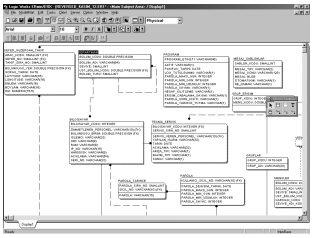
Data Modeling and Database Design Using ERwin

PRACTICAL
DATA MODELING and DATABASE DESIGN USING ERWIN

- **ERWIN and Its Main Features**
- **Data Modeling Using ERWIN**
- **Before Database Design**
- **Transforming Data Model To Database Design**
- **Reverse Engineering**
- **ERWIN Reports**
- **Samples**

What is ERWIN?

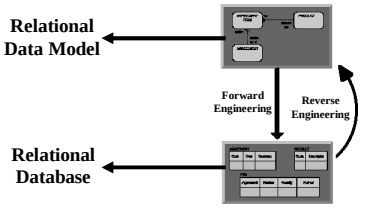
Erwin*; A Data Modeling and Database Design Application CASE Tool working on Windows



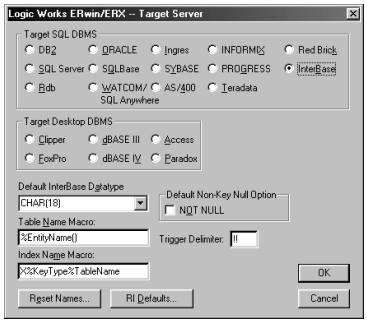
* CopyRighted by **Logic Works and Platinum Inc.**

Main Features

Forward Engineering: Transforming Data Model into Database
Reverse Engineering: Obtaining Design Info and Data Model From a Relational Database

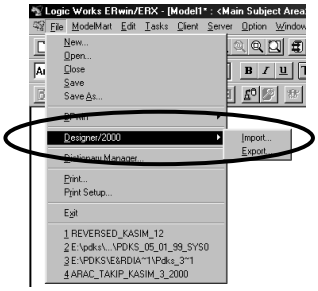


Main Features: Supported DBMSs



ERWIN and Its Main Features

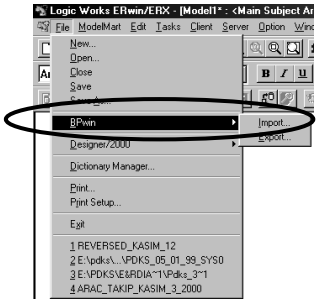
Main Features: ORACLE Designer Data Modeling Tool



ORACLE Designer 2000 Compatible

ERWIN and Its Main Features

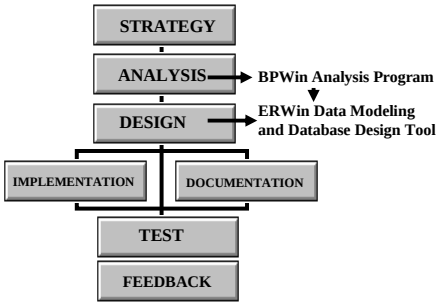
Main Features: Importing Entities from BPwin



BPWin is an activity (process) modeling, and data flow diagramming tool

ERWIN and Its Main Features

Main Features: BPWin ve ERWin



System Development Life Cycle

ERWIN and Its Main Features

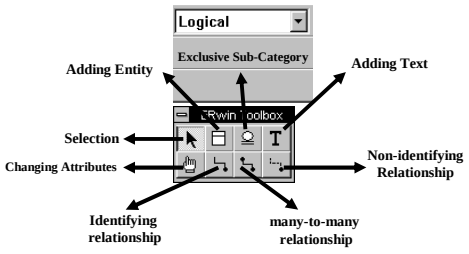
Main Features: Working Levels

1. **Logical (Conceptual) Level (Data Modeling)**
2. **Physical Level**

ERWIN and Its Main Features

Main Features: Working Levels (for Version 3.5.2)

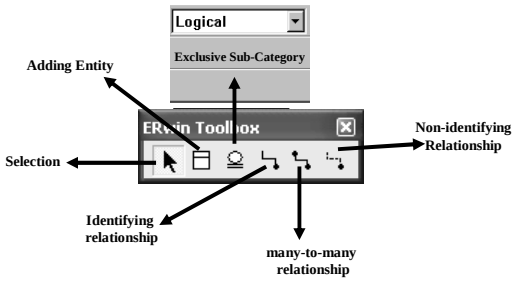
1. Logical Level (Data Modeling)



ERWIN and Its Main Features

Main Features: Working Levels (For Version 4)

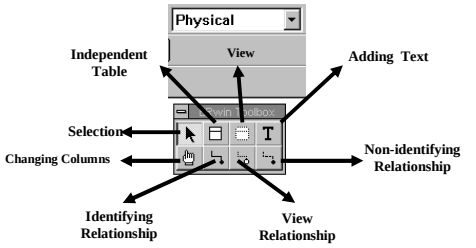
1. Logical Level (Data Modeling)



ERWIN and Its Main Features

Working Levels

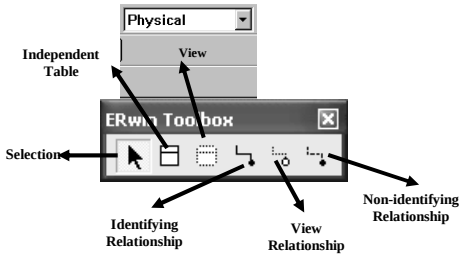
2. Physical Level (Version 3.5.2)



ERWIN and Its Main Features

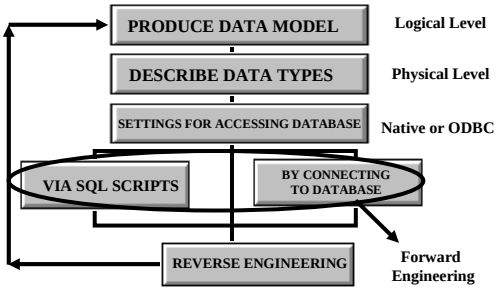
Working Levels

2. Physical Level (Version 4)



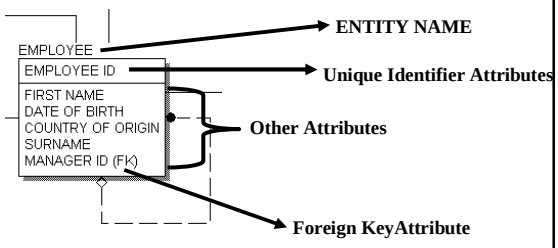
ERWIN and Its Main Features

Modeling and Design Stages for ERwin



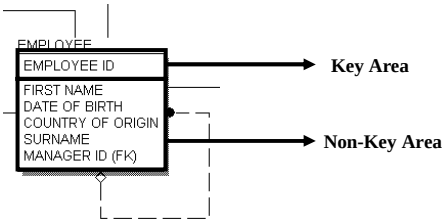
Data Modeling Using ERWIN

Representing Entities and Attributes (Logical Level)



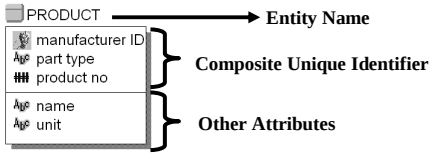
Data Modeling Using ERWIN

Representing Entities and Attributes (Logical Level)



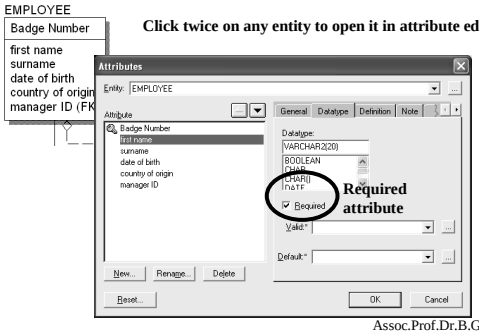
Data Modeling Using ERWIN

Composite Unique Identifiers



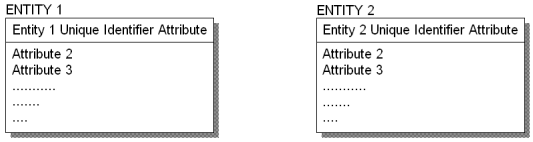
Data Modeling Using ERWIN

Optionality in Attributes



Data Modeling Using ERWIN

Representing Foreign Key Attributes

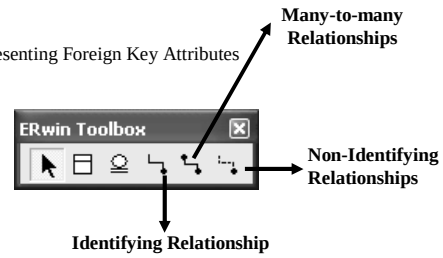


Master Entity **Child Entity**

There are two different relationships between Entity 1 and Entity 2.
1. Identifying Relationships
2. Non-Identifying Relationships

Data Modeling Using ERWIN

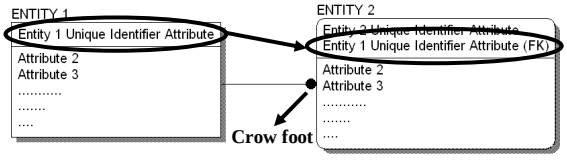
Representing Foreign Key Attributes



In both relationships, Unique Identifier in master entity goes to the second (child) entity as foreign key.

Data Modeling Using ERWIN

Representing Foreign Key Attributes

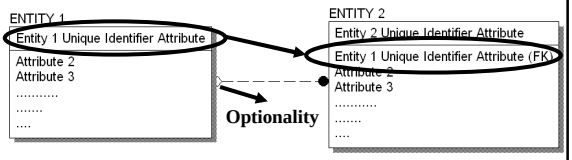


Identifying Relationship

(Unique Identifier of Entity 1 becomes the part of unique identifier in entity 2. It is placed in *key area* of second entity)

Data Modeling Using ERWIN

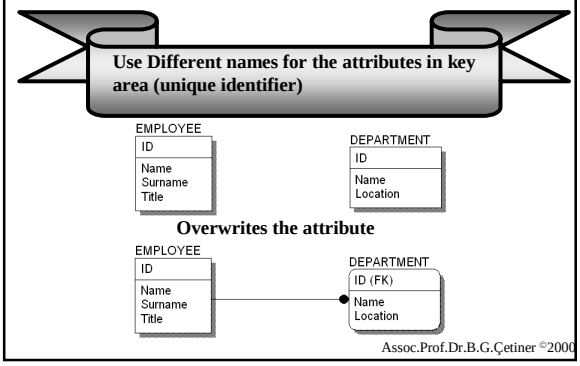
Representing Foreign Key Attributes



Non-Identifying Relationship

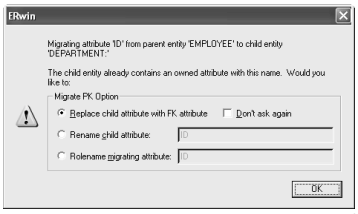
(Unique Identifier of Entity 1 becomes new attribute in entity 2. It is placed in *non-key area* of second entity)

Data Modeling Using ERWIN



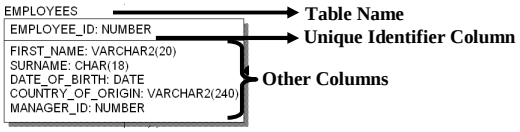
Data Modeling Using ERWIN

Use Different names for the attributes in key area (unique identifier)



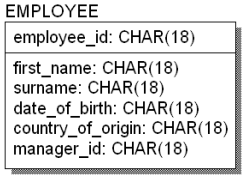
Data Modeling Using ERWIN

Tables and Columns (Physical Representation)



Data Modeling Using ERWIN

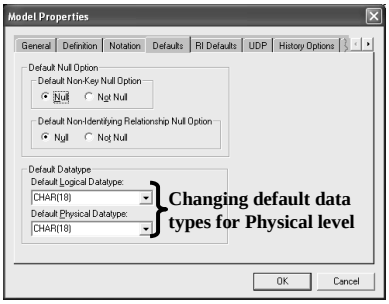
Tables and Columns (Physical Representation)



Note: All Attributes are assigned as *char(18)* (or any default value defined) when the level is changed to physical one.

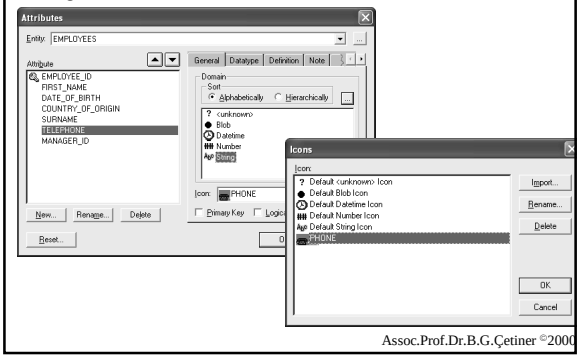
Data Modeling Using ERWIN

Tables and Columns (Physical Representation)



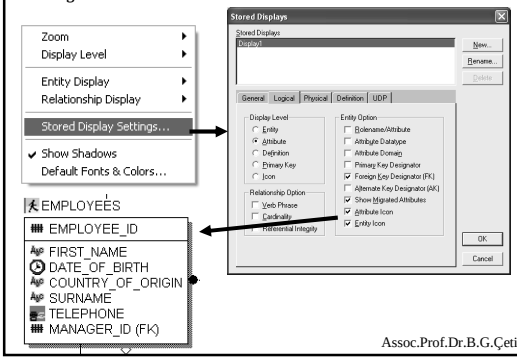
Data Modeling Using ERWIN

Adding Icons to Entities and Attributes



Data Modeling Using ERWIN

Adding Icons to Entities and Attributes



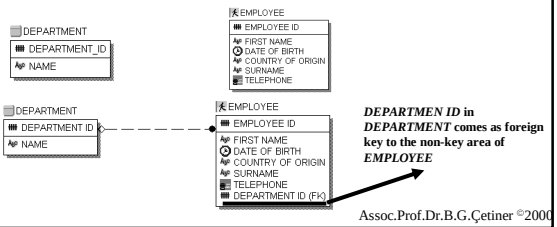
Data Modeling Using ERWIN

Establishing Non-identifying Relationship

1. Choose Non-identifying Relationship

2. Click the master entity (DEPARTMENT in this case)

3. Click the child entity (EMPLOYEE)



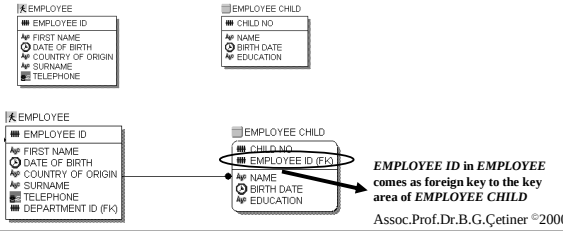
Data Modeling Using ERWIN

Establishing Identifying Relationship

1. Choose Identifying Relationship

2. Click the master entity (EMPLOYEE in this case)

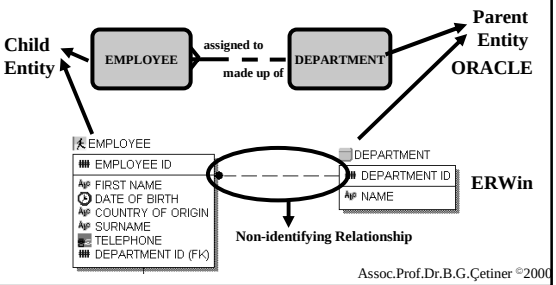
3. Click the child entity (EMPLOYEE CHILD)



Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

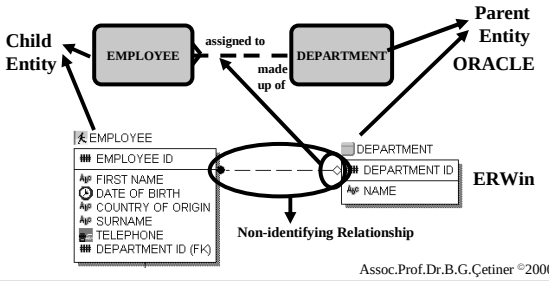
1. Many-to-one (One-to-many) Relationships (Mandatory)



Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

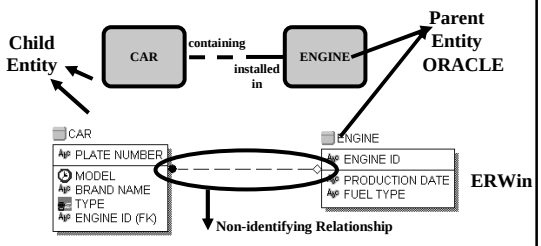
2. Many-to-one (One-to-many) Relationships (Optional)



Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

3. One-to-one Relationships (Optional)

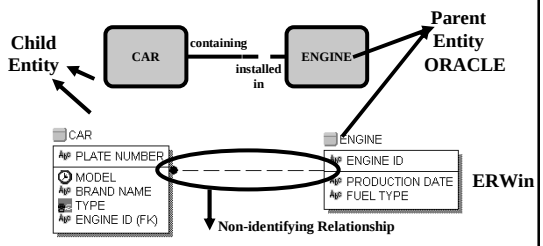


The entity covering the other entity is defined as child entity in one-to-one relationship.

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

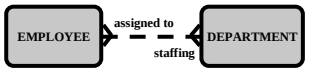
4. One-to-one Relationships (Mandatory)



The entity covering the other entity is defined as child entity in one-to-one relationship.

Representing ORACLE Designer Notations under ERwin

5. Many-to-many Relationships



ORACLE

Note: First, you have to resolve many-to-many relationships

Representing ORACLE Designer Notations under ERwin

5. Many-to-many Relationships

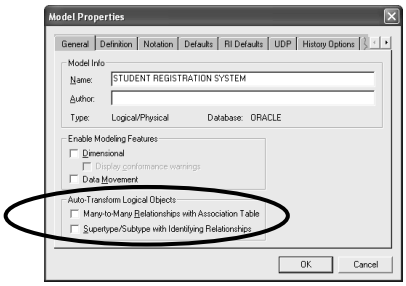
1. Logical **Choose many-to-many relationship in Logical Level**

2. Click on both entities

3. Physical **If you have chosen auto-transform, then an intersection entity will be added automatically.**

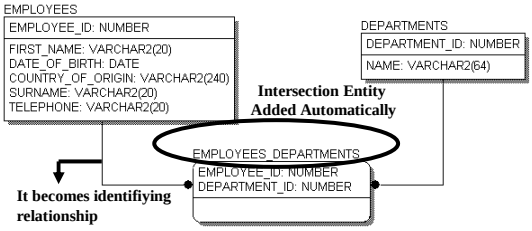
Representing ORACLE Designer Notations under ERwin

5. Many-to-many Relationships



Representing ORACLE Designer Notations under ERwin

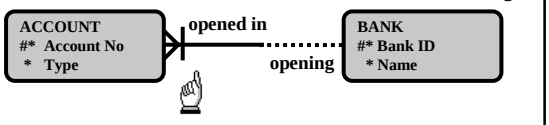
5. Many-to-many Relationships



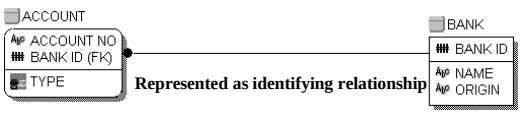
Representing ORACLE Designer Notations under ERwin

6. Obtaining Uniqueness via Relationships

ORACLE Designer



ERwin

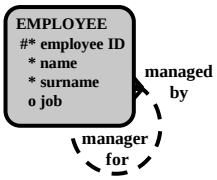


Represented as identifying relationship

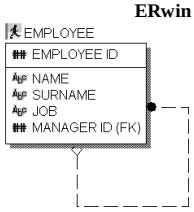
Representing ORACLE Designer Notations under ERwin

7. Recursive Relationships

ORACLE Designer



ERwin



Modeled as Non-identifying Relationship as in one-to-many relationship

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

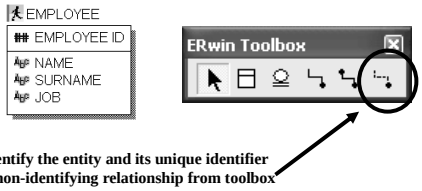
7. Recursive Relationships

1. First, identify the entity and its unique identifier

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

7. Recursive Relationships

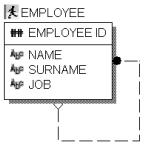


1. First, identify the entity and its unique identifier
2. Choose non-identifying relationship from toolbox

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

7. Recursive Relationships



1. First, identify the entity and its unique identifier
2. Choose non-identifying relationhip from toolbox
3. Click twice on the same entity (EMPLOYEE)

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

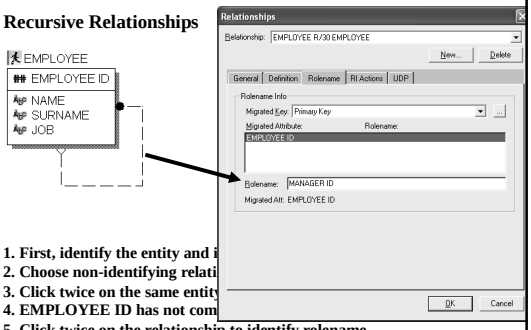
7. Recursive Relationships

1. First, identify the entity and its unique identifier
2. Choose non-identifying relationhip from toolbox
3. Click twice on the same entity (EMPLOYEE)
4. EMPLOYEE ID has not come as Foreign Key

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

7. Recursive Relationships

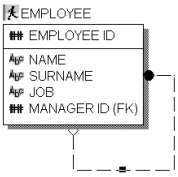


1. First, identify the entity and i
2. Choose non-identifying relati
3. Click twice on the same entit
4. EMPLOYEE ID has not com
5. Click twice on the relationship to identify rolename

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

7. Recursive Relationships

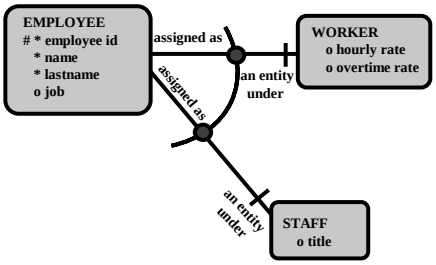


MANAGER ID comes as Foreign key.

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

8. Exclusive Relationships

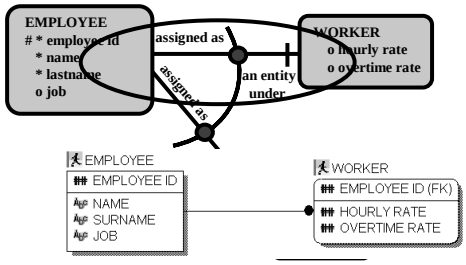


Transferred to ERwin for each pair of relationships

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

8. Exclusive Relationships

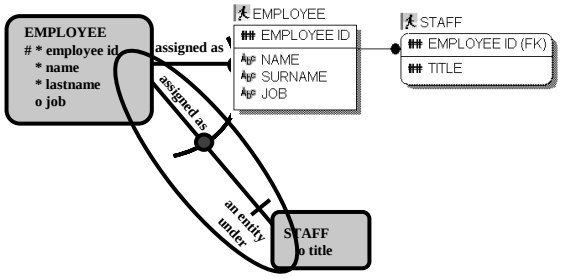


Transferred to ERwin for each pair of relationships

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

8. Exclusive Relationships

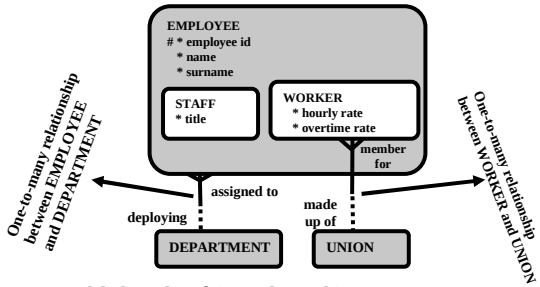


Transferred to ERwin for each pair of relationships

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

9. Supertypes and Subtypes

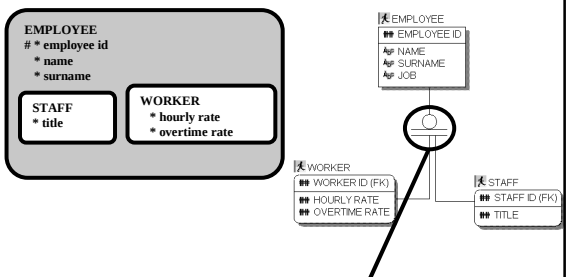


Modeled as Identifying Relationship

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

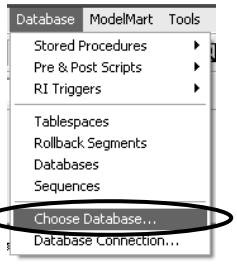
9. Supertypes and Subtypes



Modeled as Identifying Relationship

Transforming Data Model To Database Design

Choosing Target Database Management System



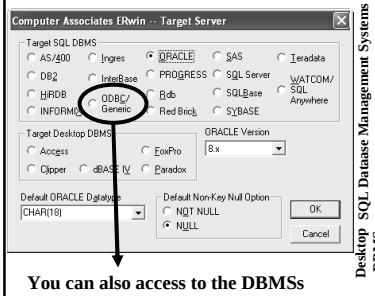
Database/Choose Database for selecting target database management system

NOTE: You have to be under Physical **level**

Transforming Data Model To Database Design

Choosing Target Database Management System

Supported Database Management Systems



SQL Dataase Management Systems

1. IBM DB2
2. IBM AS/400
3. ORACLE
4. MS SQL Server
5. InterBase
6. SYBASE
7. PROGRESS
8. INFORMIX
9. Ingres
10. Red Brick
11. SQLBase
12. Rdb
13. WATCOM/SQL Anywhere
14. Teradata

Desktop DBMS

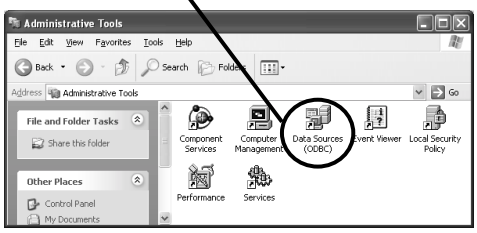
15. Clipper
16. FoxPro
17. dBase
18. Access
19. Paradox

You can also access to the DBMSs not listed here via ODBC/Generic

Before Transforming Data Model

Making Connection Settings for Target Database Management System

ODBC Settings For connecting to DBMS

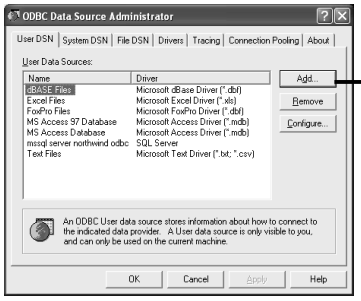


You can directly invoke ODBC Administrator from Run Dialog:

odbcad32

Before Transforming Data Model

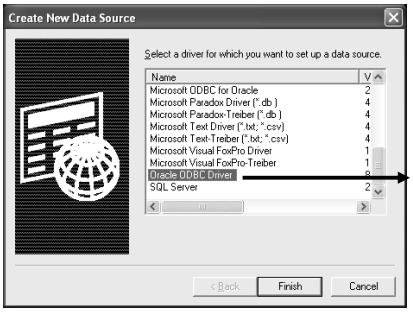
Making Connection Settings for Target Database Management System



For adding new connection

Transforming Data Model To Database Design

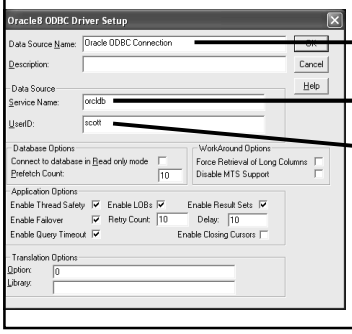
Example; Settings for connecting to Oracle DBMS through ODBC



For connecting to Oracle DBMS

Transforming Data Model To Database Design

Example; Settings for connecting to Oracle DBMS through ODBC



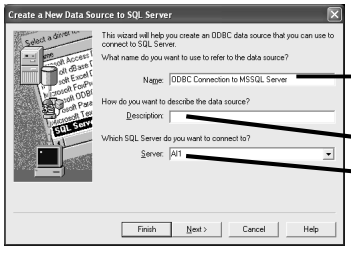
Connection name to connect (You can reach under ERWin ODBC Connection)

Service name

Username

Transforming Data Model To Database Design

Example; MS SQL Server Connections



Connection Name for SQL Server

Optional Description

***AI1* is the server name**

ODBC Settings for each DBMS is different (like printer drivers)

Transforming Data Model To Database Design

Forward Engineering

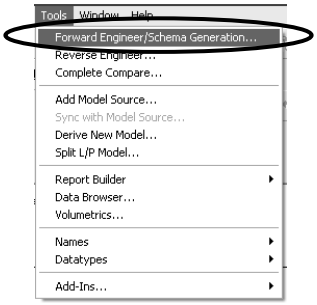
Applying the principles of Systems Development Life Cycle (SDLC) (Going from the top to bottom in Waterfall model)

Producing the Relational Database Objects via the use of Data Models

Transforming Data Model To Database Design

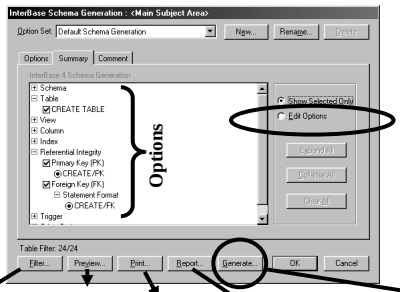
Forward Engineering

Choose *Forward Engineer* Under Tools



Transforming Data Model To Database Design

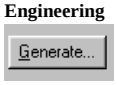
Forward Engineering: *Settings*



Filtering model objects | **Preview SQL Script** | **Print SQL Script** | **Write SQL Script to a File** | **Generate Database Objects**

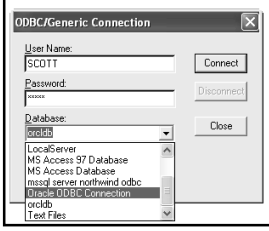
Transforming Data Model To Database Design

Forward Engineering

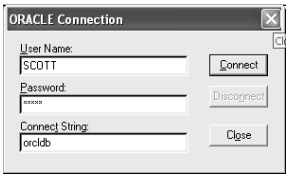
1. Press 

2. Give required security information and press *Connect* button.

ODBC Connection

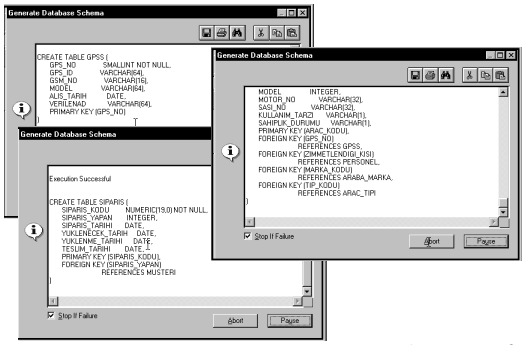


Native Connection



Transforming Data Model To Database Design

Produce the database objects from data model directly



Transforming Data Model To Database Design

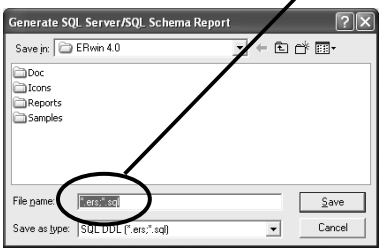
Produce the database objects from data model using SQL Script

1. Press 

button

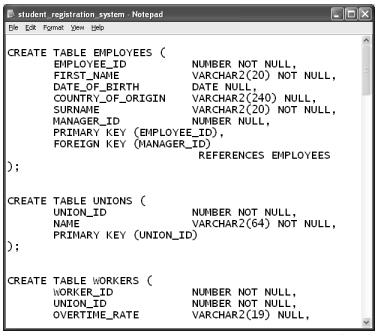
SQL Script filename

2.



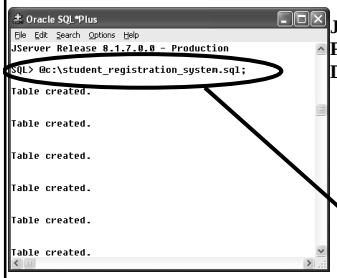
Transforming Data Model To Database Design

Produce the database objects from data model using SQL Script



Transforming Data Model To Database Design

Produce the database objects from data model using SQL Script



Just run the SQL Script File using tools under the DBMSs

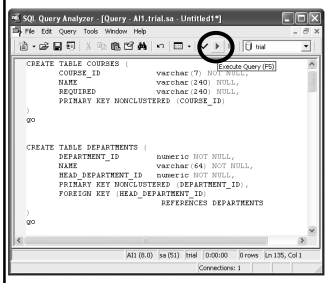
Examples
Oracle SQL*Plus for Oracle DBMS
Query Analyzer for MS SQL Server

Name of Script File

Oracle SQL*Plus for Oracle DBMS

Transforming Data Model To Database Design

Produce the database objects from data model using SQL Script



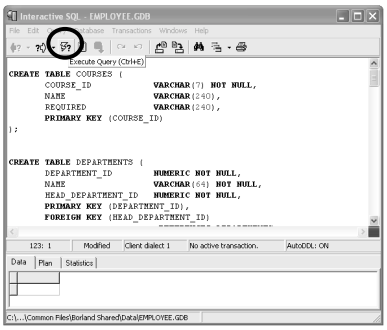
Just run the SQL Script File using tools under the DBMSs

Examples
Oracle SQL*Plus for Oracle DBMS
Query Analyzer for MS SQL Server

Query Analyzer for MS SQL Server

Transforming Data Model To Database Design

Produce the database objects from data model using SQL Script



Interactive SQL for Interbase DBMS

Reverse Engineering

Reverse Engineering

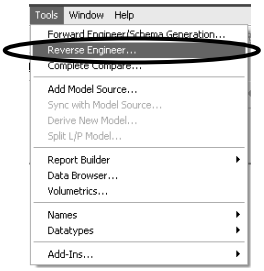
Applying the principles of Systems Development Life Cycle (SDLC) in reverse order
(Going from the bottom to top in Waterfall model)

Inferring the Data Model from Database Objects

Reverse Engineering

Reverse Engineering

Select Reverse Engineer under Tools Menu



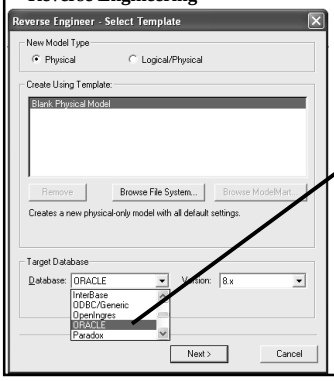
Reverse Engineering

Reverse Engineering: Settings

As in forward engineering, connections are achieved via the native or ODBC type of connection

Reverse Engineering

Reverse Engineering



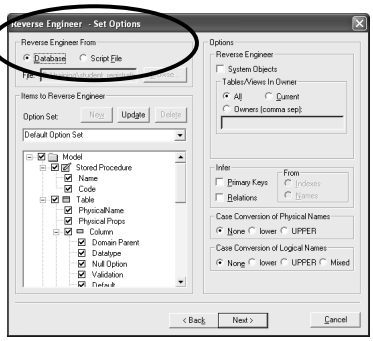
Choose the Target DBMS to reverse engineer the database

Reverse Engineering

Reverse Engineering

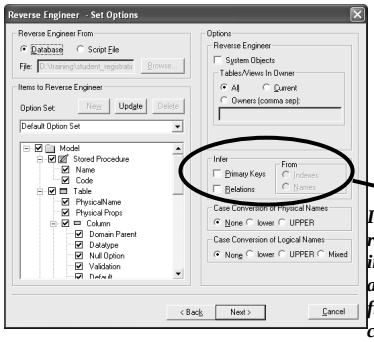
Data model can be obtained from;

1. **DBMS by direct connection**
2. **SQL Script Files**



Reverse Engineering

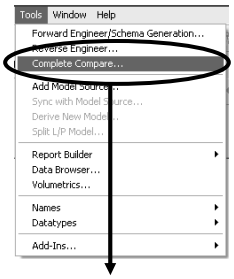
Reverse Engineering



If the database is not relational then you can infere primary keys and relationships from Indexes and the column names in tables

Reengineering

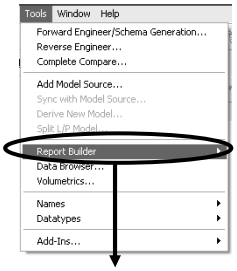
Other Functions



Reengineering: Compares the data model and physical database. You can update the data model via the database or database via the data model

ERwin Reports

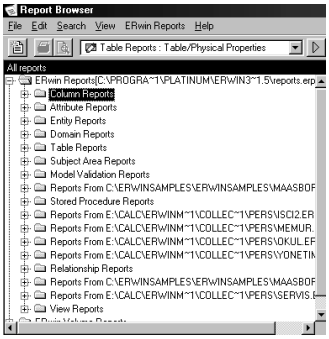
Report Generation



For report generation

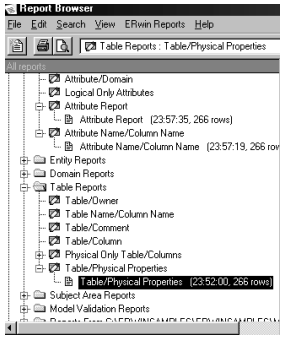
ERwin Reports

Reports



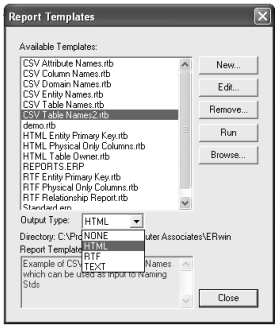
ERwin Reports

Reports: Example; Table/Physical Properties



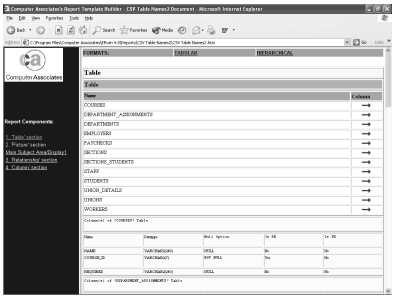
ERwin Reports

Reports: Reports are generated Using Report Templates and Output type (such as HTML)

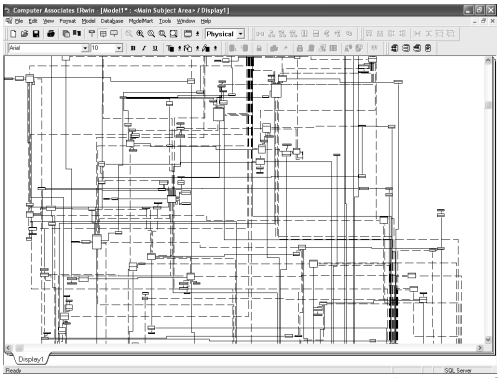


ERwin Reports

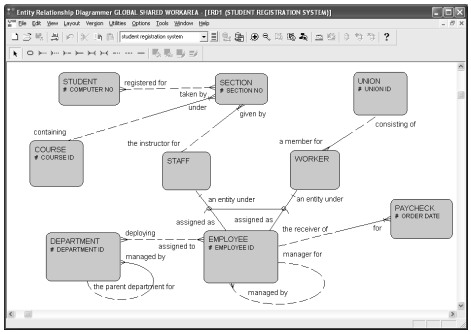
Reports: Reports are generated Using Report Templates and Output type (such as HTML)



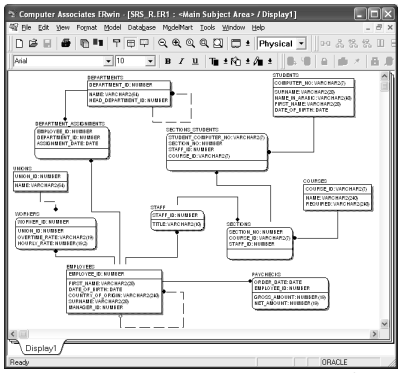
A typical data model using ERwin



Sample data model using Oracle Designer



Sample data model using ERwin



Assoc.Prof.Dr.B.G.Çetiner ©2000