

# Data Modeling and Database Design Using ERwin

Assoc.Prof.Dr.B.G.Çetiner ©2000

## 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

Assoc.Prof.Dr.B.G.Çetiner ©2000

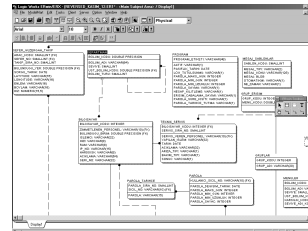
ERWIN and Its Main Features

### What is ERWIN?

Assoc.Prof.Dr.B.G.Çetiner ©2000

ERWIN and Its Main Features

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



\* CopyRighted by Logic Works and Platinum Inc.

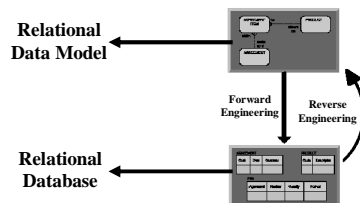
Assoc.Prof.Dr.B.G.Çetiner ©2000

ERWIN and Its Main Features

### Main Features

**Forward Engineering:** Transforming Data Model into Database

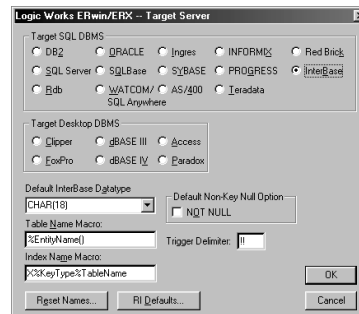
**Reverse Engineering:** Obtaining Design Info and Data Model From a Relational Database



Assoc.Prof.Dr.B.G.Çetiner ©2000

ERWIN and Its Main Features

### Main Features: Supported DBMSs



Assoc.Prof.Dr.B.G.Çetiner ©2000

ERWIN and Its Main Features

**Main Features: ORACLE Designer Data Modeling Tool**

ORACLE Designer 2000 Compatible

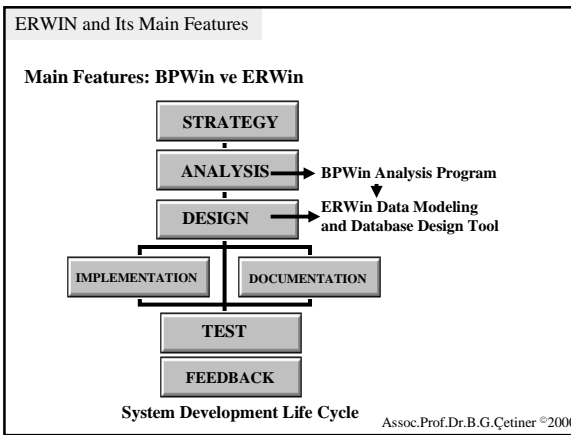
Assoc.Prof.Dr.B.G.Çetiner ©2000

ERWIN and Its Main Features

**Main Features: Importing Entities from BPwin**

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

Assoc.Prof.Dr.B.G.Çetiner ©2000



ERWIN and Its Main Features

**Main Features: Working Levels**

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

Assoc.Prof.Dr.B.G.Çetiner ©2000

ERWIN and Its Main Features

**Main Features: Working Levels (for Version 3.5.2)**

**1. Logical Level (Data Modeling)**

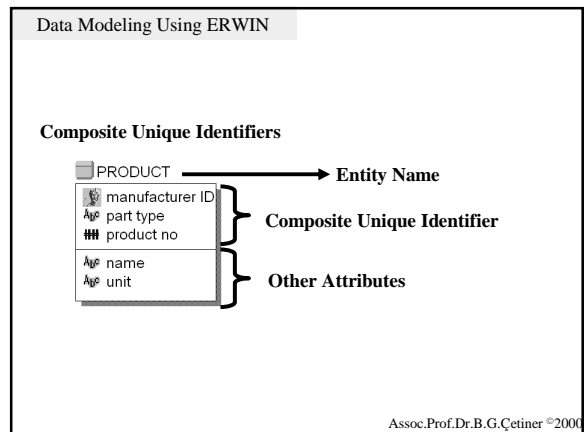
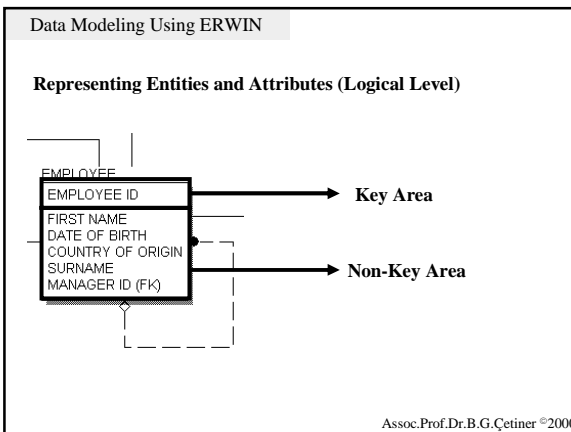
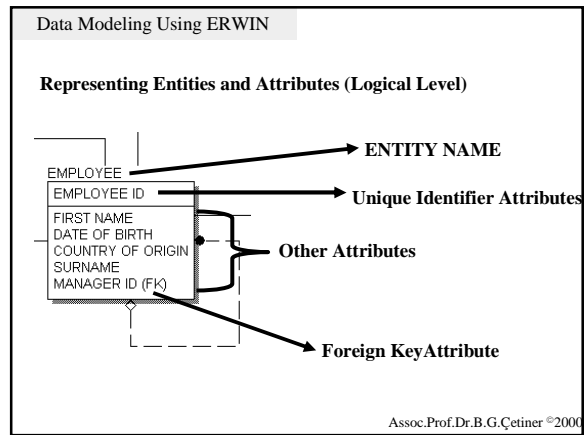
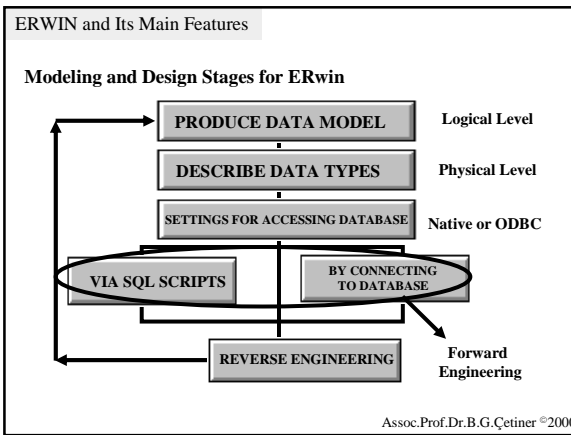
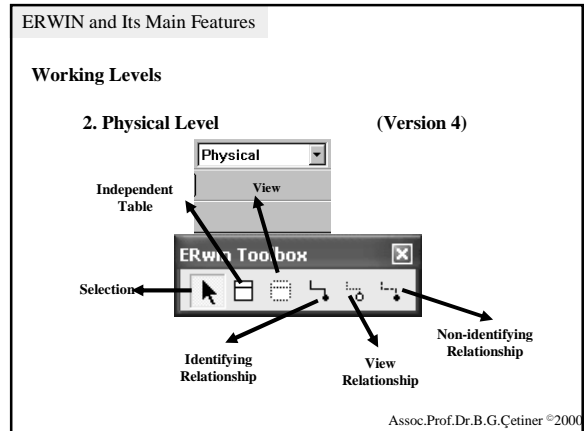
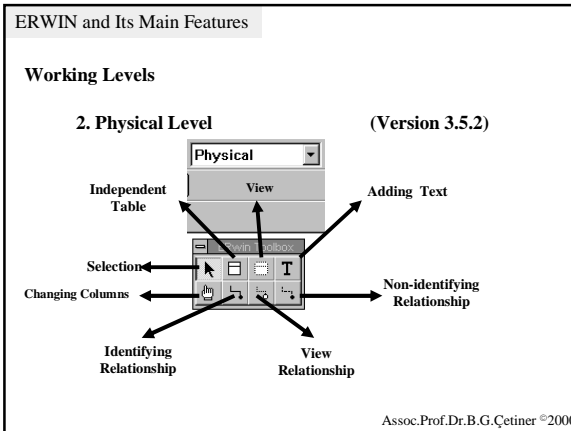
Assoc.Prof.Dr.B.G.Çetiner ©2000

ERWIN and Its Main Features

**Main Features: Working Levels (For Version 4)**

**1. Logical Level (Data Modeling)**

Assoc.Prof.Dr.B.G.Çetiner ©2000



Data Modeling Using ERWIN

**Optionality in Attributes**

EMPLOYEE  
Badge Number  
first name  
surname  
date of birth  
country of origin  
manager ID (FK)

Click twice on any entity to open it in attribute editor

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

Representing Foreign Key Attributes

ENTITY 1  
Entity 1 Unique Identifier Attribute  
Attribute 2  
Attribute 3  
.....  
.....

ENTITY 2  
Entity 2 Unique Identifier Attribute  
Attribute 2  
Attribute 3  
.....  
.....

**Master Entity**                      **Child Entity**

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

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

Representing Foreign Key Attributes

**Many-to-many Relationships**

**Non-Identifying Relationships**

**Identifying Relationship**

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

Assoc.Prof.Dr.B.G.Çetiner ©2000

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)

Assoc.Prof.Dr.B.G.Çetiner ©2000

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)

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

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

**Overwrites the attribute**

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

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

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

**Tables and Columns (Physical Representation)**

EMPLOYEES → **Table Name**  
 EMPLOYEE\_ID: NUMBER → **Unique Identifier Column**  
 FIRST\_NAME: VARCHAR2(20)  
 SURNAME: CHAR(18)  
 DATE\_OF\_BIRTH: DATE  
 COUNTRY\_OF\_ORIGIN: VARCHAR2(240)  
 MANAGER\_ID: NUMBER → **Other Columns**

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

**Tables and Columns (Physical Representation)**

EMPLOYEE

employee\_id: CHAR(18)  
 first\_name: CHAR(18)  
 surname: CHAR(18)  
 date\_of\_birth: CHAR(18)  
 country\_of\_origin: CHAR(18)  
 manager\_id: CHAR(18)

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

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

**Tables and Columns (Physical Representation)**

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

**Adding Icons to Entities and Attributes**

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

**Adding Icons to Entities and Attributes**

Assoc.Prof.Dr.B.G.Çetiner ©2000

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)

DEPARTMENT ID in DEPARTMENT comes as foreign key to the non-key area of EMPLOYEE

Assoc.Prof.Dr.B.G.Çetiner ©2000

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)

EMPLOYEE ID in EMPLOYEE comes as foreign key to the key area of EMPLOYEE CHILD

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

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

Non-identifying Relationship

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

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

Non-identifying Relationship

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

3. One-to-one Relationships (Optional)

Non-identifying Relationship

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

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

4. One-to-one Relationships (Mandatory)

Non-identifying Relationship

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

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

### 5. Many-to-many Relationships

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

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

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.

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

### 5. Many-to-many Relationships

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

### 5. Many-to-many Relationships

**Intersection Entity Added Automatically**

**It becomes identifying relationship**

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

### 6. Obtaining Uniqueness via Relationships

ORACLE Designer

ERwin

**Represented as identifying relationship**

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

### 7. Recursive Relationships

ORACLE Designer

ERwin

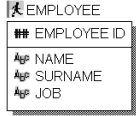
**Modeled as Non-identifying Relationship as in one-to-many relationship**

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

### 7. Recursive Relationships




1. First, identify the entity and its unique identifier

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

### 7. Recursive Relationships



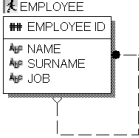
1. First, identify the entity and its unique identifier
2. Choose non-identifying relationship from toolbox

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

### 7. Recursive Relationships



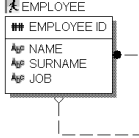
1. First, identify the entity and its unique identifier
2. Choose non-identifying relationship from toolbox
3. Click twice on the same entity (EMPLOYEE)

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

### 7. Recursive Relationships



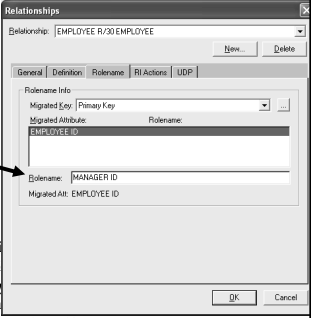
1. First, identify the entity and its unique identifier
2. Choose non-identifying relationship from toolbox
3. Click twice on the same entity (EMPLOYEE)
4. EMPLOYEE ID has not come as Foreign Key

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

### 7. Recursive Relationships



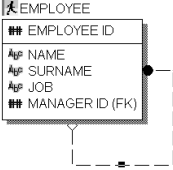
1. First, identify the entity and its unique identifier
2. Choose non-identifying relationship
3. Click twice on the same entity
4. EMPLOYEE ID has not come
5. Click twice on the relationship to identify rolename

Assoc.Prof.Dr.B.G.Çetiner ©2000

Data Modeling Using ERWIN

Representing ORACLE Designer Notations under ERwin

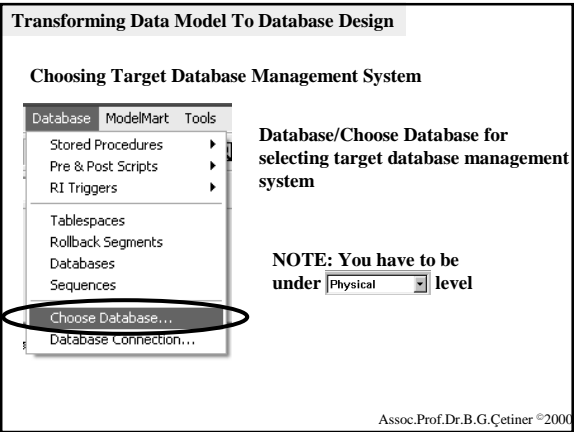
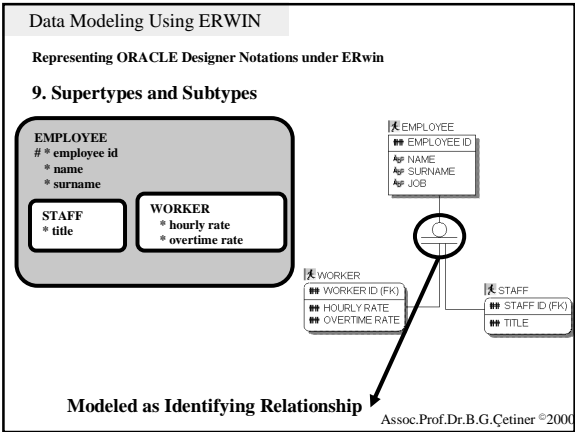
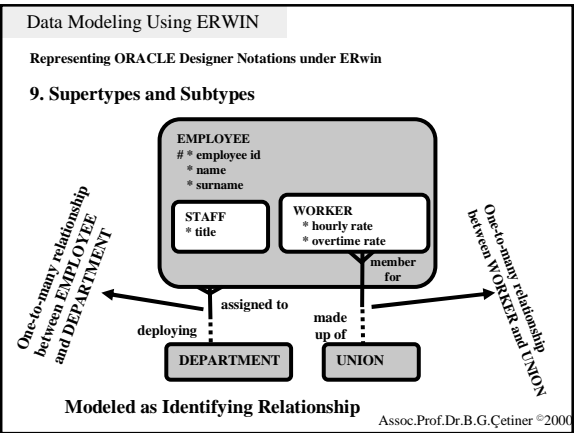
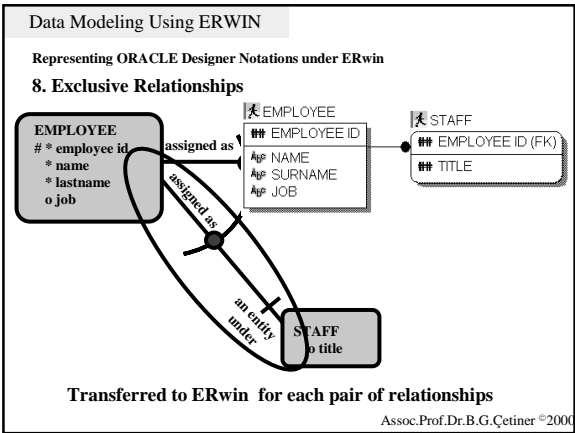
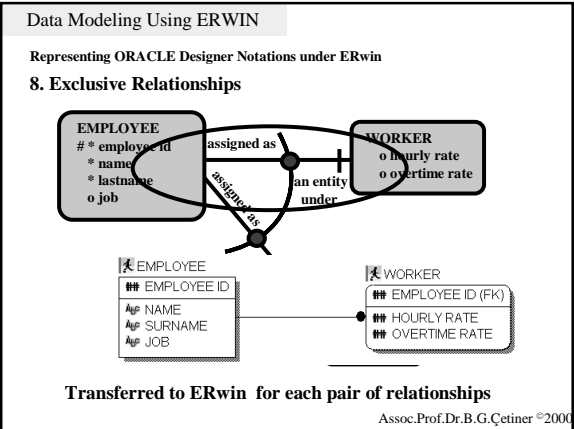
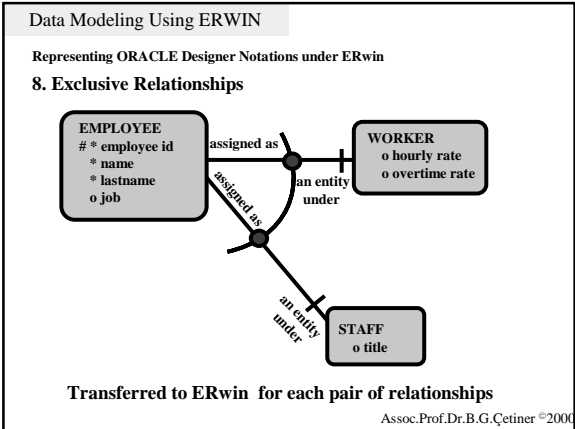
### 7. Recursive Relationships



MANAGER ID comes as Foreign key.

Assoc.Prof.Dr.B.G.Çetiner ©2000





### Transforming Data Model To Database Design

#### Choosing Target Database Management System

Supported Database Management Systems

Target SQL DBMS: AS/400, Ingres, ORACLE, SAS, Teradata, DB2, InterBase, PROGRESS, SQL Server, WATCOM/SQL Anywhere, dBase, Bdb, SQLBase, Informix, Informix, ODBC/Generic, Red Brick, SYBASE.

Target Desktop DBMS: Access, FoxPro, Oracle Version: 8.x, Clipper, dBASE IV, Paradox.

Default ORACLE Database: CHAR(18), Default Non-Key Null Option: NOT NULL, NULL.

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

Assoc.Prof.Dr.B.G.Çetiner ©2000

### 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`

Assoc.Prof.Dr.B.G.Çetiner ©2000

### Before Transforming Data Model

#### Making Connection Settings for Target Database Management System

For adding new connection

Assoc.Prof.Dr.B.G.Çetiner ©2000

### Transforming Data Model To Database Design

#### Example; Settings for connecting to Oracle DBMS through ODBC

For connecting to Oracle DBMS

Assoc.Prof.Dr.B.G.Çetiner ©2000

### 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

Assoc.Prof.Dr.B.G.Çetiner ©2000

### Transforming Data Model To Database Design

#### Example; MS SQL Server Connections

Connection Name for SQL Server

Optional Description

All is the server name

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

Assoc.Prof.Dr.B.G.Çetiner ©2000

### 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)

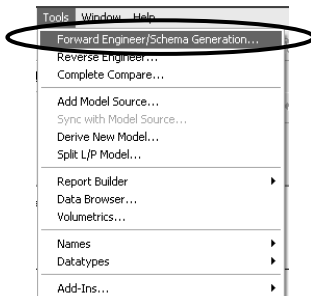
*Producing the Relational Database Objects via the use of Data Models*

Assoc.Prof.Dr.B.G.Çetiner ©2000

### Transforming Data Model To Database Design

**Forward Engineering**

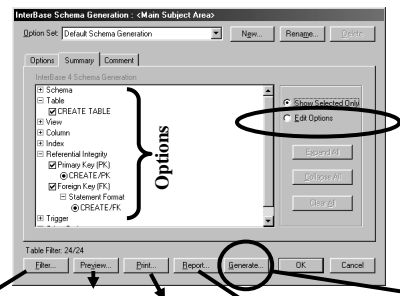
Choose *Forward Engineer* Under Tools



Assoc.Prof.Dr.B.G.Çetiner ©2000

### 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

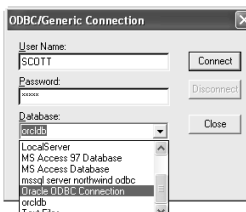
Assoc.Prof.Dr.B.G.Çetiner ©2000

### Transforming Data Model To Database Design

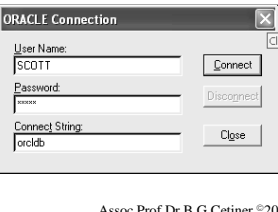
**Forward Engineering**

1. Press **Generate...**
2. Give required security information and press *Connect* button.

**ODBC Connection**



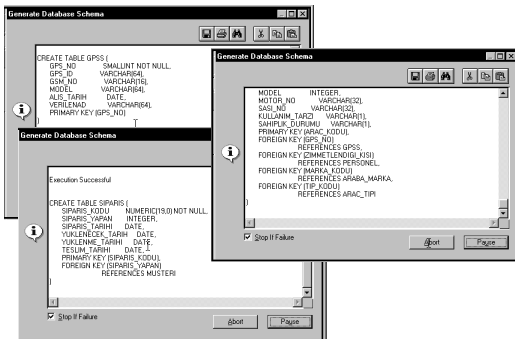
**Native Connection**



Assoc.Prof.Dr.B.G.Çetiner ©2000

### Transforming Data Model To Database Design

**Produce the database objects from data model directly**

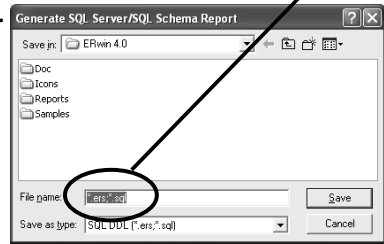


Assoc.Prof.Dr.B.G.Çetiner ©2000

### Transforming Data Model To Database Design

**Produce the database objects from data model using SQL Script**

1. Press **Report...** button
2. Generate SQL Server/SQL Schema Report



Assoc.Prof.Dr.B.G.Çetiner ©2000

**Transforming Data Model To Database Design**

Produce the database objects from data model using SQL Script

```

CREATE TABLE EMPLOYEES (
  EMPLOYEE_ID          NUMBER NOT NULL,
  FIRST_NAME           VARCHAR2(20) NOT NULL,
  DATE_OF_BIRTH        DATE NULL,
  COUNTRY_OF_ORIGIN    VARCHAR2(240) NULL,
  SURNAME              VARCHAR2(20) NOT NULL,
  MANAGER_ID           NUMBER NULL,
  PRIMARY KEY (EMPLOYEE_ID),
  FOREIGN KEY (MANAGER_ID)
    REFERENCES EMPLOYEES
);

CREATE TABLE UNIONS (
  UNION_ID             NUMBER NOT NULL,
  NAME                 VARCHAR2(64) NOT NULL,
  PRIMARY KEY (UNION_ID)
);

CREATE TABLE WORKERS (
  WORKER_ID           NUMBER NOT NULL,
  UNION_ID             NUMBER NOT NULL,
  OVERTIME_RATE        VARCHAR2(19) NULL,
);

```

Assoc.Prof.Dr.B.G.Çetiner ©2000

**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

Assoc.Prof.Dr.B.G.Çetiner ©2000

**Transforming Data Model To Database Design**

Produce the database objects from data model using SQL Script

```

CREATE TABLE COURSES (
  COURSE_ID            varchar(7) NOT NULL,
  NAME                 varchar(240) NULL,
  REQUIRED              varchar(240) NULL,
  PRIMARY KEY NONCLUSTERED (COURSE_ID)
)
GO

CREATE TABLE DEPARTMENTS (
  DEPARTMENT_ID        numeric(7) NOT NULL,
  NAME                 varchar(64) NOT NULL,
  HEAD_DEPARTMENT_ID   numeric(7) NOT NULL,
  PRIMARY KEY NONCLUSTERED (DEPARTMENT_ID),
  FOREIGN KEY (HEAD_DEPARTMENT_ID)
    REFERENCES DEPARTMENTS
)
GO

```

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

Assoc.Prof.Dr.B.G.Çetiner ©2000

**Transforming Data Model To Database Design**

Produce the database objects from data model using SQL Script

```

CREATE TABLE COURSES (
  COURSE_ID            VARCHAR(7) NOT NULL,
  NAME                 VARCHAR(240),
  REQUIRED              VARCHAR(240),
  PRIMARY KEY (COURSE_ID)
);

CREATE TABLE DEPARTMENTS (
  DEPARTMENT_ID        NUMERIC(7) NOT NULL,
  NAME                 VARCHAR(64) NOT NULL,
  HEAD_DEPARTMENT_ID   NUMERIC(7) NOT NULL,
  PRIMARY KEY (DEPARTMENT_ID),
  FOREIGN KEY (HEAD_DEPARTMENT_ID)
    REFERENCES DEPARTMENTS
);

```

Interactive SQL for Interbase DBMS

Assoc.Prof.Dr.B.G.Çetiner ©2000

**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*

Assoc.Prof.Dr.B.G.Çetiner ©2000

**Reverse Engineering**

**Reverse Engineering**

Select Reverse Engineer under Tools Menu

Assoc.Prof.Dr.B.G.Çetiner ©2000

**Reverse Engineering**

*Reverse Engineering: Settings*

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

Assoc.Prof.Dr.B.G.Çetiner ©2000

**Reverse Engineering**

**Reverse Engineering**

Reverse Engineer - Select Template

New Model Type: Physical (selected), Logical/Physical

Create Using Template: Blank Physical Model

Target Database: Oracle (selected), Informatica, ODBC/Generic, OpenIngres, Paradox

Choose the Target DBMS to reverse engineer the database

Assoc.Prof.Dr.B.G.Çetiner ©2000

**Reverse Engineering**

**Reverse Engineering**

Data model can be obtained from;

1. DBMS by direct connection
2. SQL Script Files

Reverse Engineer - Set Options

Reverse Engineer From: Database (selected), Script File

Items to Reverse Engineer: Model, Stored Procedure, Name, Code, Table, Physical Name, Physical Prop, Column, Domain Parent, Datatype, Null Option, Validation, Triggers

Options: Reverse Engineer, System Objects, Tables/Views In Owner, All (selected), Current, Owners (comma sep)

Assoc.Prof.Dr.B.G.Çetiner ©2000

**Reverse Engineering**

**Reverse Engineering**

Reverse Engineer - Set Options

Reverse Engineer From: Database (selected), Script File

File: C:\working\student\_separate

Options: Reverse Engineer, System Objects, Tables/Views In Owner, All (selected), Current, Owners (comma sep)

Items to Reverse Engineer: Model, Stored Procedure, Name, Code, Table, Physical Name, Physical Prop, Column, Domain Parent, Datatype, Null Option, Validation, Triggers

Infer: Primary Keys (selected), From, Indexes, Relations, Names

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

Assoc.Prof.Dr.B.G.Çetiner ©2000

**Reengineering**

**Other Functions**

Tools Window Help

Forward Engineer/Schema Generation...  
Reverse Engineer...  
Complete Compare...

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

Assoc.Prof.Dr.B.G.Çetiner ©2000

**ERwin Reports**

**Report Generation**

Tools Window Help

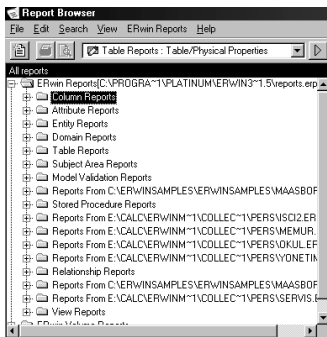
Forward Engineer/Schema Generation...  
Reverse Engineer...  
Complete Compare...  
Add Model Source...  
Sync with Model Source...  
Derive New Model...  
Split LIP Model...  
Report Builder  
Data Browser...  
Volumetrics...  
Names  
Datatypes  
Add-Ins...

For report generation

Assoc.Prof.Dr.B.G.Çetiner ©2000

ERwin Reports

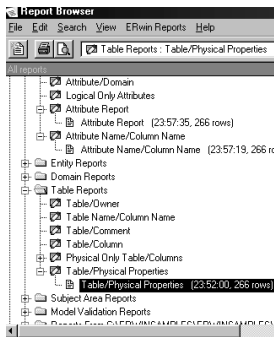
Reports



Assoc.Prof.Dr.B.G.Çetiner ©2000

ERwin Reports

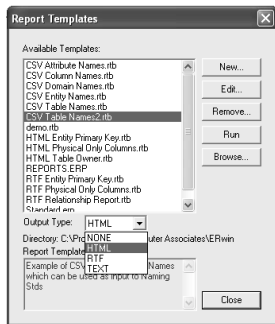
Reports: Example; Table/Physical Properties



Assoc.Prof.Dr.B.G.Çetiner ©2000

ERwin Reports

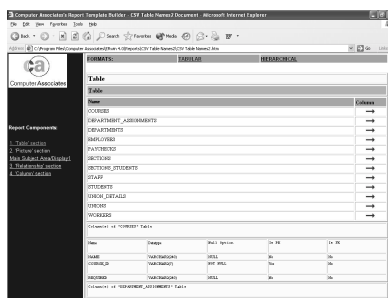
Reports: Reports are generated Using Report Templates and Output type (such as HTML)



Assoc.Prof.Dr.B.G.Çetiner ©2000

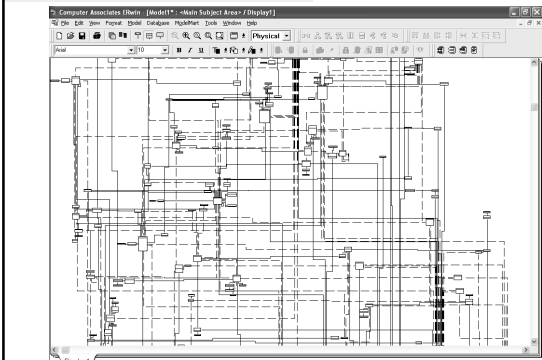
ERwin Reports

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



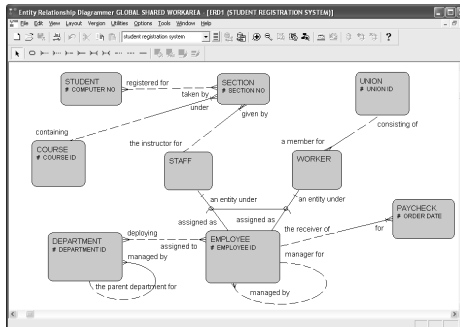
Assoc.Prof.Dr.B.G.Çetiner ©2000

A typical data model using ERwin



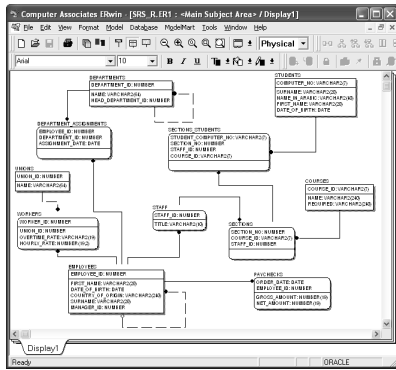
Assoc.Prof.Dr.B.G.Çetiner ©2000

Sample data model using Oracle Designer



Assoc.Prof.Dr.B.G.Çetiner ©2000

### Sample data model using ERwin



Assoc.Prof.Dr.B.G.Çetiner ©2000