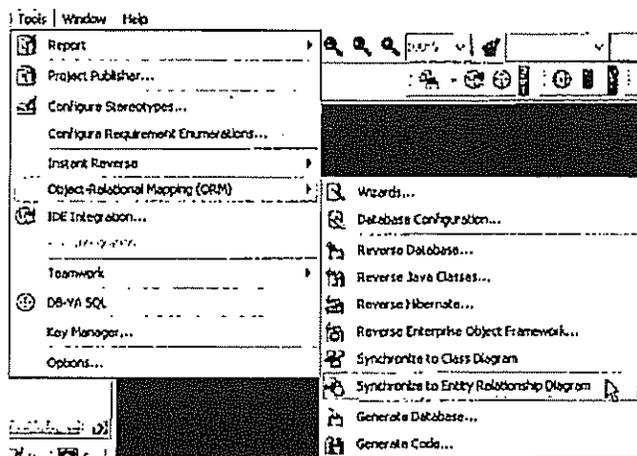


Figure 4.15 - Synchronize to ERD

- Right-click on the class diagram, select Synchronize to Entity Relationship Diagram.

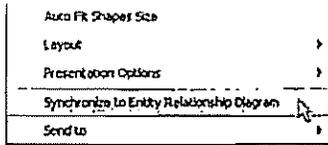


Figure 4.16 - Synchronize to ERD by click on popup menu

- On the class diagram, hold down the right-mouse button, move the mouse from left to right to form the gesture. A blue path is shown indicating the gesture.

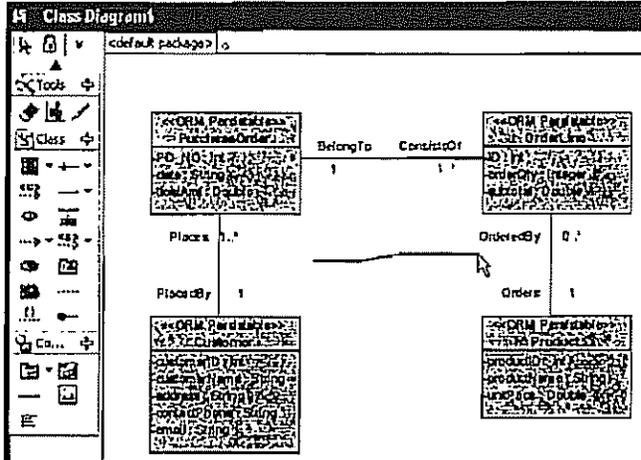


Figure 4.17 - Synchronize to ERD by using Gesture

An entity relationship diagram is generated and can be found under the Diagram Navigator.

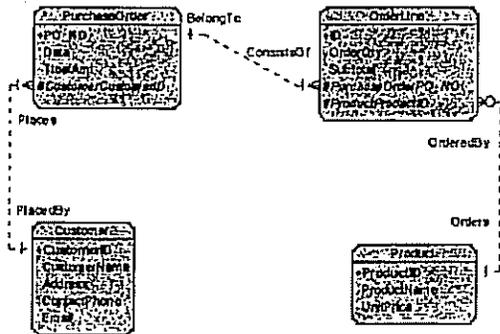


Figure 4.18 - The generated ERD

Specifying Primary Key

You can specify the column to be included in the primary key in one of the two ways:

- Specify Primary Key in Entity Specification dialog:
 1. Right-click on the entity, click **Open Specification...** to open the **Entity Specification** dialog box.
 2. Click the **Columns** tab, check the **Primary Key** option for the column that will be included in the primary key.

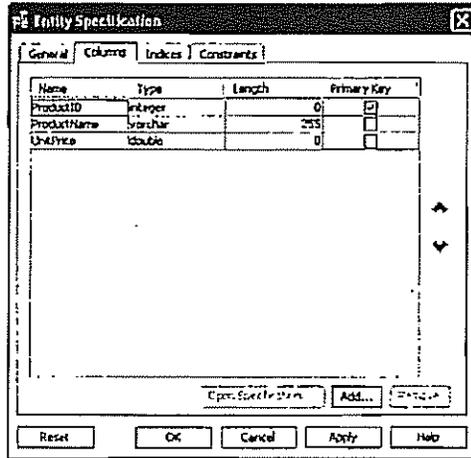


Figure 4.19 - Entity Specification dialog

- Specify Primary Key in Column Specification dialog:
 1. Right-click the column, click **Open Specification...** to open the **Column Specification** dialog box.
 2. Check the **Include in primary key** option.

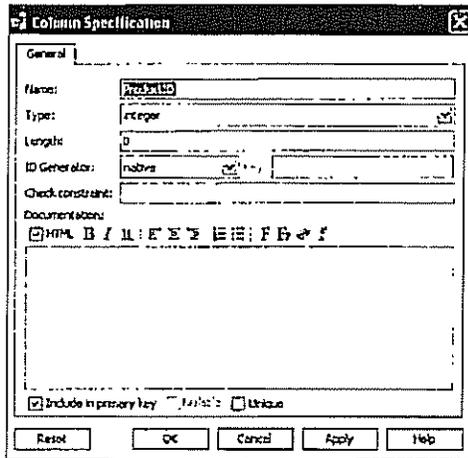


Figure 4.20 - Column Specification dialog

Note

If you assign a primary key to a column of an entity, DB-VA will automatically add a foreign key column to the related entities.

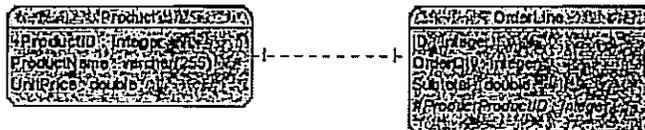


Figure 4.21 - The foreign key will be added automatically

Specifying Index Column

If a relationship with cardinality of many at one end, a corresponding collection class will be used for handling its multiple cardinality. DB-VA allows you to specify an index column to sort the collection.

1. Right-click on the connection line, click **Open Specification** from the pop-up menu.

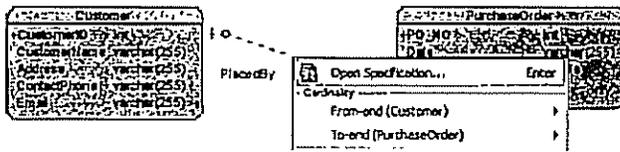


Figure 4.22 - Open relationship specification

Relationship Specification dialog box is displayed.

2. Check the option for **Ordered**.
3. Select the index column from the drop-down menu of **Index column**, click **OK**.

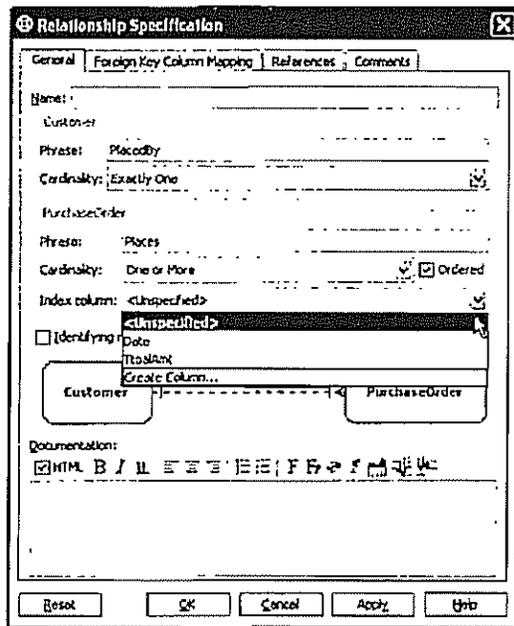


Figure 4.23 - Select the index column

Note



You can select **Create Column** from the drop-down menu to create a new index column for sorting.

Using the ID Generator

As the primary key is unique, DB-VA provides you with the generation of primary key. The ID Generator is specialized for generating a primary key value at runtime.

1. Right-click on the primary key of an entity, select **Open Specification** from the pop-up menu.

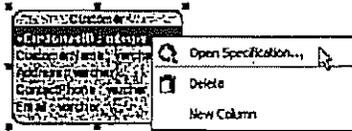


Figure 4.24 - Open the column specification

Column Specification of the primary key is displayed.

2. Select the ID generator from the drop-down menu of ID Generator, click **OK** to confirm setting.

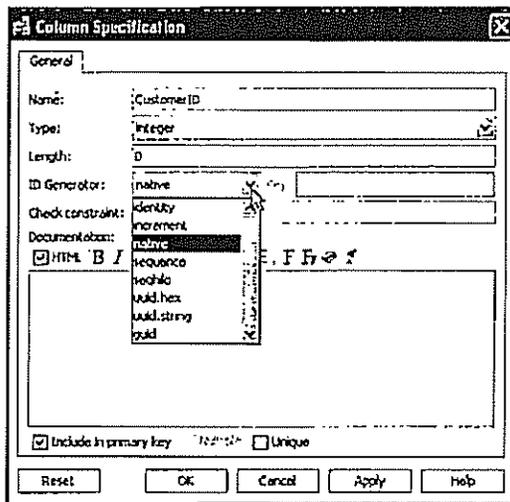


Figure 4.25 - Select the ID Generator

Note



If the ID Generator is specified as either sequence, seqhilo or hilo, you have to enter the key for the sequence/table name.

Defining Discriminator

In generalization, the superclass distributes its commonalities to a group of similar subclasses. The subclass inherits all superclass's attributes and it may contain specific attributes. DB-VA combines the entities within the hierarchy into one single entity containing all the attributes and a discriminator column for using table per class hierarchy as the inheritance strategy. The discriminator contains a unique value which is used for identifying the entity which hierarchy it belongs to. DB-VA allows you to define the discriminator in the entity and the discriminator value in the classes.

Defining Discriminator Column for Entity

You can add a new column acting as the discriminator column for an entity.

1. Right-click on an entity, select **New Column**.

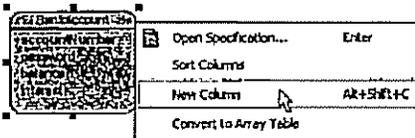


Figure 4.26 - Create a column

2. Enter the name and type for the discriminator in the form of "discriminator_name: type".



Figure 4.27 - Enter the column name and data type

3. Right-click on the entity, select **Open Specification...**

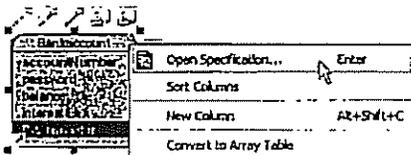


Figure 4.28 - Open the column specification

Entity Specification dialog box is displayed.

4. Select the desired column from the drop-down menu of **Discriminator Column**, click **OK** to confirm setting.

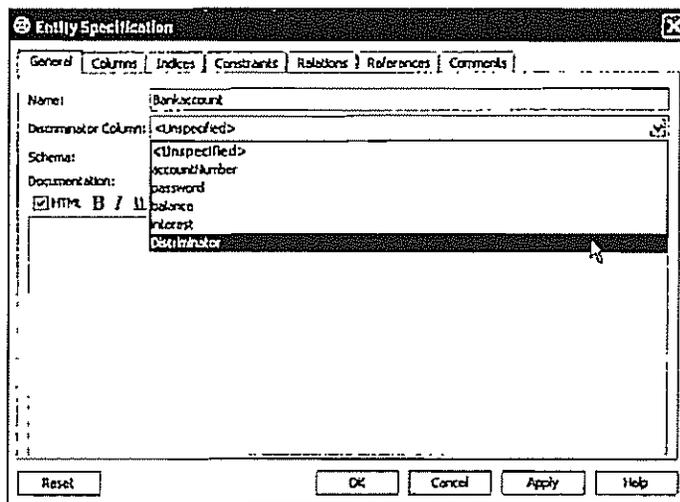


Figure 4.29 - Select the Discriminator Column

Defining Discriminator Value for Class

You can specify the discriminator value for each sub-class.

1. Right-click on the relative sub-class for adding discriminator, select **ORM > ORM Class Details...** from the pop-up menu. The **Class Specification** dialog box showing the **ORM Class Detail** tab is displayed.

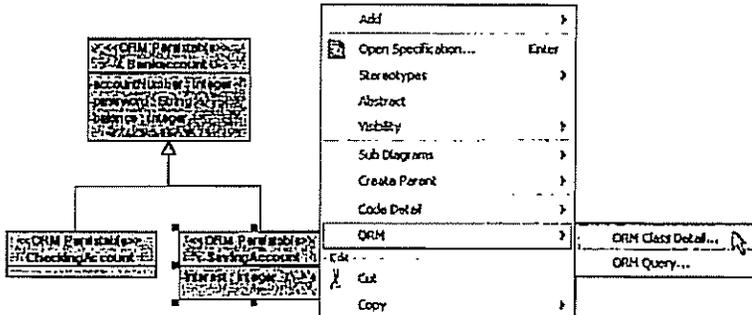


Figure 4.30 - Open the ORM Class Detail

2. Enter the discriminator value for identifying the sub-class.

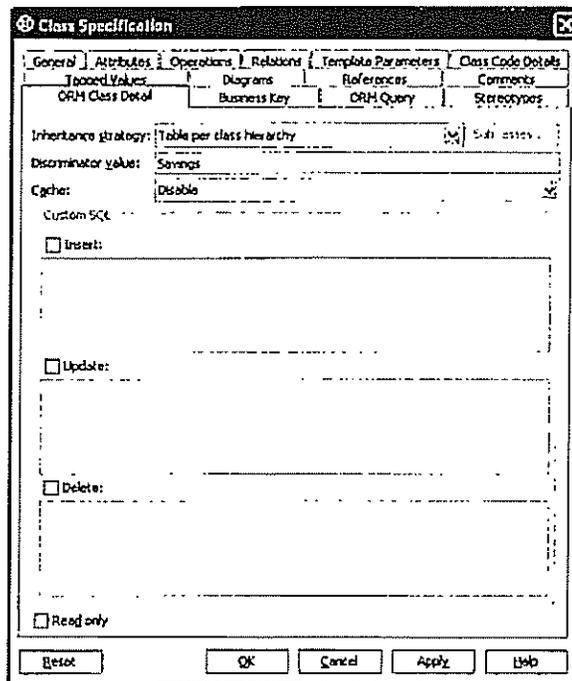


Figure 4.31 - Class Specification dialog (ORM Class Detail Tab)

Creating an Array Table

In a one-to-many relationship, a collection is used for handling the multiple objects such that it is simpler to retrieve each object from the collection one by one.

DB-VA promotes the idea of Array Table which allows users to retrieve objects in the form of primitive array, instead of a collection when handling a data column with cardinality of many.

DB-VA allows you to create an array table in the entity and define an array type in the classes.

Defining an Array Table

You can create an Array Table for the Entity with a column containing more than one instance of data.

1. Create a one-to-many relationship between the entity and one of its columns that may contain more than one instance of data.

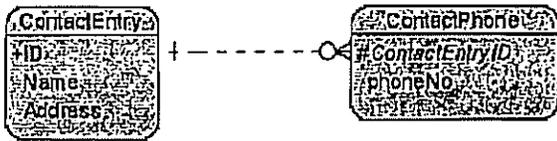


Figure 4.32 - Entities with One-to-many relationship

In the above case, the phonebook has a contact entry for each contact person. Each contact person may have more than one phone numbers. A one-to-many relationship between contact entry and contact phone can be built.

2. Right-click on the entity for the data column with cardinality of many, select **Convert to Array Table** from the pop-up menu.

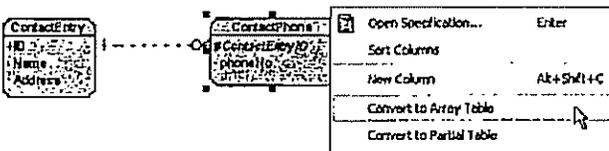


Figure 4.33 - Convert to Array Table

3. A warning message will be displayed, showing that the listed constraints are not satisfied for converting to array table. Click **Yes** to let DB-VA to resolve the constraints automatically. Click **No** to cancel the conversion to array table.

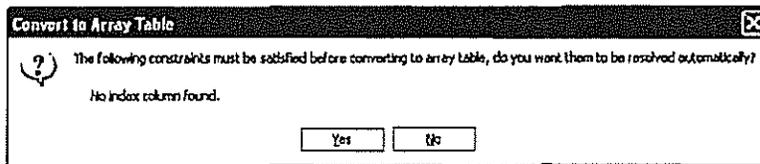


Figure 4.34 - Warning message for no index column

The conversion to Array Table is completed and the entity for the data column is stereotyped as Array Table.

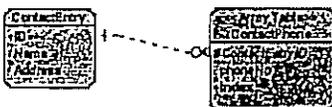


Figure 4.35 - Array Table created

Defining an Array Type for Attribute in Class

A class with an attribute of array type modifier means that the attribute may contain more than one data; thus it implies the idea of Array Table.

You can define the array type for the attribute in one of the two ways:

- Using Inline Editing
 1. Right-click on a class, click **Add > Attribute**.

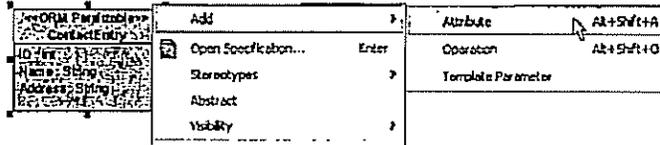


Figure 4.36 - Add an attribute by click on popup menu

2. Enter the name and type for the attribute in the form of "attribute_name :type[]", the sign, "[]" indicates the attribute is an array.

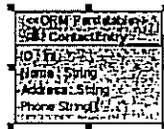


Figure 4.37 - Enter the attribute name and data types

- Using Class Specification dialog box
 1. Right-click on a class, click **Open Specification**.

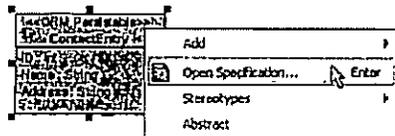


Figure 4.38 - Open the class specification

- The Class Specification dialog box is displayed
 1. Click **Attribute Tab**, click **Add**.

Attribute Specification is displayed.

2. Enter attribute name and type, select [] from the drop-down menu of Type modifier, then click OK to confirm setting.

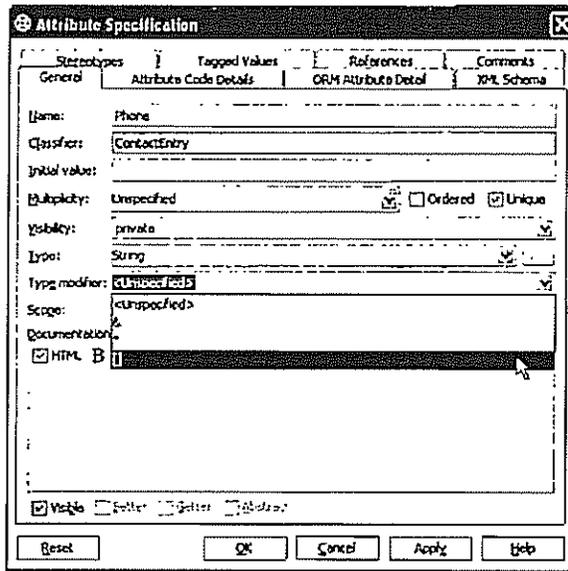


Figure 4.39 - Select the Type modifier

Creating a Partial Table

In a one-to-one identifying relationship, an entity may be a subordinate of the related entity; that is, the subordinate entity has columns which also belong to its superior entity in the real world situation.

DB-VA promotes the idea of Split Table with stereotype of Partial which allows developers to optimize the size of database, and minimizes the redundant persistent classes for handling one-to-one identifying relationship. In order to reduce the risk of appending a new column to an existing database table, Split table supports developers to add new columns to the partial table with a one-to-one identifying relationship linked to the existing table.

DB-VA allows you to split the entity into two and convert the subordinate entity to be a Partial Table in a one-to-one identifying relationship.

Splitting Table

You can split an entity into two associated with a one-to-one identifying relationship.

1. You can activate the Split Table dialog box in one of the two ways:
 - Using Pop-up Menu
 1. Right-click an entity, select Split Table.

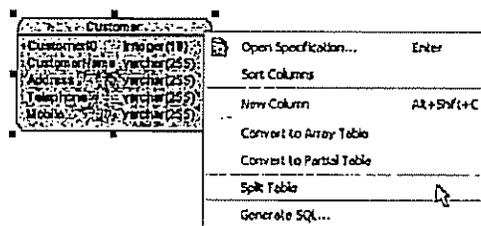


Figure 4.40 - Select "Split Table" in popup menu

- Using Resource-Centric Interface
 - Click on an entity, a group of valid editing resources are displayed around the entity.
 - Click the resource of "One-to-One Relationship -> Partial Table".

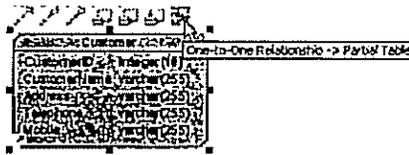


Figure 4.41 - Click on "One-to-one Relationship -> Partial Table" resource

- Split Table dialog box is displayed.

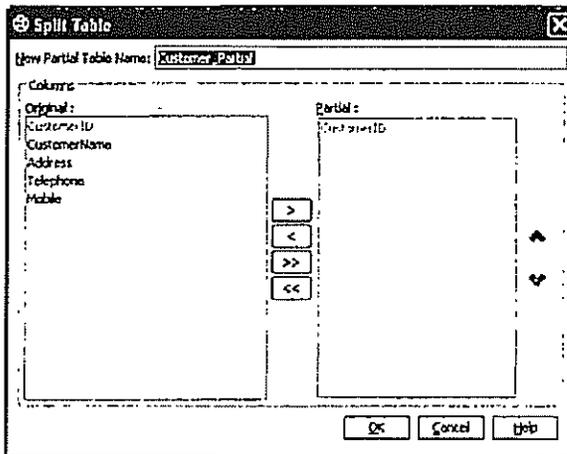


Figure 4.42 - Split Table dialog

- Edit the New Partial Table Name, select the columns from the list of Original to Partial, and click OK.

An entity stereotyped as <<Partial>> is created.

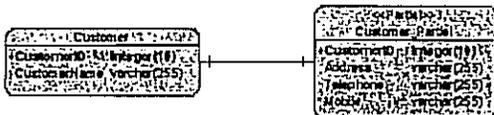


Figure 4.43 - Partial Table is created

Converting to a Partial Table

You can convert an entity to a Partial Table in a one-to-one identifying relationship.

- Right-click on the entity, select Convert to Partial Table from the pop-up menu.

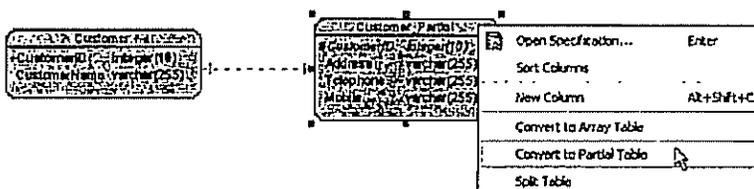


Figure 4.44 - Convert to Partial Table

The entity is stereotyped as <<Partial>>.

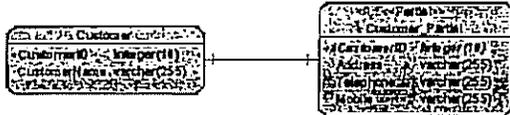


Figure 4.45 - Converted to a Partial Table

Copying SQL Statements from Tables

DB-VA provides function of copying SQL statements from the ERD entities. It allows the developers to copy the SQL statements from the entity relationship diagram easily such that developers can use and modify the SQL statement on the database directly.

In order to copy the SQL statement, you must configure the database setting in advance as DB-VA will generate the SQL statements according to the default database server type.

1. To configure database connection, on the menu, click **Tools > Object-Relational Mapping (ORM) > Database Configuration...**

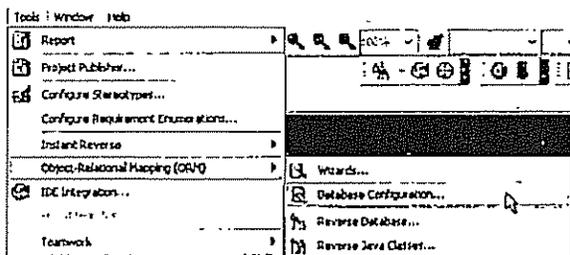


Figure 4.46 - To open the Database Configuration

Database Configuration dialog box will be displayed. To configure the database, refer to the descriptions of the Database Configuration section in the [Working with DB Visual ARCHITECT](#) chapter for more information.

Note

In DB-VA for IDE, you are allowed to configure the database connection by using **Modeling > ORM > Database Configuration** to display the **Database Configuration** dialog box.

Note

DB-VA will only provide you the function of copying SQL if the default database connection is set.

Note

If there are multiple database settings, DB-VA will generate SQL statements for all these database servers.

Example:

There are two database settings selected in the DB-VA environment.

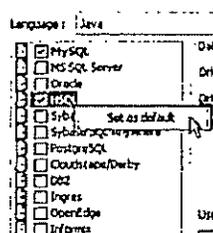


Figure 4.47 - Two database settings Selected

2. Right-click on the ERD, select **Generate SQL...** from the pop-up menu.

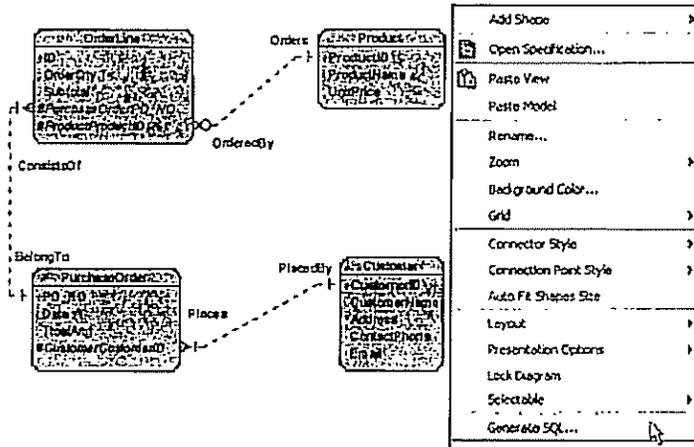


Figure 4.48 - Select generate SQL in popup menu

3. Generate SQL dialog box is displayed, select the database server from the drop-down menu of Database, the corresponding SQL statements will be displayed accordingly.

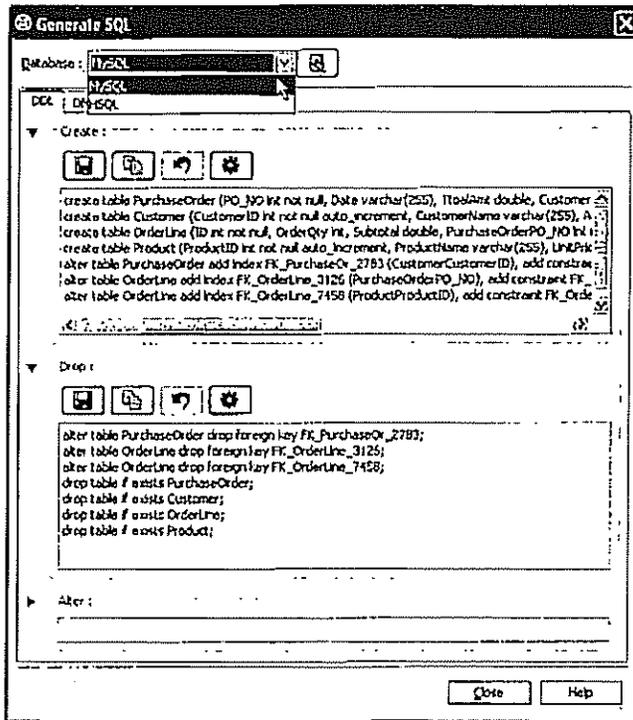


Figure 4.49 - Generate SQL dialog

You are allowed to copy the SQL statements from the Generate SQL dialog box.

Note

You can select **Create Table(s)**, **Drop Table(s)**, **Select**, **Insert**, **Update** and **Delete** from the Generate SQL dialog to directly copy the SQL statements to clipboard.

Copying SQL Statements from Specified Scope

You can specify the scope on the ERD for DB-VA to generate the SQL statements.

You can specify one of the three scopes:

- All entities on the ERD
 1. Click on an entity.
 2. On the menu, select **Edit > Select All of Same Type**.
 3. Right-click on the entity, select **Generate SQL...**

As copying SQL without specifying scope, SQL statements will be generated for all components including both entities and relationships on the ERD.

Example:

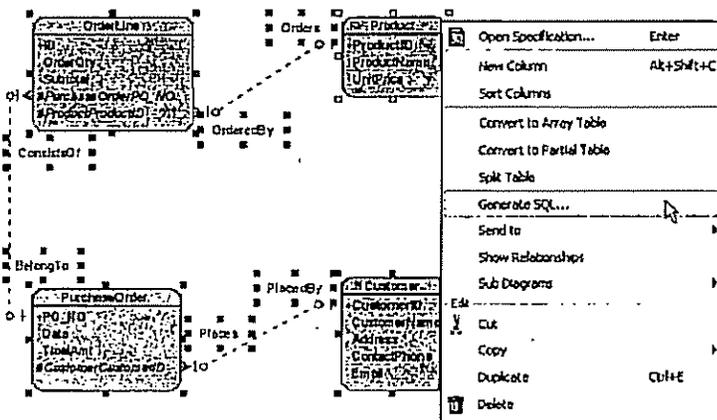


Figure 4.50 - Generate SQL for all entity

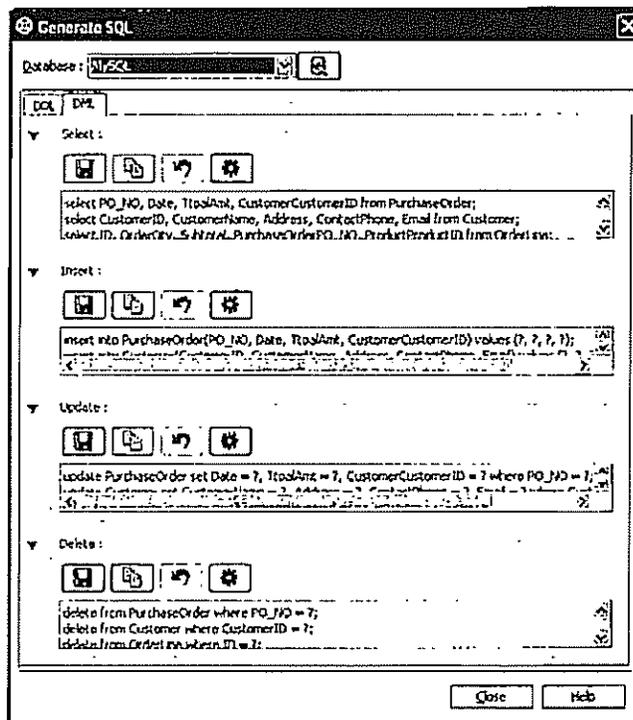


Figure 4.51 - The generated SQL for all entities

- Multiple entities and connection lines on the ERD
 - Select several entities and relationships on the ERD, right-click on the diagram pane, select **Copy SQL** from the pop-up menu.

As generate SQL with specifying a particular scope, SQL statements will be generated only for the components included in the specified scope.

Example:

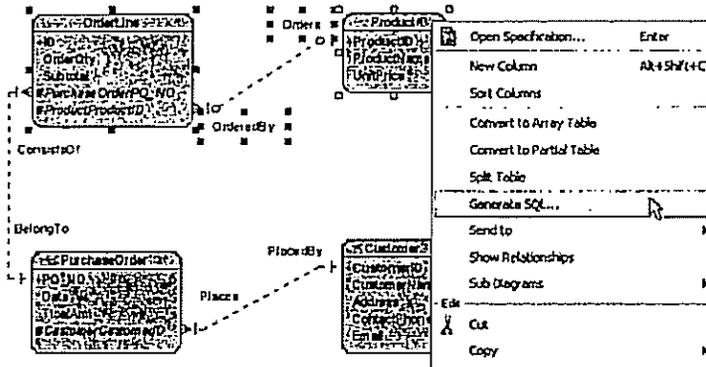


Figure 4.52 - Generate SQL for selected entities

- Connection lines on the ERD
 - Select connection line, right-click on the diagram pane, select **Generate SQL...** from the pop-up menu.

As Generate SQL... with connection lines, SQL statements for Create Constraint(s) and Drop Constraints(s) will be generated.

Example:

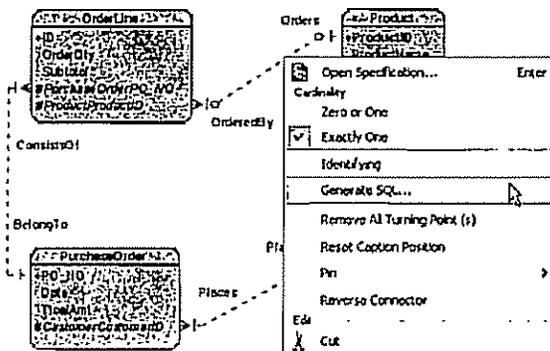


Figure 4.53 - Generate connection line's constraint

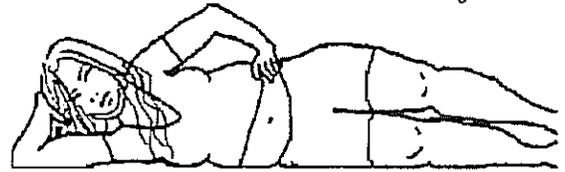
```

Create
Constraint : alter table 'OrderLine' add index 'FK_OrderLine_1969'
              ('PurchaseOrderPO_NO'), add constraint 'FK_OrderLine_1969' foreign key
              ('PurchaseOrderPO_NO') references 'PurchaseOrder' ('PO_NO');

Drop
Constraint : alter table 'OrderLine' drop foreign key 'FK_OrderLine_1969';
  
```


Advertisement

1 Rule of a flat stomach :
Cut down a bit of stomach fat every day by using this 1 weird old tip.



About.com: Databases

Entity-Relationship Diagram

[Share](#) [Print](#)

By Mike Chapple, About.com Guide

Free Databases Newsletter!

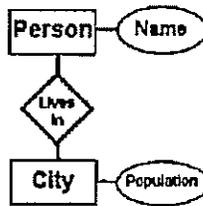
Enter email address

SIGN UP

[Discuss in my Forum](#)

See More About: [database design](#) [entity relationship](#) [er diagram](#)

Definition: An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.



E-R Diagram

Also Known As: ER Diagram, E-R Diagram, entity-relationship model

Examples:

Consider the example of a database that contains information on the residents of a city. The ER diagram shown in the image above contains two entities -- people and cities. There is a single "Lives In" relationship. In our example, due to space constraints, there is only one attribute associated with each entity. People have names and cities have populations. In a real-world example, each one of these would likely have many different attributes.

Explore Databases

See More About:

- [database design](#)
- [er diagram](#)
- [entity relationship](#)

By Category

- [Microsoft Access](#)
- [Development](#)

Must Reads

- [Databases for Beginners](#)
- [SQL Basics](#)
- [Top 10 SQL FAQs](#)
- [Top 5 Things](#)

RS714

- Learning SQL
- SQL Server
- Oracle
- Security in Databases
- MySQL
- Administration
- Design

- Data Mining
- Database Training
- Careers
- Reviews
- Certifications
- Other Databases

- Beginners Need to Know
- Database Glossary

Most Popular

- Database Normalization Basics
- SQL Server Stored Procedures
- Entity-Relationship Diagram
- Primary Key
- Microsoft Access Fundamentals

- Login
- Become a member
- RSS
- Part of the TechTarget network

SearchSQLServer.com

News

o Latest Headlines

- [SQL Server 2008 R2 Express ready for install](#)
- [Project Houston offers cloud alternative for SSMS](#)
- [eZines and eBooks for SQL Server professionals](#)
- [View All News](#)

SQL Server Topics

o Topics

■ [SQL Server Platforms](#)

[SQL Server 2008 R2](#), [SQL Server 2008](#), [SQL Server 2005](#), [SQL Server 2000](#)

[SQL Server Administration](#)

[Performance Tuning](#), [Installation](#), [Security](#), [T-SQL](#), [Stored Procedures](#), [Replication](#), [Availability and Scalability](#), [Backup and Recovery](#), [Tools and Utilities](#), [Interoperability](#)

■ [BI and Data Warehousing](#)

[BI Strategies](#), [Analysis Services](#), [Integration Services](#), [Reporting Services](#), [Data Warehousing](#), [Data Visualization](#)

[SQL Server Development](#)

[Database Design and Modeling](#), [.Net Development](#), [XML](#)

■ [SQL Server Management Issues](#)

[Migration and Planning](#), [Compliance](#), [Consolidation and Virtualization](#), [Career Advice](#), [Competitors](#)

o Hot Topics

- [SQL Server 2008 R2](#)
- [BI Strategies](#)
- [Migration and Planning](#)

Tutorials

o Advice & Tutorials

- [SQL Server Consolidation Fast Guide](#)
- [Tutorial: Performance tuning SQL Server via queries, indexes and more](#)

- SQL Server Service Broker Tutorial and Reference Guide
 - All-in-One Guide: SQL Server backup and recovery
 - Checklist: Getting ready for SQL Server 2005
 - Checklist: How to archive SQL Server backups
 - Technology Dictionary
 - [Find definitions and links to technical resources](#)
 - Powered by [WhatIs.com](#)
- Expert Advice
 - Tips
 - Exploring T-SQL metadata functions in SQL Server 2008
 - Inspecting property settings with T-SQL metadata functions
 - Uncovering database file information with T-SQL metadata functions
 - View All Tips
 - Ask a Question
 - Get help from our technical community
 - Powered by [ITKnowledgeExchange.com](#)
- Research Library
 - Research Library
 - White Papers
 - [Webcasts](#)
 - Downloads
 - Powered by [Bitpipe.com](#)
 - Product Demos
 - Try out software demos
 - Powered by [2020Software.com](#)
 - Resource Centers
 - SQL Server Management and Consolidation Resource Center
 - SHI's Microsoft Resource Center
 - Virtualization of SQL Server Databases
 - View All Resource Centers
- Blogs
 - Blogs
 - SQL Server with Mr. Denny
 - Powered by [ITKnowledgeExchange.com](#)

Search this site	<input style="width: 150px;" type="text" value="SEARCH"/>	<input style="width: 50px;" type="button" value="Search"/>
------------------	---	--



- Home
- >entity-relationship model

Print

Email This

SearchSQLServer.com Definitions *(Powered by WhatIs.com)*

Look up tech terms

Powered by: WhatIs.com

Search listings for thousands of IT terms:

 Search

Browse tech terms alphabetically:

- A
- B
- C
- D
- E
- F
- G
- H
- I
- J
- K
- L
- M
- N
- O
- P
- Q
- R
- S
- T
- U
- V
- W
- X
- Y
- Z
- #

entity-relationship model

Show me everything on SQL Server 2000

RS718

definition -

The entity-relationship model (or ER model) is a way of graphically representing the logical relationships of entities (or objects) in

9:43:32 AM Tuesday, September 21, 2010 <http://searchsqlserver.techtarget.com/definition/entity-relationship-mo...>

order to create a database. The ER model was first proposed by Peter Pin-Shan Chen of Massachusetts Institute of Technology (MIT) in the 1970s.

In ER modeling, the structure for a database is portrayed as a diagram, called an entity-relationship diagram (or ER diagram), that resembles the graphical breakdown of a sentence into its grammatical parts. Entities are rendered as points, polygons, circles, or ovals. Relationships are portrayed as lines connecting the points, polygons, circles, or ovals. Any ER diagram has an equivalent relational table, and any relational table has an equivalent ER diagram. ER diagramming is an invaluable aid to engineers in the design, optimization, and debugging of database programs.

In a logical sense, entities are the equivalent of grammatical nouns, such as employees, departments, products, or networks. An entity can be defined by means of its properties, called attributes. Relationships are the equivalent of verbs or associations, such as the act of purchasing, the act of repairing, being a member of a group, or being a supervisor of a department. A relationship can be defined according to the number of entities associated with it, known as the degree.

Learn more about SQL Server 2000

- [Tutorial: Learn SQL Server basics from A-Z](#)
Tutorial - If you're having issues with the basics of SQL Server, this learning guide will be the refresher you need. With a wide range of topics, these tutorials and tips will help you relearn SQL Server from...
- [Track changes to SQL Server 2000 and 2005 with one simple utility](#)
Tip - As a DBA, you need to be aware of what's happening within your SQL Servers. But if you have dozens of SQL Servers -- both 2000 and 2005 models -- and permissions are shared, you'll have to go beyond...
- [FAQ: Creating and altering SQL Server tables](#)
FAQ - Whether you want to select rows, move records, track activity or perform various other table-related tasks, this collection of frequently asked questions offers expert insights to help.
- [Chapter Download: Essential SQL Server 2000 maintenance](#)
Chapter Download - This chapter from the book, *Essential SQL Server 2000*, by Buck Woody discusses two types of maintenance tasks: tasks that keep the database going and tasks that keep the database going fast
- [Learning Guide: SQL Server basics](#)
Learning Guide - SQL Server is gaining ground in the database space. As more companies turn to the DBMS hyped as the low-cost big player, more DBAs will have to learn its intricacies. With that in mind, we created...
- [Track changes to SQL Server 2000 and 2005 with one simple utility](#)
Tip - As a DBA, you need to be aware of what's happening within your SQL Servers. But if you have dozens of SQL Servers -- both 2000 and 2005 models -- and permissions are shared, you'll have to go beyond...
- [Problems running SQL Server 2000 RTM without service pack](#)
Ask the Expert - Business Intelligence expert Eric Brown helps troubleshoot database growth problems.

last updated 09 Feb 2004

Do you have something to add to this definition? Let us know.

Send your comments to techterms@whatis.com

More resources from around the web:

- [Peter Pin-Shan Chen has published a white paper about the ER model \(PDF download, 1.7 MB\).](#)

- [Information Technology Services provides an overview of the ER model.](#)

SQL Server RELATED LINKS

Ads by Google

RS719

- [Dashboard Best Practices Quickly Create Valuable Dashboards. Download Your Free White Paper Now. LogiXML.com/Dashboards](#)
- [Common Data Model Provide a Common Data Model for SOA-Based Applications. web.progress.com/datatend](#)

9:43:32 AM Tuesday, September 21, 2010 <http://searchsqlserver.techtarget.com/definition/entity-relationship-mo...>

- [SPSS Data Mining Paper](#) Get This Free SPSS White Paper On The Top 3 Keys To Data Mining Today www.cognos.com/SPSS_data_mining
- [Management Dashboards](#) Create impressive Excel dashboards in hrs, from multiple data sources. www.apesoft.us
- [ETL Open Source Download](#) Fast, Powerful, Affordable, Easy Migration Tools. Free Download! www.Talend.com/Open_Source_ETL

Related Content

- [All about data purging](#) What is data purging? What technology or method can I use to purge data in SQL Server 2000?
- [FAQ: Creating and altering SQL Server tables](#) Whether you want to select rows, move records, track activity or perform various other table-related tasks, this collection of frequently asked questions offers expert insights to help.
- [Problems running SQL Server 2000 RTM](#) without service pack Business Intelligence expert Eric Brown helps troubleshoot database growth problems.

Related Glossary Terms

Terms from WhatIs.com – the technology online dictionary

- [relational database](#) (searchSQLServer.com)
- [database management system](#) (searchSQLServer.com)
- [OLE DB](#) (searchSQLServer.com)

FILE EXTENSION AND FILE FORMAT LIST

File Extension and File Format List:

- A
- B
- C
- D
- E
- F
- G
- H
- I
- J
- K
- L
- M
- N
- O
- P
- Q
- R
- S
- T
- U
- V
- W
- X
- Y
- Z
- #

Microsoft SQL Server 2000

News, tips & more

- All about data purging (*Ask the Expert*)
- Problems running SQL Server 2000 RTM without service pack (*Ask the Expert*)
- Finding the SQL Server default collation (*Tip*)
- Can I connect SQL Server 2000 with SQL Server 2005... (*Ask the Expert*)
- Pros and cons of data warehousing with SQL Server 2000 (*Ask the Expert*)
- [View more](#)

Vendor content

- Breakthrough Economics for your Business: DB2 pureScale... (*Webcast*)

Save 10%
on Oracle Database
Management Packs

Enter **PERFORM** at Oracle Store checkout.

Buy NOW

Oracle Store

ORACLE

- Capturing Customer Value in a Multichannel World:... (*White Paper*)
- Database Servers: Consolidation via Virtualization -... (*Test Results*)
- Cloud-Stored Offsite Database Backups (*White Paper*)
- Reduce Costs & Increase Oracle®

relational database Solutions - SQL White Paper Library

Feedback

News

o Latest Headlines

- SQL Server 2008 R2 Express ready for install
- Project Houston offers cloud alternative for SSMS
- eZines and eBooks for SQL Server professionals

RS721

- - View All News
 - SQL Server Topics
 - Topics
 - SQL Server Platforms
 - SQL Server 2008 R2, SQL Server 2008, SQL Server 2005, SQL Server 2000
 - SQL Server Administration
 - Performance Tuning, Installation, Security, T-SQL, Stored Procedures, Replication, Availability and Scalability, Backup and Recovery, Tools and Utilities, Interoperability
 - BI and Data Warehousing
 - BI Strategies, Analysis Services, Integration Services, Reporting Services, Data Warehousing, Data Visualization
 - SQL Server Development
 - Database Design and Modeling, .Net Development, XML
 - SQL Server Management Issues
 - Migration and Planning, Compliance, Consolidation and Virtualization, Career Advice, Competitors
 - Hot Topics
 - SQL Server 2008 R2
 - BI Strategies
 - Migration and Planning
 - Tutorials
 - Advice & Tutorials
 - SQL Server Consolidation Fast Guide
 - Tutorial: Performance tuning SQL Server via queries, indexes and more
 - SQL Server Service Broker Tutorial and Reference Guide
 - All-in-One Guide: SQL Server backup and recovery
 - Checklist: Getting ready for SQL Server 2005
 - Checklist: How to archive SQL Server backups
 - Technology Dictionary
 - Find definitions and links to technical resources
 - Powered by WhatIs.com
 - Expert Advice
 - Tips
 - Exploring T-SQL metadata functions in SQL Server 2008
 - Inspecting property settings with T-SQL metadata functions
 - Uncovering database file information with T-SQL metadata functions
 - View All Tips
 - Ask a Question

- Get help from our [technical community](#)
 - Powered by [ITKnowledgeExchange.com](#)
 - Research Library
 - Research Library
 - White Papers
 - Webcasts
 - Downloads
 - Powered by [Bitpipe.com](#)
 - Product Demos
 - Try out [software demos](#)
 - Powered by [2020Software.com](#)
 - Resource Centers
 - [SQL Server Management and Consolidation Resource Center](#)
 - [SHI's Microsoft Resource Center](#)
 - [Virtualization of SQL Server Databases](#)
 - [View All Resource Centers](#)
 - Blogs
 - Blogs
 - [SQL Server with Mr. Denny](#)
 - Powered by [ITKnowledgeExchange.com](#)

Search this site	<input style="width: 80%;" type="text" value="SEARCH"/>	<input type="button" value="Search"/>
------------------	---	---------------------------------------

More from Related TechTarget Sites

- [Business Analytics](#)
- [Data Center](#)
- [Data Management](#)
- [Exchange](#)
- [Windows Development](#)
- [Windows IT](#)
- [Windows Server](#)

- [Business Analytics](#)

How to improve business intelligence performance with BI automation

Get tips on how to improve business intelligence performance with the help of BI automation. You'll learn about the value of BI automation, potential roadblocks and more.

○

Myth: Midsize companies need specialized business intelligence skills

Find out if IT workers and business users at midsize companies need specialized business intelligence skills and how businesses can leverage the BI skills they already have.

- o [Business intelligence cost concerns shouldn't stop midsize companies](#)

Read about business intelligence cost myths for midsize companies and learn about the advantages that midsize businesses can get from in-memory analytics and other BI technologies.

- o
- **Data Center**

[Improve benchmark performance with simple optimizations](#)

While benchmarking is great for measuring system performance, benchmarking tools themselves are applications capable of overtaxing the CPU. However, simple optimizations can ensure benchmark performance by preventing

- o these tools from overwhelming the system.
[Customize data center cooling design for increased energy efficiency](#)

Choosing to customize your data center cooling design can have a big impact on data center energy efficiency. Rather than opting for one-size-fits-all products that consume an unnecessary amount of power, look to customize cooling

- o for specific components in your facility, says one expert.
[HP reloads for 'converged infrastructure' war with Cisco](#)

- o HP and Cisco customers may get caught in the crossfire of the two vendors' converged infrastructure' skirmish.

- **Data Management**

[School districts tap open source integration software to aid reporting](#)

A consortium of more than 60 school districts plans to deploy open source data integration software to simplify its

- o complex, manual processes.
[The truth about retail data modeling software](#)

Find out if there are any high-quality retail data modeling software tools and if general-purpose tools in the data

- o modeling market can help with retail data model projects.
[How to organize an enterprise conceptual data model development effort](#)

Find out why it's important and how to organize an enterprise conceptual data model development. Learn how a data

- o governance organization can help your ECDM effort.

- **Exchange**

[The Outlook Social Connector](#)

The Outlook 2010 social connector combines users' social networks into one area, making it easier to find and

- o manage this information within Outlook.
[The conversation view in Outlook 2010](#)

The conversation view in Outlook 2010 lets users sort through inbox clutter to easily view past email conversations

- o within a specific message chain.
[The ignore option in Outlook 2010](#)

Using the ignore option in Microsoft Outlook 2010 allows you to remove yourself from irrelevant email

- o conversations.

- **Windows Development**

[Using the UniformGrid panel](#)

RS724

- o UniformGrid creates a mesh of equally sized cells. It's available in WPF and Silverlight and is very handy for creating

- o game-boards for games like chess.
Screen capture in a WPF application

Using a screen capture tool is second nature to most computer users and your customers may expect to find screen

- o capture features in your application. How much work is required to add this feature to a WPF application?
The 'VB.NET or C#' debate: it's all in the syntax

Microsoft development experts say Visual Basic and C# have become nearly identical in function as the .NET Framework has matured. Choosing whether to use VB.NET or C# is more or less a matter of which syntax style you prefer. Despite the similarities, many developers still carry a stigma against VB, said Lance Keene, president of .NET

- o consultancy Keene Systems.

- Windows IT

SharePoint is a bit for Microsoft but still chunky for IT

SharePoint is Microsoft's fastest growing server software, but IT managers still grouse about its lack of acceptance

- o and the difficulties in supporting this increasingly common collaboration platform.

Should Windows users have full administrative rights?

Giving Windows users full administrator privileges can help reduce your workload, but it also leaves your

- o environment vulnerable.

Microsoft offers Windows, Exchange, SP1 betas

At TechEd 2010, Microsoft executives breezed through service pack news in an effort to assuage IT managers'

- o concerns about the company's shift to cloud computing.

- Windows Server

Increased Windows support could help spur UCS adoption

Though new to IT, the adoption of unified computing systems could accelerate as Microsoft partnerships lead to

- o better UCS support for Windows servers and apps.

Windows server management guides and learning resources

Check out our collection of guides, tutorials, FAQs and more covering a wide array of Windows server administration

- o topics and technologies.

Free Windows performance monitoring tools not to be missed

Unbeknownst to some admins, the latest versions of Windows Server include new or updated tools that can minimize

- o the need for third-party options.

- About us
- Site index
- Privacy policy
- For advertisers
- For business partners
- TechTarget events
- Media kit
- TechTarget corporate site
- Site map

Entity-relationship model

Home > Multilingual Archive Index > Entity-relationship model

Choose your language:

English | Français | Español | Italiano | Português | Deutsch | Nederlands | Svenska | Ελληνικό | Русский | 한국어 | 日本語 | 简体中文 | 繁體中文 | العربية

Common Data Model Provide a Common Data Model for SOA-Based Applications. web.progress.com/datatxtend

Custom Business Software Call now for Free Consultation! ASP .NET SQL JSP Web AJAX Experts www.dlawton.com

Entity Relationship Model Download Now! Windows, Linux, OSX Diagram Databases and ER Models www.aquaoid.com

Database Migration Tool Centerprise® - the Key to Fast & Reliable Data Migration-View Demo www.astera.com/Data-M



Ads by Google

Entity-relationship model

An entity-relationship model/diagram (or ERD) is an abstract conceptual representation of structured data. Entity-relationship modeling is a relational schema database modeling method, used in software engineering to produce a type of conceptual data model (or semantic data model) of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created using this process are called *entity-relationship diagrams*, or *ER diagrams* for short. Originally proposed in 1976 by Dr. Pin-Shan (Peter) Chen (陳品山), many variants of the process have subsequently been devised.

The first stage of Information system design uses these models during the requirements analysis to describe information needs or the type of information that is to be stored in a database. The data modeling technique can be used to describe any ontology (i.e. an overview and classifications of used terms and their relationships) for a certain universe of discourse (i.e. area of interest). In the case of the design of an information system that is based on a database, the conceptual data model is, at a later stage (usually called logical design), mapped to a logical data model, such as the relational model; this in turn is mapped to a physical model during physical design. Note that sometimes, both of these phases are referred to as "physical design".

There are a number of conventions for entity-relationship diagrams (ERDs). The classical notation is described in the remainder of this article, and mainly relates to conceptual modeling. There are a range of notations typically employed in logical and physical database design, including *IDEF1x* (ICAM DEFINITION Language) and *dimens* ^{yes - *dimens*} ~~Ateling~~.

Contents

- 1 Connection
- 2 Alternative diagramming conventions
 - 2.1 Crow's Foot
- 3 See also
- 4 Proprietary ER diagramming tools
- 5 Free software ER diagramming tools
- 6 Free software Diagram tools
- 7 References
- 8 External links

Connection

An entity represents a **discrete object**. Entities can be thought of as nouns. Examples: a computer, an employee, a song, a mathematical theorem. Entities are represented as rectangles.

A relationship captures how two or more entities are related to one another. Relationships can be thought of as verbs, linking two or more nouns. Examples: an *owns* relationship between a company and a computer, a *supervises* relationship between an employee and a department, a *performs* relationship between an artist and a song, a *proved* relationship between a mathematician and a theorem. Relationships are represented as diamonds, connected by lines to each of the entities in the relationship.

The model's linguistic aspect described above is utilized in the database query language ERROL.

Entities and relationships can both have attributes. Examples: an *employee* entity might have a *Social Security Number (SSN)* attribute; the *proved* relationship may have a *date* attribute. Attributes are represented as ellipses connected to their owning entity sets by a line.

Every entity (unless it is a weak entity) must have a minimal set of uniquely identifying attributes, which is called the entity's primary key.

Entity-relationship diagrams don't show single entities or single instances of relations. Rather, they show entity sets and relationship sets. Example: a particular *song* is an entity. The collection of all songs in a database is an entity set. The *eaten* relationship between a child and her lunch is a single relationship. The set of all such child-lunch relationships in a database is a relationship set.

Lines are drawn between entity sets and the relationship sets they are involved in. If all entities in an entity set must participate in the relationship set, a thick or double line is drawn. This is called a participation constraint. If each entity of the entity set can participate in at most one relationship in the relationship set, an arrow is drawn from the entity set to the relationship set. This is called a key constraint. To indicate that each entity in the entity set is involved in exactly one relationship, a thick arrow is drawn.

Associative entity is used to solve the problem of two entities with a many-to-many relationship [1].

Unary Relationships - a unary relationship is a relationship between the rows of a single table.

Alternative diagramming conventions

Crow's Foot

The "Crow's Foot" notation represents relationships with connecting lines between entities, and pairs of symbols at the ends of those lines to represent the cardinality of the relationship. Crow's Foot notation is used in Barker's Notation and in methodologies

BOOK-MARK

Ads by Google

Top Technology Degrees

Earn a Degree in Technology Online.

Compare Top Accredited Schools.

EducationDegreeSource.com

Enterprise Architect B

Best North American

Price for Sparx

Enterprise Architect

8.0 software

www.iug.biz

Database Migration Tool

Centerprise® - the Key to Fast & Reliable

DEMO

www.astera.com/Data-Migrat

P&C Business Intelligence

50+ reports, what-if, dashboards, ad-hoc

reports ready to go

www.4SightBI.com

Free UML Modeling Tool

Support all UML 2 diagrams Generate

XML, PDF and JPG

www.visual-paradigm.com

Ads by Google

Database Design

3D Modeling Software

Entity Diagram

Database Software

ER Data Model

RS726

such as SSADM and Information Engineering. Also this notation is gaining acceptance through common usage in Oracle texts and in tools such as Visio, PowerDesigner, Toad Data Modeler, OmniGraffle and Dia.

Three symbols are used to represent cardinality:

- the *ring* represents "zero"
- the *dash* represents "one"
- the *crow's foot* represents "more" or "many"

These symbols are used in pairs to represent the four types of cardinality that an entity may have in a relationship.

- *ring* and *dash* → zero or one
- *dash* and *dash* → exactly one
- *ring* and *crow's foot* → zero or more
- *dash* and *crow's foot* → one or more

These are notations of a side in the ER diagrams.

You can see an example of the *crow's foot* notation in the diagram to the above.

In the diagram, the following facts are detailed:

- An Artist can perform zero or more Songs
- A Song is performed by exactly one Artist

Crow's foot notation has the following benefits:

- Clarity in identifying the many, or child, side of the relationship, using the *crow's foot*.
- Concise notation for identifying mandatory relationship, using a perpendicular bar, or an optional relation, using an open circle.

See also

- Enhanced Entity-Relationship Model (EER)
- Data model
- Data structure diagram
- Object Role Modeling (ORM)
- Unified Modeling Language (UML)
- Value range structure diagrams
- Natural language Information Analysis Method (NIAM)

Proprietary ER diagramming tools

- ConceptDraw: cross platform software for creating ER Diagrams
- Casewise Corporate Modeler Suite: The Corporate Modeler Suite provides a comprehensive modeling toolset and extensible repository, assisting organizations and consultants to work with, better understand, and re-align an organization's business processes, systems and enterprise architecture.
- Sparx Enterprise Architect: full UML 2.1 which includes data modeling.
- ER/Studio: robust, easy-to use ER modelling tool from Embarcadero.
- ERwin: ERD tool, ability to generate HTML reports.
- MEGA International: The MEGA Modeling Suite is a repository-based suite of modeling products for enterprise architecture, including E/R diagramming, data modeling, and database design.
- ModelRight: innovative and complete physical modelling tool - free community edition for MySQL.
- OmniGraffle: diagramming software for Mac OS X
- Oracle Designer: Oracle's CASE tool for designing an information system and generating it
- PowerDesigner: modeling suite from Sybase which includes Data Architect for constructing or reverse engineering conceptual, logical and physical models with many of the leading RDBMS brands.
- Rational Rose: Old software designed for producing UML diagrams. Notably easy to use which has aided its longevity
- SmartDraw: point and click drawing method combined with many templates creates professional diagrams.
- Toad Data Modeler: ER modelling tool with support for both logical and physical modeling. Includes reverse engineering, SQL generation and report generation features for several db systems.
- Visio: The Enterprise Architect version supports generating and reverse engineering databases
- Zim : Entity Relationship Integrated Design and Database.

Free software ER diagramming tools

Tools that can interpret and generate ER models, SQL and do database analysis.

- BrModelo: Brazilian designer for ERMs.
- DBDesigner-Fork: a fork of DBDesigner to make it work with other databases such as PostgreSQL.
- Ferret (software): ERM tool distributed with Debian and Ubuntu.[2]
- Giffy: Online charting website.
- ModelRight: innovative and complete physical modelling tool - free Community Edition for MySQL.
- MySQL Workbench: tool for graphically creating schemas (or, only in commercial version, reverse engineering schemas) (Beta Software), works with many database engines.
- Open System Architect: ER Diagram modeller the last version dates from 2005.
- PowerArchitect: ER Diagram modeller in Java, forward and reverse engineering for several databases, open-source (originally proprietary software).
- StarUML - supports UML and ER Diagrams.
- Dia

Free software Diagram tools

These are tools that can't create ER diagrams but just draw the shapes without having any knowledge of what they mean or generating SQL.

- Kivio: flowcharting program that supports ER Diagrams.
- Dia: program to draw many kinds of diagrams, including ERDs. It mainly serves the purpose of creating any type of UML

RS727

diagrams, but there are plug-ins (e.g. tedla2sql) that generates the SQL.

References

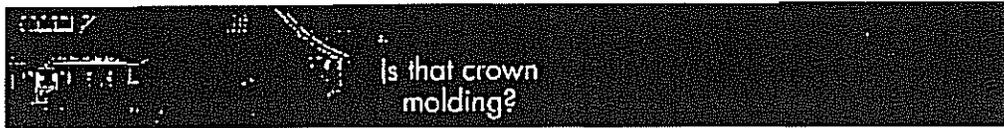
- Chen, Peter P. (1976). "The Entity-Relationship Model - Toward a Unified View of Data". *ACM Transactions on Database Systems* 1 (1): 9-36.
- Barker, Richard (1990). *CASE*Method: Tasks and Deliverables*. Wokingham, England: Addison-Wesley.

External links

- An Entity Relationship Diagram Example Demonstrates the crow's feet notation by way of an example.
- Peter Chen home page at Louisiana State University <http://bit.csc.lsu.edu/~chen/chen.html>
 - Origins of ER model pioneering
 - English, Chinese and ER diagrams (more deepened analysis of Chinese language)
- Case study: E-R diagram for Acme Fashion Supplies by Mark H. Ridley
- IDEF1X
- Notes: Logical Data Structures (LDSs) - Getting started - by - Tony Drewry
- Introduction to Data Modeling

The original article is from Wikipedia. To view the original article please click here.
Creative Commons Licence

Privacy Policy | Terms & Conditions | Free Translation
© Copyright 2010 WorldLingo Translations LLC. All rights reserved.



Home > Reference > Encyclopedia > entity relationship model

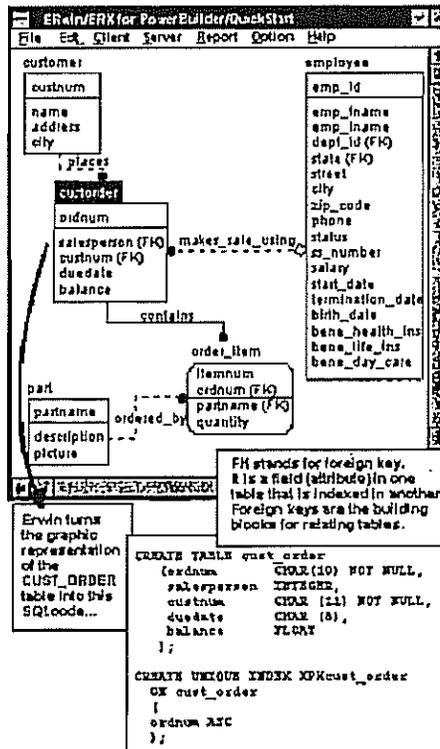


Search: Search Encyclopedia Browse the Index

Definition of: entity relationship model

A database model that describes the attributes of entities and the relationships among them. An entity is a file (table). Today, ER models are often created graphically, and software converts the graphical representations of the tables into the SQL code required to create the data structures in the database. See data model.

From Computer Desktop Encyclopedia
© 1995 The Computer Language Co. Inc.



A Database Schema

In this order processing example, the tables for customers and orders are drawn graphically, and the Erwin modeling program turns the graph into the appropriate SQL code for the target database. (Example courtesy of Logic Works, Inc.)



RELATED TERMS:

data model

Search:

[Search Encyclopedia](#)

[Browse the Index](#)



Copyright © 1991-2010
The Computer Language Company Inc
All rights reserved.

THIS COPYRIGHTED DEFINITION IS FOR PERSONAL USE ONLY.
All other reproduction is strictly prohibited without permission from the publisher.

Ads By Google

[what's this?](#)

Common Data Model

Provide a Common Data Model for SOA-Based Applications.
web.progress.com/datatend

newsletters

Get PCMag.com's FREE email newsletters delivered to your inbox.

It's easy, just follow the steps.

Want more? Check out our other newsletters here.

Manage your newsletter subscriptions here.

1. Make your selections:

- Daily News Alert
- PCMag.com Small Business Update
- PCMagCast Update
- Productwire: First Looks Update
- Security Watch
- TechSaver
- Utility Library Update
- PCMag Announcements

2. Select email format:

3. Enter email address:

Sponsors

- Google Chrome Download. Searching is fast and easy with Google's web browser. www.google.com/chrome
- Crucial memory is an affordable, effective way to boost PC performance
- Get the business phone service that gives you more. Internet fax, online meetings, call recording and softphone.
- Optimize 802.11n performance with Cisco CleanAir technology. [Learn How.](#)
- Boost productivity with cutting edge computers from HP and AMD.
- Ultimate wireless performance to power your life. Linksys E-Series.
- New Dell LED Studio monitor starting at \$159.99! Beautiful Screen Quality and Slim design!
- LogMeIn Rescue: On-demand remote support for PCs, Macs and more.
- Award-winning Norton 360 works 24/7 to protect your PC, music, & online identity. [Download a FREE Trial!](#)
- Brother Printers. [Rated #1 in Reliability. Learn More at Brother.com.](#)
- Make meetings more efficient with MindManager - start your free trial
- Save now with the best deals from CDW. Save on desktops, notebooks, LCDs and more.
- Xerox Travel Scanner 100 scans almost anything, anywhere.
- Upgrade today with new laptops and desktops from HP and Intel.
- PCMag Editors' Choice for drive imaging software - ShadowProtect 4.0.

[Home](#) > [Solutions](#) > [Encyclopedia](#) > [entity relationship model](#)

RS730



Chrome fast.

ABOUT PCMAG.COM

[Contact Us](#) [Advertise](#)
[About Us](#) [Reprints & Permissions](#)
[Newsletters](#) [Site Map](#)
[RSS Feeds](#)

PC MAGAZINE DIGITAL EDITION

[Issue Index](#)
[Customer Service](#)
[Subscribe](#)



SUBSCRIBE

Get PCMag for only \$1 an issue!



REVIEWS

[Editors' Choice](#)
[Desktops](#)
[Laptops](#)
[Cell Phones](#)
[Printers](#)
[Networking](#)
[Software](#)
[Digital Cameras](#)
[HDTV](#)
[Netbooks](#)
[Security](#)
[MP3 Players](#)
[Windows 7](#)
[Shop for Tech](#)
[Review Roundups](#)
[Inside PC Labs](#)

DOWNLOADS

[Shareware](#)
[Utilities](#)

NEWS & OPINION

[Tech News](#)
[Lance Ulanoff](#)
[John C. Dvorak](#)
[Dan Costa](#)
[Sascha Segan](#)
[Tim Bajarin](#)
[Michael J. Miller](#)

PCMAG NETWORK

[AppScout](#)
[ExtremeTech](#)
[Gearlog](#)
[GoodCleanTech](#)
[PCMag.com](#)
[Security Watch](#)
[Smart Device Central](#)
[TechSaver](#)
[PCMag Radio & After Hours](#)
[CrankyGeeks](#)
[DL.TV](#)
[PCMag Mobile](#)

TIPS

[Ask the Experts](#)
[Build It/DIY](#)
[Buying Guide](#)
[Encyclopedia](#)
[Forum](#)

PCMAG EVERYWHERE

[Twitter \(@pcmag\)](#)
[TweetMixx](#)
[Facebook](#)
[YouTube](#)
[Mixx](#)
[Google Buzz](#)

BUSINESS

[Whitepapers](#)

Use of this site is governed by our [Terms of Use](#) and [Privacy Policy](#)

Copyright © 1996-2010 Ziff Davis, Inc. All Rights Reserved. PC Magazine is a registered trademark of Ziff Davis, Inc. Reproduction in whole or in part in any form or medium without express written permission of Ziff Davis, Inc. is prohibited.



[Contact Us](#)
[Ziff Davis, Inc.](#)
[Link to Us](#)



Explore
^

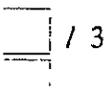
■ Documents

- Books - Fiction
- Books - Non-fiction
- Health & Medicine
- Brochures/Catalogs
- Government Docs
- How-To Guides/Manuals
- Magazines/Newspapers
- Recipes/Menus
- School Work
- + all categories
-
- Featured
- Recent

👤 People

- Authors
- Students
- Researchers
- Publishers
- Government & Nonprofits
- Businesses
- Musicians
- Artists & Designers
- Teachers
- + all categories
-
- Most Followed
- Popular

- Sign Up
- |
- Log In



/ 3

ment for Free

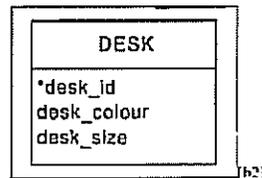
RS732

Crow's Foot Notation

A number of data modeling techniques are being used today. One of the most common is the entity relationship diagram (ERD). Several ERD notations are available. For the purposes of CS270 we will be using Crow's Foot Notation.

Components used in the creation of an ERD:

Entity – A person, place or thing about which we want to collect and store multiple instances of data. It has a name, which is a noun, and attributes which describe the data we are interested in storing. It also has an identifier, which uniquely identifies one instance of an entity. The attribute which acts as the identifier is marked with an asterisk.

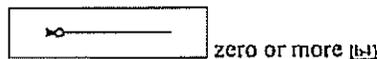


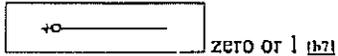
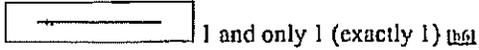
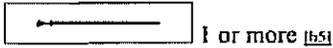
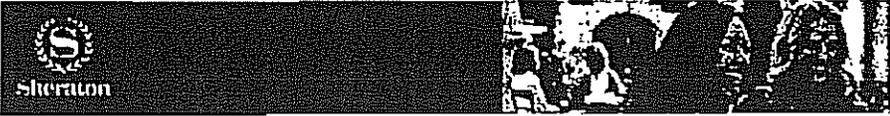
Relationship – Illustrates an association between two entities. It has a name which is a verb. It also has cardinality and modality.



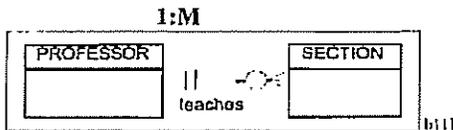
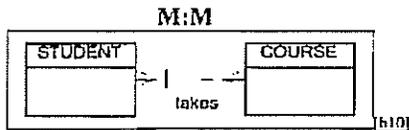
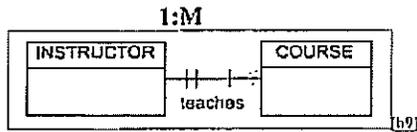
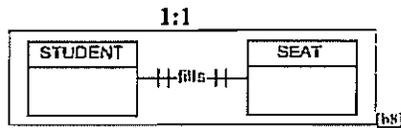
Cardinality and Modality are the indicators of the business rules around a relationship. **Cardinality** refers to the maximum number of times an instance in one entity can be associated with instances in the related entity. **Modality** refers to the minimum number of times an instance in one entity can be associated with an instance in the related entity.

Cardinality can be 1 or Many and the symbol is placed on the outside ends of the relationship line, closest to the entity. Modality can be 1 or 0 and the symbol is placed on the inside, next to the cardinality symbol. For a cardinality of 1 a straight line is drawn. For a cardinality of Many a foot with three toes is drawn. For a modality of 1 a straight line is drawn. For a modality of 0 a circle is drawn.



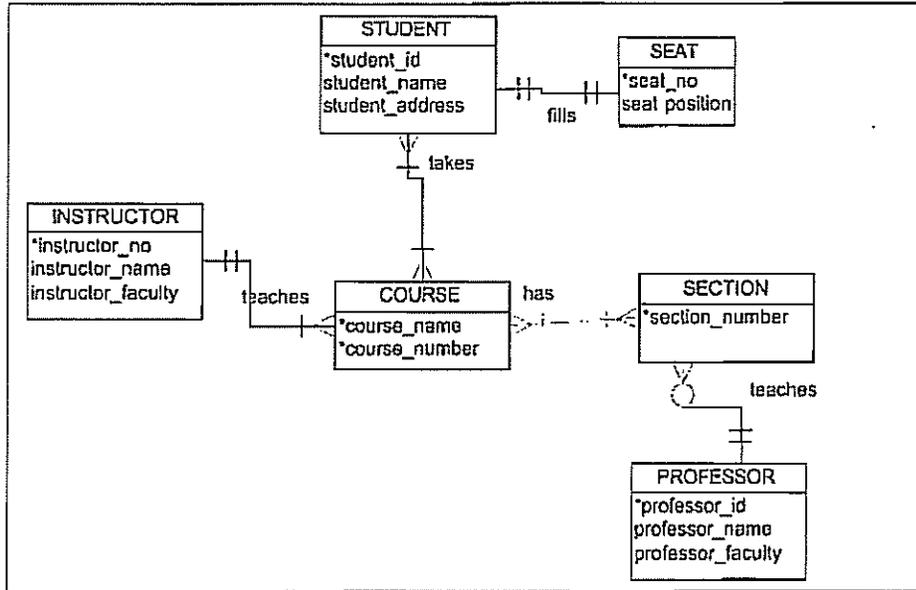


Cardinality and modality are indicated at both ends of the relationship line. Once this has been done, the relationships are read as being 1 to 1 (1:1), 1 to many (1:M), or many to many (M:M).



Typically, ERDs are much more complex than this, involving quite a number of entities and relationships. If we join all of the above relationships together and

add a few attributes, a small collection of data might be depicted in the following way using Crow's Foot Notation:



Ads by Google

Database Monitoring Tool

Monitor performance and uptime.
View reports and alerts. Try Now!
www.manageengine.com

Invention Suite - PreSys

Advanced FEA modeling software.
Highly configurable & user-
friendly
www.eta.com

Wells Fargo® Student Loan

It's Quick, Easy, and Free to
Apply
No Payments While You're In
School.
www.WellsFargo.com/student

ER Diagram convention

Download this Document for FreePrintMobileCollectionsReport Document

This is a private document. 

Info and Rating

anirelles

Follow

Ads by Google

Top Technology Degrees

Earn Your Associates in Technology
Online. Accredited Schools Only.
EducationDegreeSource.com

Ods Database

Learn How Best-In-Class Companies
Improve Sales With Quality Data
www.Jigsaw.com

SQL Server 2005 Tool

Optimize SQL Server, Save on
Costs,
Immediate ROI, Free Trial
Download!
www.Confio.com/SQL-Server

Database Federation

Standard Database Interfaces
Unlocks data from any source
www.attunity.com/data-federation

Database Software

-Describe Your Database Needs and
Receive Free Quotes from Vendors!
www.BuyerZone.com/Database_Design

ment for Free

RS736

Sign Up for an Ad-Free Scribd

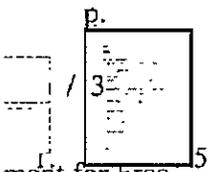
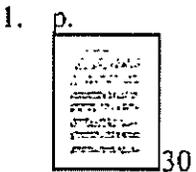
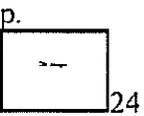
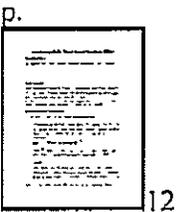
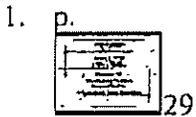
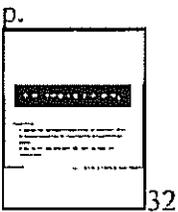
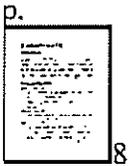
- Remove all ads.
Never see ads on Scribd again.

No Thanks

Share & Embed

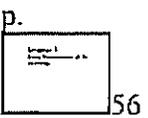
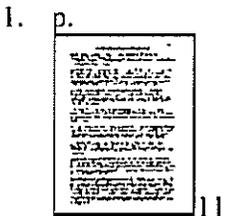
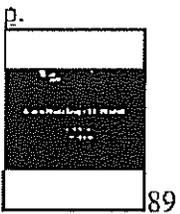
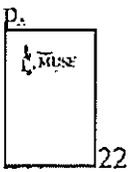
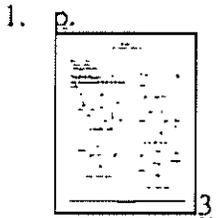
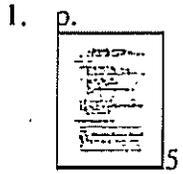
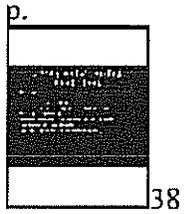
Related Documents

PreviousNext



ment for Free

RS737



1/3

ment for Free

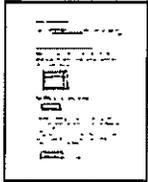


42

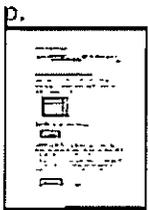
1. p.

More from this user

PreviousNext



3



3

1. p.

Add a Comment

Empty comment box with a dashed border and a small icon in the top right corner.

Submit

Ads by Google

Common Data Model

Provide a Common Data Model for SOA-Based Applications. web.progress.com/datatend

Entity Relationship Model

Run for Free! Windows, Linux, OSX

RS738

Diagram Databases and ER Models
www.aquafold.com

Convention

Ten Reasons to Hold Your Meeting
in
Las Vegas. Promote Your Event
Here.
VegasMeansBusiness.com

Database Monitoring Tool

Monitor performance and uptime.
View reports and alerts. Try Now!
www.manageengine.com

Invention Suite - PreSys

Advanced FEA modeling software.
Highly configurable & user-
friendly
www.eta.com

Wells Fargo® Student Loan

It's Quick, Easy, and Free to
Apply
No Payments While You're In
School.
www.WellsFargo.com/student

Top Technology Degrees

Earn Your Associates in Technology
Online. Accredited Schools Only.
EducationDegreeSource.com

Ods Database

Learn How Best-In-Class Companies
Improve Sales With Quality Data
www.Jigsaw.com

Upload a Document

- About
- Press
- Blog
- Partners
- Branded Reader
- Web Stuff
- Scribd Store

- Support
- FAQ
- Developers / API
- Jobs
- Terms - General
- Copyright

- Follow Us!
 - scribd.com/scribd
 - twitter.com/scribd
 - facebook.com/scribd
- ment for Free

RS739

Oracle Designer

ID #	Questions	Answers
1	When a new version is released, how can I find out about what has changed since the last version?	If you need to know what's different in a new release, you can look under the "What's New" topics in each help system.
2	How can I tell which repository elements have associations?	An easy way to tell which elements have associations is to look at the Usages node under any element definition. You can also start a new matrix diagram and select one element in the row area. The associated elements will appear in the column area on the right. This means that these two elements have an association.
3	How can I edit objects in a diagram?	You can often apply in-place editing techniques to the objects in a diagram. For example, you can click once on the entity name in an Entity Relationship Diagram. When you click again, a field will open where you can edit the value without having to open the Property Palette or property dialog. Click outside the field to close it when you are done.
4	How can I get a set of objects to fill the screen?	Use "Fit to Area" to get a particular set of objects to fill the screen. You may not be able to do this as easily with the granularity of Zoom In and Zoom Out.
5	How can I save preferences between Designer Diagrammer or utility sessions?	Click the Save button in the Preferences dialog to hold these preferences between sessions of the diagrammer or utility. If you set preferences and do not save them, those preferences will not be there when you reopen that tool. Also, the preferences are saved for each tool individually, so you may have to make the same choice in several tools.
6	The menu item or toolbar that I want to use isn't enabled. What's wrong?	If a menu item or toolbar button you need to use is not enabled, check that you have selected the proper element. You can use the "What's This" button to get help on a menu item or button as described above. This should tell you what needs to be selected before that menu item or button is enabled. Similarly, if there is a menu missing, the cursor may not have focus on the correct window. For example, in Design Editor, if you click on the Navigator window, a Navigator menu will appear, but if you click the Property Palette window, the Navigator window is replaced by a Properties menu.
7	Can I move the Entity Relationship Diagrammer toolbars to make working with them easier?	Although the default location of these toolbars is one on top of the other, you can actually drag and drop them anywhere. If you click and hold the mouse button on a non-button region within the outline of the toolbar, you can reposition the toolbar anywhere on the screen. If you drop the toolbar in the toolbar area at the top or side of the window, it will attach to the MDI window itself. If you drop it outside the toolbar, it will become a floating toolbar that you can move around inside or outside the window. This sort of toolbar is handy for maximizing the space in the drawing area because moving the toolbar out of its normal location leaves that much more area for the drawing.
8	What is meta-data?	Keep in mind that meta-data is not the actual data structure. For example, table definitions in the repository are not actual tables in the database. Rather, they only contain the information needed to create actual tables that may or may not exist in the database. The repository stores the table definition and its property values, but no table exists in the database until you run a CREATE TABLE statement to create it. This is an important, but not necessarily obvious, concept.
9	What is the difference between Edit/Cut and Edit/Delete in the ER Diagrammer?	Edit→Cut removes the element from the diagram but not from the repository. Edit→Delete (or pressing the DEL key) removes the element both from the diagram and from the repository. When you delete an entity, Oracle Designer also deletes all attributes and attached relationships.
10	How can I set the default icons used by Process Steps, Stores, and Flows?	You can set the default icons used by Process Steps, Stores, and Flows from the Options→Customize→Advanced menu item. You can also set up default icons based on various words in the name of the object. For example, if you include the word "fax" in the process step name, the Process Modeller can use a certain icon. If you include the word "create" in the process step name, it can use a different icon. You do this by creating a file that contains the key words and file names. For example: <pre>create factory1.bmp fax fax1.bmp</pre> You specify the name of this file in the same Customize-Advanced dialog. If you do not include the path, the tool looks for the file in the Windows directory or the root directory where Windows is located. This is a fast way to attach various icons without having to visit the Multimedia tab for each one.
11	In the ER Diagrammer, does the order that the relationship is drawn make a difference?	When you create a relationship, you need to draw the relationship in the order it appears on the button. Therefore, if you choose the >--- (many-to-one) relationship from the toolbar, the first entity you drop it on will be the "many" side, and the second will be the "one" side.
12	How can I easily make the relationship lines consistent in my ERD's?	Move the entities around on the diagram to the proper position without worrying about how the relationship lines wrap. When the entities are in the right place, select one relationship line, choose Edit→Select Same Type from the menu, and click the Autolayout button. This will perform an autolayout on the relationship lines alone. Be sure to Select Same Type not Select All or you could undo the careful layout you just completed.
13	What is the best way to correctly line up my diagrams?	Modify the grid size and turn Snap on using Options→Customize. Then the objects you place and lines you draw will "snap" to the grid lines (invisible or not), and you will find creating straight lines

		much easier. The Customize option also lets you display the grid, which will help you line up entities and relationships. The grid will print if it is displayed when you start printing the diagram.
14	I can't find an object I created. What's wrong?	Be sure to expand the node you think the object will be in, or Find will not locate the object, unless the node in which the object exists has been expanded at some time in your session. Once the parent node has been expanded, even if it is collapsed when you run the search, the Find command will work properly. When you run subsequent searches, RON will start a prefix search, trying to match the item as you are still typing it.
15	Why can't I see the vertical scrollbar in the Repository Object Navigator?	The vertical scrollbar may be partially hidden if your Navigator window is too small. This might even be true if your window is maximized but your monitor is set to a low resolution (under 1024x768). You can move the Navigator window up and gradually manipulate it to reveal the buttons or turn off the display of the top toolbar buttons and status line (in the View menu), but the long-term solution is a higher resolution.
16	How can I include a certain subset of elements on a particular report even if the elements do not have a similar name?	Some reports include a parameter for the name of a diagram on which the elements appear. For example, the Entity Definition report includes an ER Diagram parameter, which lists the existing diagrams. If you select one, Oracle Designer uses it as the source for the entities it reports on. Thus, suppose a diagram called ERD PERSON contains the entities PERSON, ORGANIZATION, and PURCHASE ORDER. You can fill in the name of the diagram in the ER Diagram parameter, and the report will include only the PERSON, ORGANIZATION, and PURCHASE ORDER. The wildcard % works here too, so if you have similarly named diagrams and specify the wildcard name in this parameter, Oracle Designer will report on all entities on all diagrams with names that match the parameter.
17	How can I change the name of a user-defined repository report?	You can edit the name of a user-defined report by clicking on its name in the Navigator and editing the existing text.
18	How can I capture the design of an object that already exists in the repository?	You can capture the design of an object that already exists in the repository. If you do that, the utility will add any columns that are not in the repository that are in the database. It will not modify columns that already exist even if they have different datatypes or sizes. It will also not drop columns from the repository that no longer exist in the database. To reconcile the database definitions with the repository definitions, run the Generate Database from Server Model item in the Generate menu of the Design Editor. Specify that you want the target to be a Database. When you run this utility, it will generate a script with DDL ALTER statements as well as a reconcile report (described in Chapter 20) that you can view from dialogs in the utility. These files will show you the differences between the repository and the database so you can determine how to synchronize them.
19	How should I design capture a large number of objects?	If the number of objects to design-capture is large, you may find it easier to run the Capture Design of Server Model from Database Utility more than once with subsets of objects. If the objects are owned by different schemas (users), you will have to run the utility once for each user.
20	How can I easily create multiple elements of the same type in the Repository Object Navigator?	If you need to create multiple elements of the same type in the Repository Object Navigator, use the Fast Create dialog in the Design Editor's Edit menu. This is a way to get a list of definitions sketched out in the repository quickly. You call up this utility after selecting the Modules node, and it will give you a list to fill in of module names and languages. This is the minimum amount of information needed to create a module. After you press the OK button, the utility will create the definitions, and you can fill in the details.
21	The button for the action that I want to perform is dimmed? What can I do?	In all Oracle Designer diagrammers, if the button or menu choice you wish to use is disabled (dimmed), you have not performed the prerequisites for that task. Usually, all the diagrammer requires is for you to select an object on the drawing, but you may have to do something else as well. The quickest way to find out what to do is to press the Context-Sensitive Help button and then click the help cursor on the button (or select the menu item) in . A help screen with information on that particular activity will appear.
22	Can I base a domain on another domain?	Domains can be based on domains. The Definition tab of the Domains dialog has a <i>Supertype</i> property, which allows you to specify an existing domain name that is the parent of the one you are defining. This documents that the domain is a subset of another domain and has characteristics in common with it. The <i>Supertype</i> property is used for documentation only; you have to specified the subset of allowable values in both the child and parent domain.
23	How can I quickly change the cardinality, optionality, or transferability of a relationship?	You can quickly change the cardinality, optionality, or transferability of a relationship by right-clicking on the end you want to change and selecting from the popup menu.
24	How long should entity names be?	Although a short name for an entity can contain up to ten characters, it's better to use seven or fewer, especially if you'll be generating Oracle Developer forms from your application. The Database Design Transformer uses the short name as the table's <i>Alias</i> property, which eventually becomes the block name in a generated form, and forms coding is easier if the block names are short.
25	How can I pin a button drawing?	All diagrammers in Oracle Designer allow you to pin a drawing button so you do not have to reselect it if you are drawing multiple objects of the same type. If you want to draw three functions, for example, you can hold down SHIFT when you click the function button. This pins the button so you can draw as

		many functions as you want without having to press the function button before drawing each one. When you want to stop placing functions, just click the Select toolbar button.
26	I tried to perform a Design Capture but it failed. What's wrong?	You should be sure the definitions exist in the repository for the tables and columns used by the form or report you are design capturing. Otherwise, the Design Capture utility will fail.
27	I removed some filters in a previous Designer session but they were reactivated when I restarted the application. What happened?	If you remove filters that have previously been saved as the default, the next time you open the Design Editor, the filters will be in place. To remove the filters permanently, clear the filters, select Options→Customize, and check the Save Settings as the Default checkbox in the dialog.
28	Why is it important to start the Database Design Transformer from the Repository Object Navigator (RON)?	Start the Database Design Transformer from RON, because if you select entities in the RON Navigator, those entities will automatically have the <i>In Set</i> box checked when DDT starts. This could save a significant amount of time if you are running the utility for a subset of the entities. Starting DDT from RON is also recommended when you are just getting started with Oracle Designer or with a new team of developers, so you can check the output of DDT and make corrections to the analysis-level objects before all the design-level objects are created.
29	How can I see more characters in my table name columns in the Database Design Transformer?	In the Table Mappings tab, you can reorder and resize the columns if you need to see more characters. For example, the values in the table name column may not be totally visible, but you can move the column to the left by dragging and dropping the Table column heading. You can then resize the column by dragging the side of the heading left or right. The Database Design Transformer remembers the new layout the next time it's used.
30	I need to restart the Application Design Transformer but I've already created some modules. What do I need to do?	If you need to start over with the Application Design Transformer (ADT) but have already created modules, you must delete the modules first or you will get naming conflict errors when you run the utility again. The fastest way to delete a group of modules is to select them in RON or DE and use Utilities→Force Delete. If you simply try to delete the modules, you might have to run the delete process a number of times, as the delete will fail on modules that are called by other modules.
31	How can I assign different preference sets to different levels?	While you cannot assign a preference set to another preference set, you can assign different preference sets to different levels. For example, if you have PREFERENCE SET 1 that you assign to MODULE1, all module components, items, item groups, and constraint usages under MODULE1 will inherit that preference set. If you want to apply PREFERENCE SET 2 to that module as well, you could attach it to all module components for that module. In effect, you would have two active preference sets governed by the precedence and inheritance rules.
32	How can I set up the Design Editor to display both dialog and Property Palette styles of dialogs?	If you set the default for properties to the Property Palette (using the toolbar button or Options menu item) and display the properties, you will see the palette window open. If you then switch to Property Dialogs (button or menu item), when you double-click an object in the Navigator, the dialog will appear, even though the palette window is still open. Once you dismiss the dialog, you can set any properties in the Property Palette that were not in the dialog. For example, you prefer the dialogs but can only set text properties like <i>Description</i> or <i>Notes</i> in the palette, so you use the dialog to do what is possible there and enter the text using the Property Palette.
33	What is the best way to do table-level validation on column values?	Database procedures called from database triggers are often a better way to do table-level validation on column values than using check constraints. You can easily perform most validation with a check constraint because it can use a database function as part of its SQL expression. However, check constraints are not a <i>normal</i> place to look for code; if someone tries to determine where validation is firing for a column, she or he may not find that code easily. In addition, you can do operations in a procedure that might not be possible with a function that has SQL restrictions. For example, writing a record to an audit table if validation failed is straightforward in a database procedure called by a trigger but may not be possible using a function in a check constraint.
34	When should I use the Fast Create utility?	You can enter a number of definitions for a particular type very quickly with the Fast Create utility. Run this from the Edit→Fast Create menu item after selecting an element type in the Navigator. You will be able to enter a number of definitions at once. The only property you fill in for this dialog is the <i>Name</i> , so you will need to complete the definition later, but this will give you a starting point.
35	How can I create foreign key constraints in the Repository Object Navigator?	You can create foreign key constraints by dragging and dropping in the Navigator. Drag the primary key table's node and drop it on the foreign key table's node. This will create a new foreign key in the table that was the target of the drop. You still have to assign columns to this constraint, but the constraint will be created. This drag-and-drop facility can be a problem if you accidentally drop a table on another table when you meant to drop it on a free space to create a diagram. While you can always delete the foreign key you accidentally create, it is best to exercise caution when performing drag and drop with tables. Another tip for using foreign keys in RON: click on the blue arrow to the left of the foreign key name in the Navigator. The selection will jump to the table definition of the table to which the foreign key points.
36	Besides the Help system, where can I find information about what a particular property of a database object is intended for?	Check the Oracle server documentation, which is optionally installed with the server software. This is a set of HTML files (for Oracle versions 7 or 8) that contains all manuals describing the server. The HTML files are searchable and have sample syntax and descriptions of all clauses. A good document to start with is the "SQL Language Reference," since most of the objects you are creating from

		repository definitions will produce CREATE syntax.
37	The package installed by my Table API script does not compile correctly. What's wrong?	Be sure you have the CG\$ERRORS package installed or accessible. The Table API procedures call this package, but it may not be installed into your schema. This is documented in the help system, but you may get the error before you see the note in the help system. You can install it by running the CDSAPER.PKS and CDSAPER.PKB scripts in SQL*Plus from the ORACLE_HOME/CGEN\$70/SQL directory.
38	How can I create Type Model definitions from Server Model definitions?	There is no utility to create Type Model definitions from Server Model definitions, but you can create DDL scripts from the Server Model using the Generate Database from Server Model utility. You can then use these DDL scripts as a source for the Capture Design of Type Model from Database. This will effectively create Type Model definitions from objects in the Server Model.
39	What can I do if my drawing contains a stray pixel or two of a line that remained when something was moved or when I scrolled the window?	Since there is no redraw action in any of Oracle Designer's diagrammers, you have to force the tool to perform a redraw operation. The easiest way to do this is to use the window icons in the top-right part of the window to minimize the drawing window (not the MDI window) and then maximize it again.
40	How do I move a pig's ear (self-referencing) foreign key line?	To move a pig's ear (self-referencing) foreign key line, click it to select it and move the whole line by clicking the middle of the arc (not the end). You can drop it close to a corner, and the line will span the two adjacent sides.
41	After initially laying out the existing objects, how can I reduce the number of blank pages?	Select Layout Minimize Number of Pages to reduce the number of blank pages. Remember that you can change the page orientation in the File Print Setup dialog box at any time if you prefer to use landscape mode.
42	How can I find out ahead of time what DDL statement or clause a particular property will generate?	The key to success in the Server Model Diagram (indeed, in all of Oracle Designer) is knowing what the result will be when you set or ignore a particular property. When in doubt regarding what DDL statement or clause a particular property will generate, create a test table or element and run the generator before and after specifying the setting to note the differences in the script that the Generate Database from Server Model utility creates. The extra time it takes to learn the properties will benefit you in the long run.
43	How can I save a server model diagram to a file?	<p>Several solutions were suggested:</p> <p>A. From Alex Haratampiev, Ph.D, M.Eng, CNA, CCP Nationwide Financial Services My experience with copy/paste is slightly different from the earlier suggested ones:</p> <ol style="list-style-type: none"> 1. Select all elements in the Design Editor diagram 2. Ctrl-C for copy to clipboard 3. Switch to blank document in MS-Word'97 4. Go to Edit Paste Special (don't do direct Ctrl-V) 5. In the pop-up box there are three choices; select 'Picture' 6. The result is diagram in WMF format that can be resized without loss of resolution (or even edited in Word) <p>The main difference in my procedure are steps 4 and 5; I never had success pasting directly the diagram as an OLE object (which is the default behavior of Paste in this case). Even on machines with lot's of memory I was getting message about "Insufficient Memory" and only the names of the tables in the diagram are pasted as a paragraph of text. (One additional exception is the "Legend", which shows as selected in step 1. is never transferred to the picture in Word) These results are from server installation of Designer (R2.1.2 and R6.0) and Windows NT clients (PIII 400Mhz, 128M RAM)</p> <p>B. From Paul Sunners Screen captures will only give you what's displayed on the screen, if your diagram doesn't fit then you'll have problems. You can use copy/paste instead (I think this was mentioned a few days back on this list). Just make sure that nothing is selected in the diagram, then select Edit/Copy, switch to Word and Paste. You can then resize the image to fit the page.</p> <p>C. From Paul Dorsey An alternative is to use PDFs. To do this you will have to a) buy the full Adobe Acrobat, which comes with a PDF printer driver that allows you to "print" from any application to a PDF file, or b) use Ghostscript (a freeware program that can convert from PostScript files to PDF). To use Ghostscript, you simply set up a "dummy" printer in Windows using the Postscript driver, and set the destination to FILE instead of a port. The resulting Postscript file can then be passed through</p>

RS743

		<p>Ghostscript and converted to a PDF. You can download Ghostscript from http://www.cs.wisc.edu/~ghost/</p> <p>D. From Alex Campbell Download a shareware copy of Snagit from www.snagit.com. Great screen captures and you can save them to BMP(Paintbrush), etc.</p>
44	<p>How do I get started with Oracle Designer?</p>	<p>Training is NOT enough to get you started. This product has a VERY large learning curve. If you get training, it will only teach you product features.</p> <p>The Oracle Designer Handbook (2nd Edition) that Peter Koletzke and I and wrote tried to address how to use the product, but I would still not assert that it is a complete answer to the question. It is a very big question.</p> <p>Without a mentor, your first project (or 5) will be a throw away learning exercise that will cost you more in frustration and pain than any benefits you will receive from using Designer. There are lots of ways to use the product. How to best use it depends on your available talent and your specific requirements. Only a mentor can really help to guide you.</p> <p>You should also be aware that there is a sizable contingent that has abandoned using whole parts of the product. Probably the only parts of the product that everyone agrees are useful are the ER/database diagrammers that can be used to design and generate your tables.</p> <p>In terms of finding a mentor, there are not a lot of people whose opinions I would trust to help guide you in joining the wide world of Designer. Try to contact one of the senior people on the ODTUG forum and try to get their help. If you want to do it locally, talk to your user group president and ask who they would suggest to provide help (you want the best Designer person in your area). Then bring in that person for 2-3 days to meet with you and your team and help you to decide on your next steps.</p>
	<p>Question submitted by Kent Graziano</p> <p>In your conclusions you state as one of the reasons to not use generation: "...if your team is relatively inexperienced and you are trying to train new people, generating applications is not likely to produce satisfactory results. Using Designer successfully requires a huge learning curve."</p> <p>No argument there (on the learning curve issue). Given that, what approach to application building (in the Oracle world) do you advocate that does NOT have a huge learning curve and can be used by an inexperienced team to produce satisfactory results?</p>	<p>Question submitted by Kent Graziano</p> <p>In your conclusions you state as one of the reasons to not use generation: "...if your team is relatively inexperienced and you are trying to train new people, generating applications is not likely to produce satisfactory results. Using Designer successfully requires a huge learning curve."</p> <p>No argument there (on the learning curve issue). Given that, what approach to application building (in the Oracle world) do you advocate that does NOT have a huge learning curve and can be used by an inexperienced team to produce satisfactory results?</p> <p>This is an empirical question. I submit that in the existing projects, there is no more consistency in Designer-generated apps than in hand coded apps. This is particularly true in shops that are less experienced. The norm is to generate and break the link by doing post generation modifications. Such shops clearly have as much (if not more) non-maintainability in their applications as shops that have hand-built their applications.</p> <p>It really comes down to standards and standards enforcement. This is true for either environment. If you don't have solid, enforced standards, you will have chaos. It doesn't matter which environment you use.</p> <p>Even in the rules based environment that I now use, there is the ability to ignore the environment and place hard code the rules into the applications (I can't PREVENT someone from being stupid and doing that).</p> <p>To paraphrase the gun lobby "Software doesn't enforce standards; people enforce standards." However, we all know that papers on project management and standards enforcement are usually ignored. The comment that will forever make me roll my eyes (even in memory) was when Peter Koletzke and I were criticized by a reader on our first Designer/2000 Handbook as follows: "Why do you have all this nonsense about building a system in your book, I KNOW how to build systems - just show me what I have to click on to use the tool." And we wonder at the high failure rate of projects.</p>

Oracle Designer FAQs
Oracle Developer FAQs
Oracle RDBMS FAQs
PL/SQL FAQs
Data Modeling FAQs
JDeveloper FAQs
BRIM[®] FAQs
Business Rules FAQs

[Oracle Consulting Services](#) | [Application Development](#) | [Custom Training](#) | [Data Migration](#) | [Data Warehousing](#)

[About Dulcian](#) | [Papers & Presentations](#) | [Related Links](#)

[Publications](#) | [Employment Opportunities](#) | [Products](#) | [Home](#) | [Contact Us](#)

©2000 Dulcian, Inc.

Database Management Systems II

Entity Relationship Modeling and Normalization

What You'll Learn This Week

- Introduction to the course
- Background Materials
- Review of Entity-Relationship Model
 - Set of E-R modeling constructs
 - Entities
 - Attributes
 - Identifiers
 - Relationships
- E-R Diagrams
 - Variation One - What the Kronke book uses
 - Variation Two - Chen notation as used in the Elmasri/Navathe Textbook
 - Variation Three - Oracle Designer CASE
 - Variation Four - Visible Analyst
 - Variation Five - Sybase PowerDesigner
 - Variation Six - Popkin System Architect
 - Variation Seven - Popkin System Architect - UML Notation
 - Displaying Attributes
 - Weak Entities and ID Dependent Entities
 - Subtype Entities
- Entity to Relation Conversion

Introduction to the course

- Syllabus
- Computer Labs and Software
- Course Logistics

Background Material

You may wish to briefly review Chapters 2, 10, 11, 12, 13 in the Connolly/Begg textbook (Chapters 1 through 5 in the Elmasri/Navathe book) to refresh your memory of basic database concepts. *You are responsible for this material.*

Our goal is to provide *information* for the purposes of decision making in business. The database is one way to provide such information.

However, we must organize and develop a database in a structured fashion to meet the needs of an organization and reflect the way it operates.

Recall the database design process we follow:

1. Gather user/business requirements.
2. Develop the Entity Relationship (E-R) Model (shown as an E-R Diagram) based on the user/business requirements. The E-R model is the *conceptual model* of the database.
3. Convert the E-R Model to a set of relations in the relational model. We call this the *Logical model*
4. Normalize the relations to remove any anomalies.
5. Implement the database schema by creating a table for each normalized relation. We call this the *Physical model*
6. Develop applications (forms, reports, queries, scripts, procedures) that work with these tables.

Note that in the second step, we may also develop additional models such as data flow and functional models.

Review of Entity Relationship Model

Elmasri/Navathe 3rd ed.	Kroenke 7th Ed.	McFadden 5th ed.	Connolly/Begg 3rd ed.
Chapters 3 and 4	Chapter 3	Chapters 3 and 4	2, 10, 11, 12

- Entity Relationship Modeling: A Set of *constructs* used to interpret, specify and document data requirements for database processing systems.
- E-R Models are *Conceptual Models* of the system. They can not be directly implemented in a database.
- Many variations of E-R Modeling used in practice.
- Mainly differences in notation, symbols used to represent the constructs.

E-R Modeling Constructs

RS746

- E-R Modeling Constructs are: Entity, Relationship, Attributes, Identifiers
- It is important to get used to this terminology and to be able to use it at the appropriate time. For example, in the ER Model, we do not refer to *tables*. Here we call them *entities*.
- Entity: Some identifiable object relevant to the system being built. Examples of Entities are:
 - EMPLOYEE
 - CUSTOMER
 - ORGANIZATION
 - PART
 - INGREDIENT
 - PURCHASE ORDER
 - CUSTOMER ORDER
 - PRODUCT

An *instance* of an entity is like a specific example:
 Bill Gates is an Employee of Microsoft
 SPAM is a Product
 Greenpeace is an Organization
- Attribute: A characteristic of an Entity. Properties used to distinguish one entity instance from another. Attributes of entity EMPLOYEE might include:
 - EmployeeID
 - Social Security Number
 - First Name
 - Last Name
 - Street Address
 - City
 - State
 - ZipCode
 - Date Hired
 - Health Benefits Plan
- Identifier: A special attribute used to identify a specific instance of an entity.
 - Typically we look for *unique* identifiers:
 - Social Security Number uniquely identifies an EMPLOYEE
 - CustomerID uniquely identifies a CUSTOMER
 - We can also use two attributes to indicate an identifier: ORDER_NUMBER and LINE_ITEM uniquely identify an item on an order.
- Relationship: An association between two entities.
 - A CUSTOMER *places* a CUSTOMER ORDER
 - An EMPLOYEE *takes* a CUSTOMER ORDER
 - A STUDENT *enrolls* in a COURSE
 - A COURSE is *taught* by a FACULTY MEMBER
 - Relationships are typically given names.
 - A relationship can include one or more entities
 - The *degree* of a relationship is the number of Entities that participate in the relationship.
 - Relationships of degree 2 are called *binary relationships*. Most relationships in databases are binary.
 - Relationship Cardinality refers to the number of entity instances involved in the relationship. For example:
 - one* CUSTOMER may place *many* CUSTOMER ORDERS
 - many* STUDENTS may sign up for *many* CLASSES
 - one* EMPLOYEE receives *one* PAYCHECK

1:N "One to Many"

N:M "Many to Many"

1:1 "One to One"

Beware of 1:1 relationships. The two entities involved might be coalesced into one. Also called HAS-A relationship.

Beware of N:M relationships. Typically split these into two 1:N relationships with an intersection entity.

 - *Participation* of instances in a relationship may be mandatory or optional.
 - For example,
 - one* CUSTOMER *may* place *many* CUSTOMER ORDERS
 - one* EMPLOYEE *must* fill out *one* or more PAY SHEETS
 - This is also called "minimal cardinality" or the "optionality" of a relationship.

E-R Diagrams

- The most common way to represent the E-R constructs is by using a diagram
- There are a wide variety of notations for E-R Diagrams. Most of the differences concern how relationships are specified and how attributes are shown.
- In almost all variations, entities are depicted as rectangles with either pointed or rounded corners. The entity name appears inside.
- Relationships can be displayed as diamonds (see below) or can be simply line segments between two entities.
- For Relationships, need to convey: Relationship name, degree, cardinality, optionality (minimal cardinality)

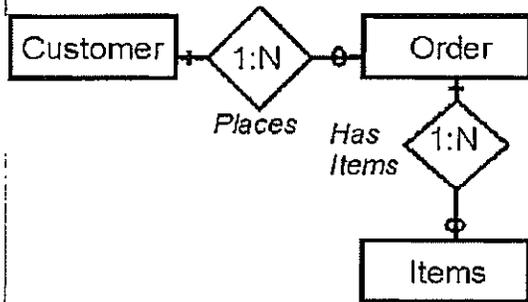
RS747

- Here we will give examples several variations. The point is not that you memorize all of these variations. Pick one you are most comfortable with and use it consistently.

Kronke textbook Notation

(NOTE: We no longer use this notation so you can safely ignore this example)

- Relationship Name: Displayed just outside of the relationship diamond.
- Degree: Shown by line segments between the relationship diamond and 2 or more entities.
- Cardinality: Displayed inside the relationship diamond.
- Optionality: Mandatory participation indicated by an intersecting hush mark made perpendicular to the relationship line segment. Optional participation indicated by a 0 intersecting the relationship line segment.

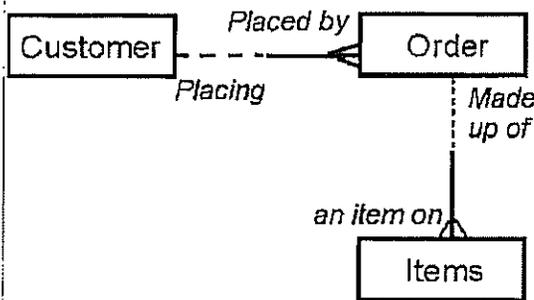


For this diagram:

- An ORDER must be placed by one and only one CUSTOMER.
- A CUSTOMER may place zero or more ORDERS.
- An ORDER may have zero or more ITEMS.
- An ITEM must have one and only one ORDER.

Note: This notation is rarely used if at all since the notations are essentially backwards from all other notations. Also, when written vertically, the cardinalities become ambiguous.

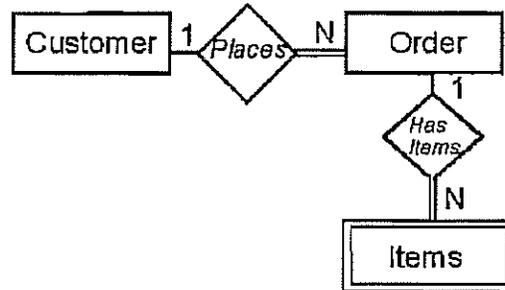
Oracle Designer CASE Notation



- Also called the "Information Engineering" notation.
- In Oracle Corporation's Designer, relationships are expressed in a rigid sentence format. For example: An ORDER must be placed by one and only one CUSTOMER. The "be" is mandatory making the verb difficult to get right.
- Relationship diamonds are not used.
- Relationship Names: Are expressed as a verb phrase starting with "be". There are two phrases, one for each direction of the relationship. This phrase is then written along the line segments for the relationship.
- Degree: Shown by line segments between any two entities. As such, ternary (3 way) relationships as described in the Kronke and Elmasri/Navathe textbooks (Chen notation) can not exist.
- Cardinality: Single participation ("1" in the previous example) is indicated by a single line segment. Multiple participation ("N") is indicated by *crow's feet*
- Optionality: Mandatory participation is indicated by a solid relationship line segment. Optional participation is indicated by a dotted line segment.
- One ORDER must be placed by one and only one CUSTOMER.

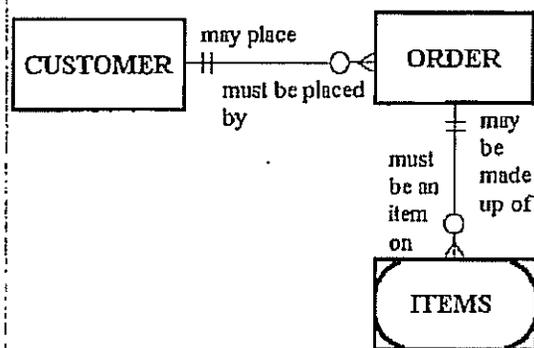
Chen notation as used in the Elmasri/Navathe Textbook

- Relationship Name: Displayed just inside the relationship diamond.
- Degree: Shown by line segments between the relationship diamond and 2 or more entities.
- Cardinality: Displayed between the participating entity and the relationship diamond next to the relationship line. Split up the cardinality.
- Optionality: Mandatory participation indicated by double relationship line. Optional participation indicated by a single relationship line.



This is also called the "Chen notation" after the author who first proposed it.

Visible Analyst Notation

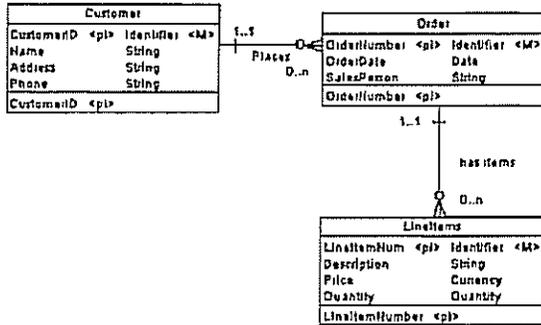


- Visible Analyst Workbench (VAW) uses the rounded box to show an *Attributive Entity* - one that depends on the existence of a *fundamental entity* (noted by just the rectangle).
- The relationships use the following symbols:
 - For cardinality, the crow's feet are used to show a "Many" side of a relationship.
 - A single line indicates a "One" side of the relationship.
 - Optional participation is shown with an open circle. Thus in the above diagram, a Customer *May* place one or more Orders.
 - Mandatory participation is shown with two hush marks. Thus in the above diagram, an Order *Must* be placed by one and only one Customer.

- One CUSTOMER may be placing zero or more ORDERS.
- One ORDER may be made up of zero or more ITEMS.
- One ITEM must be an item on one and only one ORDER.

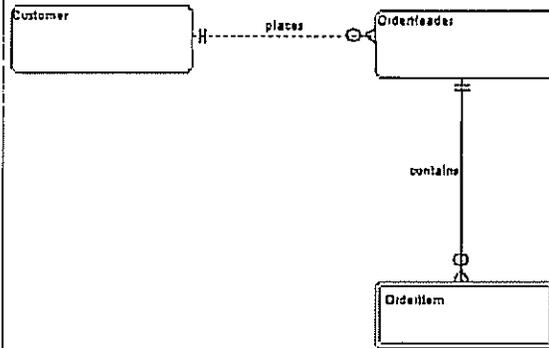
There are a set of tools within the Oracle CASE environment that can print these "relationship sentences".

Sybase PowerDesigner Notation



- Single hash mark indicates mandatory participation
- Hollow circle indicates optional participation
- Crows-foot indicates *Many* while single line indicates *One*
- Cardinalities can also be shown next to the relationship anchors such as 0..n

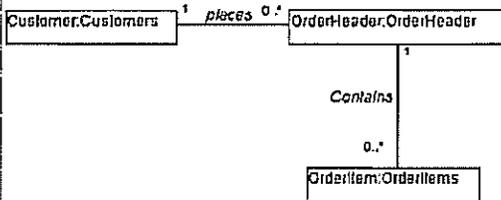
Popkin System Architect - Crow's Feet Notation



- Two hash marks indicates mandatory participation
- Hollow circle indicates optional participation
- Crows-foot indicates *Many* while single line indicates *One*
- Dashed line indicates non-associative relationship
- Solid line indicates associative relationship (e.g., that the Identifier of OrderHeader should be part of the composite identifier of Order Items)

Popkin System Architect - UML Notation

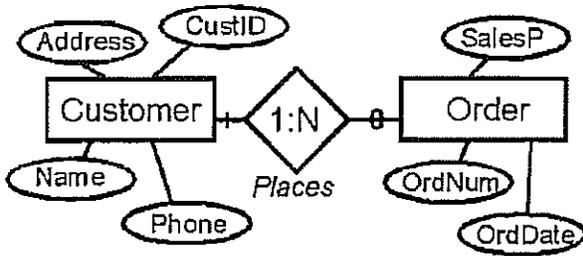
- The Unified Modeling Language (UML) has gained in popularity
- UML draws upon several different Object Oriented modeling techniques developed in the 1980's and 1990's.
- Learn more at <http://www.uml.org/>



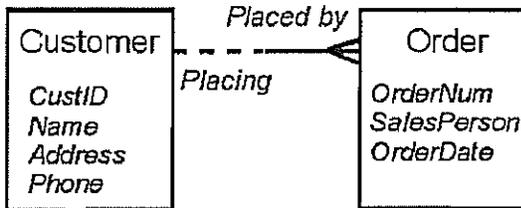
- In UML notation, entities are referred to as *classes* or *object types* and are represented as rectangles.
- Note the naming convention used: Customer is a member of the class Customers
- Associations (relationships) can be binary, ternary or higher (*n-ary*)
- Associations are shown by connecting lines with a single term as a label
- Cardinalities can be displayed (as above) e.g., 1 indicates *one and only one* 0..* indicates a minimum of 0 and a maximum of Many
- Note that only relationships of degree 1 and degree 2 are supported in UML (no ternary relationships)
- The Connolly/Begg textbook 3rd edition now uses UML throughout.

Displaying Attributes

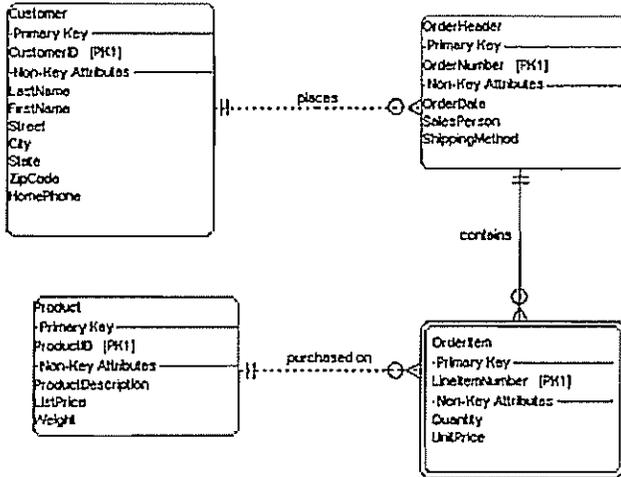
- Technically, an *Entity-Relationship* diagram should show only entities and their relationships.
- Consider: *Entity-Relationship-Attribute (ERA)* model.
- Two main ways to display attributes associated with an entity.
 1. Attributes appear in ovals attached to the entity. Gets messy.



2. List attributes inside of the entity box.



Note that some CASE and drawing tools include additional information in the entity box such as the primary keys. Example from Popkin System Architect CASE tool:



- Two additional important Notes:
 - The "Relationships" screen in MS Access is not an ER modeling tool. This screen represents a physical model of the database, not a conceptual one.
 - Be aware that many CASE tools automatically display the foreign keys in the entities on the many side of one to many relationships. This is not correct ER modeling technique as such propagation of foreign keys should be a result of the conversion to relational model and not present at the conceptual modeling stage.

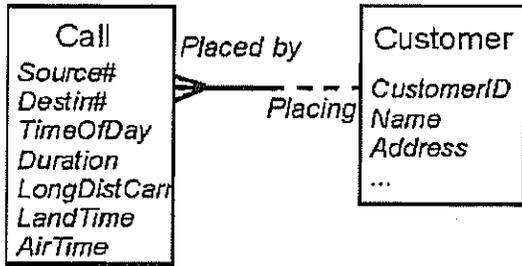
Weak Entities and ID Dependent Entities

- Vague definition: **Weak Entity:** An entity that depends on another for its existence.
- Elmasri/Navathe definition: An Entity with no identifying attributes
- ID Dependent Entity:** A weak entity that includes the identifier of the related strong entity. The weak entity depends upon the identifier from the strong entity for its existence.
- Examples of strong entities: People, Employees, Customers, Clients, Vendors, Students
Products, Services, Parts, Resources, Materials
- Weak entities are typically shown with a double-box.

Subtype Entities

- Attributes of two or more Entities may overlap significantly but not completely.
- Consider:
Phone Call (Source#, Destination#, Time of day, Duration)
LongDistance Call (Source#, Destination#, Time of day, Duration, Long distance Carrier)
Cell Phone Call (Source#, Destination#, Time of day, LandTime, AirTime)

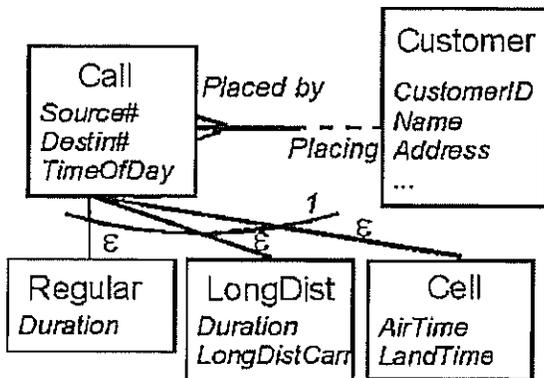
- One approach would be to put all of the attributes into a single entity.



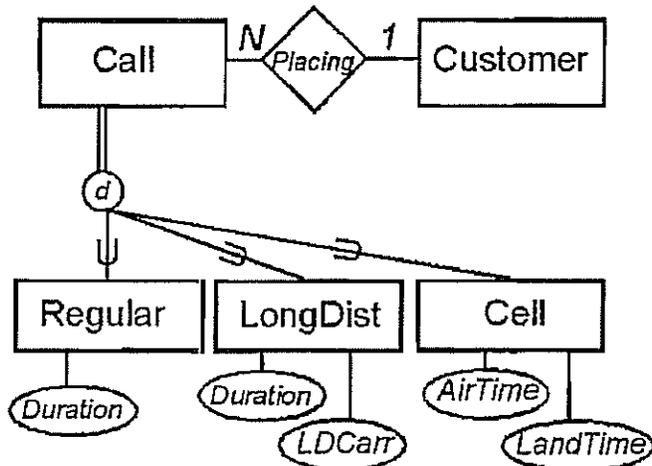
- Second approach, put common attributes into a *parent* or *supertype* entity and then have 3 *subtype* entities.
- Relationship is called an IS-A relationship.

Subtype Entity Notations

- As with ER models, there are a number of different notations for super/subtype relationships
- The following diagram uses the Oracle Designer symbols for Supertype/Subtype.



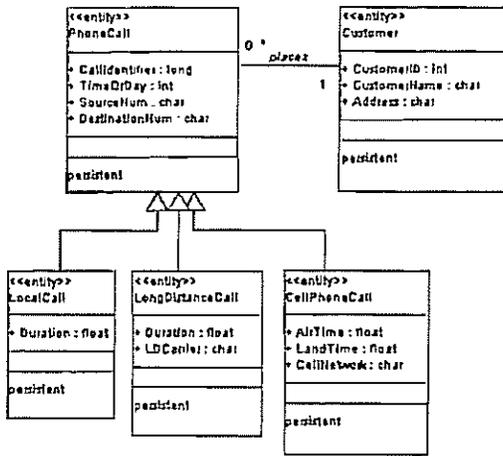
- Below is the same diagram drawn using E-R symbols from the Elmasri/Navathe book.



The *d* in the circle indicates the subtype entity is *distinct*. Only one subtype entity can participate in an instance.

As before, the double line between the Call entity and the *d* in the circle indicates the relationship is mandatory.

- In UML, we can use the superclass/subclass relationships (inheritance) as shown below:



Review of the Relational Model

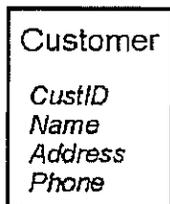
Elmasri/Navathe 3rd ed.	Kroenke 7th ed.	McFadden 5th ed.	Connolly/Begg 3rd ed.
Chapter 7	Chapter 9	Chapter 6	3, 13

- Recall, the Relational Model consists of the elements: relations, which are made up of attributes.
- A relation is a set of attributes with values for each attribute such that:
 - Each attribute value must be a single value only (atomic).
 - All values for a given attribute must be of the same type (or domain).
 - Each attribute name must be unique.
 - The order of attributes is insignificant.
 - No two rows (tuples) in a relation can be identical.
 - The order of the rows (tuples) is insignificant.
- Domain: The set of allowable values an attribute may take. Also includes the data type and size/length.
- A Key in a relation is a set of attributes that have unique values across all tuples. There may be several candidate keys in a relation.
- A Primary key is a key that uniquely identifies a tuple.
- A Foreign key is a set of attributes that act as a primary key in R1 (with their associated domains) but also appear in another relation, R2.
- Relational integrity:
 - Entity Integrity: A primary key may not contain NULL values.
 - Referential integrity: A relationship between two relations used to maintain consistency of values. The Foreign key is a mechanism to enforce referential integrity.

Entity to Relation Conversion

- For a majority of ER Models, entities and weak entities convert easily into relations.
- Entities - In general, each entity will be converted directly to a relation. The attributes of the entity become the attributes of the Relation.

The Identifier of the Entity becomes a Key of the Relation.
(not primary key - just "key")

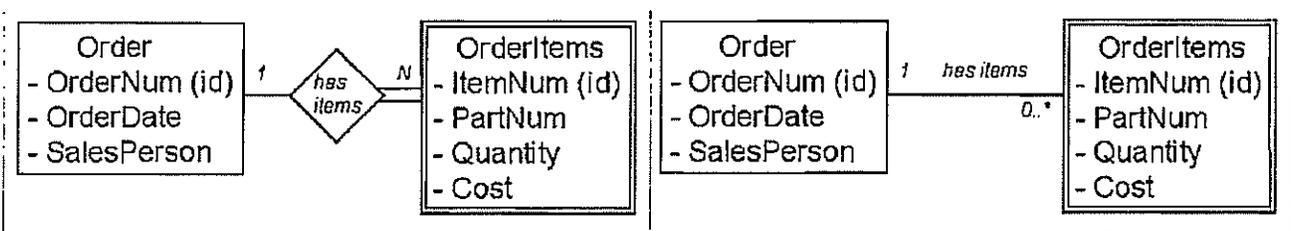


CUSTOMER (CustID (key), Name, Address, Phone)

- Weak Entities - If a weak entity is not ID Dependent, then treat it like any other entity - Application must enforce referential integrity.

If entity is ID Dependent, then the parent relation's key (Identifier) is copied into the dependent relation and is combined with the dependent relation's identifier to form a composite key.

Chen Notation	UML Notation

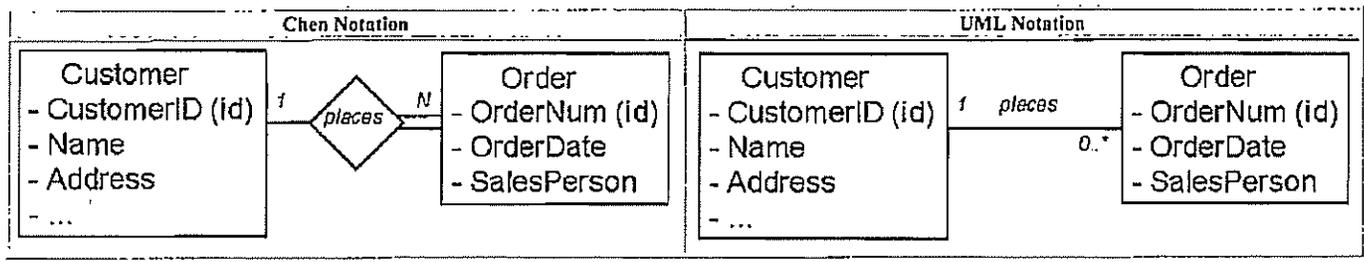


ORDER (OrderNum (key), OrderDate, SalesPerson)
 ORDERITEMS (OrderNum (key){fk}, ItemNum (key), PartNum, Quantity, Cost)

In the above example, in the ORDERITEMS Relation: OrderNum is the Foreign Key and OrderNum plus ItemNum is the Composite Key.
 In the ORDER Relation: OrderNum is the Key.

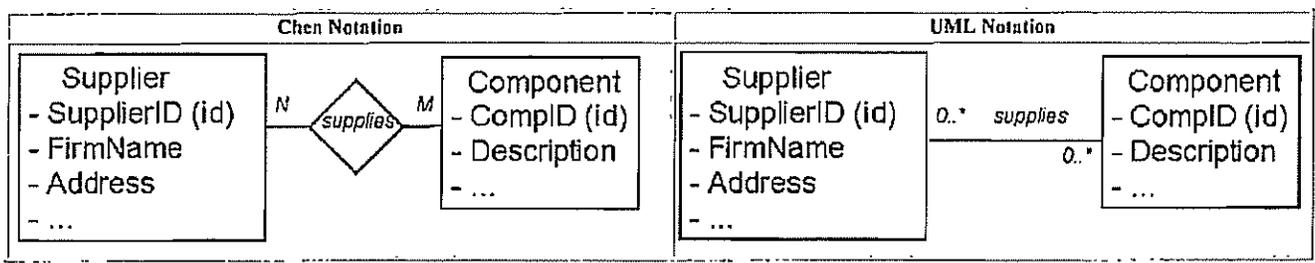
Representing Relationships

- 1:1 Relationships. The key of one relation is stored in the second relation. Look at example queries to determine which key is queried most often.
- 1:N Relationships.
 Parent - Relation on the "1" side.
 Child - Relation on the "Many" side.
- Represent each Entity as a relation.
 Copy the key of the parent into the child relation.



CUSTOMER (CustomerID (key), Name, Address, ...)
 ORDER (OrderNum (key), OrderDate, SalesPerson, CustomerID (fk))

- M:N Relationships. Many to Many relationships can not be directly implemented in relations.
- Solution: Introduce a third Intersection relation and copy keys from original two relations.

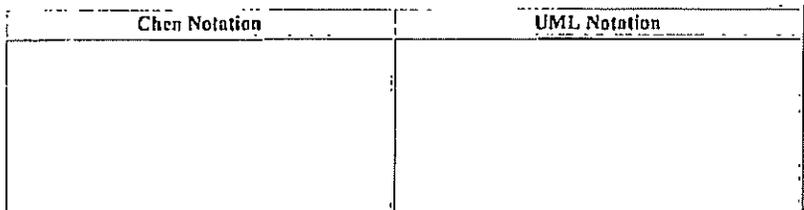


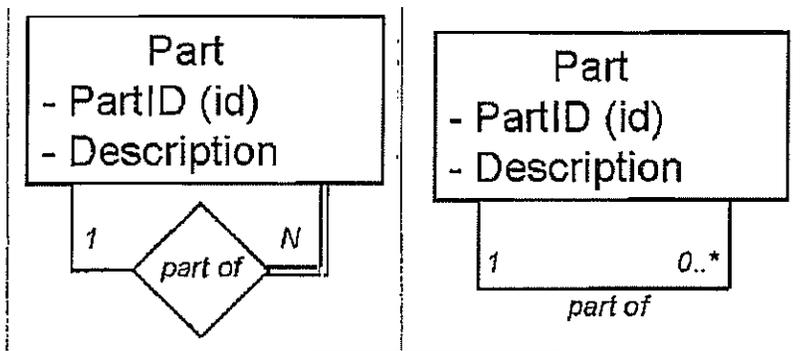
SUPPLIER (SupplierID (key), FirmName, Address, ...)
 COMPONENT (CompID (key), Description, ...)
 SUPPLIER_COMPONENT (SupplierID (key), CompID (key))

- Note that this can also be shown in the ER diagram. Also, look for potential added attributes in the intersection relation.

Recursive Relationships

- Consider assembling an automobile - a collection of basic parts are combined together to make a complete car.
- In this case, a part or component might be made up of one or more other parts. This forms a recursive 1:N relationship.





- We can implement this directly by including another copy of the key in the relation.
PART (Part_ID (key), Parent_Part_ID (fk), Description)
- Here are some example data:

Part_ID	Parent_Part_ID	Description
1000	null	Complete Car
200	1000	Engine
500	1000	Body
510	500	Doors
520	500	Hood
530	500	Quarter Panels
540	500	Trunk
550	500	Roof
512	510	Front Passenger Door
514	510	Front Driver's Door
516	510	Back Passenger Door
518	510	Back Driver's Door etc.

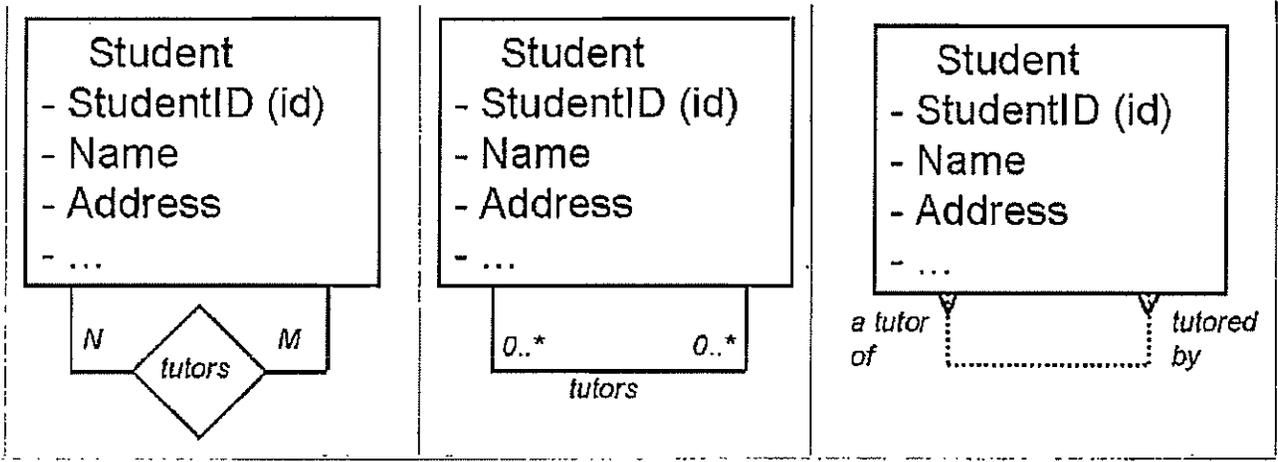
- This forms a *Tree* structure:

```

Complete Car           1000
  Engine               200
  Body                 500
    Doors              510
      Front Passenger Door 512
      Front Driver's Door  514
      Back Passenger Door  516
      Back Driver's Door   518
  Hood                 520
  Quarter Panels       530
  Trunk                 540
  Roof                 550
  
```

- *null* = No Value present
- Other kinds of recursive relationships:
1:1 CUSTOMER is referred by one and only one other CUSTOMER
N:M STUDENT tutors one or more other STUDENTS - Also, that STUDENT can be tutored by one more other STUDENTS.

Chen Notation	UML Notation	Crow's Foot Notation



STUDENT (StudentID (key), Name, Address, ...)
 STUDENT_TUTOR (StudentID (key), Tutored_StudentID (key))

• Example Data:
 STUDENT

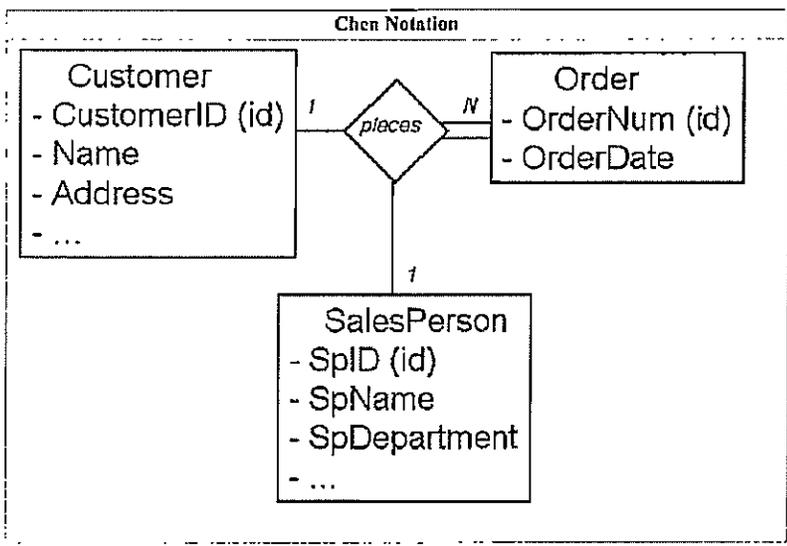
StudentID (key)	Name	Address
101	B. Smith	1234 Smith St.
202	A. Green	12 Grant St.
303	D. Jones	98 Short St.
404	P. Ewing	999 Tall St.

STUDENT_TUTOR

StudentID (key)	StudentTutoredID (key)
101	202
404	202
202	303
404	303

Ternary Relationships

- Ternary relationships are likely ambiguous.
- It is best if we can represent a ternary relationship as two binary relationships between the three entities.
- If we can re-write the ternary relationship as several binary relationships, then the above steps can be used to convert the entities into relations.



RS755

• One Example: ORDER, CUSTOMER, SALESPERSON becomes:

CUSTOMER (CustomerID (key), Name, Address...)
 SALESPERSON (SpId (key), SpName, SpDepartment, ...)
 ORDER (OrderNum (key), OrderDate, CustomerID (fk), SpId (fk))

This represents the business model where a salesperson is assigned to specific customers.

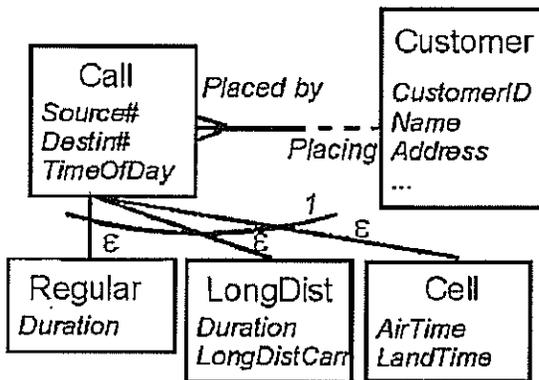
- Another Example: ORDER, CUSTOMER, SALESPERSON becomes:
 CUSTOMER (CustomerID (key), Name, Address..., SpId (fk))
 SALESPERSON (SpId (key), SpName, SpDepartment, ...)
 ORDER (OrderNum (key), OrderDate, CustomerID (fk))

This represents the business model where a salesperson is assigned to individual orders.

- Note that with other notations such as Crow's feet and UML, one must write these as two binary (1 to many) relationships. This removes the ambiguity of the ternary relationship.

IS-A Relationships

- Recall that some entities may have *subtypes* associated with them.



- Strategy is:
 1. Convert the *supertype* entity directly into a relation using only those attributes in the supertype - note the key.
 2. Convert each of the subtype entities into relations - also only with the attributes they contain.
 3. Copy the key for the supertype entity into each of the subtypes.
- So our example becomes:
 - CUSTOMER (Customer_ID (key), Name, Address, ...)
 - CALL (Call_Identifier (key), Customer_Id (fk), Source_Number, Destination_Number, TimeOfDay)
 - REGULAR_CALL (Call_Identifier (key), Duration)
 - LONG_DISTANCE (Call_Identifier (key), Duration, LongDistanceCarrier)
 - CELL_CALL (Call_Identifier (key), Air_Time, Land_Time)

WINDOWS

MAC

LINUX

SCRIPTS

APPS

MOBILE

DRIVERS

CAMERA

TUTORIAL

NEWS

 You are here: [Home](#) - [Miscellaneous](#) - [entity relationship diagram](#)

entity relationship diagram

Popular Searches

 Popular searches about
 Miscellaneous

[abby fine reader 6.0](#)
[anonymous proxy list](#)
[directshow decoder](#)
[directx 12](#)
[dj virtual full...](#)
[download](#)
[nero...](#)
[free download, .](#)
[free download...](#)
[kai s power goo](#)
[matlab free](#)
[download](#)
[pdf to docx](#)
[public proxies .](#)
[spss 14.0 trial...](#)
[spss 16 download](#)
[xlsx to xls...](#)

[more](#) ▶

Virtualization by Dell [www.TalkToDell.com](#)
 Improve Utilization. Simplify IT Powered with
 Intel® Xeon

Free Flowchart Diagrams [www.SmartDraw.com](#)
 Create Flowchart Diagrams fast See Examples.
 Free Download!

Download Open Source ETL [www.Talend.com/Open_E](#)
 Leading Open Source ETL Tools. Complete
 Suite. Download Copy Now!

Application Management [www.ca.com](#)
 Improve Performance & Availability of Critical
 Web-Based Applications



Ads by Google

An entity-relationship model (ERM) in software engineering is an abstract and conceptual representation of data. Entity-relationship modeling is a relational schema database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.

There are a lot of ER diagramming tools. Some of the proprietary ER diagramming tools are Avolution, ConceptDraw, ERStudio, ERwin, DeZign for Databases, MEGA International, OmniGraffle, Oracle Designer, PowerDesigner, Rational Rose, RISE Editor, SmartDraw, Sparx Enterprise Architect, SQLyog, Toad Data Modeler, Microsoft Visio, and Visual Paradigm.

Some free software ER diagramming tools that can interpret and generate ER models. SQL and do database analysis are MySQL Workbench and StarUML. Some free software diagram tools which can't create ER diagrams but just draw the shapes without having any knowledge of what they mean or generating SQL are Kivio and Dia.



ER diagram editor
 ER diagram editor Ease your work with Eclipse with this tool.
 Tags: [edit diagram](#) , [generate code](#) , [Eclipse plugin](#) , [editor](#) , [generator](#)



Aggregate Profiler
 Import from Flat, XML file

RS757

DBVA for Visual Studio Professional Edition DBVA for Visual Studio Professional Edition - Visual Studio UML Tool with Code Generation and Reverse Engineering

Tags: Visual Studio, Visual Architect, Database Architect, VS, SQL



DBVA for Visual Studio .NET Edition

DBVA for Visual Studio .NET Edition DBVA .NET Edition for Visual Studio - Object Relational Mapping Tool, builds database applications faster, better and cheaper

Tags: Visual Studio, Visual Architect, Database Architect, VS, .NET

Searches Related to entity relationship diagram

diagram entity relationship diagrams Data Flow Model Diagram entity UML diagram flow diagram tm diagram jk diagram control data flow diagram wan diagram marketing diagram neuron diagram for .net EPC Diagram TQM Diagram relationship diagram



Home

Software Download
Freeware Download
New Software
Weekly Recommendations
Editor's Reviews
Editor's Picks
Solution Guides
Online Manuals

DVD & Video

Free DVD Decoder
DVD Copy for Mac
DVD Ripper
DVD Player
DVD Copy Software
DVD to iPod Converter
Video Converter
3GP Video Converter
iPod Video Converter
Youtube Video Downloader
FLV to Video Converter

Windows 7

Windows 7
Windows 7 release date
Windows 7 antivirus
Windows 7 compatibility
What's new in Windows 7
Windows 7 upgrade
Windows 7 vs Windows Vista
Windows 7 vs Windows xp
Win 7 & Direct3D
Windows 7 drivers
Daemon tools windows 7

PDA Mobile

iPhone Converter
iPhone Transfer
iPhone to PC
YouTube to iPhone
CPhone Converter
CPhone vs BlackBerry
DVD to CPhone Converter
Video to CPhone Converter
BlackBerry Converter
BlackBerry curve PDF reader
BlackBerry flash player

Links

Flash Player
PDF Reader
Internet Explorer 8
PDF Converter
iPhone Ringtone Maker
CPhone vs iPhone
DVD Maker
Video Capture Software

RS759

This article may contain URLs that were valid when originally published, but now link to sites or pages that no longer exist. To maintain the flow of the article, we've left these URLs in the text, but disabled the links.



[<http://www.pinnaclepublishing.com/sq/default.aspx>]

When All You Want is an ER Diagram

Karen Watterson

Paul Munkenbeck's great review of ERwin last month inspired me to reflect on less expensive alternatives we can use to crank out "quick and dirty" ER diagrams. Here's what I came up with. If you've got other ideas, please share them as a letter to the editor, tip, or mini-review. We'll all benefit! Also, in forthcoming issues, look for reviews of Embarcadero's ER/Studio and Sybase's PowerDesigner.

I suspect that almost all *SQL Server Professional* readers are familiar with ER diagrams, entity-relationship diagrams that are sometimes referred to as ERDs. For small projects, you probably sketch out the entities and relationships of your logical data model on a sheet of notebook paper—or even on a proverbial cocktail napkin. For larger ones, however (say, databases with 100+ tables—or databases you inherit), it's nice to have a "crow's foot" (or similar IDEF1X) diagram that you can just print out. You might want to thumb-tack it onto an appropriate empty section of cubicle wall, flash it at your boss, or even (gasp) use it as part of project documentation. Hence, the appeal of products such as ERwin, reviewed by Paul Munkenbeck in last month's issue.

Unfortunately, however, ERwin's current owner, Computer Associates, doesn't exactly give ERwin away, and not all organizations will choose to invest several thousand dollars in ERwin or competing products such as Embarcadero's ER/Studio, Sybase's PowerDesigner, or Popkin's System Architect. Indeed, why should you if all you want is a "quick and dirty" ER diagram? In this article, I'll show you how you can use SQL Server itself, Microsoft Access, or Visio 2000 Enterprise/2002 Professional to create logical data models that you can print out.

SQL Server's own Diagrammer

For SQL Server 7 or higher user databases (though not for master, model, or msdb for some reason), you can simply right-click on the "diagrams" feature in the Enterprise Manager hierarchy tree to launch the Create Database Diagram Wizard. (Another way to start most wizards, but not this one, is to click on the EM's "wizard" icon, the wand.) You select the tables to add (opting to have the wizard automatically select related tables if you so desire), and bingo—instant ER diagram (see Figure 1).

As you might expect, you can manually arrange the tables, modify the relationships, opt whether or not to display attributes, and so on. You can even add text annotations and, of course, print out your masterpiece. No, you don't have a lot of flexibility when it comes to importing, exporting, and moving EM diagrams around, but a `SELECT * FROM dtproperties` (for the database where you saved your diagram) gives you the clues you need to copy a diagram's "recipe" to another server, for example. You won't, however, be able to opt for different diagramming notations. You're "stuck" with EM Diagrammer's use of the key and infinity symbols to depict one-to-many relationships, for example. Commercial products let you choose among a variety of diagramming notations and methodologies.

Another SQL Server tool you can harness to help you generate ER diagrams is DTS. Remember, you can use DTS to get virtually any data or database into SQL Server. Once you've imported it (or just its schema), it's often a matter of just a few seconds to have EM's Diagrammer generate a diagram for you. You can even use DTS to get a database into Access—that is, move an Oracle8i database into Access, for example, bypassing SQL Server altogether. And don't forget that the SQL Server 2000 version of DTS also supports XML. The ReadFile and WriteFile transformations give you a way to import and export XML files as part of your transformations, and the parallel data pump task lets you process XML's hierarchical recordsets. (For more on using XML with SQL Server, see Tom Moreau's March 2001 column, "Feeding XML to Stored Procedures," and consider purchasing Graeme Malcolm's excellent new book, *Programming Microsoft SQL Server 2000 with XML* [ISBN 0-7356-1369-9].)

What about Access?

RS761

If you use a sample Microsoft database like Northwind and then select Tools | Relationships (see Figure 2), you'll

probably think Access's Relationships designer is even faster than Enterprise Manager's.

Figure 3 illustrates another example of Access's prowess, here modeling SQL Server's Analysis Services FoodMart2000 database. Like SQL Server, there are limitations to what you can do with the diagrams. You can print an Access "relationships" diagram, but you can't change the notation (a "1" and an infinity symbol for 1:M relationships, for example) or add annotations.

What you don't realize until you try to generate a diagram for a non-Microsoft sample database is that "someone" has to manually set up all of the relationships (EM automates the process by making intelligent "guesses" based on primary and foreign keys). Granted, Access's Relationships designer is reasonably flexible (you can define relationships as 1:1, 1:N, or 1:M and specify whether referential integrity is to be enforced), and we've been able to save Access Relationships at least since Access 2.0, but the Relationships designer isn't smart enough to figure out the relationship itself.

The Relationships designer and menu in Access 2002/XP (see Figure 4) seems to be the same as it is in Access 2000. The main feature that potentially extends its features is Access XP's ability to import and export XML.

What about Visio 2000?

Visio 2000 Enterprise Edition SR-1, which is included in the MSDN Universal Subscription as disks 672 and 673, is no longer available commercially. If you're not an MSDN subscriber and want a copy of any version of Visio 2000, you can either try to find an unsold copy in the channel or buy Visio 2002 and then "downgrade" to Visio 2000. (For information about Visio 2000's SR-1, you can read "Visio 2000: How to Obtain the Visio 2000 Service Release 1 [SR-1] Update" at <http://support.microsoft.com/support/kb/articles/Q2637/21.ASP> [<http://support.microsoft.com/support/kb/articles/Q2637/21.ASP>] and/or Q264400, "What Is Fixed in the Service Release 1 [SR-1] Update for Visio 2000 [All Editions].") Actually, you might not need EE to do database diagramming, since Visio 2000 Professional lets you create six types of database diagrams: Bachman, Chen ERD, Database Model, Express-G, Martin ERD, and ORM Diagramming.

Visio 2000 EE supports seven types of diagrams: Bachman, Chen ERD, Database Model Diagram, Express-G, Martin ERD, ORM Source Model, and ER Source Model.

The Bachman, Chen, Martin, DBM, and ER Source model diagrams are undoubtedly familiar to most DBAs as variations on the ER theme of depicting tables with annotated lines between them. Less familiar, perhaps, is Express-G, a graphical notation for information models. (See www.steptools.com/support/stdev_v8_docs/devtools/devtools_9.html [http://www.steptools.com/support/stdev_v8_docs/devtools/devtools_9.html] for information about Express and its support of STEP. STEP, the Standard for the Exchange of Product Model Data, is a comprehensive ISO standard [ISO 10303] that describes how to represent and exchange digital product information. STEP was born in 1983, and was based on previous national efforts such as IGES, VDAFS, SET, CAD*I, and PDDI.) ORM refers to Dr. Terry Halpin's Object Role Modeling methodology, which Visio inherited when it acquired InfoModeler. See www.orm.net/halpin.html [<http://www.orm.net/halpin.html>] for more about ORM.

Visio, of course, is a commercial modeling product, so you'd expect it to offer more than SQL Server's EM Diagrammer or Access's Relationships designer, and indeed it does. In addition to creating databases or data models in Visio Professional or EE, you can also reverse engineer database models from existing databases. You can also import ERwin ERX diagrams (but not the newer ER1 models—see the "ERwin Filenames" sidebar).

To test Visio 2000, I decided to try to reverse engineer a data model. I opted to create a new Database Model Diagram (you can only reverse engineer three of the seven diagrams: Database Model Diagram, ER Source Model, and the ORM Source Model because the others are "left over" shape-only stencils from earlier versions of Visio) and was rewarded by Visio 2000 EE's Reverse Engineer Wizard. The first step is to select the Visio database driver for the source DBMS, and the wizard will guide you through the process of configuring and testing the ODBC data source if you need to.

Next, you check the boxes for the type of information you want to extract (see Figure 5).

Then you check the tables (and views, if any) that you want to extract (fortunately, there's a Select All button). If you checked Stored Procedures, you'll see a screen asking you to check the procedures that you want to extract (again, there's a Select All option to extract them all).

The wizard extracts the selected information and displays notes about the extraction process in the Output window and lists the tables in the Tables window (see Figure 5).

At this point, the human work begins. You have to drag the tables you want from the Tables window onto the drawing page and manually create the relationships using the icons from the toolbox on the left side of Figure 5. Visio is a rich product and gives you all kinds of options such as keeping your drawing and the original database schema in sync. When you save a Visio 2000 EE drawing, you can opt to have your VSD file saved for printing, for PPT output, or for HTML/GIF output. For information about ways to include Visio 2000 diagrams in other docs, refer to www.microsoft.com/technet/prodtechnol/visio/maintain/spot1.asp.

RS762

Visio 2000 Professional and EE also support UML 1.2 diagrams and reverse engineering from Visual Studio, though you can only generate Visual Studio code from UML (Unified Modeling Language) models with the EE. If you want to use UML with Visio 2000, you should download the 240KB Microsoft Visio 2000 UML Shapes Update from <http://office.microsoft.com/downloads/2000/visUMLShapesddl.aspx> [<http://office.microsoft.com/downloads/2000/visUMLShapesddl.aspx>] . The download replaces some of your existing Visio 2000 Professional edition UML shapes with shapes you can manually resize. You might also want to read Frank Sternberg's white paper on "Iterative Development Using UML and MS Visio 2000 Enterprise Edition" at www.microsoft.com/technet/prodtechnol/visio/maintain/visiodev.asp?frame=true.

The best sources of current Visio 2000 information are www.microsoft.com/office/previous/visio/default.htm [<http://www.microsoft.com/office/previous/visio/default.htm>] and the 20 or so newsgroups (including ones in French, German, Dutch, Spanish, and Japanese) at public.microsoft.com/visio.... You can search for Microsoft Online Seminars [<http://www.microsoft.com/seminar/>] dealing with Visio. You can also subscribe to Visio's bi-monthly Visio SmartPages newsletter from Microsoft and work through several online database-related tutorials at www.microsoft.com/presspass/guides/visio/professional/enttutorials/defaultd.asp [<http://www.microsoft.com/presspass/guides/visio/professional/enttutorials/defaultd.asp>] .

Interesting bit of trivia: Microsoft BizTalk customers and developers might have noticed that the BizTalk Orchestration Designer is based on Visio.

Visio 2002 Professional

Microsoft began shipping Visio 2002 Standard and Professional editions earlier this year for \$199 U.S. and \$499 U.S., respectively. There's no downloadable evaluation version available for these products, but U.S. and Canadian customers can order a 30-day trial version CD for \$7.50 (\$10 Canadian). There's no Enterprise edition because "the advanced database and software modeling solutions previously found in Visio 2000 Enterprise Edition will be delivered in Microsoft Visual Studio .NET," presumably as part of the recently announced VS.NET Enterprise Architect (<http://msdn.microsoft.com/vstudio/productinfo/overview/eaoverview.asp> [<http://msdn.microsoft.com/vstudio/productinfo/overview/eaoverview.asp>]). Here's the main URL for info about Visio 2002: www.microsoft.com/office/visio/evaluation [<http://www.microsoft.com/office/visio/evaluation>] . The main URLs for Visio programmers and developers are <http://msdn.microsoft.com/library/default.asp?url=/nhp/Default.asp?contentid=28000456> [<http://msdn.microsoft.com/library/default.asp?url=/nhp/Default.asp?contentid=28000456>] and <http://www.microsoft.com/technet/prodtechnol/visio/default.mspx> [<http://www.microsoft.com/technet/prodtechnol/visio/default.mspx>] .

I'd been unsuccessful installing the Visio 2002 Marketing beta on either 1) any of three different Windows 98 systems or 2) a Windows 2000 system that had Office XP installed, but the shipping release product did install on both. I wouldn't, however, recommend using it on a Win98 box unless you're desperate. (It brings new meaning to the notion of "slow.")

Visio 2002 Professional does provide you with features that let you generate ER diagrams, but, instead of Visio 2000 Professional's six, or Visio 2000 EE's seven, you only get three choices: Database Model Diagram, Express-G, and ORM Diagram (see Figure 6). Basically, the Bachman, Chen, and Martin ER diagrams have been subsumed into the Database Model Diagram, but Visio 2002 Professional does provide all of the shapes you need to create logical models using today's popular ER and IDEF1X notations. (IDEF1X is a method for designing relational databases with a syntax designed to support the semantic constructs necessary in developing a conceptual schema. For more about IDEF1X, see www.edef.com/idef1x.html [<http://www.edef.com/idef1x.html>] or buy Thomas Bruce's 1992 classic *Designing Quality Databases With IDEF1X Information Models* [ISBN 0932633188].)

As with Visio 2000 Enterprise, I was able to successfully reverse engineer pubs and generate a Database Model Diagram. Unlike Visio 2000 Enterprise, however, I couldn't generate DDL (a big minus, IMHO), nor could I reverse engineer databases from DDL scripts. (These features will be offered in what amounts to a Visio 2002 Enterprise, which apparently will be available in the form of the Enterprise Edition of Visual Studio .NET. VSEA is *not* available in the general release of .NET Beta 2, but was distributed in mid-July to TechEd attendees and MSDN Universal subscribers. And speaking of Visual Studio, it too offers developers a "quick and dirty" ER diagram generator of sorts. In VB6, for example, you can use the Data Environment's data diagrams, and a similar functionality is available in VS.NET via the Data Form Wizard, which actually stores the schema as a VSD file.)

New in Visio 2002 Professional, however, is XML support, both in the guise of an optional new file format (VDX) for Visio diagrams and for reverse engineering. See the June white paper "XML for Visio Scenarios," which illustrates how XML for Visio can be used to extract Visio data for use in solution development, data analysis, text localization, Web publication, and database interoperability for more on Visio's XML.

Here are some additional resources for Visio 2002:

- COM Add-In Designer for Visio 2002.
- Visio MVP Graham Wideman's Unofficial Visio (still mainly Visio 2000) FAQ site at

RS763

www.diagramantics.com/diagenvs/visio/faq/index.htm

[<http://www.diagramantics.com/diagenvs/visio/faq/index.htm>] . (I also *highly* recommend Graham's extremely useful *Visio 2002 Developer's Survival Pack*, ISBN 1-55212-682-X. Owners can download a 7MB ZIP of samples and tools.) Note from Graham: "One currently missing opportunity is that, though Visio in general is programmable by Automation, the DB and UML modeling solutions have *not* been. So, you can use Automation to visit and read the shapes of a UML or DB diagram, but that doesn't get you very far, as there's much more information in the model than in the shapes. So counter to expectations, you can't write your own code to do something (useful) automatically with a Visio DB or UML diagram."

- Short animated illustration of how to use database modeling and UML features:
www.microsoft.com/office/visio/evaluation/techtour/page4.htm
[<http://www.microsoft.com/office/visio/evaluation/techtour/page4.htm>] .

Conclusion

This just represents tools that I use to generate low-cost ER diagrams when I'm in a hurry or at the site of a client who doesn't have one of the high-end commercial tools. If you know of others you think should be included, please let us all know.

Link to www.microsoft.com/office/visio [<http://www.microsoft.com/office/visio.aspx>]

Sidebar: Get ORMed

(Adapted from the June 20 issue of Karen's SQL eXTRA eNewsletter #84. Subscribe at www.FREEeNewsletters.com [<http://www.freeenewsletters.com/default.aspx>] .)

Perhaps I'm reading too much between the lines here at TechEd, but it seems to me that Microsoft is favoring its own database modeling software—software acquired when it bought Visio (which had itself acquired InfoModeler)—over competing tools such as Computer Associates' ERwin. Indeed, Visio's ORM also seems poised to supplant Visual Modeler, a subset of Rational Rose that ships with Visual Studio 6.0 Professional and Enterprise editions.

Here's what Microsoft was saying as of June 18: "Visual Studio .NET Enterprise Developer (VSED) will include functional testing tools for XML Web services, full Unified Modeling Language (UML) 1.2 software modeling, industry-leading database modeling (spanning conceptual, logical, and physical models), and enterprise frameworks and templates—a set of technologies that enable software architects to create an application recipe (ingredients and instructions) and provide it directly to developers within the Visual Studio .NET environment. The enterprise features aren't provided in the public Visual Studio .NET Beta 2 release due to timing issues."

So what's ORM and where did it come from? Object Role Modeling (ORM) is a methodology for designing and documenting databases, and www.orm.net [<http://www.orm.net/default.aspx>] is "the official site for conceptual data modeling." As the site's owner, Dr. Terry Halpin, now program manager in Microsoft's Database Modeling, Enterprise Framework and Tools Unit, explains, "Object Role Modeling (ORM) is a method for designing and querying database models at the conceptual level, where the application is described in terms easily understood by non-technical users. In practice, ORM data models often capture more business rules, and are easier to validate and evolve than data models in other approaches." This site features a number of technical papers and articles on ORM, as well as other approaches such as the UML. Permission has been obtained from the relevant publishers to reproduce these publications on this Web site, and to make them freely downloadable.

Halpin, whose new book *Information Modeling and Relational Databases* has just been published (ISBN 1-55860-672-6), formalized the Object Role Modeling notation and has authored four books and more than 90 technical papers.

Related: The former ORM tool known as VisioModeler (essentially InfoModeler) is now freely available as an unsupported product from Microsoft Corporation (as a 26.5MB download). Models developed in VisioModeler should be able to be exported to Microsoft's current and future ORM solutions, which are based on the Visio drawing engine. (Visio MVP Graham Wideman recommends using this rather than the incomplete version of ORM that shipped with Visio 2000 Enterprise.)

See <http://msdn.microsoft.com/library/default.asp?url=/downloads/list/visio.asp> for the 26.5 MB download.

Sidebar: Conceptual, Logical, and Physical Models

RS764

- *Conceptual database design* is a process that might be described as "getting your head around" the data you need to model. The result is a high-level model of the data that you can use to describe the database to high-level managers and end users, for example.

- *Logical database design* is the process of constructing a model according to the conventions of a specific modeling methodology, but independent of a particular DBMS and physical considerations.
- *Physical database design* is the process of producing a description of the database (often in terms of DDL scripts, for example) in terms of secondary storage. It will include descriptions of the base tables and views, the indexes, constraints, and so on.

Sidebar: ERwin Filenames

ER1 and ERX are both filename extensions associated with ERwin (Entity Relationship for Windows), a popular database modeling program originally created by Logic Works, Inc., and subsequently purchased by Platinum Technology, which was itself acquired by Computer Associates (www.cai.com [<http://www.cai.com/default.aspx>]). After a long wait (the previous version, ERwin 3.52, hadn't been updated since 1998), Computer Associates finally released ERwin 4.0.

One of the outputs of all of these programs is an ER diagram that graphically illustrates database tables (entities) and the relationships (one-to-one, one-to-many, or many-to-many) between them. At any rate, if you run across references to ER1 or ERX, they're probably referring to ERwin files:

- ER1—Standard ERwin diagram file format.
- ERX—Text-based version of ERwin diagram file format. The text-based ERX file format is designed to transfer model data and comments between ERwin and other tools.

Those aren't the only types of files ERwin creates. Others include:

- ERT—ERwin diagram file saved as a template.
- ERV—ERwin diagram saved as an Intersolv PVCS archive file. You must have Intersolv's PVCS version 5.1 or later installed to open an ERV file. (Intersolv is now owned by Merant, www.merant.com [<http://www.merant.com/default.aspx>] .)
- EAX and BPX—Files with these extensions are associated with BPwin, the business process model tool.
- ERS—ERwin physical database schema script text file.
- SQL—SQL DDL (Data Definition Language) schema script text file.
- SML—Structured Modeling Language text file. Similar to a SQL data file used to store information about an ERwin model in text format.
- CMT—Text file containing logical entity/attribute and physical table/column definitions (comments).
- DF (SQL)—A special version of SQL files used for Progress Software databases.

Conceptual, Logical, and Physical Models >

To find out more about Microsoft SQL Server Developer and Pinnacle Publishing, visit their website at <http://www.plnpub.com/html/main.isx?sub=57> [<http://www.plnpub.com/html/main.isx?sub=57>]

Note: This is not a Microsoft Corporation website. Microsoft is not responsible for its content.

This article is reproduced from the August 2001 issue of Microsoft SQL Server Developer. Copyright 2001, by Pinnacle Publishing, Inc., unless otherwise noted. All rights are reserved. Microsoft SQL Server Developer is an independently produced publication of Pinnacle Publishing, Inc. No part of this article may be used or reproduced in any fashion (except in brief quotations used in critical articles and reviews) without prior consent of Pinnacle Publishing, Inc. To contact Pinnacle Publishing, Inc., please call 1-800-493-4867 x4209.

Name: _____

SSN/ID: _____

Section & Group: _____

Creating ER Diagrams with MS Visio

This experiment introduces the student to using Microsoft Office's Visio Application to create ER Diagrams.

Objectives:

1. What is Visio?
2. Getting Started with MS Visio
3. Entities
4. Relationships
5. Fine-tuning the Drawing

What is Visio?

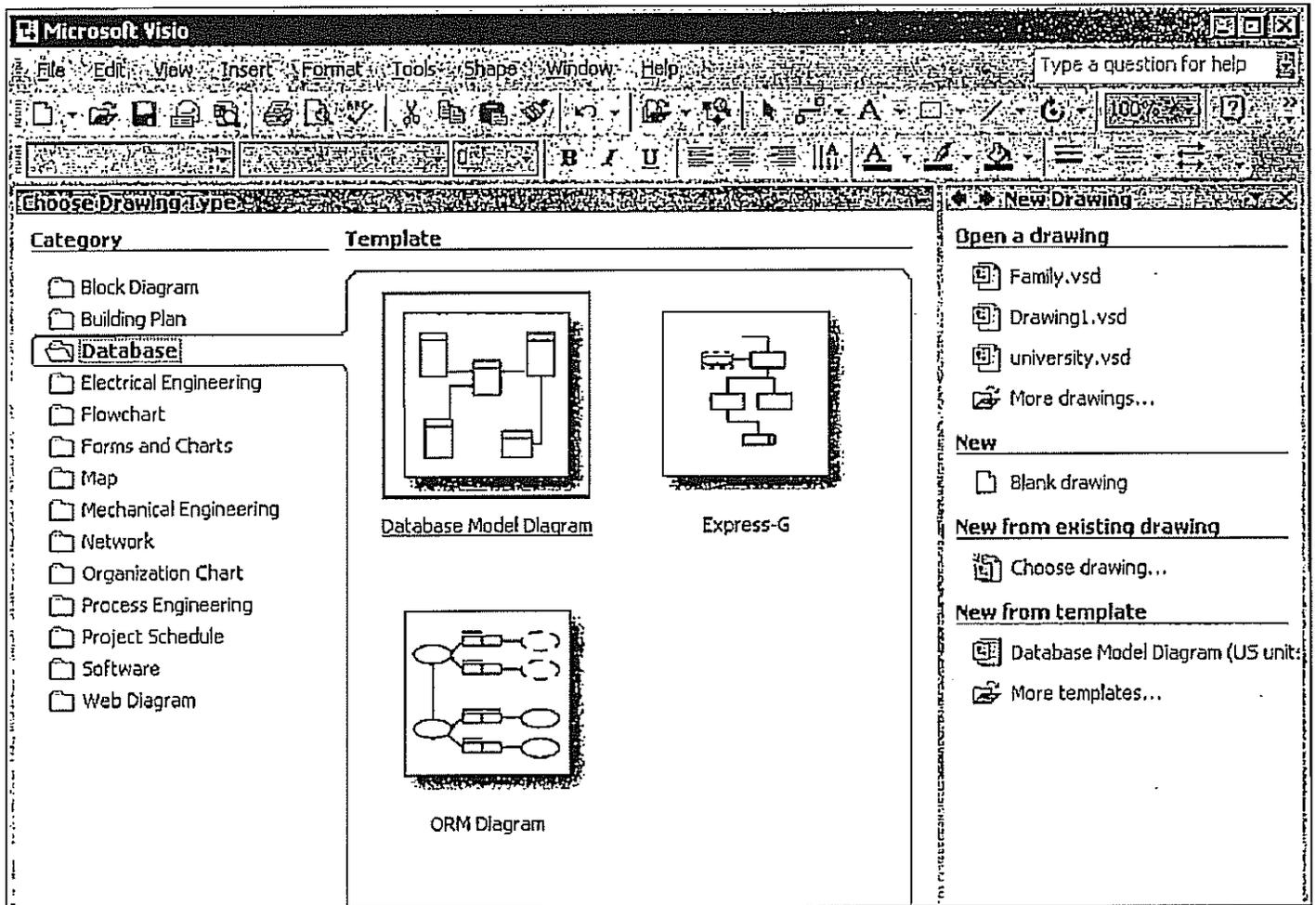
Visio Corporation was founded in 1990 by former software developers from Aldus Corporation (the PageMaker folks). The first version of their business drawing program was marketed in 1992. Microsoft acquired the Seattle-based company in 1999, and today the product line includes four major versions: Visio Standard, Professional, Technical, and Enterprise Editions. Collectively, the Visio products are the market-share leader for business drawing tools.

Visio can be used to quickly create a wide range of drawings: building blueprints, street maps, project timelines, program flowcharts, network diagrams, data flow and process flow diagrams, ER Diagrams, etc.

This lab tutorial will illustrate the creation of crow's foot ER Diagrams with the Professional Edition of Visio, which is available as part of your student software bundle. If you need to download the software bundle, please use this link to proceed to the The Fulfillment Center. In addition, a free 30-day trial version can be downloaded directly from Microsoft.

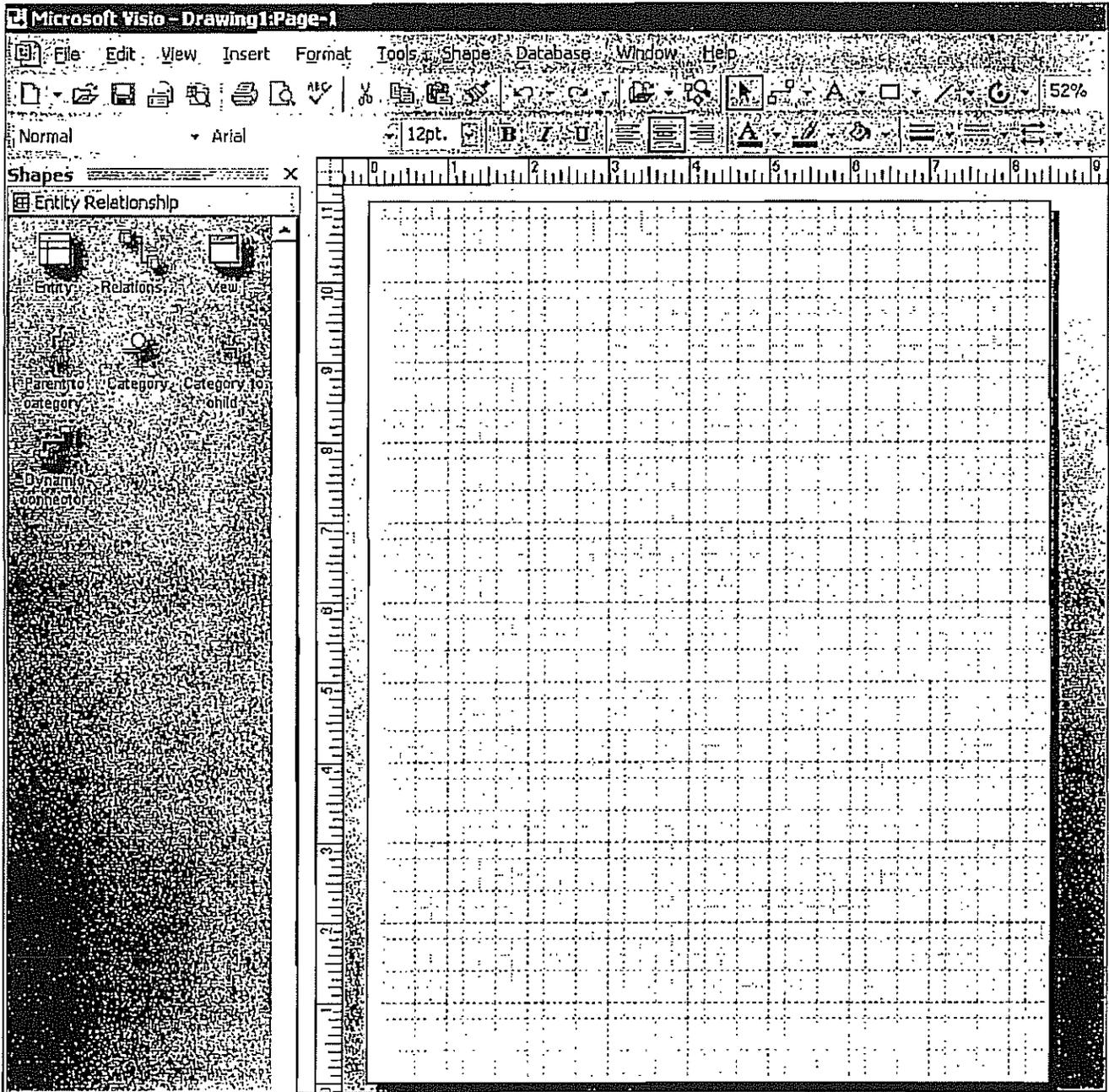
Getting Started with MS Visio

As soon as you start Visio you will either need to identify a new drawing type (category) or select an existing file. For this example, select a new drawing with the Database category and the Database Model Diagram template.

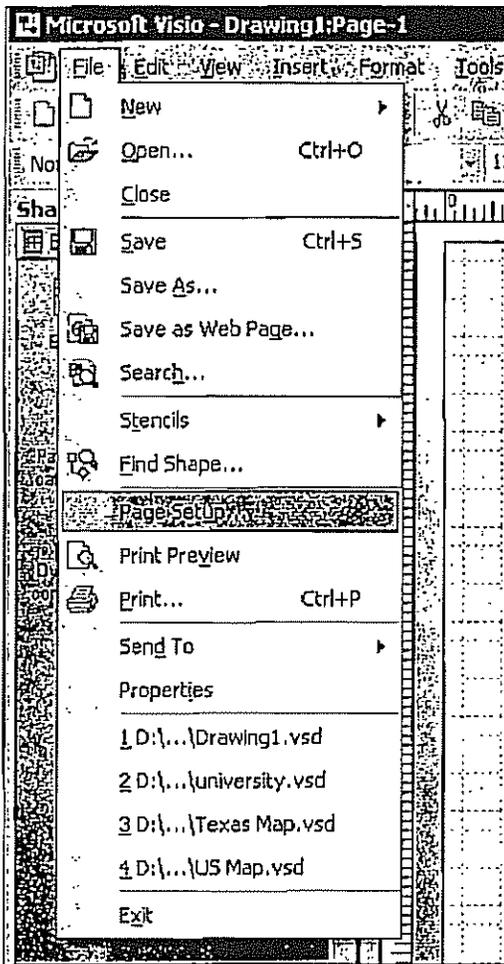


A template (stencil) is simply a set of common shapes and connectors used in the type of drawing specified. Some templates only have a couple of shapes while others have dozens. Visio lets you use multiple templates on the same drawing if necessary. Only one template will be necessary to draw our ER diagrams.

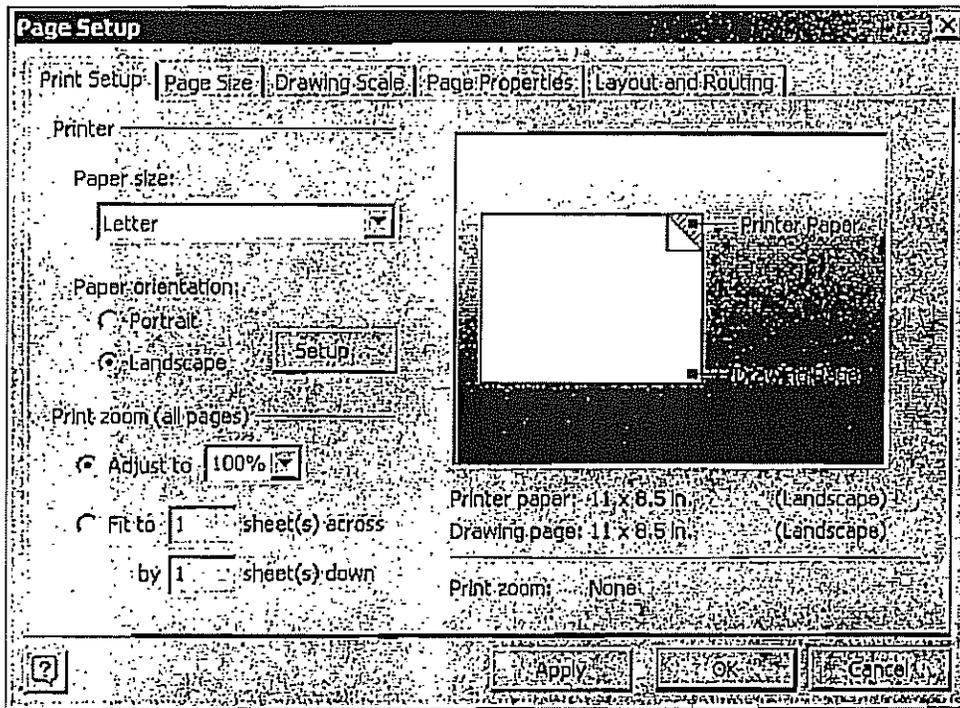
Visio will then give you a blank worksheet (like graph paper) and the template containing the basic shapes and connectors for the chosen drawing type. Most crow's foot ER diagrams can be created with just two of the seven basic database shapes and connectors: entity and relationship.



If your initial worksheet appears in the portrait orientation (as in this example), then you might want to rotate it to landscape orientation to make more effective use of your rectangular screen. To do this, select the Page Setup option from the File drop-down menu.



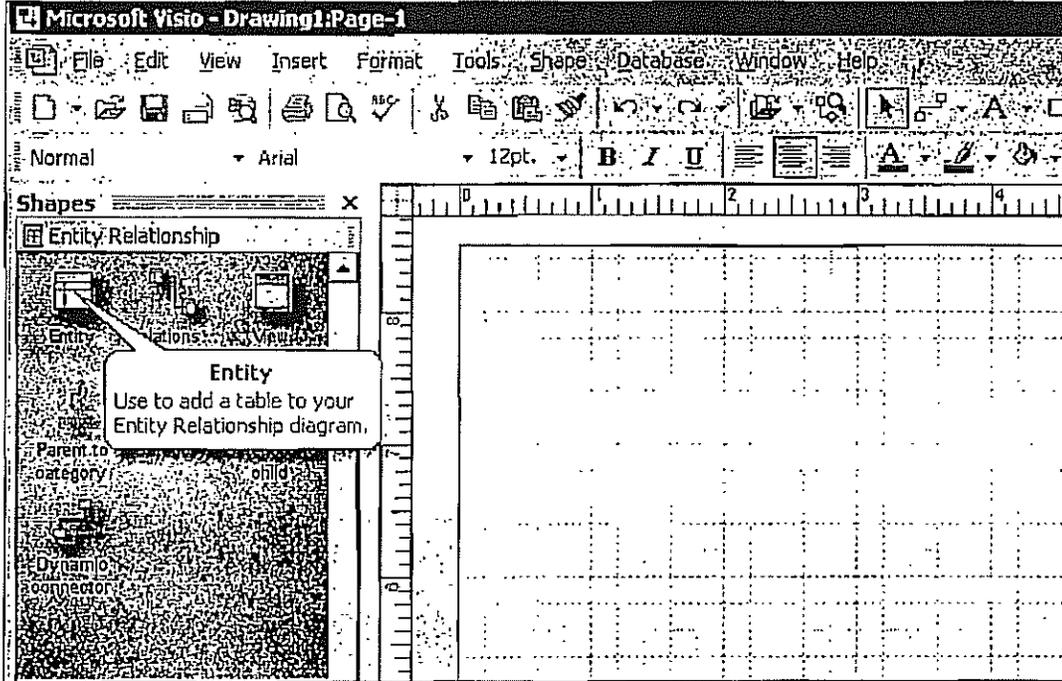
In addition to the worksheet orientation, Page Setup allows you to change the scaling, paper size, margins, etc.



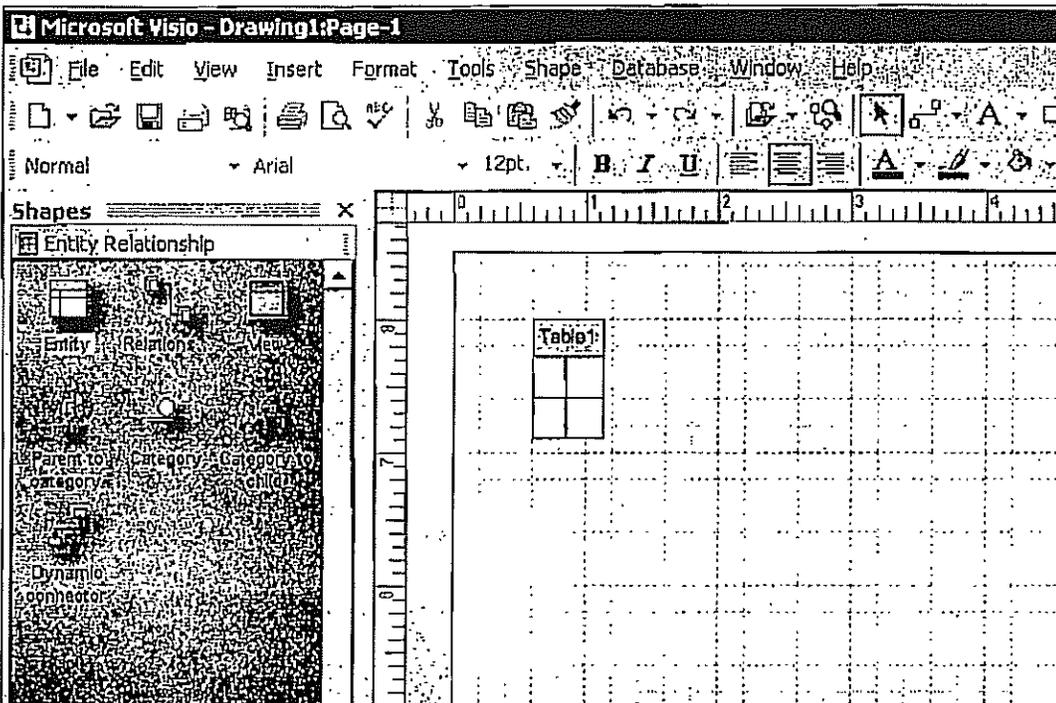
RS769

Entities

Begin creating your ER diagram by clicking the Entity (table) shape on the green template.

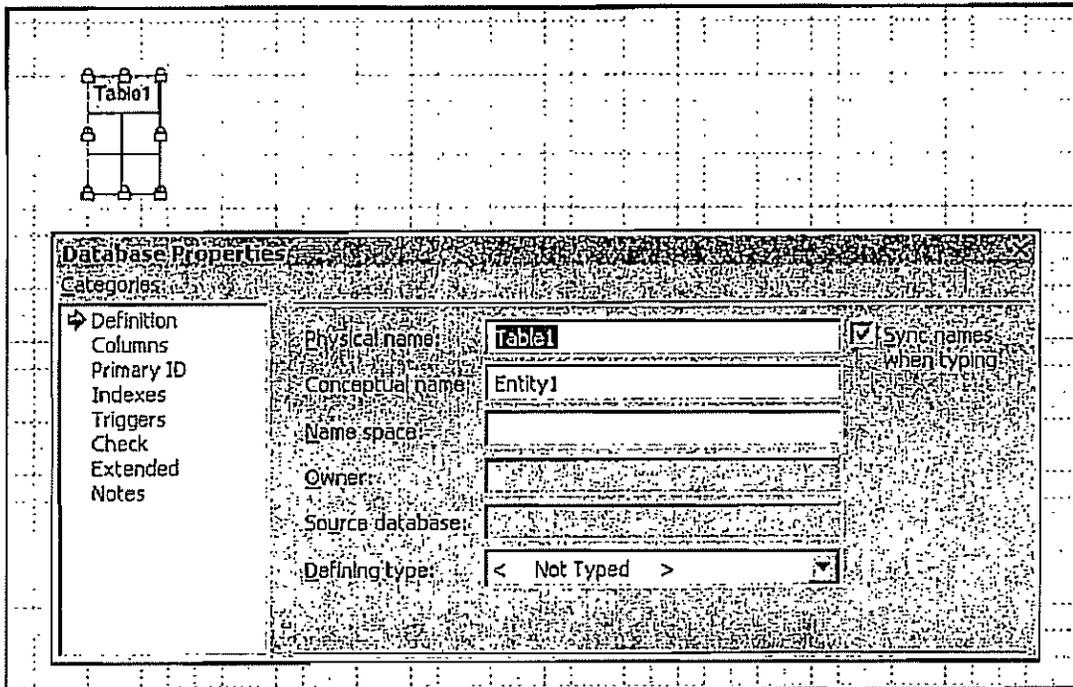


Drag the entity shape to your worksheet. You can reposition it at any time. You can delete any shape by clicking it and pressing the delete key.

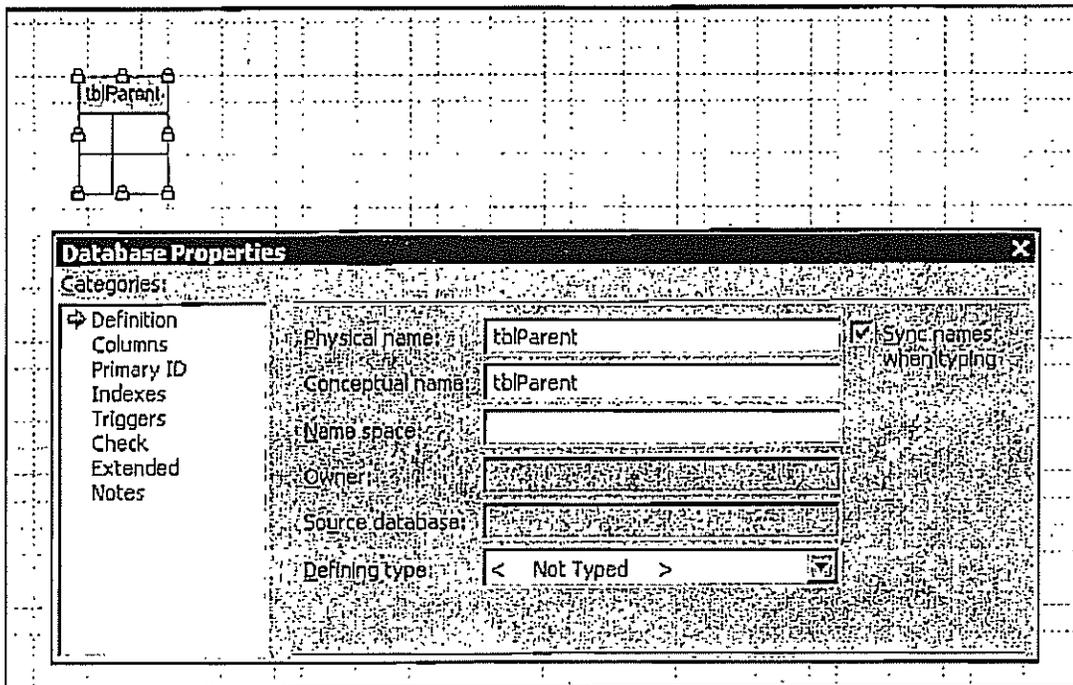


RS770

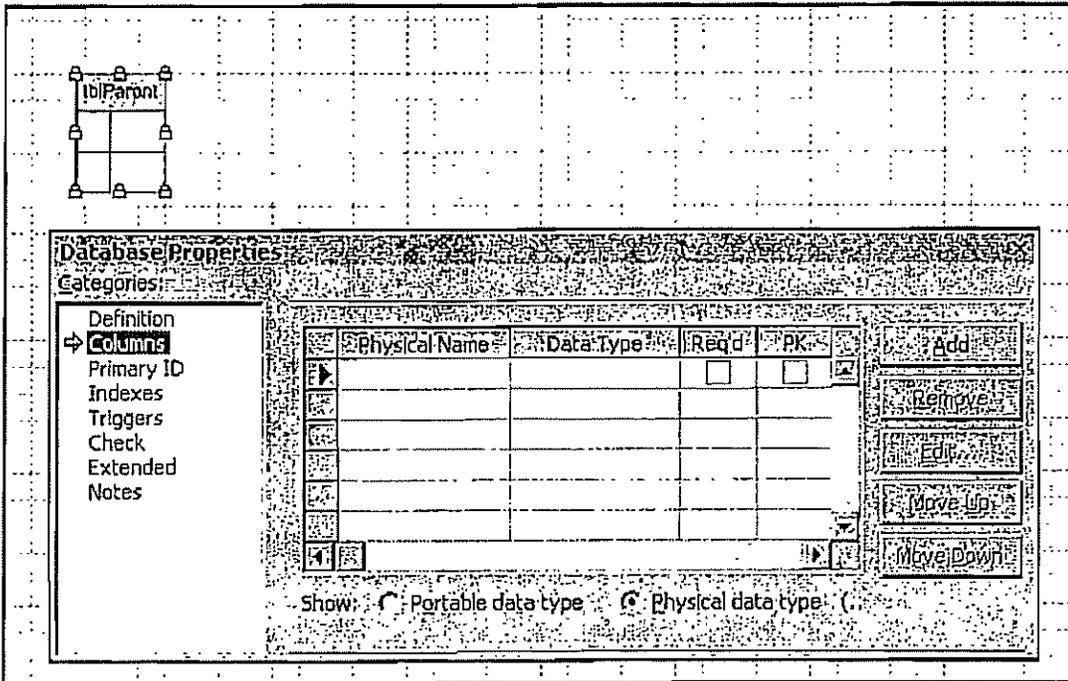
Double click the entity to view/edit its name, columns (fields), primary key, secondary indices, etc.



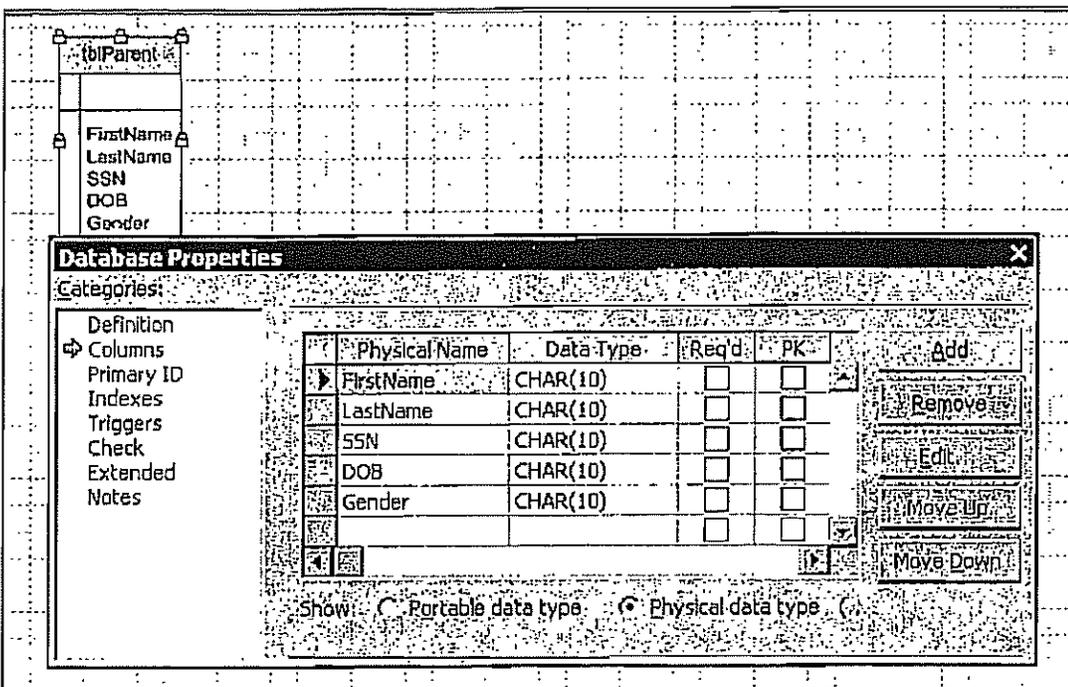
Change the default name (Table1) of the entity to tblParent. Visio allows you to specify different physical and conceptual names for each entity. Since the Sync Name property has been selected, you only need to enter one new name for the table.



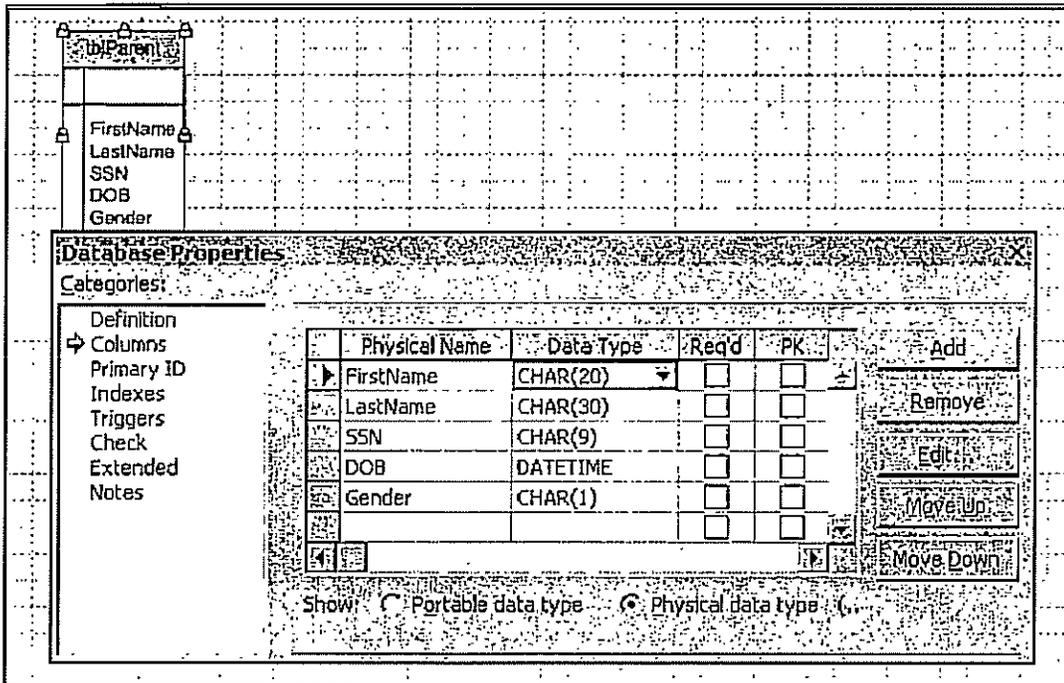
Switch to the Columns (fields) category.



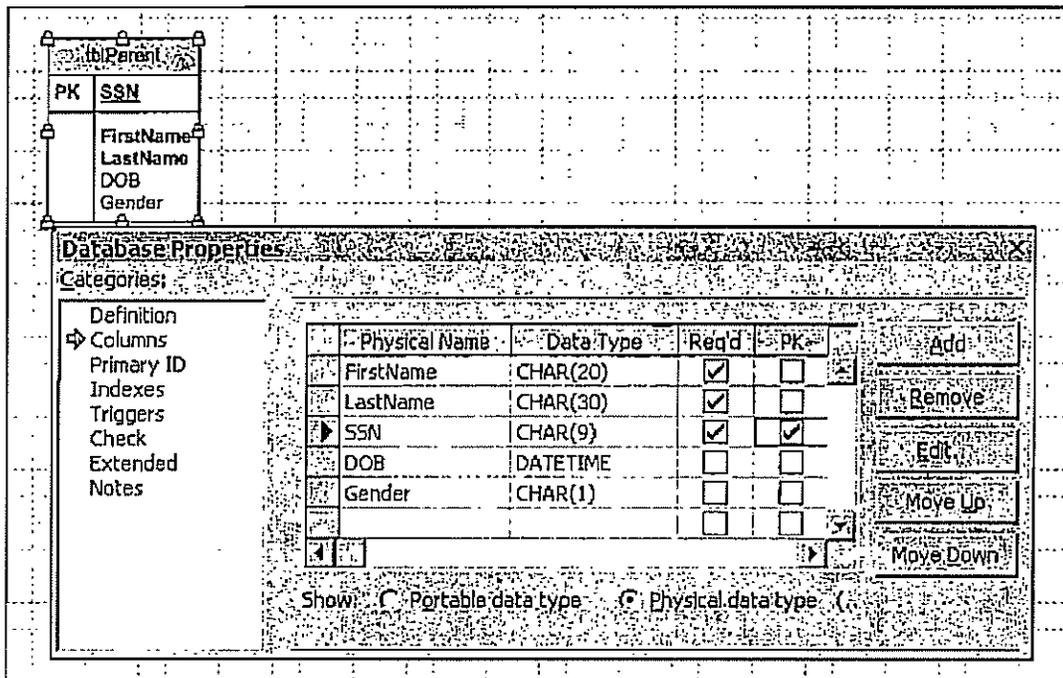
Enter the names of the five fields in the table. The order isn't particularly significant.



Visio assigns a default data type of ten character text to all new columns. If desired, you can change the columns to their correct data types. Of course, this will have no impact on the appearance of the drawing.

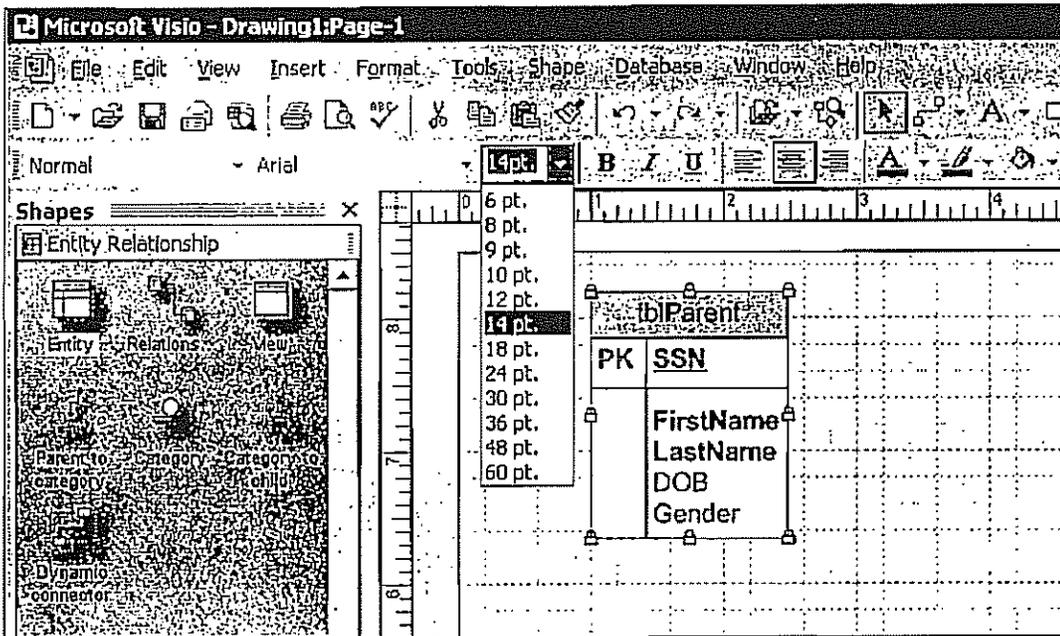


Indicate the Required and Primary Key (PK) status for each field. Notice that Visio underlines the key field and moves it to the top of the diagram. We will change this shuffling behavior later.



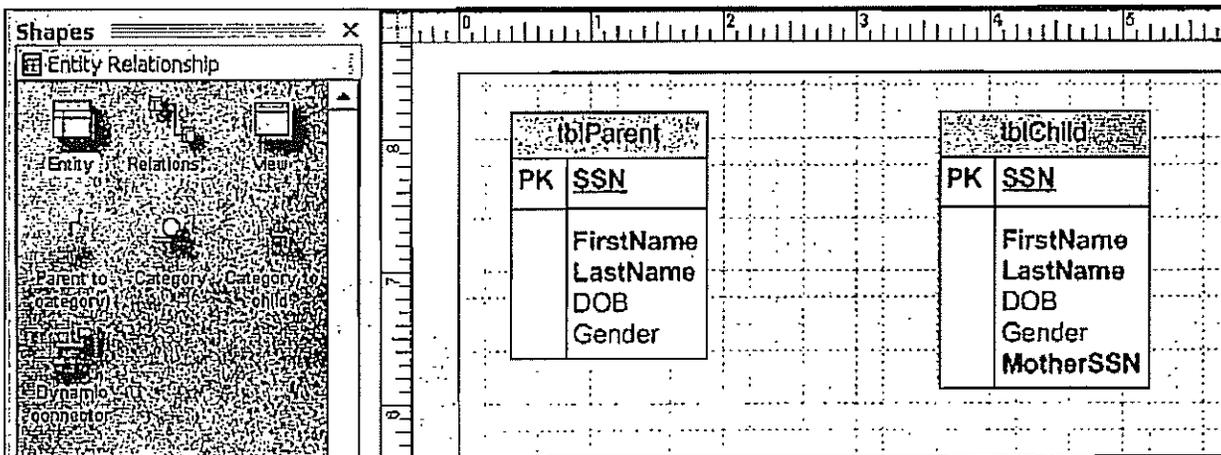
The three required fields of the table are shown in boldface and the two optional fields are shown in standard face.

You can change the font size by selecting the entity then picking a size from the drop-down menu. You can also change the font family, color, alignment, etc.

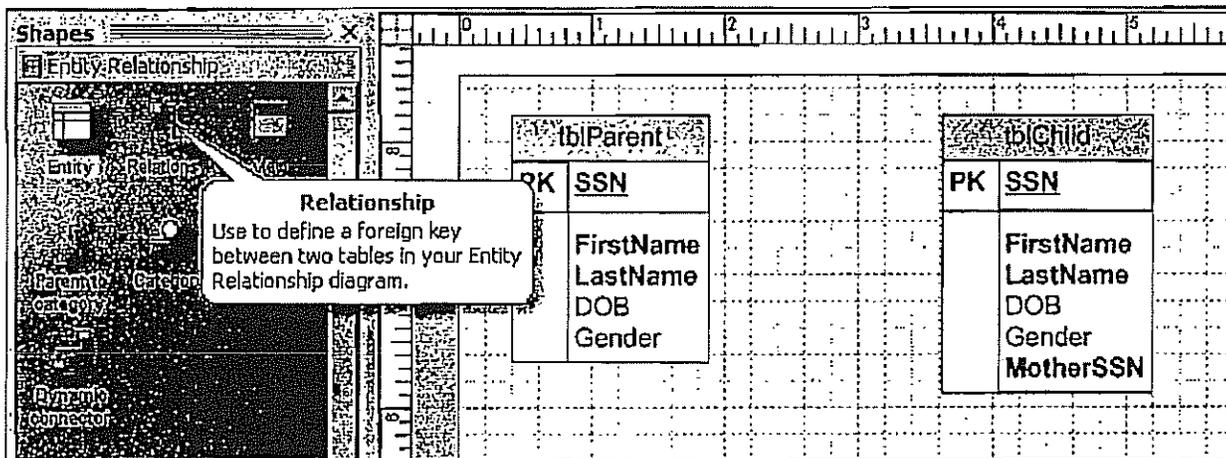


Relationships

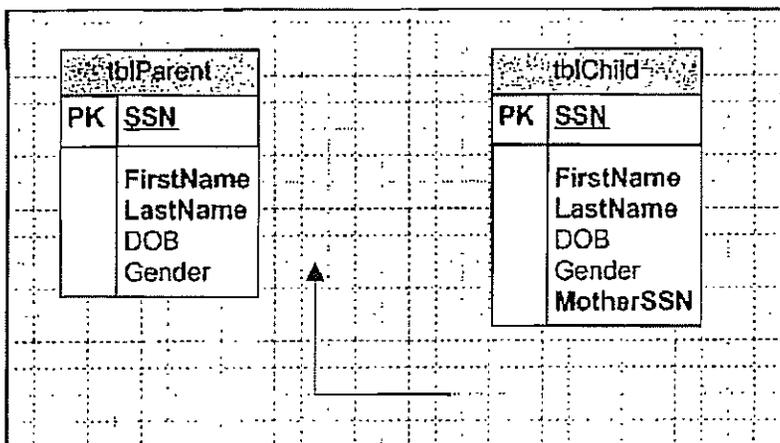
Follow the same basic steps to create a second table. Notice that the tblChild table contains two social security number fields. SSN is the unique ID for the child and MotherSSN is the ID for the child's mother. MotherSSN will be a foreign key (pointer) to tblParent.



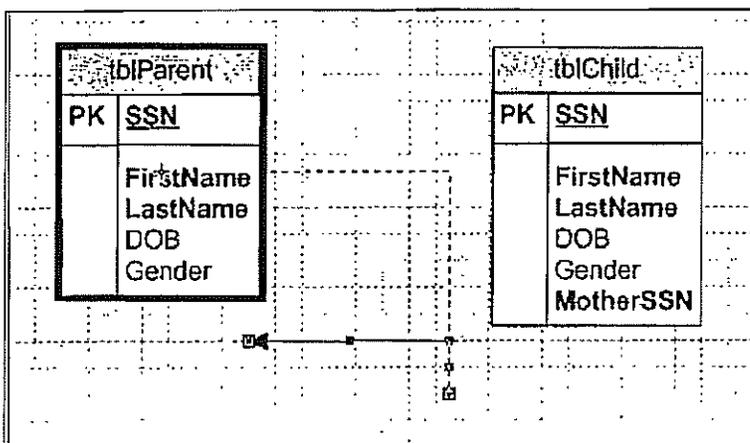
Click the Relationship connector in the template.



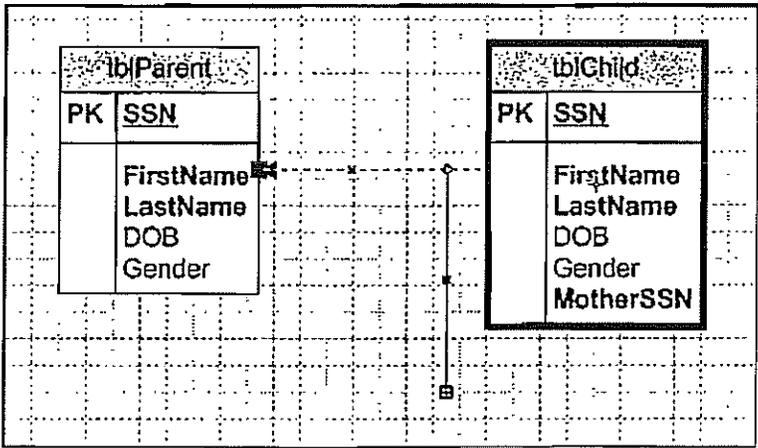
Drag and drop the relationship connector to any blank space in your diagram. Notice that it appears as an arrow. We will change it to a crow's foot later.



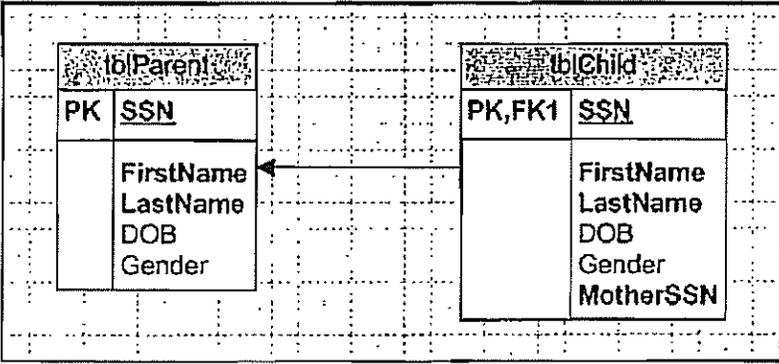
Click the end of the relationship connector with the arrowhead. Drag it over 'tblParent' until the border turns red.



Drag the other end of the relationship connector over 'tblChild'.

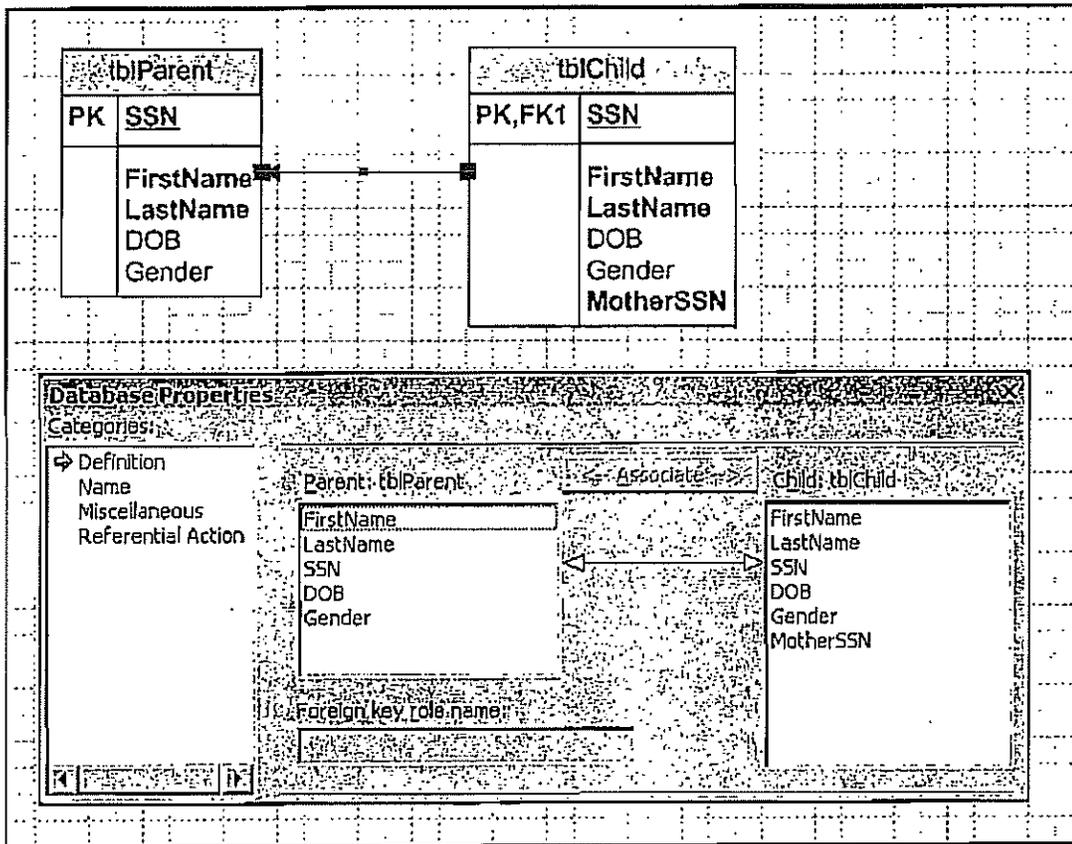


It is very important that you connect the tables in the proper direction. The arrowhead must be on the one-side (i.e., parent-side) and the tail must be on the many-side (i.e., child-side) of every relationship. (Remember: In Codd's relational database model, children point to their parents. Parents do not point to their children.)

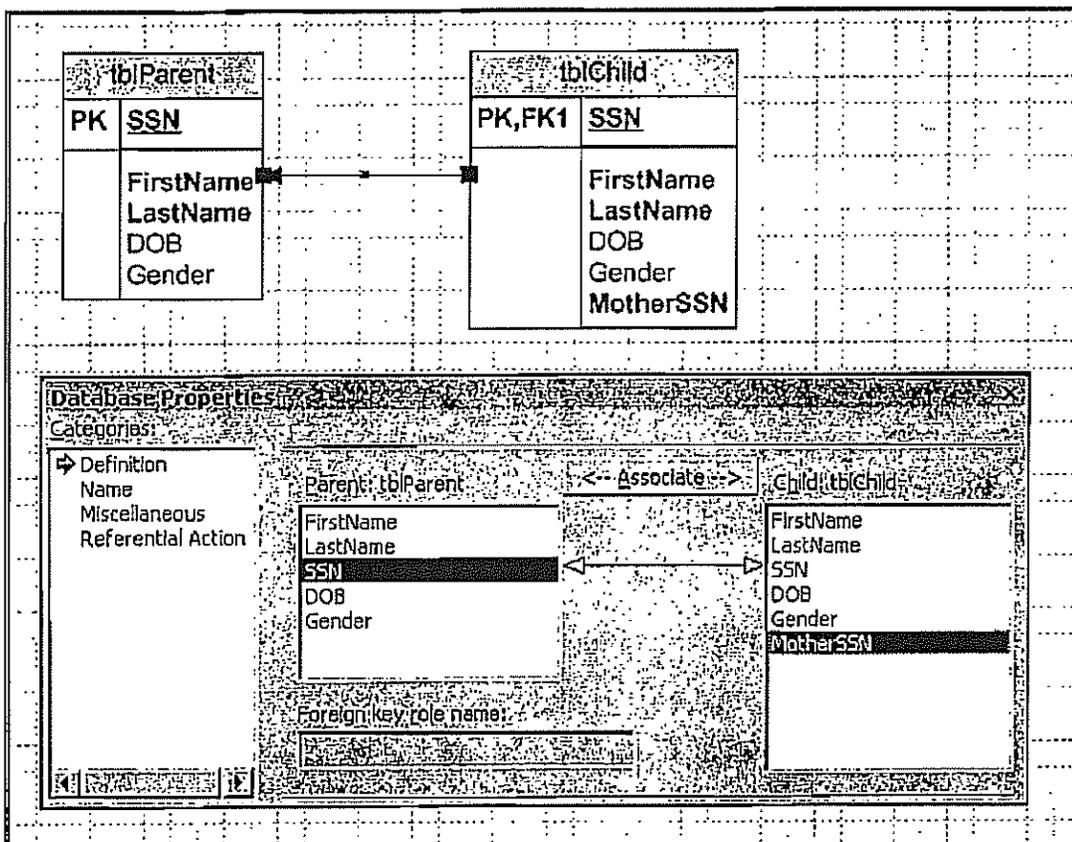


It takes a sharp eye to realize that something has gone wrong. Do you see it?

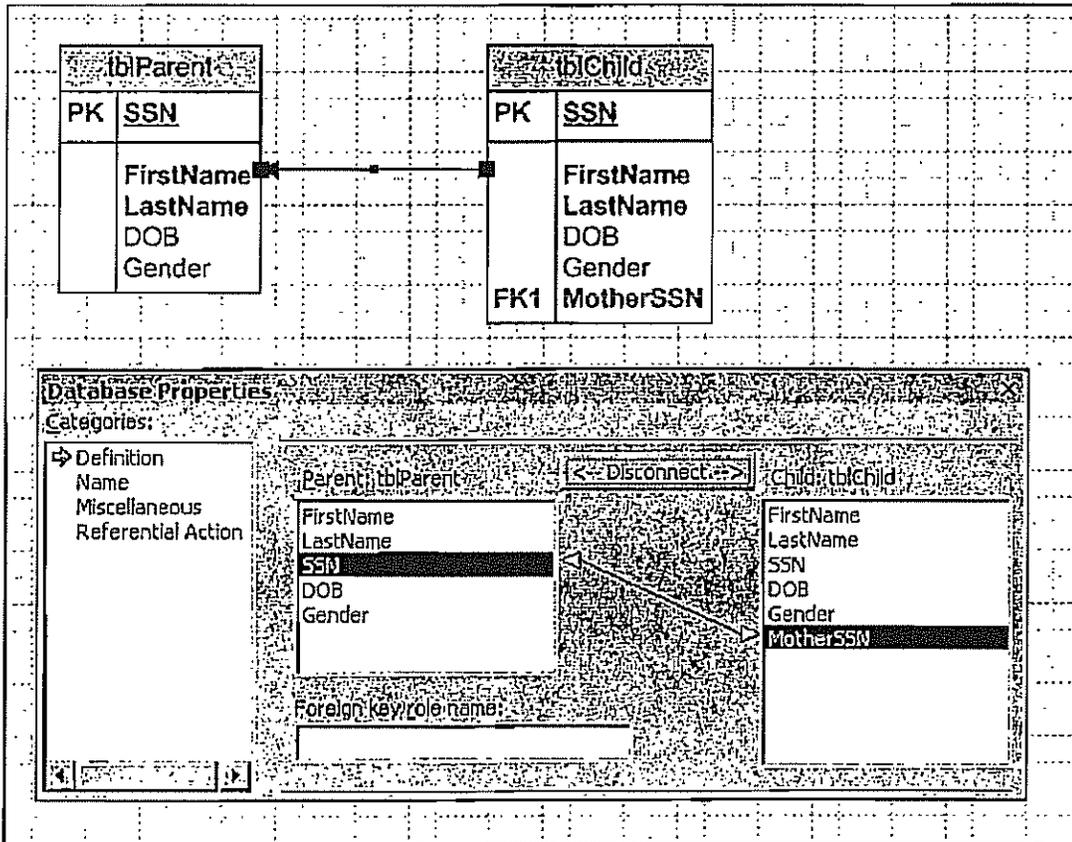
When you created the relationship between tblParent and tblChild, Visio assumed you wanted to use SSN as the matching fields. Unfortunately, that assumption is only half right. You want SSN in tblParent to match MotherSSN in tblChild. To correct this foreign key mistake, double click the relationship.



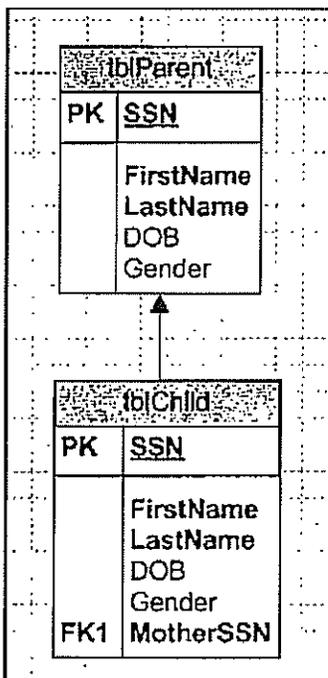
Highlight SSN in tblParent and MotherSSN in tblChild, then click the Associate button.



Now MotherSSN correctly appears as a foreign key.



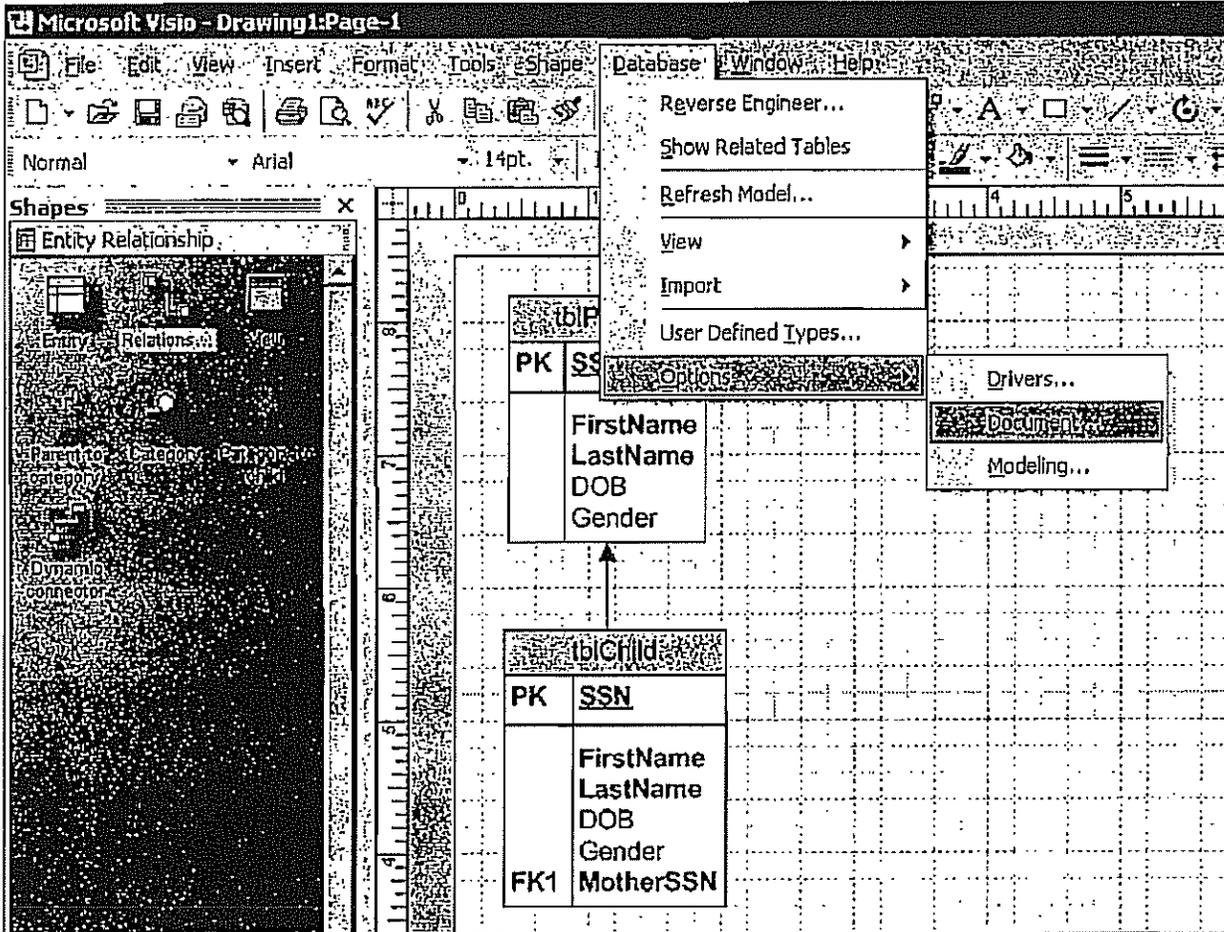
One of the nicest features of Visio is that you can reorganize your diagram by simply moving the entities around. The relationships are automatically adjusted for you. Reorganizing a diagram in PowerPoint or Word, on the other hand, can be very tedious.



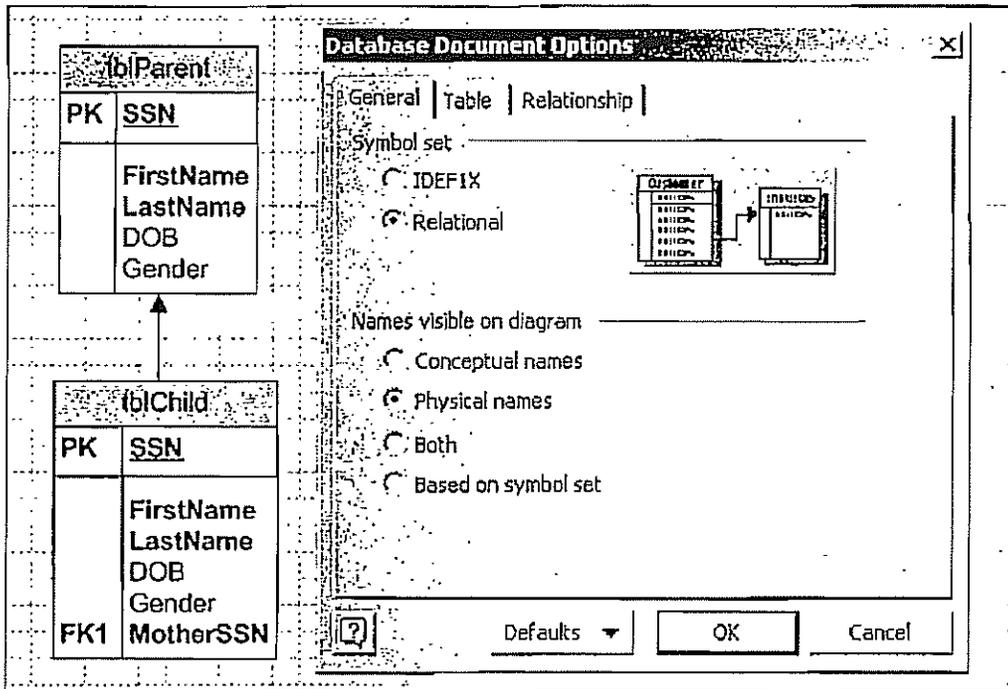
RS778

Fine-tuning the Drawing

Many of the basic drawing style options can be accessed by selecting the Database/Options/Document drop-down menu.

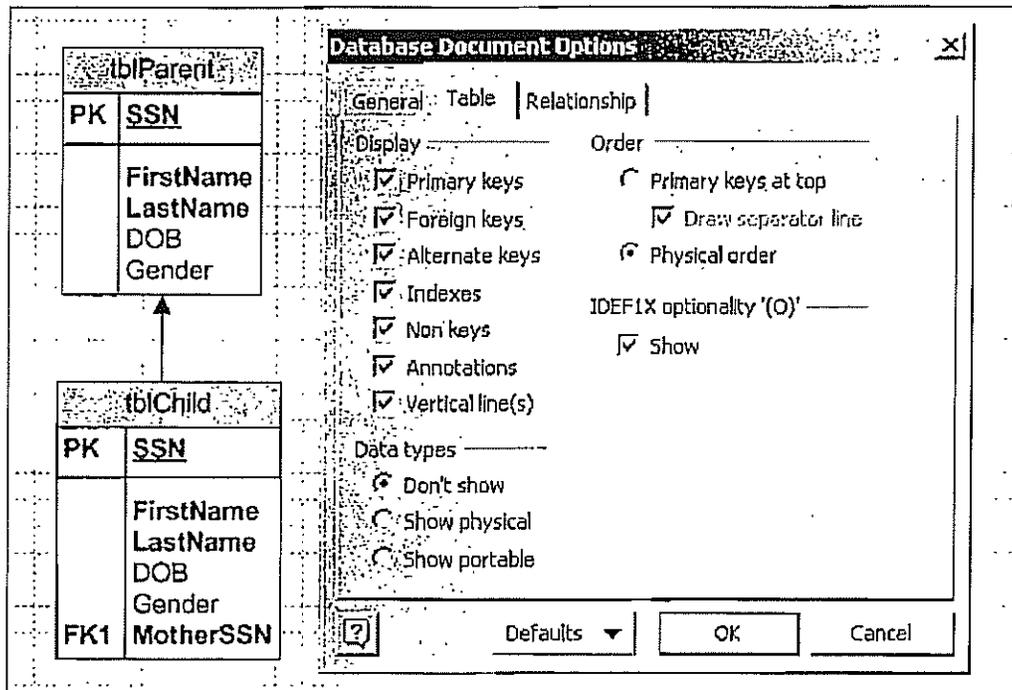


The General tab lets you select the IDEF1X or Relational symbols. It also lets you select which set of entity names to display.



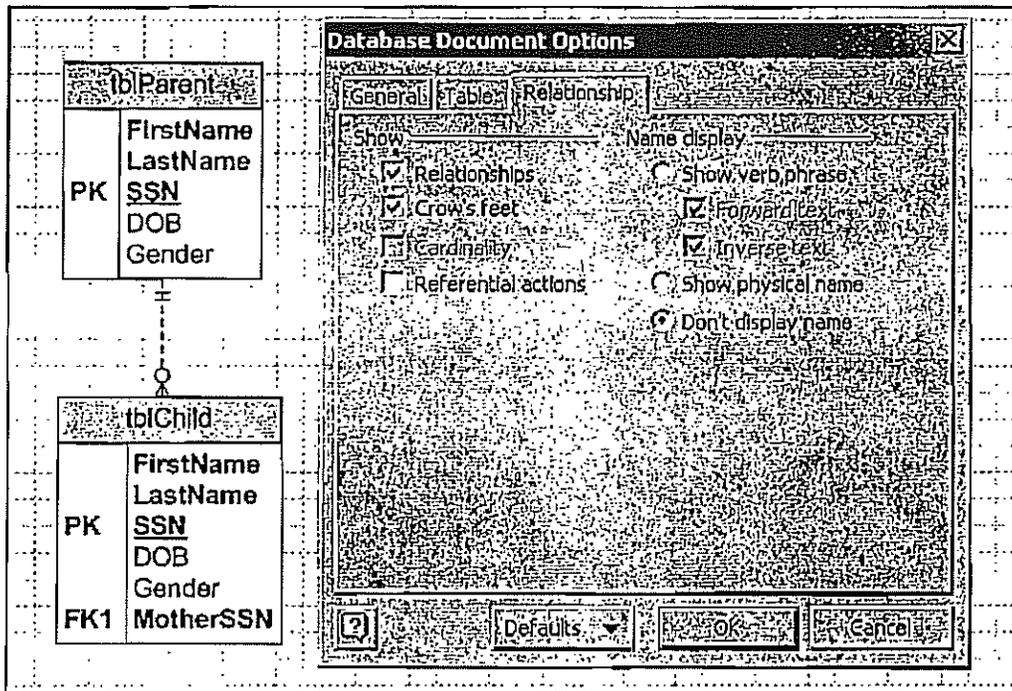
The Table tab lets you select the items to display (e.g., PK, FK, secondary indices, non-key fields, data types). You can also elect to display the primary key fields at the top of each entity or in their actual physical order.

Select the Physical Order option in order to have the two SSN fields displayed in their original (third) positions instead of at the top of tblParent and tblChild above the horizontal lines.

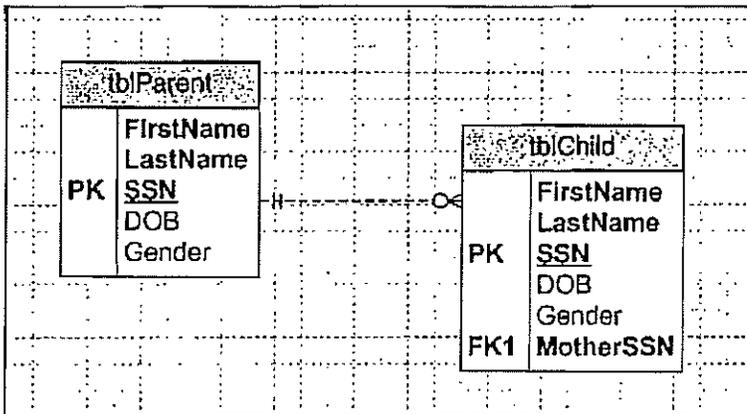


The Relationship tab lets you display the relationships as arrows or as crow's feet.

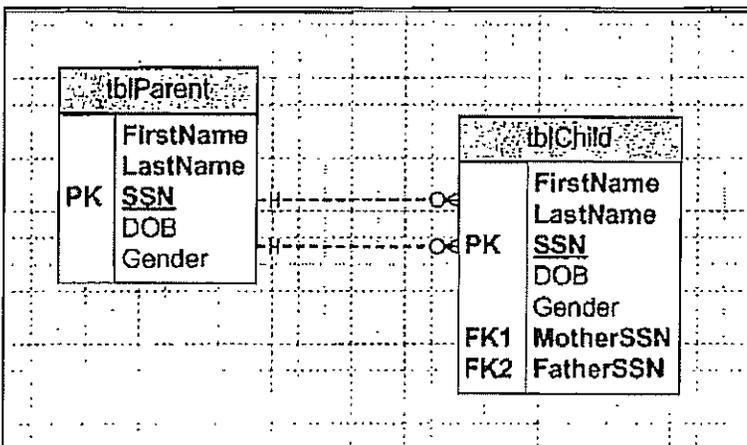
Select the Crow's Feet option. Notice that the mother-child relationship is shown as mandatory on the one-side and optional on the many-side. We will see how to change this later.



You can reposition the entities on the worksheet and the relationships will automatically be updated.

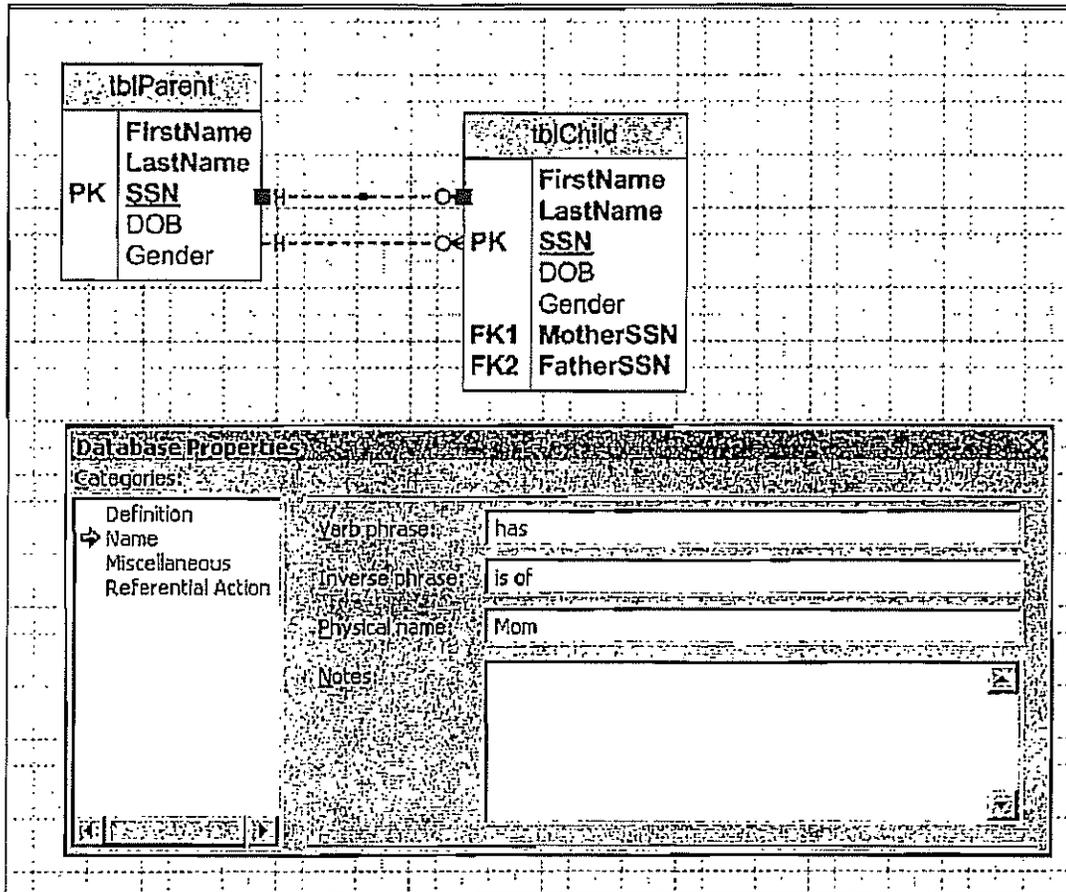


Create a new FatherSSN field in tblChild, then create a one-to-many relationship between SSN and FatherSSN.

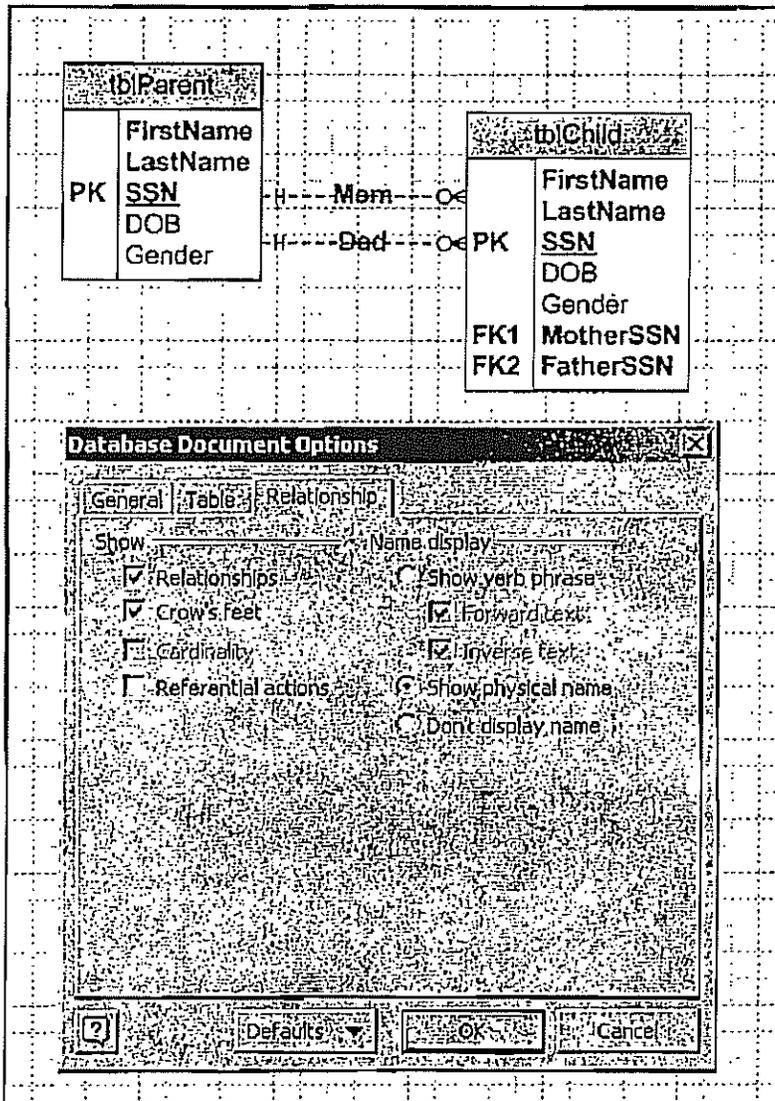


This illustrates one of the problems that Peter Chen tried to solve with his invention of ER diagrams. In particular, our drawing has two relationships that appear identical. We know that one represents motherhood and the other represents fatherhood, but the difference isn't particularly clear in the diagram. Chen proposed that all relationships should be labeled. Unfortunately, his diamond style labels tend to clutter a diagram.

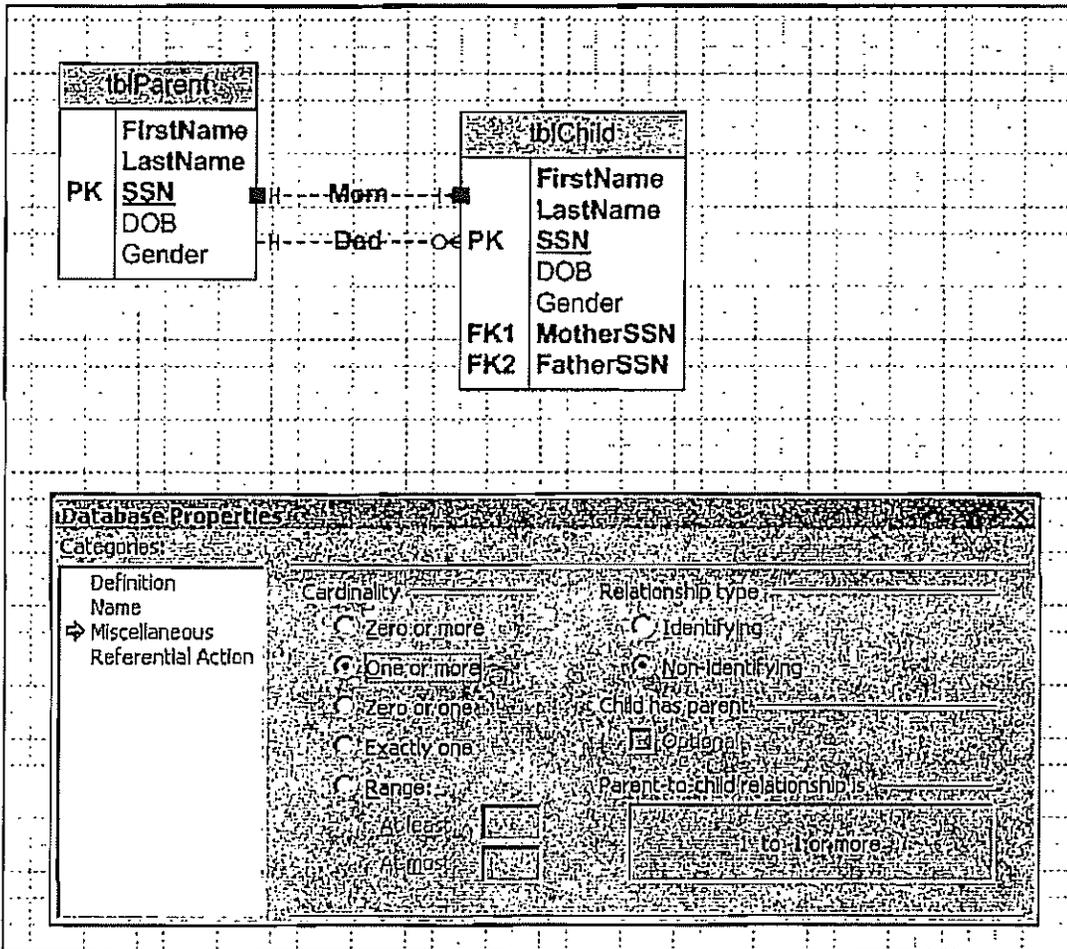
Visio offers a compromise. It lets you add simple labels to individual relationships. To do this, double click a relationship, then enter a descriptive name. In order to minimize clutter, you might want to use short labels like Mom and Dad.



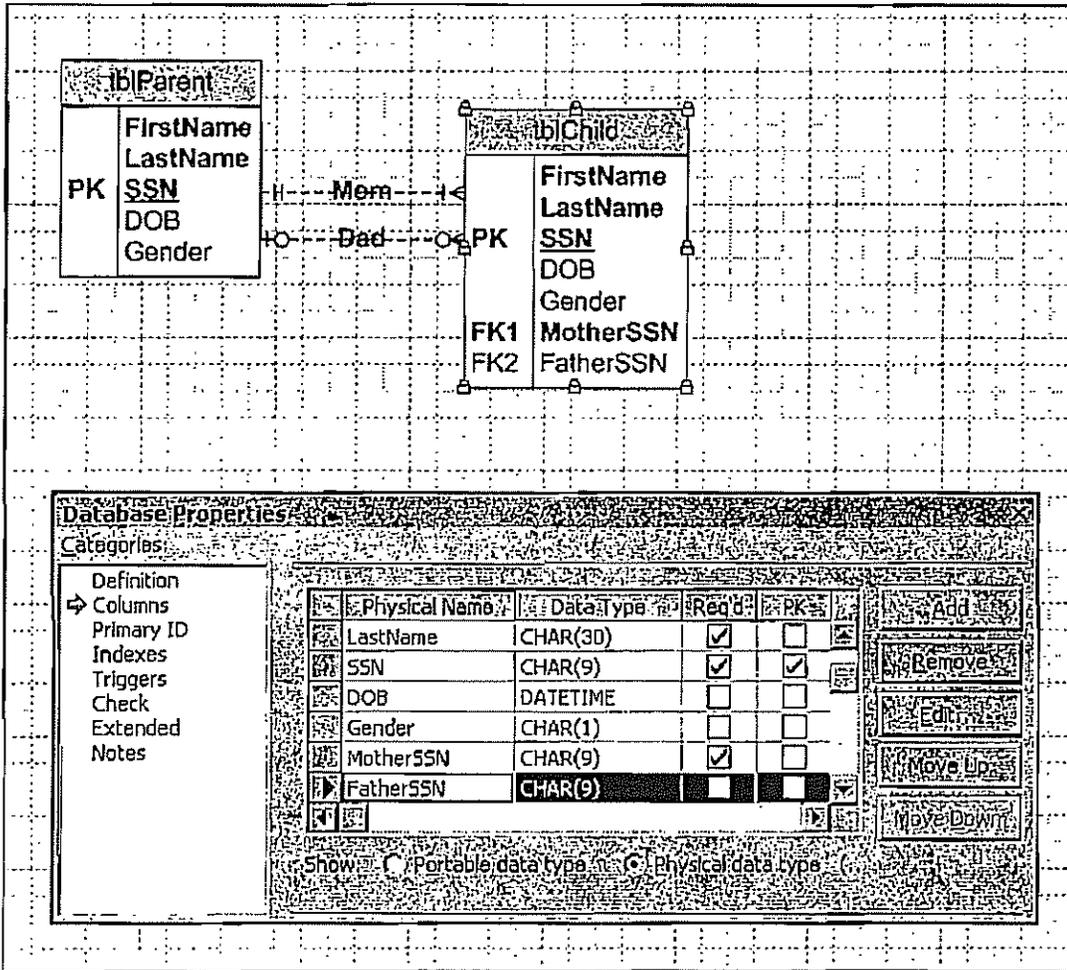
Open the Database/Options/Document drop-down menu, then select the Show Physical Name option from the Relationship tab.



The one-to-many mother and father relationships were created as mandatory on the one-side and optional on the many-side. You can change this by double clicking the relationship. For instance, you can force every **tblParent** record to have at least one child by changing the mother relationship to One or More cardinality. You can force every parent to have between two and five children by selecting the Range cardinality then specifying the At Least and At Most frequencies.

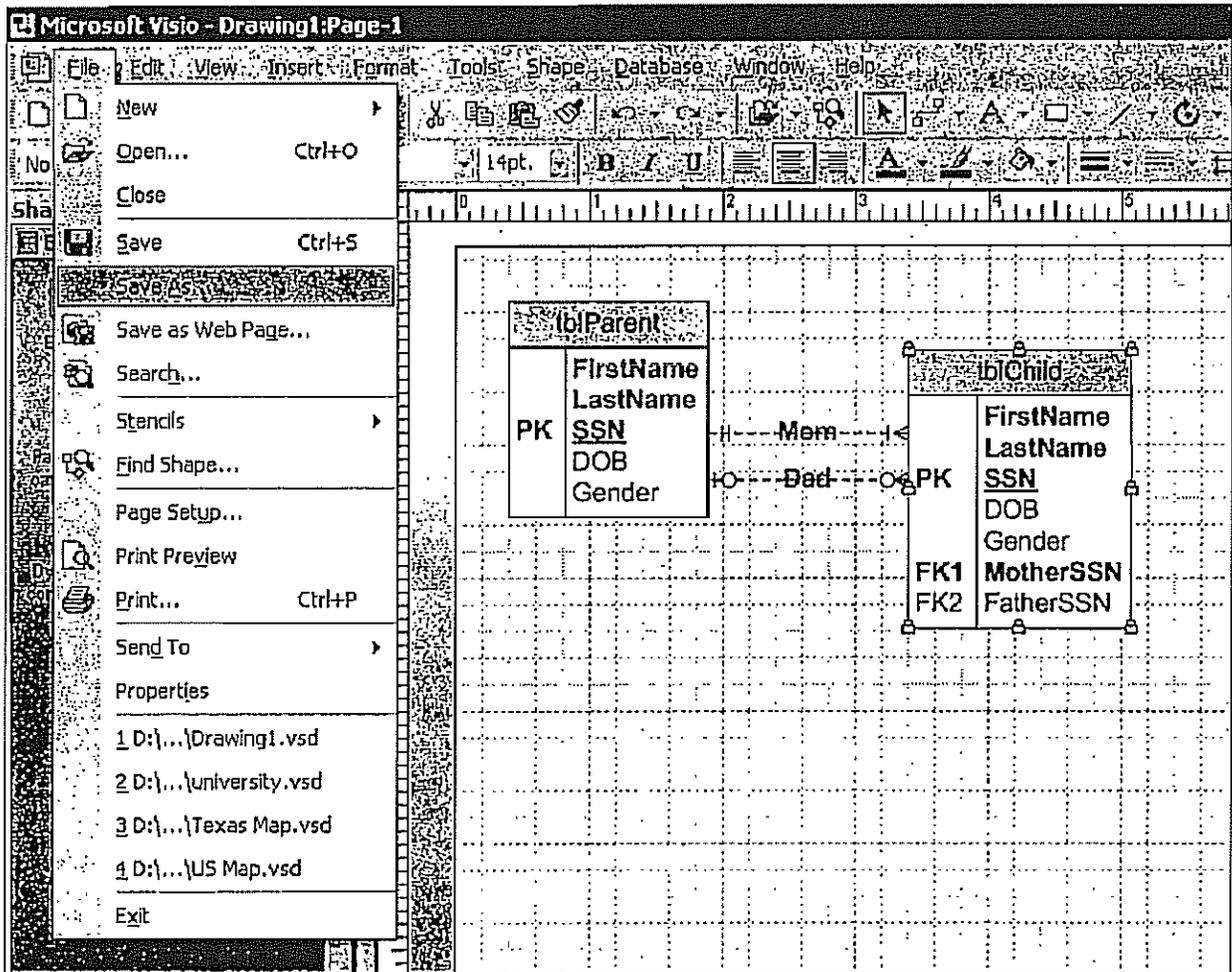


Initially, the mother and father relationships were defined as mandatory on the one-side because MotherSSN and FatherSSN were defined as required fields. If you want to allow a child to not have a father, then de-select the Req'd checkbox for FatherSSN. Now the relationship is optional in both directions.

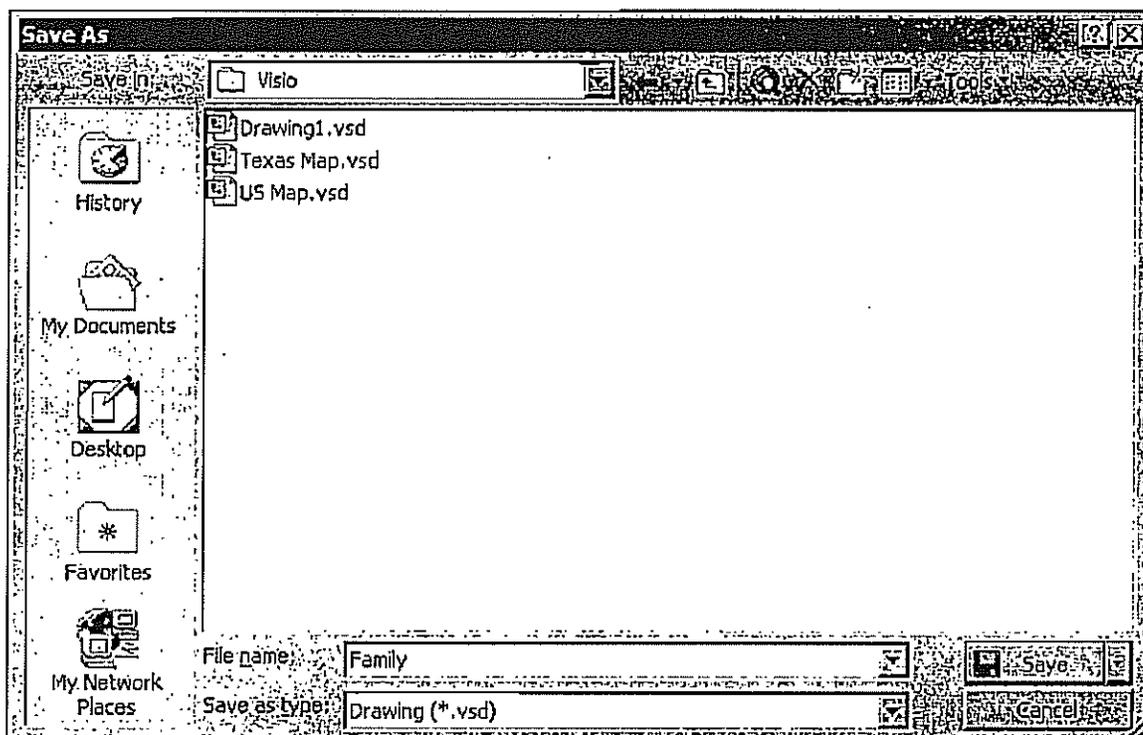


Wrapping Up

Select the File/Save or File/Save As drop-down menu when you are finished with your diagram.



Identify a subdirectory and file name for your new drawing. Note that Visio will use vsd as the extension for all drawing files. In this example, the new file will be named Family.vsd and it will be saved in the Visio subdirectory.



Lab Report Questions

Please answer and submit answers to the following exercises as specified below.

Use Visio Pro to create a crow's foot style ER diagram of a database for a typical online or mail-order retail business. The database consists of five tables and four parent-child relationships.

tblCustomer Table

Each customer record has a FirstName, LastName, CustomerID, Street, Apartment, City, State, PostalCode and Country field. The LastName, CustomerID, Street, City and Country fields are required (i.e., no missing data allowed). The other fields are not required. The CustomerID field is the primary key of the table.

tblSupplier Table

Each supplier record has a SupplierName, SupplierID, Street, City, State, PostalCode and Country field. All fields are required except State and PostalCode. The SupplierID field is the primary key.

tblProduct Table

Each product record has a Description, ProductID, SupplierID, Price, Weight and QOH field. All six fields are required. The ProductID field is the primary key and the SupplierID field is a foreign key pointing to the parent record in tblSupplier.

tblPurchaseMaster Table

Each purchase master record has a PMID, CustomerID, PurchaseDateTime, ShipDateTime, AlternateAddress, Street, Apartment, City, State, PostalCode and Country field. The PMID, CustomerID, PurchaseDateTime and AlternateAddress fields are required. The other seven fields are not required. The PMID field is the primary key and the CustomerID field is a foreign key pointing to the parent record in tblCustomer.

tblPurchaseDetail Table

Each purchase detail record has a PDID, PMID, ProductID and Quantity field. All four fields are required. The PDID field is the primary key. The PMID field is a foreign key pointing to the parent record in tblPurchaseMaster and the ProductID field is another foreign key pointing to the parent record in tblProduct.

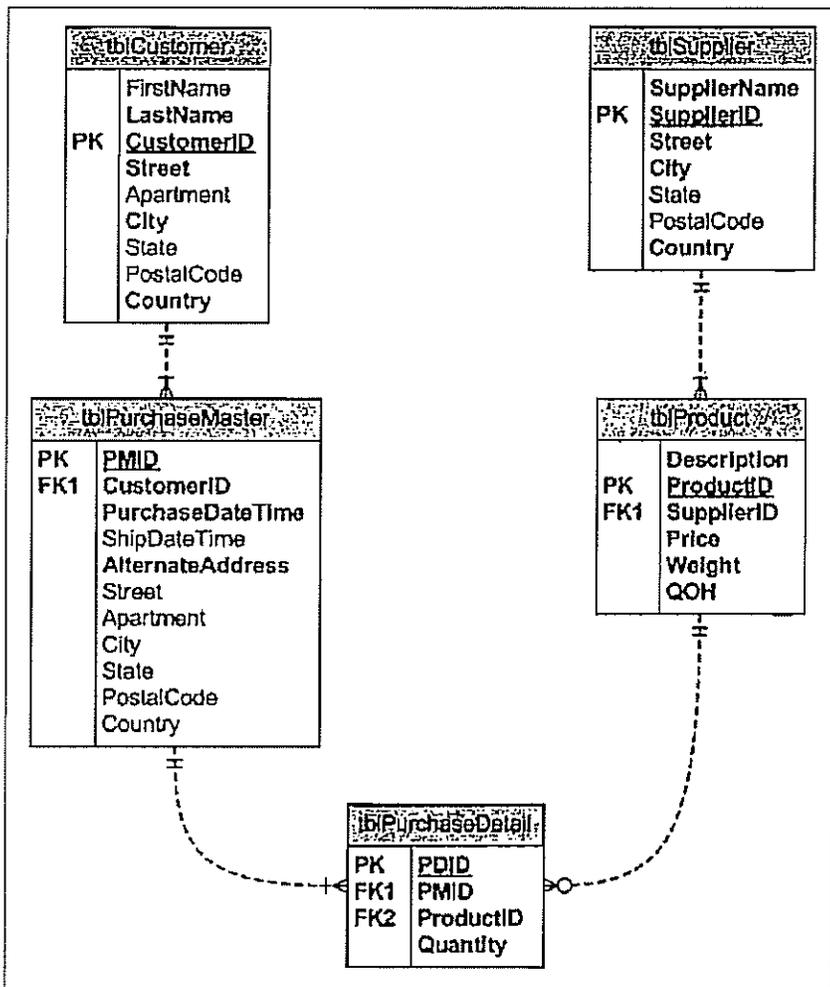
Relationships

RS787

There are four parent-child relationships as described by the four foreign keys given above. Each relationship has a mandatory parent (i.e., no orphans, referential integrity). Every tblCustomer record must have at least one related tblPurchaseMaster record (i.e., child). Every tblPurchaseMaster record must have at least one related tblPurchaseDetail record. Every tblSupplier record must have at least one related tblProduct record. It is possible, however, for some tblProduct records to not have any related tblPurchaseDetail records.

ER Diagram

This is the completed ER diagram for the database. Be sure you fully understand it.



Deliverable

Replicate this ER diagram in Visio. Use the same table and field names specified above. Use the crow's foot style for the representation of relationships. Be sure to correctly indicate the required/non-required status of each field and the mandatory/optional status of both "ends" of each relationship. Note that the relationship between tblProduct and tblPurchaseDetail is slightly different than the other three relationships. This is not a mistake. Grading in this lab may seem harsh at times. Be sure to do everything that the lab specifies. Since this assignment actually provides you with the correct solution (i.e., the diagram above), there is no excuse for a less than perfect score.

 **Hand In:** Please submit your Visio (.vsd) file via the dropbox on eCollege.

Introduction to Data Modeling

Introduction

Introduction

Data Modeling

- Overview
- E-R Model
- Database Design
- Data Objects
- Basic Schema
- Refining the E-R
- Primary Keys
- Attributes
- Hierarchies
- Integrity Rules
- Bibliography

This document is an informal introduction to data modeling using the Entity-Relationship (ER) approach. It is intended for someone who is familiar with relational databases but who has no experience in data modeling. The basic techniques described are applicable to the development of microcomputer based relational database applications as well as those who use relational database servers such as MS SQL Server or Oracle.

The document is a practical guide, not an academic paper on either relational database design or data modeling. Readers interested in a rigorous treatment of these topics should consult the **bibliography**.

Relational Model

- Overview
- Data Structure
- Notation
- Relational Table
- Relationships
- Data Integrity
- Relational Data
- Normalization
- Advanced

[Windows Services](#) > [Enterprise Solutions](#) > [Windows Database Services](#) > [Introduction to Data Modeling](#)

ITS Information Technology Services *Infrastructure. Innovation. Integrity.*



Last updated February 29, 2004.

Copyright © 1994-2010, Information Technology Services at The University of Texas at Austin.
All rights reserved. For privacy concerns read our [privacy policy](#).

To submit questions or comments regarding this page, use the [online Comment Form](#).

Welcome

Education & Experience

Papers Download

Courses

Honors & Activities

New Practical Applications

Photos

How to Contact



Welcome to the home page of Dr. Peter Chen (陳品山) at Louisiana State University where he holds the position of M. J. Foster Distinguished Chair Professor of Computer Science since 1983. Prof. Peter Chen received his Ph.D. from Harvard University and has held regular and visiting faculty appointments at MIT, UCLA and Harvard. He is the originator of the **Entity-Relationship Model (ER Model)**, which serves as the foundation of many systems analysis and design methodologies, computer-aided software engineering (CASE) tools, and repository systems including IBM's Repository Manager/MVS and DEC's CDD/Plus. After years of efforts of many people in developing and implementing the entity and relationship concepts, now "Entity-Relationship Model (ER Model)," "Entity-Relationship Diagram (ER Diagram)," and "Peter Chen" have become commonly used terms in "online" dictionaries, books, articles, web pages, course syllabi, and commercial product brochures.

At LSU, he is the Director of a research lab and also, a member of the top management team of the National Center for Security Research and Training (NCSRT), a large training and research organization with an annual \$50 million budget.

Dr. Peter Chen's original paper on the Entity-Relationship model (ER model) is **one of the most cited papers** in the computer software field. His original ER model paper was selected as **one of the 38 most influential papers in Computer Science** according to a survey of 1,000 computer science college professors (Table of Contents, *Great Papers in Computer Science*, edited by P. Laplante, West Publishing, 1996). It was selected for inclusion as one of the important publications in the "Science Pearls" project of Wikipedia. Based on one particular citation database, Chen's paper is the **35th most cited article in Computer Science**. It is the **4th most downloaded paper** from the ACM Digital Library in January 2005 (*Communications of ACM*, March 2005) even though the paper was published 30 years ago.

The ER model was adopted as the meta model for the **ANSI Standard in Information Resource Directory System (IRDS)**, and the ER approach has been ranked as the **top methodology for database design** and **one of the top methodologies in systems development** by several surveys of **FORTUNE 500 companies**.

Dr. Chen's work is a cornerstone of software engineering, in particular **Computer-Aided Software Engineering (CASE)**. In the late 80's and early 90's, **IBM's Application Development Cycle (AD/Cycle) framework** and **DB2 repository (RM/MVS)** were based on the ER model. Other vendors' repository systems such as Digital's CDD+ were also based on the ER model. Prof. Chen has made significant impact on the CASE industry by his research work and by his lecturing around the world on structured system development methodologies. Most of the major CASE tools including **Computer Associates' ERWIN**, **Oracle's Designer/2000**, and **Sybase's PowerDesigner** (and even a general drawing tool like **Microsoft's VISIO**) are influenced by the ER model.

The ER model also serves as the foundation of some of the recent work on **Object-Oriented analysis and design methodologies** and **Semantic Web**. The **UML modeling language** has its roots in the ER model.

The **hypertext concept**, which makes the World Wide Web extremely popular, is very similar to the main concept in the ER model, and Dr. Peter Chen is currently investigating this linkage. From 1999 to 2006, he served as an **invited expert** of several XML working groups of the World Wide Web Consortium (W3C).

Professor Peter Chen's work is cited heavily in a book published in 1993 for general public called *Software Challenges* published by **Time-Life Books** as a part of the series on "Understanding Computers".

Dr. Chen is a **Fellow of the IEEE**, the **ACM**, and the **AAAS**. He is a member of the **European Academy of Sciences**. He has been listed in *Who's Who in America* and *Who's Who in the World* for approximately 18 years.

He is the recipient of prestigious awards in several fields of Information Technology (IT): data management, information management, software engineering, and general information science/technology:

- The **Data Resource Management Technology Award** from the Data Administration Management Association (NYC) in 1990.
- The **Achievement Award in Information Management** in 2000 from DAMA International, an international professional organization of data management professionals, managers, and Chief Information Officers (CIO's). Dr. E. F. Codd (the inventor of the Relational data model) was the winner of the same award in 2001.
- Inductee, the **Data Management Hall of Fame** in 2000.
- The **Stevens Award in Software Method Innovation** in 2001, and the award was presented at IEEE International Conference on Software Maintenance in Florence, Italy on November 8, 2001.
- The **IEEE Harry Goode Award** at the IEEE-CS Board of Governors meeting in San Diego, February 2003. The previous winners of the Harry Goode Award include the **inventors of computers, core memory, and semiconductors**.
- The **ACM/AAAI Allen Newell Award** presented at the ACM Award Banquet in San Diego, June 2003. He was also introduced as the award winner at the opening ceremony at the 2003 International Joint Conference on Artificial Intelligence (IJACI-03) on August 11, 2003 in Acapulco, Mexico. The previous 7 winners of the Allen Newell Award include a Nobel Prize and National Medal of Science winner, 2 National Medal of Technology winners (one of them is also an ACM Turing Award winner), and other very distinguished scientists who either have made significant contributions to several disciplines in computer science or have bridged computer science with other disciplines.
- The **Pan Wen-Yuan Outstanding Research Award** in 2004 (Taiwan's prestigious high-tech award). Starting 1997, the awards have been given to usually three individuals each year (one in Taiwan, one in Mainland China, and one in "overseas" – outside of Taiwan and mainland China) in the high-tech fields (including electronics, semiconductors, telecommunications, computer science, computer hardware/software, IT, and IS). In 2003, the overseas winner was Prof. Andrew C. C. Yao of Princeton University, who is also a winner of the ACM Turing Award.

Prof. Chen is also listed in several "online dictionaries" such as:

- He ("Peter Chen") is one of approximately **80 persons in the computer and applied math fields (living or deceased, including John von Neumann, A.K. Erlang, Alan Turing, and Bill Gates)** listed in probably the most popular dictionary of computer words/terms: The Free Online Dictionary of Computing, edited by Denis Howe since 1985.
- He is listed in *Who's Who in Internet and Computer Technology* together with **founders of major computer and Internet companies** in Webopedia.com.
- He is listed in the popular free encyclopedia: Wikipedia.com.
- He is listed in the popular "general dictionary" for all fields: dictionary.com.
- His entries in the above dictionaries can also be found at: OneLook.com.

Dr. Peter Chen was recognized as a "software pioneer" in the "Software Pioneers" Conference, Bonn, Germany, June 27-28, 2001, together with a group of very distinguished scientists including winners of President's Medals of Technology, ACM Turing Awards, ACM/AAAI Allen Newell Awards, or IEEE distinguished awards such as Harry Goode Awards. The **streamed video and slides** of the talks in the "Pioneers" Conference may be available at the [conference website](#). All the speeches in the conference are documented in a book (with 4 DVD's) published by Springer-Verlag, and how to order the book can be found in the section on [Papers-Online](#).

The Entity-Relationship model is described in **most textbooks on databases, software engineering, and information systems analysis**. It is included as a fundamental topic in the **ACM/IEEE recommended curriculum on computer science and information systems**. Today, it is very likely to find at least one chapter on the ER model when a person randomly picks up a college textbook on information system design or databases. It is also very likely to walk into a college classroom to attend a class on information management and see that the ER modeling is being taught there. For example, at LSU, the ER model is being taught in 3 different colleges: the Computer Science department in **College of Basic Sciences**, the Information Systems and Decision Sciences Department in **College of Business**, and the Industrial Engineering and Manufacturing Systems Department in the **College of Engineering**. In other universities, the ER model is also taught in a variety of departments and colleges. For example, at Berkeley, the ER model is being taught in 2 or 3 courses at the **School of Information Management**. As another example, the ER model is being taught in the **Computational-Biology/bioinformatics programs** at University of Pennsylvania, Drexel, University of Virginia, and Hong Kong University. There are more examples of college courses covering the ER model.

From 2005 to 2009, Professor Peter Chen was a member of the U.S. [Air Force Scientific Advisory Board](#). From 2004 to 2006, he was a member of the Advisory Committee of the Computer and Information Science and Engineering (CISE) Directorate of the U.S. [National Science Foundation \(NSF\)](#). Dr. Peter Chen was a member of the **Airlie Software Council**, which consists of software visionaries/gurus and very-high-level software organization executives, organized by U.S. Department of Defense (DoD). He was an advisor to the President of Taiwan's largest R&D organization, [Industrial Technology Research Institute \(ITRI\)](#), with over 6,000 employees, which has been the driving force of Taiwan's high-tech growth in the past 3 decades. Since 2004, he has served as an advisor of [DAMA International Foundation](#), the advisory body of a large international professional organization of data management professionals and Chief Information Officers.

Dr. Peter Chen was one of 5 main US delegates to participate in the **1st IEEE USA-China Computer Conference**, which was held in Beijing in 1984, and to meet with PRC high-level leaders and government officers in the Science and Technology fields and the Education area.

Dr. Chen has been a keynote speaker for more than 30 international conferences and workshops. He had teamed up with an Executive VP of **Software A.G.** (one of the largest software companies) giving seminars around the world on information technologies.

Prof. Peter Chen is also the Editor-in-Chief of *Data & Knowledge Engineering*, the Associate Editor for the *Journal of Intelligent Robotic Systems*, *Electronic Government*, and other journals. In the past, he was the *Associate Editor* for *IEEE Computer*, *Information Sciences*, and other journals.

At MIT, UCLA and Harvard, Dr. Chen has taught courses in *Information Systems*, *Systems Simulation*, *Telecommunication*, *Computer Performance Modeling and Evaluation*, *Systems Analysis and Design*, *Database Management & Programming Languages*, and *Office of the Future*. At LSU, he has been doing research and teaching on *Information Modeling*, *Software Engineering*, *Data/knowledge Engineering*, *Object-Oriented Programming*, *Internet/Web*, *Java*, *XML*, *Data Warehousing*, *E-commerce (B2B and B2C)*, *Homeland Security*, *Identity Theft*, *System Architecture*, *Digital Library*, and *Intelligent Systems for Networking (Sensors Networks, Wi-Fi, and Cellular)*.

Professor Peter Chen is the Principal Investigator of a large NSF-funded multi-disciplinary project on profiling of terrorists and malicious cyber transactions for counter terrorisms and crimes. For more detailed information about

this project and its participants and publications, please browse the webpage: www.csc.lsu.edu/~chen/NSF-Project.htm .

He is also a Co-Principal Investigator of another large project on “**Secure Cyber Space Center**,” sponsored by Board of Regents (BoR) of Louisiana, LSU, and Louisiana Tech University. The research focus is on enforcing cyber security using sensor networks. One of the objectives of this project is to perform research work to support the provisional **Air Force Cyber Command** in the defense of cyber attacks.

In addition to the NSF project and the BoR project, Dr. Peter Chen has been the Principal Investigator or Co-Principal Investigator of various projects in system architecture, information/knowledge management, software engineering, and performance analysis sponsored by many government agencies and commercial companies.

He is also active in Digital library research and construction. He is a member of a group of experts led by Dr. Raj Reddy of Carnegie-Mellon University (CMU) to create a large digital library of over one million books. In this project, he serves as the **Executive Director of US-China million book project**.

More details of various honors and awards received by Prof. Peter Chen can be found in the section on Honors & Professional Activities.



- [Home](#)
- [Screenshots](#)
- [Downloads](#)
- [Documentation](#)
- [Support](#)
- [Licensing](#)
- [Contact](#)

- [Oracle](#)
- [DB2 iSeries](#)
- [DB2 LUW](#)
- [DB2 z/OS](#)
- [SQL Server](#)
- [Sybase ASE](#)
- [Sybase Anywhere](#)
- [Sybase IQ](#)
- [Teradata](#)
- [InCluster](#)
- [Informix](#)
- [PostgreSQL](#)
- [MySQL](#)
- [Apache Derby](#)

Entity Relationship (ER) Modeler, Database Diagrams and Database Design - Aqua Data Studio

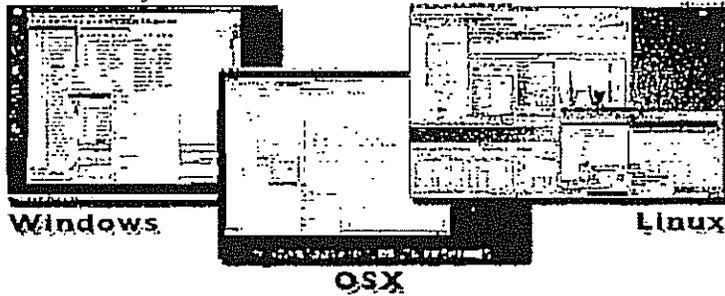
Aqua Data Studio offers an Entity Relationship (ER) Modeler for all major RDBMSes. It's a full-featured Schema Model tool for database administrators and database developers to diagram databases and visualize schema relationships. The Entity Relationship (ER) Modeler, advanced query tools, visual query building tools, schema compare tools and a broad range of features help database developers create ER diagrams and work efficiently and productively. Any database you connect to in Aqua Data Studio can be diagrammed with the ER Modeler!



For Windows, Linux and OS X, Aqua Data Studio's features and capabilities include:

- DB2 SQL Debugger
- Oracle SQL Debugger
- MS SQL Server Debugger
- Sybase SQL Debugger
- ER Modeler
- Visual Editing
- Subversion (SVN) client
- CVS client
- SVN Repository Browser
- CVS Repository Browser
- Autocompletion across DBs
- Server Side Comments
- Morph to Delimited List
- SQL Formatter
- Visual Explain Plan
- Procedure/Package Editor
- Script Generator
- Server Script Generator
- Execution Monitor
- Explain Diagram
- Import Tool
- Export Tool
- Object Search
- Parameterized Scripts and much more...

Aqua Data Studio ER Modeler



Benefits

Script Database Design Diagrams into Databases: Edit database structures in a database design diagram, modify schema objects, alter entity relationships and then automatically script the changes into a database from the ER model.

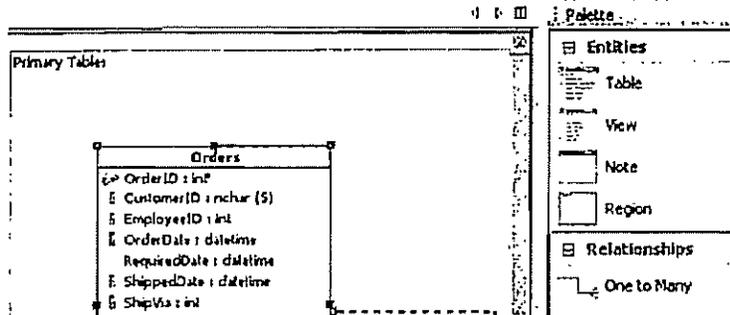
Share Database Design: With Aqua Data Studio's built in Subversion (SVN) and CVS clients, share your ER Models and database entity relationships with coworkers so they can collaborate on your database models in version control.

SQL (History) Archive: The SQL History Archive can automatically store every query run (over days, months and years) and can be searched and filtered later for easy reuse of queries. This archive is also extremely useful for auditing purposes (Sarbanes-Oxley Compliance).

Includes Database Administrator Tools For Oracle, DB2, MS SQL Server, Sybase and MySQL

Instance Managers: Provides manageability of

Entity Relationship (ER) Models



[Full Documentation](#)

[Community Mailing List](#)

[L337 Support](#)

[Who uses Aqua Data Studio?](#)

the database instance
allowing the user to view
and modify server
parameters and viewing
server version
information.

Storage Managers:
Provides manageability of
databases, database
devices, storage and dump
devices and caches,
allowing a user to
visualize and maintain
storage.

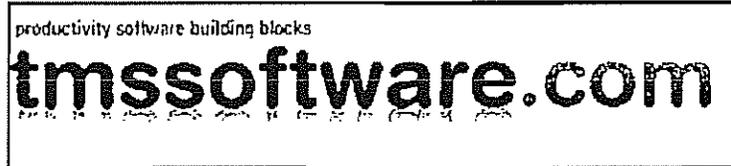
Security Manager:
Provides manageability of
logins, server roles, users
and roles allowing the
user to manage
permissions, roles and
security of databases.

Session Managers:
Provides manageability of
database sessions
including user locks which
allow the user to monitor
and kill sessions.

and much more!

- Home
- Products
- Blog
- Customers
- Consulting
- Support

73



Login



PRODUCTS » Quick Links

- Delphi, C++Builder VCL component sets
- Delphi, C++Builder VCL components
- .NET components
- ASP.NET components
- Silverlight - WPF components
- IntraWeb components
- ~~Develop Tools~~
- Windows Mobile components
- Free Tools
- ONLINE ORDERS (Your cart is empty)

FEATURED PRODUCT

- TMS Advanced Charts



DB-aware and non DB-aware feature-rich charting components for business, statistical, financial & scientific data

License only 75 EUR

[See More](#)

LOGIN

Customer login to access products, support information & special benefits.

Your registration e-mail Your registration code

[Forgot code?](#)

SEARCH

Search text 

- [Sitemap](#)

STAY IN TOUCH



Add your e-mail address here to receive the monthly TMS Software alerts.

Your e-mail address

[Subscribe](#) [Unsubscribe](#)

LINKS

- [TMS Partners](#)
- [Merchandise](#)
- [Resources](#)



[PRODUCTS](#) » [DEVELOPER TOOLS](#) [COMPONENTS](#) » [TMS DATA MODELER](#)

v1.5 (Aug 31, 2010)



SUPPORT

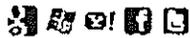
- [Contact Support](#)

TRIAL DOWNLOADS

- [Demo](#)

[Purchase a license](#)

☆BOOKMARKS



TMS Data Modeler

The ultimate tool for modeling databases with an easy and simple interface.

- [Overview](#)
- [Supported databases](#)
- [Features](#)
- [User comments](#)
- [Purchase a license](#)

TMS Data Modeler is the ultimate tool for modeling databases with an easy and simple interface. It provides powerful features you need, such as diagramming, reverse engineering, SQL script generation and version control. It keeps track of your changes and generates SQL scripts to update your database according to them. It has smooth integration with development tools like Delphi and C++Builder. And all of these features are accessible through a very pleasant user interface.

Overview

Design your database model

- Entity relationship diagram editor (ER diagram)
- Create tables in diagram and/or in tab forms
- Multi-diagram interface
- Project validation allows you to trace errors in your model
- Supports logical (application-level) domains
- Design tables, fields, domains, indexes, triggers, relationships, constraints, views, procedures, generators and other database objects
- Several database systems supported and more will be added

Database development lifecycle

- Reverse engineering of existing databases
- Generate SQL (DDL) scripts to create databases
- Keep track of your changes by archiving and managing versions
- Compare models between versions or between projects
- Generate Alter SQL scripts to update database according to changes
- Convert model to a different database system
- TMS Data Modeler can be integrated with Delphi and C++Builder

Simple and easy

- Automatic online updates
- Modern ribbon interface

- Tables and other objects organized in tabs, all accessible through one click
- Easy and quick navigation. Create tables and fields swiftly
- Multi-project interface in Microsoft Office 2010 style, with cutting edge design

Supported databases

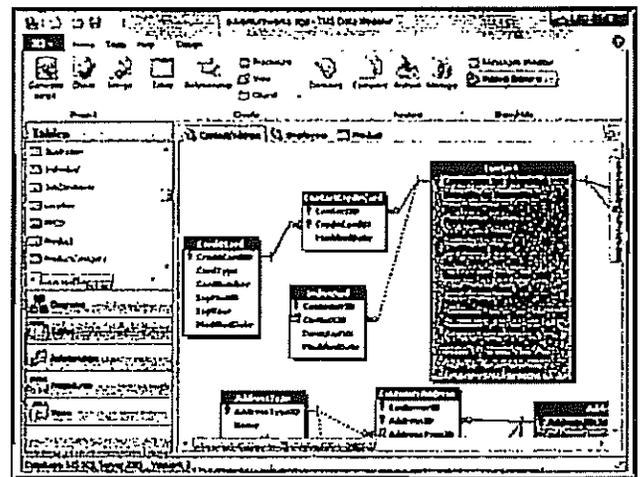


- Absolute Database
- Firebird 2
- MS SQL Server 2000
- MS SQL Server 2005
- MS SQL Server 2008
- MySQL 5.1
- NexusDB V3
- Oracle 10g

Features

Entity relationship diagram editor (ER diagram)

Design ER diagrams using Data Modeler. You can add tables to the diagram, see the relationships, change several display options, and much more.



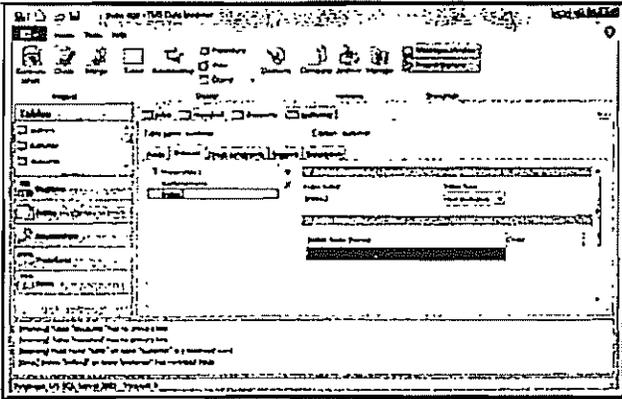
Create tables in diagram and/or in tab forms

If you don't want or don't need to model your database using diagrams, you don't have to. Data Modeler allows you to create and manage database objects directly, without having to deal with visual diagram designing. It's up to you.

Multi-diagram interface

Create different ER diagrams in a single project. You can put different tables in each diagram, making it easy to document and separate your application if it's too big.

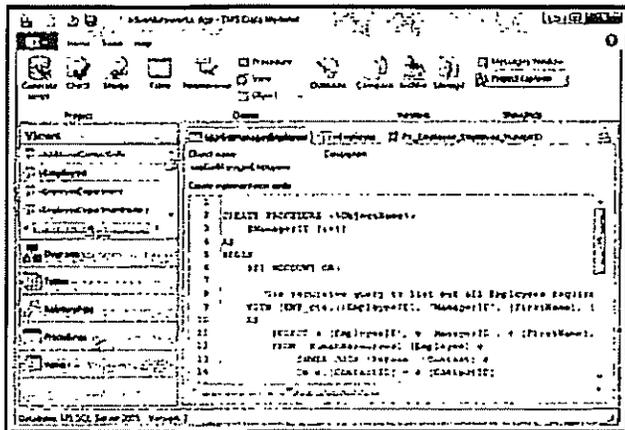
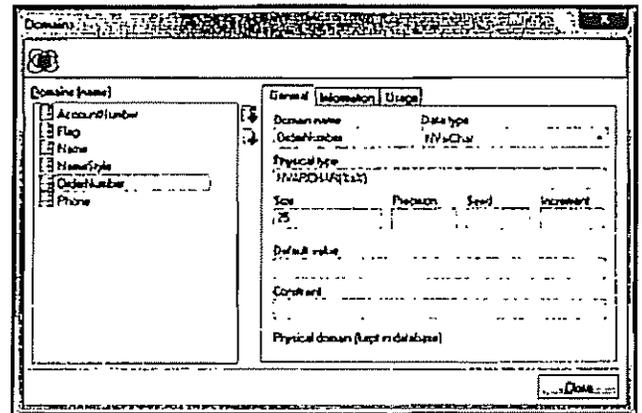
Project validation allows you to trace errors in your model



Data Modeler performs a validation in your database model checking for design errors. It also displays warnings about your project, such as usage of reserved words and tables with no primary keys, among others. This validation allows fine tuning of your database model, and avoids future problems after your software is in production.

Supports logical (application-level) domains

Use domains as an abstraction layer for your data types. Even if your database system doesn't support domains/user-defined types, you can do it at application level in Data Modeler and make it easy to maintain your databases.



Manage many database objects

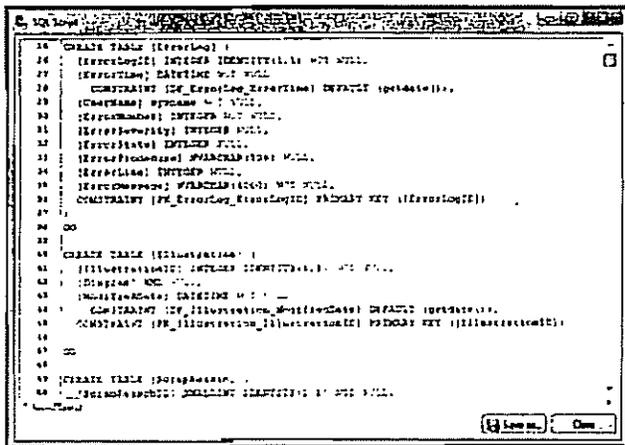
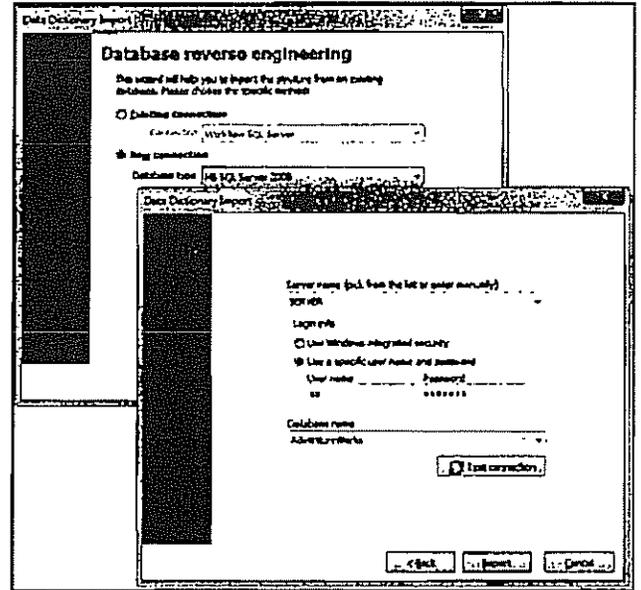
You can use Data Modeler to manage not only tables and fields, but several other database objects. Data Modeler supports indexes, triggers, relationships, constraints, views, procedures, generators, etc. Each database system has its own set of objects available.

Several database systems supported and more will be added

Data Modeler supports a number of database systems and new ones are in our roadmap to be included in future versions.

Reverse engineering of existing databases

Import your existing databases and start using Data Modeler quickly. The reverse engineering tool connects directly to your existing database and extracts all the information you need, such as tables, fields, foreign keys and other objects.

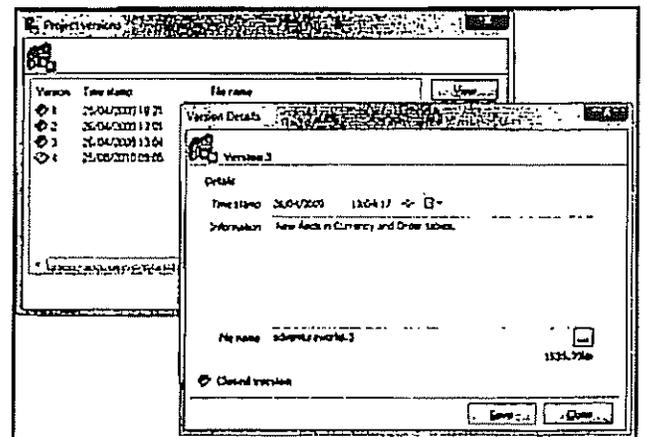


Generate SQL (DDL) scripts to create databases

Use Data Modeler to generate the SQL script that will create all your database objects. You can choose which object types to generate.

Keep track of your changes by archiving and managing versions

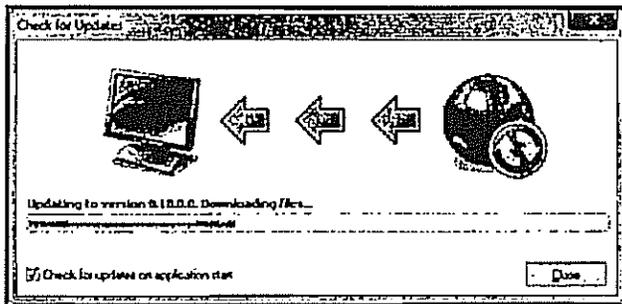
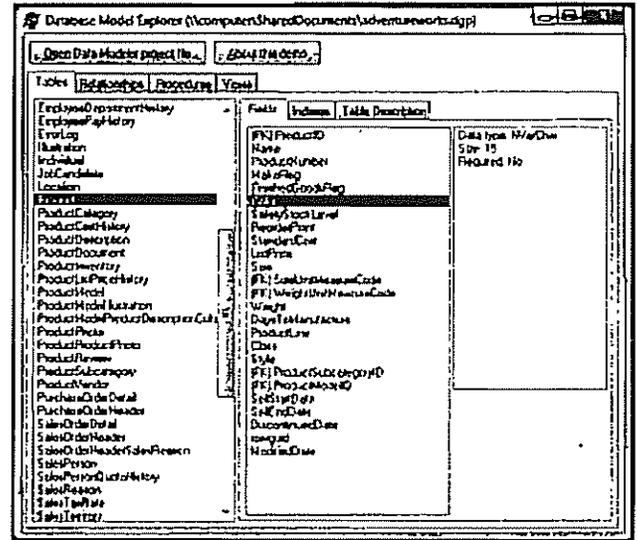
With Data Modeler you can archive project versions/add checkpoints. This way you can have the history of all development stages of database modeling, and also compare between versions and see what has changed from one version to another



Compare models between versions or between projects

TMS Data Modeler can be integrated with Delphi and C++Builder, thanks to the component package TMS Data Modeler Library.

TMS Data Modeler Library (DMLib) is a collection of classes that allows you to read the whole structure of a database (tables, fields, procedures, indexes, etc.) from a TMS Data Modeler project file, using a Delphi / C++ Builder application. After modeling your database with TMS Data Modeler, you can use TMS Data Modeler Library classes to have the database structure information available in your Delphi application.

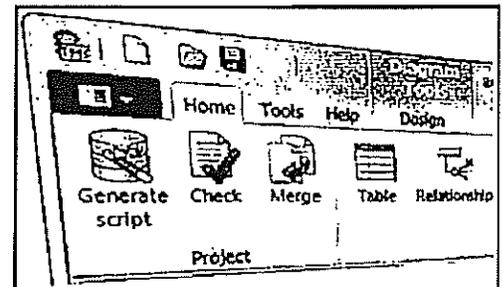


Automatic online updates

Like any modern software, TMS Data Modeler checks for new versions in the internet and auto-updates online. You will always have the latest version of your software.

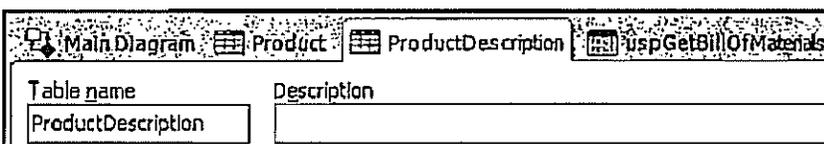
Modern ribbon interface

Data modeler is one of the few database modeling tools that offers the modern ribbon interface. It's much easier to use, nicer to look, and all options and features are available with less menu clicks.



Tables and other objects organized in tabs

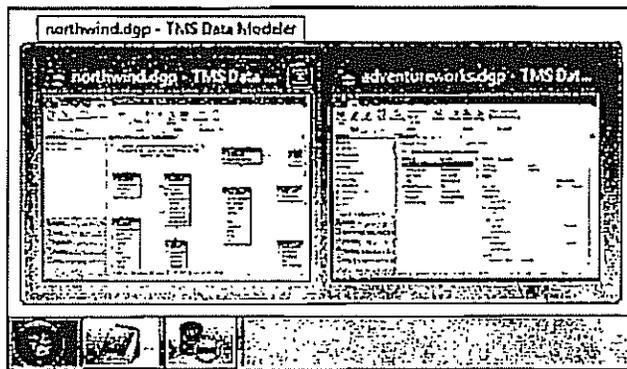
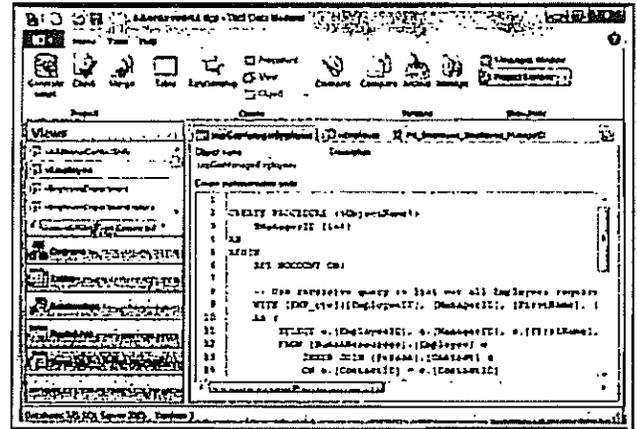
Don't edit your tables, fields and objects in modal forms or windows that overlap each other. Most tables and objects in data modeler are displayed in tabs, making navigation very similar to browsers like Firefox and Internet Explorer. You can have



many objects open at the same time, all accessible through one click.

Easy and quick navigation. Create tables and fields in a glance

Everything in Data Modeler was designed to be easy. Objects are displayed in tabs, and creating and editing fields don't require you to open another window. Everything is simple, clean and accessible through keyboard in a single window. Create your tables and fields quickly.



Multi-project interface

Open several projects at the same time. And you don't need to deal with MDI interfaces. Every project has its own taskbar button and you can easily switch between projects using regular Windows Alt-Tab function.

User comments

I am very happy with your product, it meets just about all of my needs and is great value for money.

Mike Hamilton, via email

- Your product is by far the most pleasant tool to use. No other product even comes close to its ease of use.
- It is currently the easiest to use especially for those who neither want or need to have to spend lots of time learning some of the competitor's products.
- Data Modeler makes this an almost pleasant experience where you can work with a minimum of technical knowledge or be able to delve deeper where necessary.

Bill Marshall, via email

Buy online



TMS Data Modeler Single developer license for commercial use



free updates for a full version cycle (from current version to v2.4) and free priority support: **95 EUR**

Please choose your language and [order immediately](#) or [add to cart](#)

Copyright © 1995 - 2010 TMS Software v3.3

[Home](#)
[Products](#)
[Live demo](#)
[Download](#)
[Order](#)
[Support](#)

Toad™ Data Modeler 3

Database design and entity relationship diagrams (ER diagrams)

Reverse engineering

SQL code generation

HTML and RTF reports

Maintenance

Customization

Supported database systems

Requirements

What's new in v. 3.6

Toad™ Data Modeler BETA

BETA Program - expect more...

Toad™ Data Modeler 3 - database design tool

This software is for all developers, database designers and for all who want to create and maintain their database structures effectively. Key features include:

- Entity relationship diagrams - both Physical and Logical modeling (incl inheritance)
- Support for various databases (Oracle, MS SQL Server, MySQL, Sybase Ase, PostgreSQL, DB2)
- Generation of SQL (DDL) scripts
- Reverse engineering
- Generation of Alter Scripts
- Model Update and Model Merge
- Generation of detailed HTML and RTF documentation
- Conversion from Logical to Physical model
- Support for Unicode
- UNDO / REDO
- Modeless dialogs, user-friendly GUI
- Editable forms, scriptable objects...

Entity relationship diagrams - ER diagrams

Using the graphically very well arranged Entity Relationship Diagrams (ER Diagrams) you can easily create and maintain your database structures. In your models you will have all the entities, attributes, domains, primary keys, foreign keys, constraints, relationships, indexes, descriptions, notes and other physical and logical data, laid-out in a transparent order. While creating models you will have a perfect view of all database elements

Triggers, Procedures, Views....

Toad™ Data Modeler supports Functions, Procedures, Triggers, Views, Packages, Package bodies, Object types, Object type bodies, Sequences and Synonyms etc



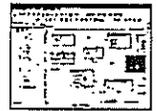
Physical ERD



Logical ERD



Modeless dialogs



In-place editing



Loupe and Overview panes



Changeable notation

English, Czech

Copyright 2000 - 2010 Quest Software Inc. CHARONWARE, s r o
 Uletlova 267-5, Ostrava, 760 36, Czech Republic (European Union)
 Http://www.queststudio.com Database design and modeling tools

[More](#)

Testimonials

Fred Hildenbrand:

"I have purchased MySQL Maestro and like it very much, it has made it much easier to deal with DBA maintenance and table replication as well as a quick way to create SQL statements with the graphical query tool. Good job on a fine product".

Padster Reynolds:

"Good software that has become one of our must have tools - wouldn't like to be without it these days. Keep up the good work".

[More](#)

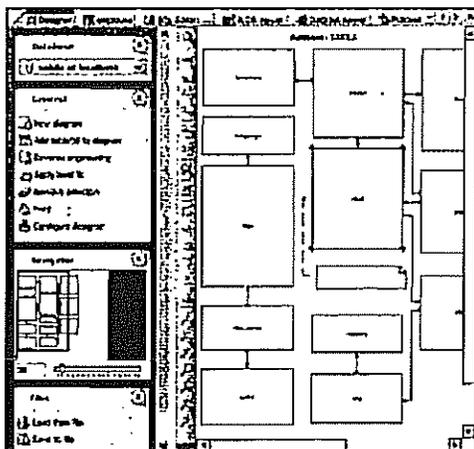
[Add your opinion](#)

Customers



[More](#)

[Become a customer](#)



[Click to enlarge](#)

aligning are also available.

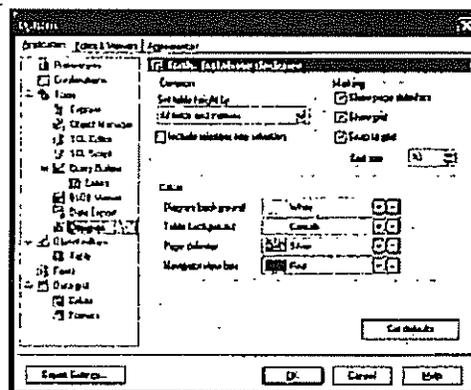
Besides the Database Designer allows you to **Load a diagram from file**, **Save to file**, and **Save as image** (Bitmap, GIF and JPEG formats are supported).

All the ER Diagram elements are customizable.

You can set table height option to resent it on the diagram as caption only, key fields, etc.

You can enable showing of the **grid** and **snap to grid function**. Then diagram will be covered by points disposed on the same interval between and when you will move the table or other object, its upper left corner "snaps" to the nearest grid point. Interval between grid points can be changed.

The **Colors** options are to attune the Designer appearance to your taste.



[Click to enlarge](#)

[Back](#)

[Next](#)

Products Resources Solutions Forum Support Partners Registered Users Enterprise Architect UML Tutorial About Us

See Also

Work with HDG Technologies
 Import HDG Technologies
 Extensions - MDG Technologies
 Archimate
 Data Flow Diagrams
 Entity Relationship Diagrams (...
 Eriksson-Penker Extensions
 GoF Patterns
 ICONIX
 Mind Mapping
 Sc4HL

Index Modeling Languages Specialized UML Models MDG Technologies - Using Entity Relationship Diagrams (ERDs)

Entity Relationship Diagrams (ERDs)

Note:

Entity Relationship Diagrams are supported in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect.

The following text is derived from the Entity Relationship Model entry in the online Wikipedia:

An entity-relationship model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created by this process are called Entity-Relationship Diagrams, ER Diagrams, or ERDs.

For further information on the concepts of Entity Relationship Diagrams, refer to the Wikipedia item and its linked sources.

Entity Relationship Diagrams in Enterprise Architect

Entity Relationship Diagrams in Enterprise Architect are based on Chen's ERD building blocks: entities are represented as rectangles, attributes are represented as ellipses and relationships are represented as diamond-shape connectors. ERD technology in Enterprise Architect assists you in every stage from building conceptual data models to generating Data Definition Language (DDL) for the target DBMS.

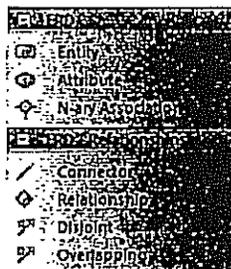
Enterprise Architect enables you to develop Entity Relationship diagrams quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect Installer. The Entity Relationship diagram facilities are provided in the form of:

- An Entity Relationship diagram type, accessed through the New Diagram dialog
- An Entity Relationship Diagram page in the Toolbox
- Entity Relationship element and relationship entries in the Toolbox Shortcut Menu and Quick Linker.

Enterprise Architect also provides transformation templates to transform Entity Relationship Diagrams into Data Modeling Diagrams, and vice versa.

Entity Relationship Diagram Toolbox Page

You can access the Entity Relationship Diagram page of the Toolbox through the More tools | Entity Relationship Diagrams menu option. The following icons are available:



- **Entity** is an object or concept that is uniquely identifiable. The property of *Multiplicity* in the SourceRole and TargetRole definitions for the *Relationship* connector (below) can be used to define the cardinality of an Entity that participates in this relationship.
- **Attribute** is a property of an entity or a relationship type.
- **N-ary Association** represents unary (many-to-many recursive) or ternary relationships and can also be used to represent relationships that have attributes among the entities; Note that the N-ary Association element should always be at the target end of a connector.
- **Connector** is a connector between an Entity and an Attribute, and between two Attributes.
- **Relationship** is a diamond-shape connector, representing the meaningful association among entities.
- **Disjoint** and **Overlapping** represent the relationships between the super-class Entity and the sub-class Entity.

Tagged Values

Some of the Entity Relationship diagram components can be modified by Tagged Values, as indicated below:

Component	Tagged Value	Notes
Entity	IsWeakEntity	If true, this entity is a weak entity.

RS810

Attribute attributeType Four options:

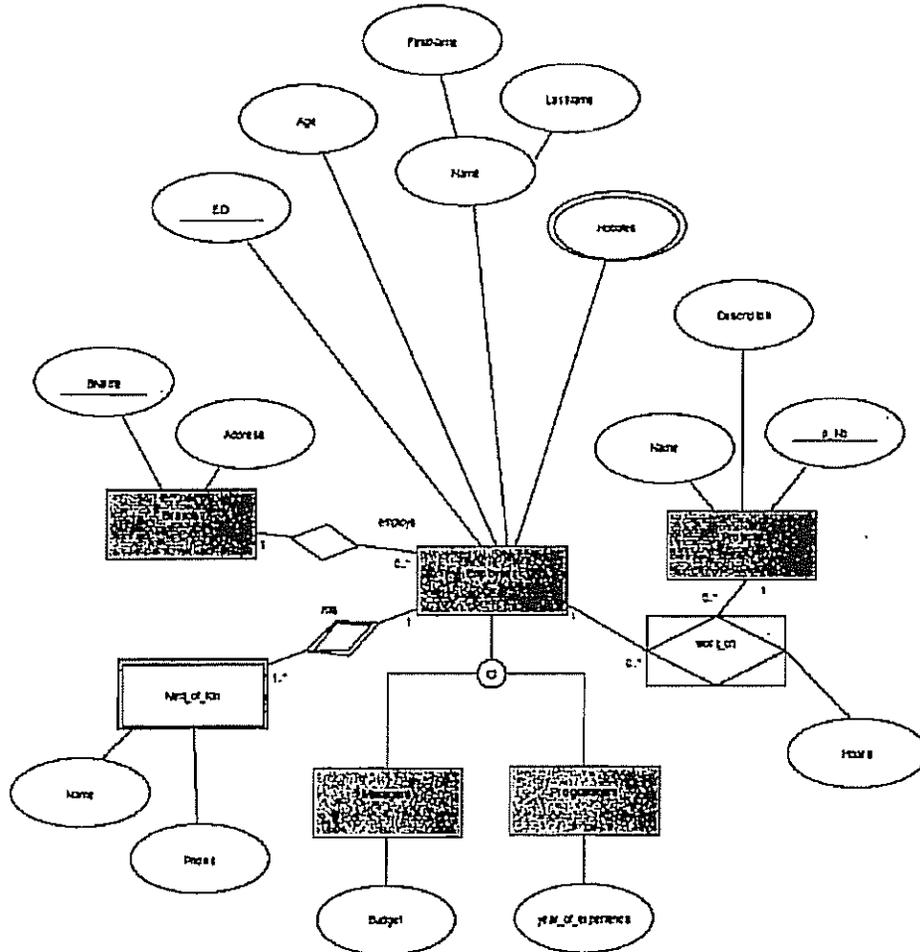
- normal Attribute
- primary key attribute
- multi-valued Attribute
- derived Attribute.

commonDataType Defines the common data type for each attribute.

	dbmsDataType	Defines the customized DBMS data type for each attribute. Note: You must define the customized type first through the Settings Database Datatypes menu option. Also, set the <i>commonDataType</i> tag to <i>na</i> to activate the <i>dbmsDataType</i> tag.
N-ary Association	isRecursive	If true, the N-ary Association represents the many-to-many recursive relationship. For one-to-many and one-to-one recursive relationships, we suggest using the normal <i>Relationship</i> connector.
Relationship	isWeak	If true, the Relationship is a weak relationship.
Disjoin Overlapping	Participation	Two options: <i>partial</i> and <i>total</i> .

Diagram

A typical Entity Relationship Diagram is represented below:



RS811

Tip:

Sometimes you might want to limit the stretch of the diamond-shape Relationship connectors. Simply pick a Relationship connector, right-click to display the context menu, and select the **Band Line at Cursor** option.

Disable Entity Relationship Diagrams

If you prefer not to use Entity Relationship Diagrams in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (**Settings | MDG Technologies**).

The material ("material" includes all pages, documents and on-line graphics) on this web site is protected by copyright law. (C) 1998-2010 Sparx Systems Pty Ltd.



Products

Enterprise Architect
Eclipse Integration
Visual Studio Integration
SysML Technology
B2M Technology
GDSPs Link
Additional Addons

UML at a Glance

UML
UML Tools
DIP UML Modeling
Business Process Modeling
Model Driven Arch. Solution
Requirements Management
Software Development

Solutions

Corporate
Government
Small/Medium Enterprise
IT Professionals
Trainers
Academic

Resources

UML 2.0 Tutorial
MDG Technologies
Corporate Resources
Developer Resources
Media Resource

Support

Online Manual
User Forum
Report a Bug
Feature Request
Change Your Email

Global Partners

Trainers
Value Added Resellers
Resellers
Sister Companies
Technical Partners
Standards Organizations

Home > Products > VP-UML > What VP-UML Provides

Visual Paradigm
for **UML 8.0**
Model-Code-Deploy Platform

[Buy Now](#) [Try it Free](#) [Get a Free Community Edition](#)

[What VP-UML Provides?](#) [What's New?](#) [Tutorials](#) [Editions](#) [Resources](#)

What VP-UML Provides?

-  UML Modeling >
-  Requirements Capturing >
-  **Database Design** >
-  Business Process Modeling >
-  Mind Mapping >
-  Modeling Toolset >
-  Team Collaboration >
-  Diagramming Toolset >
-  Code and DB Engineering >
-  Impact Analysis >
-  Report Generation >
-  Interoperability & Integration >
-  Supported Standards >
-  Screenshots >

Database Design



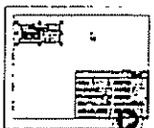
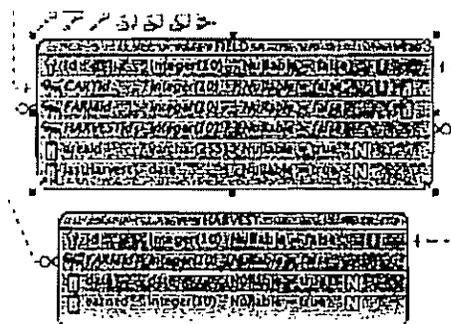
Design database with entity relationship diagram. Generate UML class diagram.

Share |    

Entity relationship diagram

Entity relationship diagram (ERD) is used to present the Entity-Relationship Model in software engineering industry. ERD can be used for visualizing conceptual data structure and physical database schema. Visual Paradigm for UML supports conceptual, logical and physical data modeling.

The following movie shows how to create conceptual data model and transit the conceptual data model to physical. The movie shows also the modeling of physical database schema, stored procedure and generates DDL from physical database design.



Tutorial

- Drawing entity relationship diagram

User's Guide

- Drawing entity relationship diagram

Online training

- Basic database modeling technique (Free)
- Database modeling with Oracle (Paid)

Know-how

- Pinning open specification in ERD

VP-UML Quick Start
Download PDF
1.9 MB
August 25, 2010



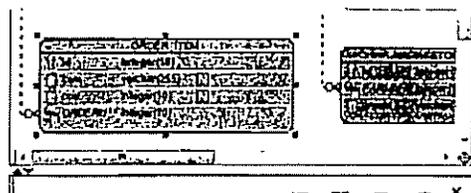
VP-UML User's Guide Online



VP-UML Datasheet
Download PDF
1.0 MB
August 16, 2010

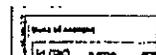


Visual Paradigm



Define sample data in ERD **NEW**

Input sample data or initial database data when design database with Entity Relationship Diagram (ERD). The generated database initialization script includes insert sample data to the database.



RS813



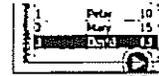
product adoption
roadmap
Download PDF
1.0 MB
August 17, 2010

Visual Paradigm International
User Meeting 2010
November 9, 2010 - Munich Germany

Register NOW

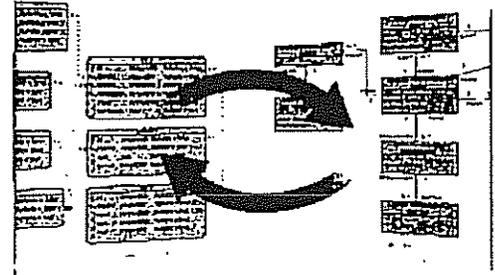
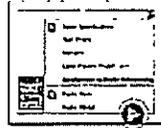
Tutorial

- Specifying default data in database design



Synchronization between ERD and class diagram

Entity relationship diagrams (ERD) presents persistent structure of the database, and class diagrams presents object structure in memory. There is alignment between ERD and class diagram. For example the column in entity can map to attribute in class. Visual Paradigm for UML supports generating class diagram from ERD and synchronize changes from ERD to class diagram, and vice versa.



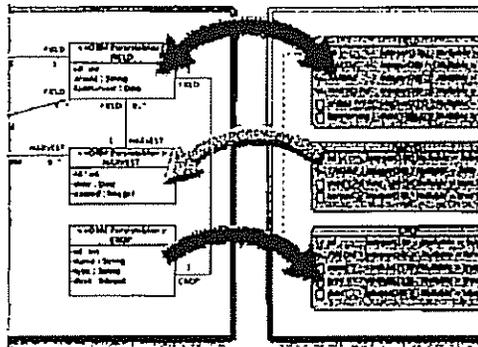
Tutorial

- Generate class diagrams from entity relationship diagrams

User's Guide

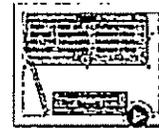
- Generate class diagram from ERD

Class Diagram Entity Relationship Diagram



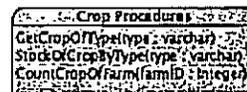
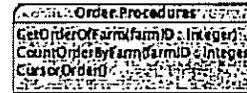
Object-Relational Mapping diagram

Object-relational mapping diagram (ORM diagram) presents the mapping between classes and entities visually. Visualization between classes and entities can help developer to gain more understanding of how a row of data in database transform to/from object in memory. The following movie shows how to visualize the mapping between classes and entities.



Stored procedure and trigger

Visual Paradigm for UML can design, reverse and generate stored procedures and database triggers. Flexible grouping helps to group stored procedures and triggers in meaningful way.



User's Guide

- Drawing stored procedures

← Requirements Capturing

Business Process Modeling →

Tell us what you think?

Support

- Contact Sales Team
- Contact Support Team
- Discussion Forum

Products

- Business Process Visual ARCHITECT
- Agilan
- Smart Development Environment
- Visual Paradigm Teamwork Server
- DB Visual ARCHITECT
- A* Platform

All rights reserved. [Legal](#) [Privacy statement](#)

[Site map](#) | [About Visual Paradigm](#) | [Hot News \(RSS\)](#) | [Contact us](#)

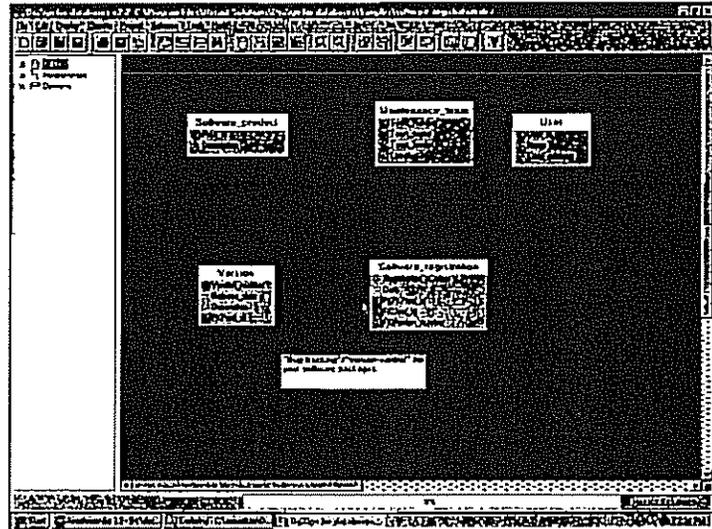
DATABASE DESIGN

Index > Other 3rd Party Tools > Software Tools > DeZign for Databases

DeZign for databases to create ER diagrams

Topics

- Summary
- History
- What is great about this product
- Supported Operating Systems and Versions
- Installation
- What it does
- Who should use it?
- Competitive products
- Detailed review
- Shortcomings
- Cost and where to buy
- Support



Click to enlarge

Rating



19-March-2001
Author: Ashutosh_Gaur

Summary

For a database to support the activities of a business, it needs a good blueprint and a sound foundation: the Data Model. Data modeling (usually) occurs in three phases: conceptual design, logical design, and physical design. The conceptual design phase uses an entity relationship diagram (ERD) to graphically represent the business' data and information requirements and is the most crucial phase of the software development life cycle.

An entity-relationship diagram is a technique that creates a graphical representation of the entities, and the relationships between them, within an information system. For more information on Entity-Relationship Diagrams, visit

- http://devworld.apple.com/techpubs/webobjects/System/Documentation/Developer/EnterpriseObjects/DevGuide/Apple_ERMd.html
- <http://panoramix.univ-paris1.fr/CRINFO/dmrg/MEE/misop003/>

Datanamic provides the database designers with an effective tool, DeZign for Databases, to create ER diagrams and then, to generate schema scripts for the same; effectively taking you through all the design phases to arrive at a physical design. It's an excellent tool, especially for database programmers who want to consider database designing as the next step in their professional development.

The tool is capable of creating schema scripts for a variety of target databases including Oracle, ANSI Level2, dBase, IBM DB2, Informix, Ingres etc.



Usefulness to developers	★★★★
Functionality, how much does it do	★★★★
User interface, intuitive/friendly	★★★★★
Software quality, integrity, robustness	★★★★★
Documentation quality and scope	★★
Technical support availability	★★★★
Value for money	★★★★
Ease of integration with other tools and systems	★★★★★
Overall rating	★★★★

- [Back to top of file](#)
-

History

DeZign was developed by Rick Van Dijk from Netherlands. The tool was developed so as to keep the task of data modeling as simple as possible for database architects. As a result DeZign was kept very simple and intuitive. One doesn't even need to go through the 'Readme' file (I could just start off with the tool once it was installed).

- [Back to top of file](#)
-

What is great about this product

DeZign provides the database designers with an effective tool to create ER diagrams and to generate schema scripts. It's an excellent tool, especially for database programmers who want to consider database designing as the next step in their professional development.

Since the tool is capable of creating schema scripts for a variety of target databases including Oracle, ANSI Level2, dBase, IBM DB2, Informix, Ingres etc. there have been some compromises made to the functionality provided but these generally do not act as an impediment for most database designs.

- [Back to top of file](#)
-

Supported Operating Systems and Versions

DeZign runs on all 32-bit Windows platforms including win95 / 98 / 2000 / me and NT. Unix / Linux are not supported yet. "DeZign for databases" requires a minimum of 4MB RAM and will run on a 486 processor. Disk space required is 4 MB.

- [Back to top of file](#)
-

Installation

The licensed version of the program can be downloaded from Heraul's web site here. Installation is pretty simple and straightforward. Once downloaded from Internet, installation takes just a matter of minutes.

- [Back to top of file](#)
-

What it does

DeZign provides a design environment that is quite intuitive. The Help is well organized and can act as a ready reference.

Coming back to the application, there is a large space kept out for the actual diagrams and that makes it easier to work on, compared to Oracle Designer ER Diagrammer. Simplicity of the user interface keeps the designer more focussed.

The tool comprises of numerous utilities and aids such as

- **An entity relationship diagram (ERD) editor** : Simple and efficient compared to other similar tools. What I liked most was the provision of a large amount of customizable space for the actual design.
 - **Support for ER modeling techniques** : Modeling techniques related to the entity relationship design are well supported. There are some functions that are absent but they're the ones that are generally not required.
 - **Fast graphics manipulation** : This is one area where DeZign outperforms many other database design tools. Those of you who've worked with Oracle Designer will get a welcome break.
 - **Automatic schema-generation (DDL)** : This capability is more or less required by any database design tool, and is present in DeZign as well.
 - **Full control over schema-output** : The DDL generated are flushed to a text file that can be edited before running the script to create the database.
 - **Ability to define indexes** : Indexes can be easily defined during the design phase itself. The user interface for index generation is very simple and intuitive. Simple or composite indexes can be generated with ease.
 - **Automatic foreign key migration at design-time** : Parent-child relationship among entities is automatically converted to referential integrity constraints (Foreign Key Constraints) when the DDL scripts are generated.
 - **Automatic resolution of many-to-many relationships** : Many-to-Many relationships are automatically resolved into two one-to-many relationships and the DDL for the third table gets automatically generated in the script.
 - **Template driven schema-creation** : The physical database (the SQL script) is generated based on the contents of the template files (*.tem). You can edit it in notepad since it is a normal text file. If you need to adhere to some standardization or a naming convention, all you need to do is to modify this file. The same technique is used for the reports (*.rep files). Working this way makes it very easy to support a new database. All you need to do is to create / modify the *.tem file based on your target database.
 - **User Defined data types** : Users can define their own datatypes that can then act as domains. These domains can then become the datatype for attributes of different entities defined in the diagram.
 - **Design Version Control** : With the version-control system you can maintain several versions of your diagram. You can switch between versions making any of the previously saved versions as the current one.
 - **Reporting** : The tool comes with a powerful report engine. Reports can be generated as text/HTML files. Reports detailing the changes made to the data model across versions are very useful. The entire ER diagram can also be saved as a bitmap, GIF file or as windows metafile.
-
- [Back to top of file](#)

Who should use it?

Database architects can benefit most from DeZign. It is also a great tool to database administrators who run the scripts. Even the design changes made over time can be easily incorporated and viewed diagrammatically.

This tool can act as an excellent launch pad for database programmers who wish to take to database design as a career advancement move. It is very simple to use and the reference help is straightforward enough to understand.

- [Back to top of file](#)
-

Competitive products

The major competitor for DeZign is 'ERWin' from Computer Associates. It does more or less what DeZign is capable of. However, DeZign has a far better reporting capability that is very essential. The number of supported databases by ERWin is far greater than those supported by DeZign.

The second major competitor is Oracle's Designer/2000. Though the Designer/2000 is a much bigger and complex application compared to DeZign, it doesn't have much support among the database architect community. Further, it doesn't support databases other than Oracle. Designer/2000 assists in the entire software development life cycle, right from database design to front-end GUI generation. However, this tool is extremely slow even on fast processors with sufficient amount of RAM.

PowerDesigner from Sybase is yet another competitor in this field. However, this again, like Designer/2000, is a very 'Heavy' application. One cannot just start off with modeling databases without going through the user manuals and/or Help.

DeZign scores better than all these applications if the purpose of purchase is just database design. It's cheaper than ERWin and costs peanuts compared to Designer/2000 or PowerDesigner.

- [Back to top of file](#)
-

Detailed review

A detailed review of the current day GUI applications would easily fill up any book. The same holds for DeZign. There are so many things this simple looking application can perform that it's actually quite difficult to summarize. I've selected a few important features that are extremely useful:

ER Modeling : This is the major function performed by DeZign. Some advanced features like resolution of many-to-many relationship and 'weak entities' are also supported. Once the design architecture is complete, scripts can be generated for any of the supported target database for creation of tables, constraints and indexes.

Version Control : DeZign helps you maintain the versions of your database designs. Older versions are automatically saved. Any version can be made the current version to generate the schema. This way you can compare between different schema scripts generated for each of the design.

Domains : You can create user defined datatypes that can then be used to define the datatypes for attributes of entities. If a change is made to the domain definition, all attributes under this domain (or rather, having this domain as their datatype) are automatically changed. Data characteristics can thus be standardized across the database with minimal efforts.

Help File : The help file acts more like a reference book and guide rather than the usual context sensitive help. It consists of detailed documentation about the ER diagram, entities, relationships, cardinality, domains, indexes etc.

Schema Generation : Once the database architecture is complete, scripts can be automatically generated. 'Create Table' statements create the tables with details about the check constraints, unique constraints, indexes, primary and foreign keys. Even the scripts for various 'Domains' defined in the database model are generated based on the nature and type of the target database. All the datatypes of various attributes in the ER diagram get automatically converted to the corresponding datatype in the target database. A 'Drop Table' script is also generated simultaneously. This is specifically required if there is a change/modification to the earlier design version for which the script was generated using DeZign.

Reports : DeZign has a powerful reporting engine that can generate reports of varying levels of complexity. The generated reports can either be in Text or HTML formats. The reports can also be used to compare the differences between two design versions.

Supported Databases : DeZign supports schema generation capability for quite a few different databases like

- ANSI SQL2
- dBase
- DB2 (IBM)
- Informix
- Ingres
- Interbase
- MS SQL Server
- MS Access
- Oracle
- Paradox
- PostgreSQL
- SQL-Anywhere
- Sybase

"DeZign" uses open type mapping. You can perform the datatype mapping from what you have defined in the diagram to the actual supported datatype of the target database from within the application.

The "auto number"-data type will result in a trigger-definition with a sequence, trigger-definition with a generator, default value, datatype or a special code for the target database.

- [Back to top of file](#)
-

Shortcomings

There are certain limitations with the current version of DeZign. Absence of a 'Search' facility in 'Help' is disappointing. The one

provided is documentation style Help.

Supertype and Subtype entities are not supported. Though there is a workaround for such entities and the relationship that exists among them, the ability of the application to support them would have been beneficial in designing complex business applications.

The 'Arc' relationship is also not currently supported. However, DeZign Application Architect, Rick van Dijk tells me that future versions will have these features.

Tablespaces and Storage parameters aren't supported at the moment but will be available in a future release. The number of entities and relationships that DeZign can support depends on your internal memory. Even large database designs are very well supported. However, you might run into problems if you try to save very large diagrams as bitmap files since windows has got a limitation in the size of a bitmap. Therefore, it is recommended to export large diagrams to a Windows metafile. A Windows metafile is vector based and therefore much smaller.

- [Back to top of file](#)
-

Cost and where to buy

A free trial version of DeZign can be downloaded from [here](#)

The complete product for a single license can be purchased for as little as \$139 from www.datanamic.com

Rates vary with the number of licenses purchased.

- Datanamic
- Guido Gezellestraat 18
- 2394 TV Hazerswoude
- The Netherlands
- Tel: +31(0)71 3410483
- Fax: +31(0)71 3410484
- info@datanamic.com

- [Back to top of file](#)
-

Support

The price of DeZign includes 1 year of maintenance and support, including minor version changes

The support provided is excellent and during the testing of DeZign, I bombarded Heraut with numerous questions and queries, all of which were answered promptly and politely.

- [Back to top of file](#)
-

About the Author

Ashutosh Gaur is a consultant software engineer working for Rsystems, Inc in El Dorado Hills, CA. He has extensive experience designing and developing databases for different kind of businesses, including Insurance, Health Care, Shipping and Asset Management. He can be reached at ashutosh.gaur@rsys.com.

Can you write reviews and articles like this one? Why not become an editor or reviewer for the Oracle FAQ? Find out how you can make money, get noticed and advance your career [click here](#)

<small>FREE AD BY ADDESIGNER.COM</small>
REVIEW SOFTWARE PRODUCTS AND WRITE ABOUT ORACLE, MAKE MONEY AND BECOME FAMOUS!
<small>RSB</small>

MDM & DATA GOVERNANCE Presented by: information
Summit 2010 October 3-5, 2010 | Crowne Plaza | Times Square Manhattan | New York, NY

Is Dead
 A fresh look at what business analytics means for today's organizations

information management
 How Your Business Works

E-Mail Address: Password: [LOGIN](#)
[Forgot your password?](#)

- [HOME](#) | [NEWSLETTERS](#) | [eBOOKS](#) | [EVENTS](#) | [WHITE PAPERS](#) | [VENDOR SHOWCASE](#) | [IM FOCUS](#) | [RSS/XML](#)
[DISCUSSION BOARDS](#) | [BLOGS](#) | [DM RADIO](#) | [WEB SEMINARS](#) | [VIDEOS](#) | [eLEARNING](#) | [DATA CENTER EXCHANGE](#)

Register Now for FREE Site Access and More

Enter Email Address [REGISTER](#)

Entity Relationship Modeling with UML

Information Management | 2010

Robert A. Maksimchuk, Eric J. Naiburg

Like what you see? Click here to sign up for Information Management's daily newsletter to get the latest news, trends, commentary and more.

- Printer Friendly
- Email
- Reprints
- Reader Comments (NEW)
- [Share](#) | [in](#) | [e](#) | [f](#) | [t](#)

If you don't know your destination, any road will lead you there. The goal of entity relationship (ER) modeling is to create a valid representation of the entities, their attributes and their relationships that will fulfill the needs of the business. While most of the available modeling notations can serve this purpose, the unified modeling language (UML) allows you more flexibility to achieve this goal, especially when dealing on the business side of the equation.

Getting Started

You don't need to understand every aspect of the UML to benefit from its use in creating your logical ER models. When starting out with UML, you could begin creating your ER diagram by simply using classes (classes that have a stereotype of "entity," behave as any entity would), attributes and their associations (i.e., relationships) to other entities. This is how many data analysts get their feet wet using UML for logical data modeling. As you can see in Figures 1 and 2, the diagrammatic difference between ER models represented in the UML vs. other notations is inconsequential.

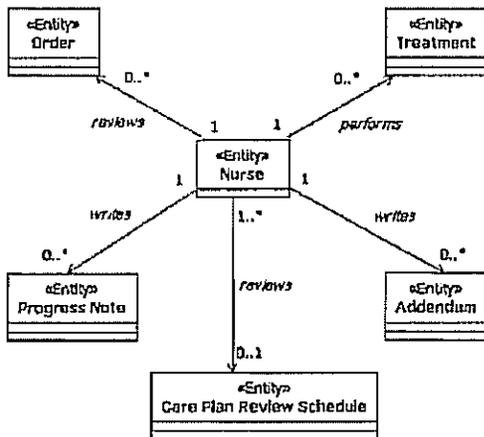


Figure 1: ER Diagram Using UML Notation

ADVERTISEMENT

Integrating Data: An Open Source Approach
 Any technology deployed in the information system needs to interoperate with existing applications or databases. Learn through real-life scenarios how Open Source data integration solves the interoperability challenge. [Free White Paper](#).

RS822

INTERACTIVE eBook

Where Will Your Data Take You? The Benefits of a Single Platform for Data Management

[DOWNLOAD NOW](#)

MOST READ | MOST EMAILED

1. Hadoop for Operational Business Intelligence
2. IBM To Acquire Netezza
3. Developing a Business Intelligence Roadmap
4. Is Predictive Analytics on the Horizon?
5. Microsoft Releases "Thousand-Node" HPC Server

**Information Management BuzzBox
 Comments from Oracle OpenWorld**

[IM BuzzBox](#)
 slyamvaran #OOW10 Oracle openworld attendance 41,000 this time! Big conference by any means but San Francisco can handle it! #ORACLE #IOUG
 6 hours ago

Twitter Join the conversation

MARKETPLACE

- Find Data Modeling products & services on Information Management Marketplace

CHANNEL RESOURCES

- Articles from this Site
- \$100 Million Solution; Basel III Leading to

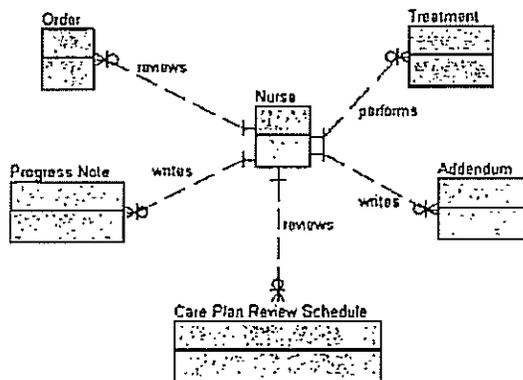


Figure 2: ER Diagram Using Crow's Foot Notation

However, UML provides you with a host of additional useful capabilities. For example, you can directly represent many-to-many associations in your data models along with the appropriate association (a.k.a. intersection) entity. This enables you to have a true visualization of the model elements including additional attributes defined within the association entity.

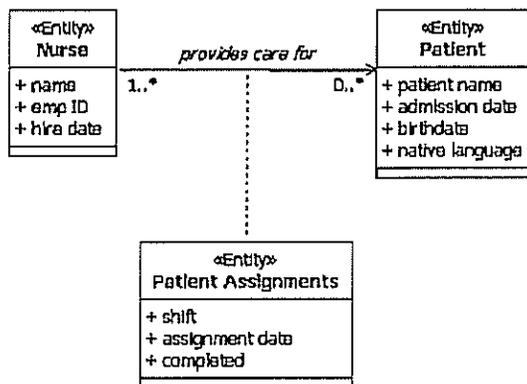


Figure 3: Many-to-Many Relationship Using UML Notation

This representation clearly highlights to all team members that this relationship needs to be examined further. What business needs are being realized by this many-to-many relationship? Does the association entity contain additional attributes (beside the PKs of the other entities) that support the business needs? Are there other attributes that can be added to association entity that can simplify queries? Should this really be a many-to-many relationship? Making such relationships explicit enables all stakeholders to understand these entities and to resolve them in ways that create a good database structure that is responsive to the business requirements.

The Bigger Picture

With all the discussion on notation, the bigger picture is often forgotten. Where do all these entities and relationships originate? They spring from the business requirements that they are meant to serve. For years, studies have shown the most frequent reasons causing project failure are: business people not participating enough in the development of the system, developers not understanding the business or its objectives and unclear or constantly changing business requirements. Here is where the UML provides valuable capabilities beyond other traditional notations. For example, who will use the database and how? What data security restrictions need to be enforced and which will be managed by different user views? A traditional logical data model examines the data that is needed, but does not provide a business or system description of why and how the data will be used.

UML helps address such areas beginning earlier in the development lifecycle where you develop your conceptual model. You can use the UML's use cases to create a model of the system's existing and/or desired functions. The simplicity of use case diagrams allows the business, database team, and development team to easily understand these models. Use case modeling is a simple way to a) understand the current business, b) elicit the desired requirements for the new system you are creating, and c) establish who will be interacting with the system and how.

In this way, you reach a clear understanding of what is expected of the system you must build and gain agreement from everyone involved.

- ▶ "Massive System Upgrades"
- ▶ Pentaho Reveals Data Integration
- ▶ Models Getting a Little Long in the Tooth
- ▶ Stanford To Open High-Security Data Center For Census Research
- ▶ HP, Microsoft in \$250 Million Partnership
- ▶ More Articles from this Site

White Papers

- ▶ Market-Leading Data-Modeling Tools
- ▶ How Data Warehousing Challenges Can Be Resolved Through Modeling
- ▶ Tightening the Alignment Between Business and IT through Modeling
- ▶ Combining Data Profiling and Data Modeling for Better Data Quality
- ▶ Modeling Unstructured Data: An Independent Perspective
- ▶ More White Papers

Web Seminars

- ▶ The Learning Curve: BI and DW Data Modeling
- ▶ The Learning Curve: Conceptual Data Modeling
- ▶ Creating a Safer, Smarter Pharmaceutical Supply Chain
- ▶ Streamlining the Business of Healthcare Through Effective Information Delivery
- ▶ Clinical Analytics and the Unified Electronic Health Record
- ▶ More Web Seminars

eLearning

- ▶ The Data Model Scorecard
- ▶ More eLearning

Books

- ▶ Data Modeling for the Business: A Handbook for Aligning the Business with IT using High-Level Data Models
- ▶ Information Modeling and Relational Databases
- ▶ Data Modeling Fundamentals: A Practical Guide for IT Professionals
- ▶ Arc Marine: GIS for a Blue Planet
- ▶ Data Modeling Theory and Practice
- ▶ More Books

ADVERTISEMENT

HealthData
Management PRESENTS
HEALTH IT
WRESTLING WITH THE FEDS

NOVEMBER 14-16, 2010
SWISSOTEL CHICAGO | CHICAGO, IL

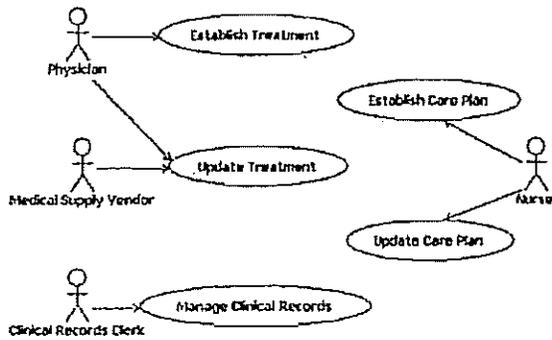


Figure 4: Use Case Diagram

As you develop your use case models, you will discover the important business entities that are needed and come to a common agreement of their definition, early in the lifecycle. How many times have you seen disagreement on key domain entities in your business such as: Customer, Account or Agent? How many times has this gone unnoticed until it was too late, causing you to degrade your database implementation?

Taking this approach to conceptual modeling, you may also choose to use the UML's sequence diagrams that show how the users of the system interact with the important entities you have defined. Not only does this elaborate the system design further, it also helps to begin the definition of the user transactions into the database. For example, in the Figure 5 you can see what entities (e.g., Order, Treatment, etc.) the medical supply vendor needs to access.

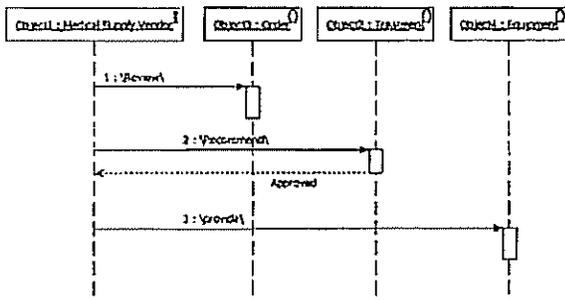


Figure 5: Sequence Diagram

Only the Beginning

Of course, there is much more you could do with the UML when designing databases (see *UML For Database Design*; Naiburg and Maksimchuk). By just using the UML to create a valid conceptual model of the system and then a logical model that realizes those business needs, you will have an excellent foundation for (automatically) transforming this logical model into an initial physical data model upon which you can elaborate and build your databases.

Simply taking these beginning steps will empower you, the data analyst, to really become a data steward. Since you can now talk a common language with the business people and application developers, you can protect the data assets of your company from architectural decay, redundancy and other data degradation that causes additional maintenance, increased defects and even project failure.

By starting with the UML at the beginning of the development life cycle, all stakeholders (e.g., business sponsors, application developers, database developers) can have a common understanding of the project goals. Your UML models are the maps to your final destination. Once everyone understands where they are going and how to get there, your chances of successfully arriving at that destination are much greater.

Robert A. Maksimchuk is the data modeling evangelist for Rational Software Corporation. Maksimchuk coauthored UML For Database Design, Addison-Wesley, 2001, ISBN 0-201-72163-5 with Eric J. Naiburg.

Eric J. Naiburg is the group manager of business segments for IBM Rational Software. Naiburg is responsible for managing the product direction for supporting business and IT applications. Previously, he was manager of product management for Rational, focusing on the Rational Rose and Rational XDE Professional product lines, he extended the ability of Rational products to support database design and e-business solutions within the visual modeling tools space and the UML. He and Robert A. Maksimchuk coauthored UML For Database Design, Addison-Wesley, 2001, ISBN 0-201-72163-5. You can contact him via e-mail at ejnaiburg@us.ibm.com.

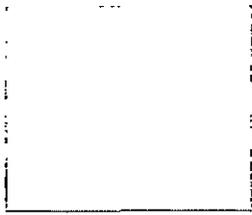
For more information on related topics, visit the following channels:

- Data Modeling

An IBM Web Seminar Series



REGISTER NOW



Connecting the Dots —
Using Data Quality to Achieve
Your Business Goals

SPONSORED BY



TRILLIUM
SOFTWARE

DOWNLOAD
NOW



360⁺plus
SourceMedia
Custom Solutions

The **DATA QUALITY** eBook

WHEN BAD DATA HAPPENS TO GOOD COMPANIES



[About Us](#) | [Advertising/Media Kit](#) | [Awards](#) | [Contact Us](#) | [Customer Service](#) | [Editorial Calendar](#) | [Press Releases](#) | [Privacy Statement](#)



Business Technology & Securities Group

©2010 Information Management and SourceMedia, Inc. All rights reserved.
SourceMedia is an Investcorp company.
Use, duplication, or sale of this service, or data contained herein, is strictly prohibited



Development Cycles

26

[Home](#) / [Development Cycles](#) / [Entity Relationship Modeling](#)

Baseline
Confessions of IT
Job Seekers
For some lucky folks,
it's anything but a
game.
[CLICK HERE >>](#)

Does the price of a database equal
the performance of a database?

Dev Articles Forums
[ADO.NET](#)
[Apache](#)
[ASP](#)
[ASP.NET](#)
[C#](#)
[C++](#)
[ColdFusion](#)
[COM/COM+](#)
[Delphi-Kylix](#)
[Design Usability](#)
[Development Cycles](#)
[DHTML](#)
[Embedded Tools](#)
[Flash](#)
[Graphic Design](#)
[HTML](#)
[IIS](#)
[Interviews](#)
[Java](#)
[JavaScript](#)
[MySQL](#)
[Oracle](#)
[Photoshop](#)
[PHP](#)
[Reviews](#)
[Ruby-on-Rails](#)
[SQL](#)
[SQL Server](#)
[Style Sheets](#)
[VB.Net](#)
[Visual Basic](#)
[Web Authoring](#)
[Web Services](#)
[Web Standards](#)
[XML](#)
[Mobile Linux](#)
[App Generation ROI](#)
[IBM developerWorks](#)

DEVELOPMENT CYCLES



Entity Relationship Modeling

By: [Alf Pedersen](#)Rating: **★★★★☆** / 61

2004-04-05

[Search For More Articles!](#)[Disclaimer](#)[Author Terms](#)

Table of Contents:

- Entity Relationship Modeling
- The Entity
- Other Business Contacts
- Attributes in entities
- Business Rules
- Three types of relationships
- Supplier Entity
- A Weak Relation
- A Useful Relation
- Involved (or recursive) relationships
- Many-to-Many
- The Database Analysis Team - A Teamwork
- Level of Knowledge
- Experience vs. Inexperience
- Complete Model?
- Building Queries
- Other Common Errors in ER Modeling
- Second Normal Violation
- More Specific
- Generic or Specific Models?
- Analysts Experience

Rate this Article: Poor Best Rate

ADD THIS ARTICLE TO:

- Del.icio.us
 Digg
 Blink
 Slmpy
 Google
 Spurl
 MyWeb
 Furl

Email Me Similar Content
When Pasted

Add Developer Shed Article
Feed To Your Site

Email Article To Friend

Print Version Of Article

PDF Version Of Article

Entity Relationship Modeling (Page 1 of 21)

Entity Relationship Modeling (ER modeling) is by far the most common way to express the analytical result of an early stage in the construction of a new database. In this ebook, Alf Pedersen describes the principles for ER modeling, as well as the most important terms used in modeling a new database.

In the analysis phase, we are concerned about finding out about the business. At this early stage, we normally do not talk about performance-specific issues. The main goal of the Analysis phase is to determine what information the business needs to perform a certain task (accounting, invoicing, customer support), how that information should be best possibly organized, and what are the relationships between the different sets of information. We start with a few definitions:

Entity-An entity is a specific object of interest to the business area. We might say that in an accounting system, two things are basically necessary: Accounts and transactions. In a customer support system, we would need to have some information about customers, for a start. Each such unit of essential information is named 'an entity', with a name and attributes.

Attributes-Each entity will normally have one or more attributes. Attributes may be thought of as smaller pieces of information within an entity. Together they describe our entity to the degree we find necessary. An example is a customer entity:

PayPal
MERCHANT SERVICES

PayPal lets you accept credit
cards on your website

VISA

1-800-833-7269

ADVERTISEMENT

RS826

ETL Open Source Download
Fast, Powerful, Affordable, Easy Migration Tools. Free Download!
www.Talend.com/Open_

VIRTUALIZATION Optimize, Extend, & Accelerate Your Infrastructure. Free EMC Whitepaper
www.emc.com/virtualizat

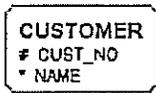
Informatica: Free Demo
Help your IT team deliver projects efficiently & at a lower cost.
www.informatica.com

Common Data Model
Provide a Common Data Model for SOA-Based Applications.
web.progress.com/data*

Weekly Newsletter
Your E-mail [GO]
Developer Updates
Free Website Content

TechDirect
Stop Searching. Start Finding.

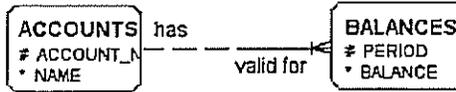
Articles
Forums
All Feeds
Write For Us Get Paid
Request Media Kit
Contact Us
Site Map
Privacy Policy
Support
USERNAME
PASSWORD
[GO]
>>> SIGN UP!
Lost Password?



(2005-08-17)
Examples and Tools for Database Design (2007-12-06)

The entity is named CUSTOMER, and has two attributes, that we have found to be important to know about the customer: Customer number and customer name. Most likely, the customer will have several other attributes as well, but that will show up during the analysis phase. The attributes have a # and a * in front of them: A # means that the attribute is (part of) the primary key for that entity. The primary key is a unique value for every new customer we get so that we can uniquely identify him from all the other customers. The attribute NAME has a * in front: This indicates that it is a mandatory attribute; for every new customer, name MUST be filled in. The opposite is also possible; if an attribute has an o in front of it, it is an optional attribute, and a value is not required.

Relationships-In a relational database, all entities have bonds between them, expressed as relationships. A relationship is a link between two entities, and it tells us something about which relationships exists between our entities. The following example illustrates this:



This sketch tells us that a given account MAY have zero or more balances, but a given balance MUST BE valid for one and only one account. That sounds reasonable. Reading ER diagrams with relationships, gives us a very quick view of the logical structure of a database in the making, without thinking about the physical implementation of it, at this point.

These are the basic elements you need to know about before starting out on your ER modeling: Entities, their attributes, what uniquely identify an occurrence of an entity, and what the relationships between the different entities are.

Next: The Entity >>

More Development Cycles Articles
More By Alf A. Pedersen

Comments On: Entity Relationship Modeling

>>> Be the FIRST to comment on this article!

DEVELOPMENT CYCLES ARTICLES

- Division of Large Numbers
- Branch and Bound Algorithm Technique
- Dynamic Programming Algorithm Technique
- Genetic Algorithm Techniques
- Greedy Strategy as an Algorithm Technique
- Divide and Conquer Algorithm Technique
- The Backtracking Algorithm Technique
- More Pattern Matching Algorithms: B-M
- Pattern Matching Algorithms Demystified: KMP
- Coding Standards
- A Peek into the Future: Transactional Memory
- Learning About the Graph Construct using Gam...
- Learning About the Graph Construct using Gam...
- Learning About the Graph Construct using Gam...
- How to Strike a Match

Find More Development Cycles Articles

PayPal
MERCHANT SERVICES

DID YOU KNOW...
You can increase sales by tapping into **84** million PayPal buyers

LEARN MORE