

CPSC 450 Database Design Exercise Lab 1

- Where: Technology Building #113 (Tech. Lab)
- Before you come to lab, you need to know the followings:
 - Read text book Chapter 3 and 4: ER and EER modeling.
 - Read ERwin documents at <http://www.ca.com/us/data-modeling.aspx>
 - Valid UB account and Oracle Database account

Contents

About ERwin	1
SALES Database Definition	2
Tutorial 1: Configure ER Display Environment	3
Tutorial 2: Create New Entity Types	4
Tutorial 3: Create 1-n Relationships	8
Tutorial 4: Create m-n Relationships and Apply Transformation	10
Tutorial 5: Generate DB Schema	11
Tutorial 6: Create Enhanced ER Diagram	12
In-class Exercise Solution – The Complete ER Diagram of the SALES database	13
Lab Assignments	14
How to submit Lab1 (Exercise 1 to 3)	17

About ERwin

ERwin is one of the most popular software for data modeling throughout worldwide. ERwin, manufactured by CA (Computer Associates), supports many database related works including database design, logical data modeling, physical data modeling, and reverse engineering for a variety of DBMS such as Oracle, DB2, Sybase, SQL Server and others. ERwin is not free software, but commercial product. ERwin is installed on computers in Tech 113 lab under University Associate program with CA.

The objective of this lab exercise is for you to acquire the essential skills for creating ER/EER diagrams using ERwin. You will learn the skill to carry out the following tasks:

- Configure ERwin display environment (Tutorial 1)
- Create new entity types (Tutorial 2)
 - Define key/non-key attributes.
 - Set data type for each attributes
 - Create new data type and define its domain
- Create 1-n relationship among entity types (Tutorial 3)
 - Set the cardinality
 - Choose between identifying/non-identifying
- Create m-n relationship and apply transformation (Tutorial 4)
- Generate DB schema (Tutorial 5)
- Create EER diagram (Tutorial 6)

For more information on using ERwin Data Modeler, please refer to the ERwin DM Implementation Guide available under 'Start → All Programs → CA → AllFusion → ERwin Data Modeler r7 → ERwin DM Implementation Guide'

Let's look at the description of SALES database and try to understand it. This includes entity types, their attributes and data types, key attribute (underlined), and some constraints. Notice that the key attribute must not be null. We are going to create ER diagram using this example.

To start ERwin: 'Start → All Programs → CA → AllFusion → ERwin Data Modeler r7 → ERwin Data Modeler r7' {Clicking this shortcut will open up ERwin}

SALES Database Definition

CUSTOMERS

```
(
    cust_id          varchar (10)  NOT NULL ,
    cust_name        varchar (50)  NOT NULL ,
    cust_address     varchar (50),
    cust_city        varchar (50),
    cust_state       varchar (5) ,
    cust_zip         varchar (10),
    cust_country     varchar (50),
    cust_contact     varchar (50),
    cust_email       varchar (255)
)
```

ORDERITEMS

```
(
    order_item      number          NOT NULL ,
    order_num       number          NOT NULL ,
    prod_id          varchar(10)     NOT NULL ,
    quantity         number          NOT NULL ,
    item_price       number(8,2)     NOT NULL
)
```

ORDERS

```
(
    order_num       number          NOT NULL ,
    order_date       date            NOT NULL ,
    cust_id          varchar(10)     NOT NULL ,
)
```

PRODUCTS

```
(
    prod_id         varchar(10)     NOT NULL ,
    vend_id          varchar(10)     NOT NULL ,
    prod_name        varchar(255)    NOT NULL ,
    prod_price       number(8,2)     NOT NULL ,
    prod_desc        varchar(255)
)
```

VENDORS

```
(
    vend_id         varchar(10)     NOT NULL ,
    vend_name        varchar(50)     NOT NULL ,
    vend_address     varchar(50),
    vend_city        varchar(50),
    vend_state       varchar(5) ,
    vend_zip         varchar(10),
    vend_country     varchar(50),
)
```

Try to create the ER diagram on your own so that you learn how to create ER diagram based on a given database. If you have trouble understanding this example, a final ER diagram is available for your reference at the end of this document.

Tutorial 1: Configure ER Display Environment

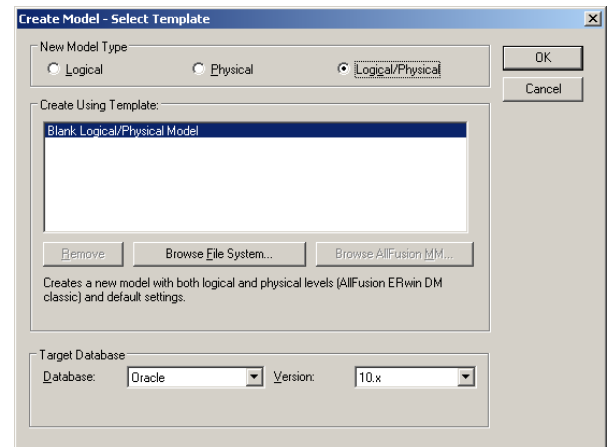
1. Create a new model with 'File → New'

- New Model Type: Select 'Logical/Physical'
- Target Database: Select 'Oracle' and '10.x'
- Click 'OK'



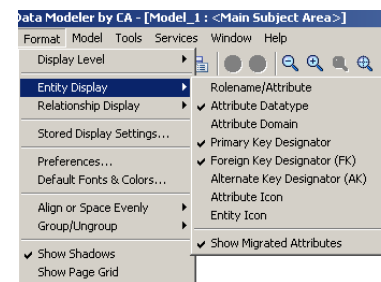
Logical: A conceptual model that contains objects such as entities, attributes, and key groups.

Physical: A database-specific model that contains objects such as tables columns and datatypes.



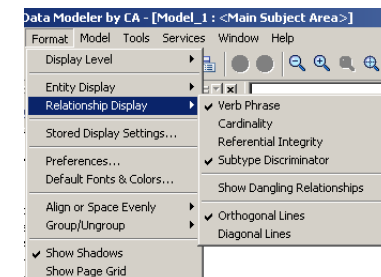
2. From 'Format → Entity Display' menu, change "Entity Display" as follows:

- Check 'Attribute Datatype'.
- Check 'Primary Key Designator'.
- Check 'Foreign Key Designator (FK)'.
- Check 'Show Migrated Attributes'.



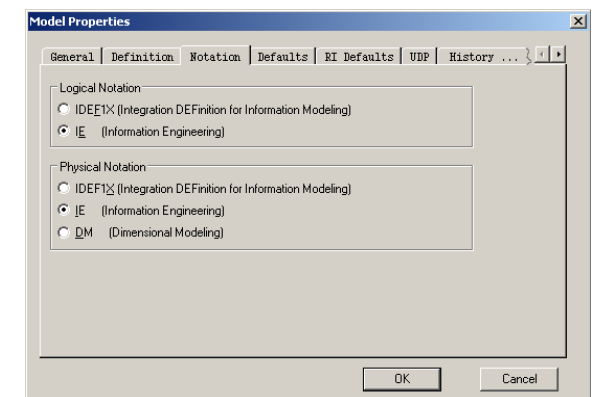
3. From 'Format → Relationship Display' menu, change "Entity Display" as follows:

- Check 'Verb Phrase'.



4. From 'Model → Model Properties' menu, select the 'Notation' tab as follows



- Logical Notation: Select 'IE', (i.e., Crow Foot notation, for both logical and physical notations.)
- Physical Notation: Select 'IE'

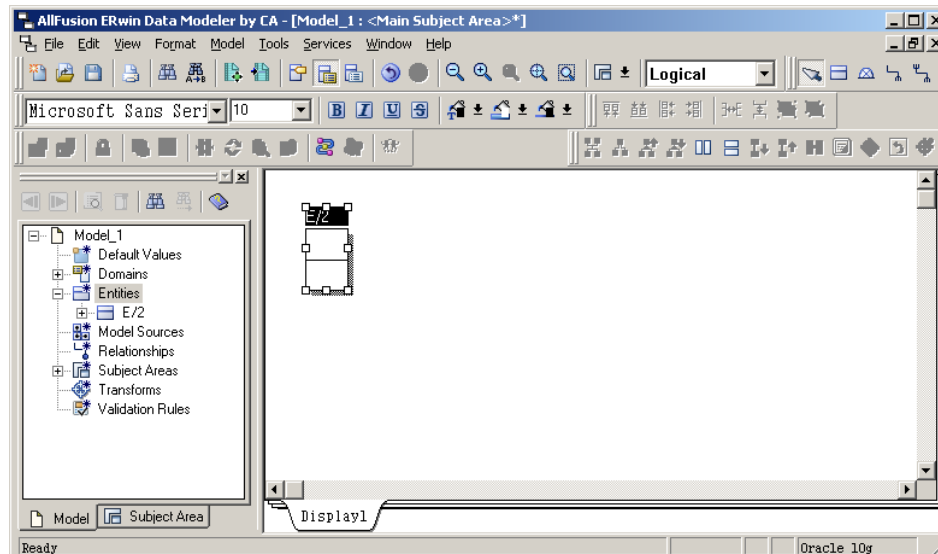


5. Save.

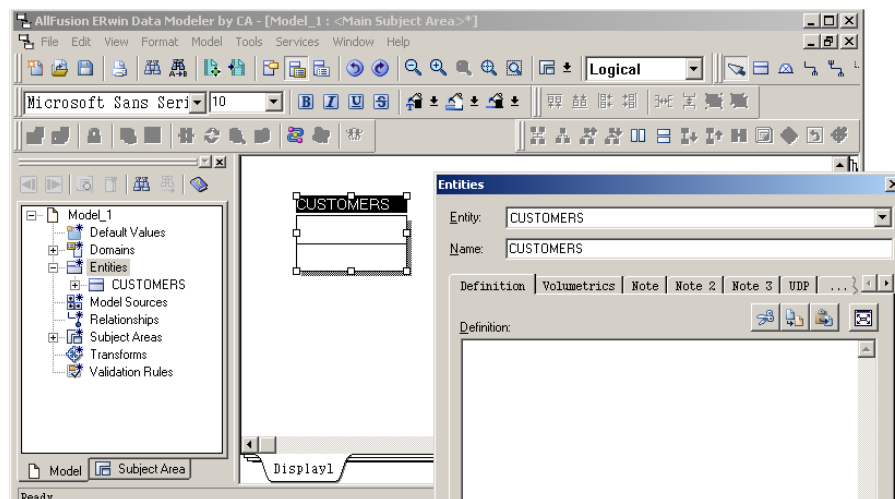
Tutorial 2: Create New Entity Types

Step 1: Create Entity Type

- To place a new entity onto the model, click the entity icon () on the tool bar (), or right-click on the word Entity in the Model Navigator. If you click on the entity icon, you then should click on the Display Window where you would like the entity to appear.

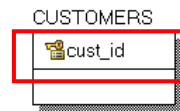


- The default name will be [E/X] where 'X' is sequence number, e.g. '2' in this example. The entity consists of three parts, 'Entity Name', 'Primary Key Attribute(s)' and 'Non-Primary Key Attribute(s)'. You can move each part by pressing 'TAB' key.
- Press tab key until the entity name is highlighted, and then enter the entity name, 'CUSTOMERS'.
- To edit an existing entity, right-click on the entity and select 'Entity Properties....'. Then, double click on the entity. The Entity Editor will be shown.

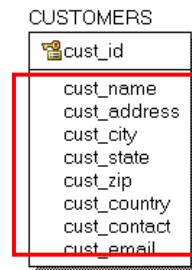


Step 2a: Attributes with Built-In Datatypes

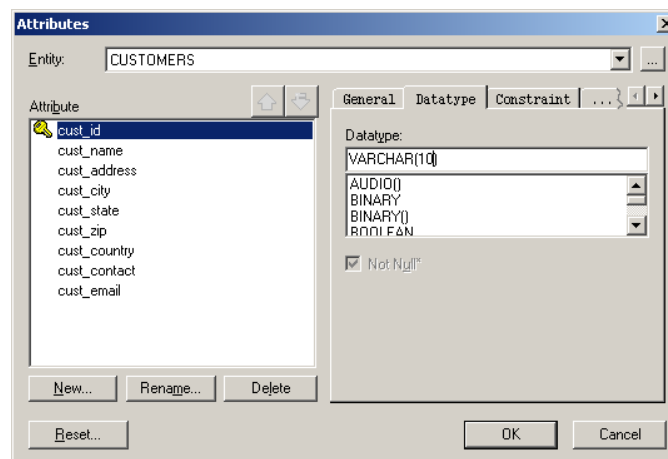
- To add primary key, enter the name of primary key attribute, i.e., cust_id.



- To add non-primary keys, press the tab key to highlight attribute box, where you can add non-primary key attributes. Type the attribute name and enter, continuously, i.e., cust_name, ..., cust_email, then enter ESC.



- To change the DataType of an attribute, right-click on the entity and select 'Attributes..'. Then, double click on the entity. The Attribute Editor will be shown.



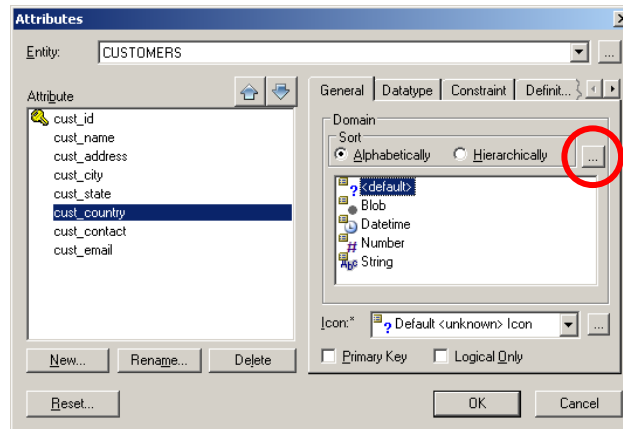
- Select the attribute and choose 'Datatype' tab. Then, select appropriate DataType from the list. If the assigned datatype has parameters (e.g. VARCHAR(xxx)) enter the wanted value between (). Repeat this for all table columns
- Check 'Not Null' if the attribute should not be null.
- Once you have created the 'CUSTOMERS' entity, save the file in your home directory, and then define all the other entities. (You need to save your file frequently)

Step 2b: Attributes with User-Defined Domain (AKA. Validation Rules)

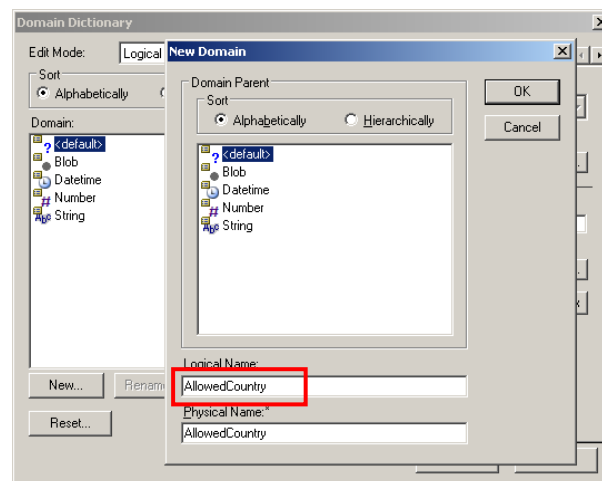
Often we need to restrict our domain to specific values. For instance, if we want to restrict the values of 'cust_country' in the CUSTOMERS to 'US' and 'Canada' only, we can define a new data type by following the following steps:

- Double click on CUSTOMERS entity type to bring up the Attributes dialogue box

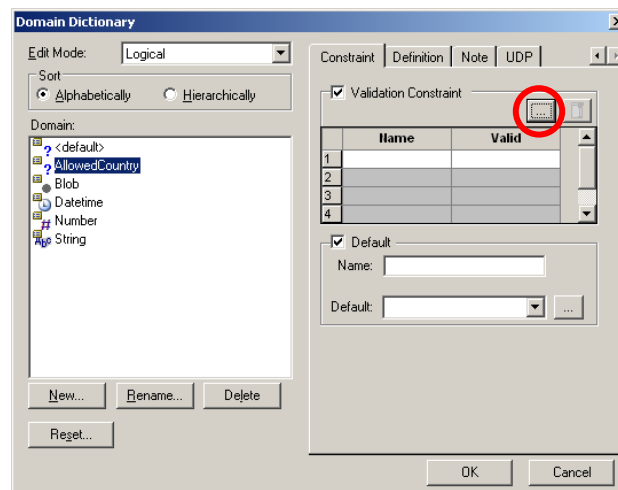
- Click on the 'General' tab and click on [...] to bring up the 'Domain Dictionary' dialogue box



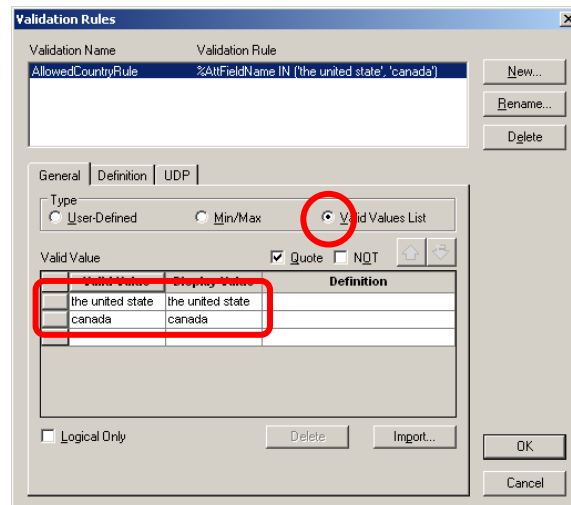
- Click on 'New' button to bring 'New Domain' dialogue box. In 'Logical Name' text field, type 'AllowedCountry' then click 'OK'. 'Physical Name' will automatically be filled.



- From 'Domain Dictionary' dialogue, select 'AllowedCountry' from the left-side Domain list and select 'Constraint' tab. Make sure 'Validation Constraint' is checked and then click on the [...] next to it.



5. From 'Validation Rules', click on [New] to define a new validation rule as following then click [OK].
Logical textfield → AllowedCountryRule.
6. Select the validation rule that we just created. In the 'General' tab, set 'Type' to 'Valid Values List'. In the Valid Value table, type 'the united state' and 'canada' then click [OK].



7. Click [OK] to close the Domain Dictionary dialogue.
8. In Attribute dialogue box, make sure the 'cust_country' is mapped to 'AllowedCountry'. Select 'Constraint' tab, you will see the new rule is now bounded to the 'cust_country'. Click [OK] to close the attribute dialogue.
9. Save.

Tutorial 3: Create 1-n Relationships

Once we have designed the entities, we need to define the relationships between them. We must link primary and foreign keys, and link the relations together using the correct cardinality. Here are the relationships (you can give them your own names):


- Order has a 1..N relationship with OrderItems, so FK_OrderItems_Orders FOREIGN KEY (order_num) REFERENCES Orders (order_num);
- OrderItems has a 1..N relationship with Products, so FOREIGN KEY (OrderItems_order_item) in Products REFERENCES OrderItems (Order_order_item);
- Customers has a 0..N relationship with Orders so FOREIGN KEY (Customers_cust_id) in Orders REFERENCES Customers (cust_id);
- Vendors has a 1..N relationship with Products, so FK_Products_Vendors REFERENCES Vendors (vend_id);

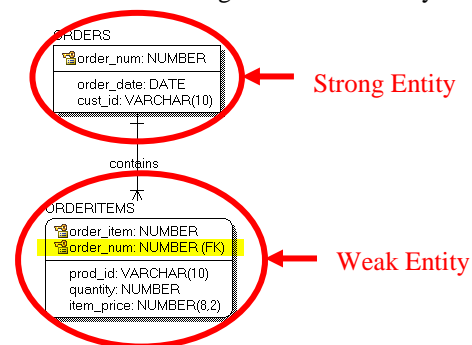


Identifying / Non-identifying relationship


Upon the creation of an entity, all the entities are created as strong (independent) entity by default, shown as rectangle. To create weak entity, we use 'identifying relationship' to turn the child entity into weak entity. Once an identifying relationship was created, the primary keys of the parent entity will be migrated into the child (weak) entity's primary key area as shown in yellow color in this figure. A weak entity is identified as round-corner rectangle.

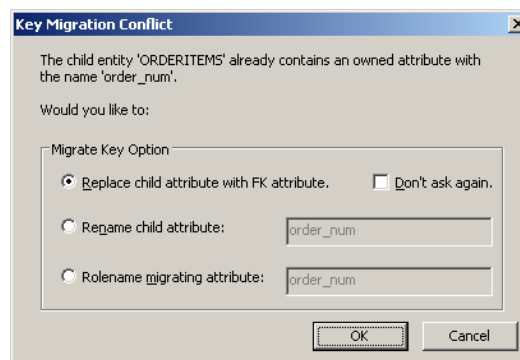
A weak entity can be created in 3 steps:

1. Click on the 'identifying relationship' icon ()
2. Click on parent entity (strong entity)
3. Click on child entity (weak entity)



Create “ORDER contains ORDERITEM” Relationship

1. Complete Tutorial 2.
2. Click on the Identifying Relationship icon ()
3. First click on ‘ORDERS’ (target entity, or strong entity) then click on ‘ORDERITEMS’ (destination entity, or weak entity). In case that destination entity (ORDERITEMS) has the same attribute of parent entity (ORDERS), i.e., order_num, you will see the following pop-up window. You can select ‘Replace child attribute with FK attribute’, and then ‘OK’.



4. Relationship a Name: ORDER contains ORDERITEM

5. Cardinality: One or More

6. Verb Phrase: Set 'Parent-to-Child' to 'contains' and ignore the 'Child-to-Parent' field.

Relationships

Relationship: R/3 (ORDERS to ORDERITEMS)

Name: ORDER contains ORDERITEM

General | Definition | Rolename | RI Actions | UDP

Verb Phrase

Parent-to-Child: contains

Child-to-Parent:

Relationship Cardinality

Summary: One-to-One-or-More (P)

Cardinality

☐ Zero, One or More

☒ One or More (P)

☐ Zero or One (Z)

☐ Exactly:

Relationship Type

☒ Identifying

☐ Non-Identifying

Nulls

☐ Nulls Allowed

☒ No Nulls

☐ Logical Only

Reset Cardinality

OK Cancel

7. Click [OK].

To change the relationship name, cardinality, or relationship type at a later time, simply double-click on the relationship to bring up the "Relationship Properties" dialogue box to modify the values.

8. Save

Tutorial 4: Create m-n Relationships and Apply Transformation

Step 1: Create m-n Relationship


1. Create EMPLOYEES and DEPARTMENTS entities using the following definition:

EMPLOYEES

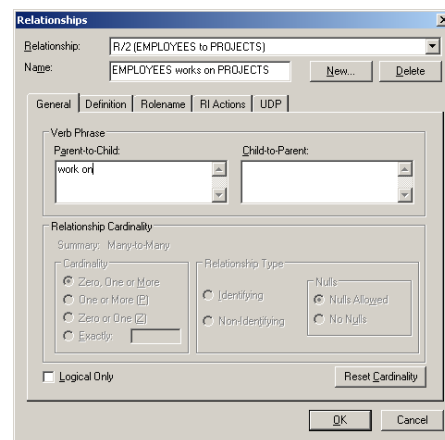
```
(
  ssn          varchar(10)    NOT NULL,
  emp_address  varchar(10),
  emp_fname    varchar(50),
  emp_lname    varchar(50)
)
```

PROJECTS

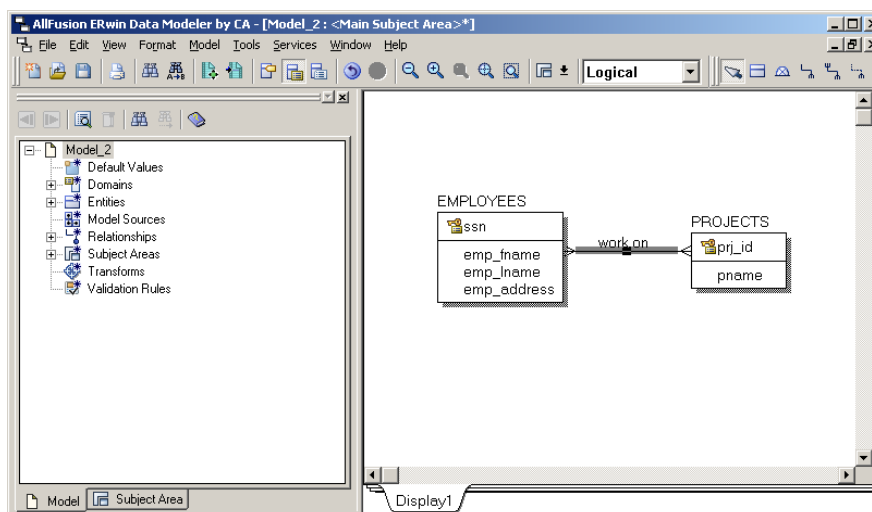
```
(
  prj_id       number          NOT NULL,
  prj_name     varchar(50),
)
```

2. Click on the 'Many-to-many relationship' icon ().


3. First click on EMPLOYEES then PROJECTS.
 - Name: EMPLOYEES works on PROJECTS
 - Parent-to-Child: work on
 - Click [OK]

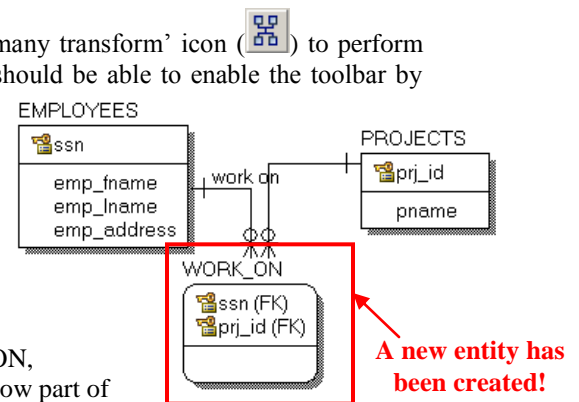


Your screen should look like this:



Step 2: Invoke 'Many-to-Many Transform Wizard'


1. Select the 'work on' relationship and then click on the 'Many to many transform' icon () to perform transformation. In case you don't see the transform toolbar, you should be able to enable the toolbar by checking on menu 'View→Toolbars→ Transforms '.
2. Follow the 'Many-to-Many Transform Wizard',
 - Click on [Next]
 - Click on [NEXT]
 - Entity Name: WORKS_ON, then click on [NEXT]
 - Click on [Finish]



Now your screen should look like this. A new entity type, WORK_ON, is created by the wizard and the primary key from both entities are now part of the primary key of the new entity. You can further modify the cardinality by double clicking on the relationship you intent to change. See Tutorial 2 for how to do this.

3. Save

Tutorial 5: Generate DB Schema**Steps:**

1. Complete Tutorial 2
2. Switch to Physical Model. From menu bar click 'Model → Physical Model'
3. From menu bar click 'Tools → Forward Engineering → Schema Generation'
4. In Forward Engineering Schema Generation dialogue box, feel free to navigate the 'Option' tab for all available options. You can specify whether to generate certain options simply by enable or disable them.
5. Click on [Preview]. If you are happy with what you are seeing, click on the 'Save' icon () to save the schema to your disk.

```

CREATE TABLE ART_OBJECT
(
    id_no          CHAR(18) NOT NULL ,
    year           CHAR(18) NULL  ,
    title          CHAR(18) NULL  ,
    description     CHAR(18) NULL  ,
    aname          CHAR(18) NOT NULL
);

CREATE UNIQUE INDEX XPKART_OBJECT ON ART_OBJECT
(id_no ASC,aname ASC);

ALTER TABLE ART_OBJECT
ADD CONSTRAINT XPKART_OBJECT PRIMARY KEY
(id_no,aname);

```

Table Filter: 10/10

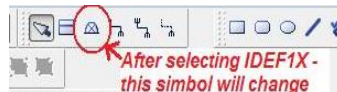
Generate... Close


Tutorial 6: Create Enhanced ER Diagram

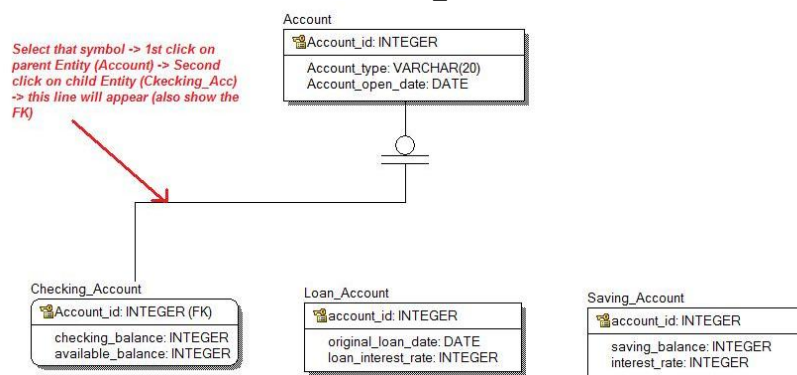
Step1: Create EER Diagram




Following are the steps that can help you to complete the EER diagram:

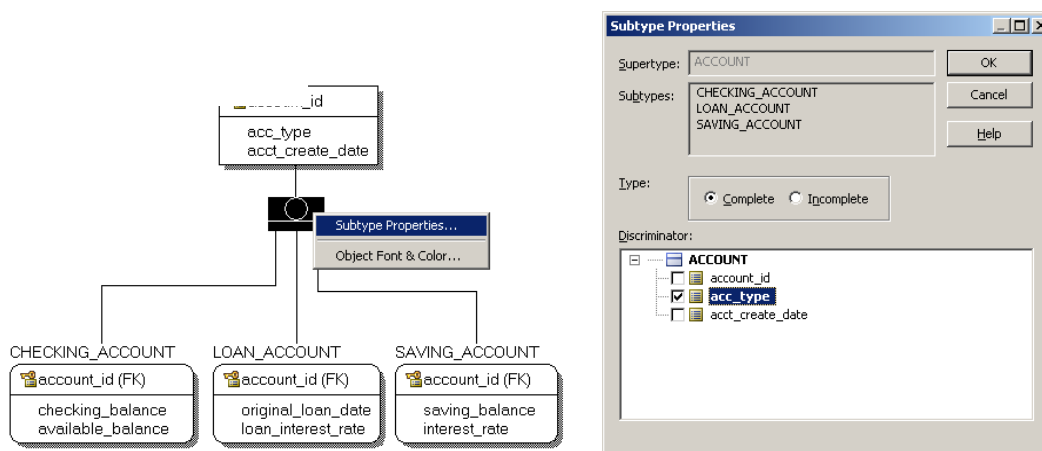
1. File → New
2. Model → Model Properties. Select 'IDEF1X' in the 'Logical Notation' for using EER Diagram




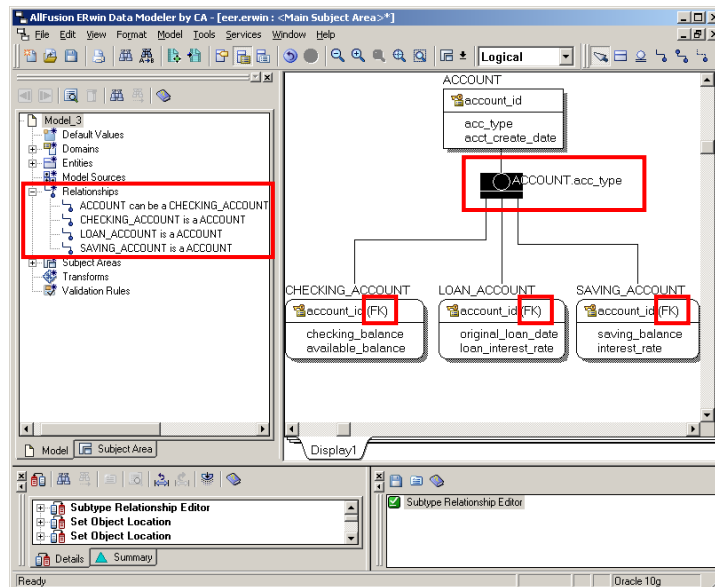
3. Create four entities namely: ACCOUNT, CHECKING_ACCOUNT, LOAN_ACCOUNT, and SAVING_ACCOUNT
4. From Toolbar, select 'Exclusive sub-category' icon ().
5. Click on 'ACCOUNT' first then 'CHECKING_ACCOUNT'.



6. To join super and sub classes, select 'Exclusive sub-category' icon () again.
7. Click on the sub-type circle () then click on the child entity. Repeat 6, 7 until done.
8. To specify the discriminator attribute, right click on the sub-type circle () and select 'Subtype Properties' from the popup menu. Select 'acct_type' as the discriminator.

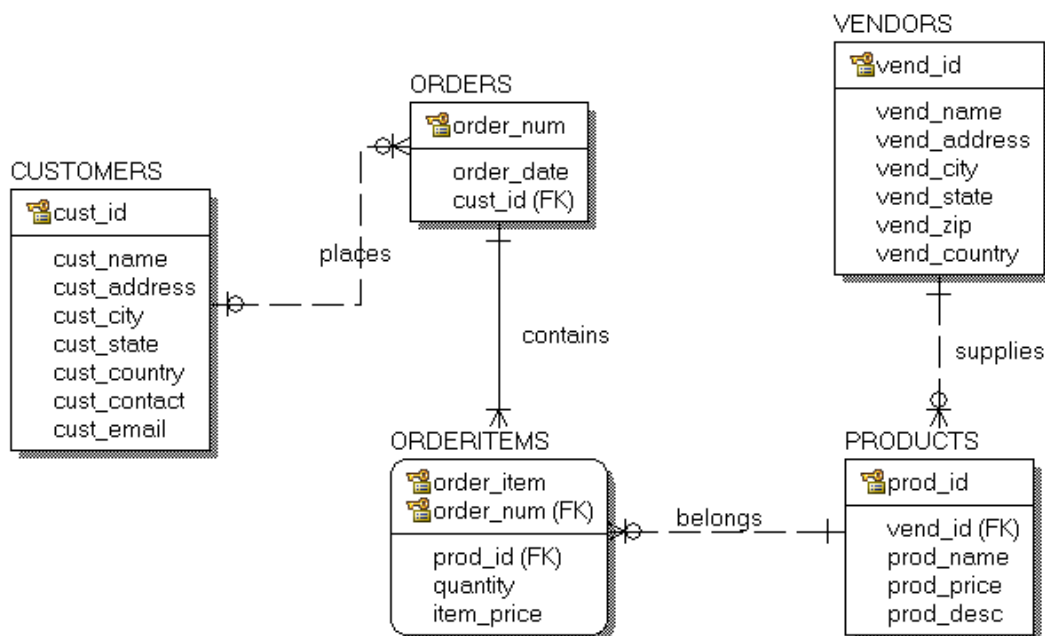


9. Click on the sub-type circle () then click on the child entity. Repeat 6, 7 until done.
10. Once done, in the Model Explorer on the left-hand-side, you will see the following relations
 - Relationships:
 - ACCOUNT can be a CHECK_ACCOUNT, LOAN_ACCOUNT, SAVING_ACCOUNT
 - CHECK_ACCOUNT is a ACCOUNT
 - LOAN_ACCOUNT is a ACCOUNT
 - SAVING_ACCOUNT is a ACCOUNT



11. Save

In-class Exercise Solution – The Complete ER Diagram of the SALES database



Lab Assignments

Exercise 1: ER Diagram for COMPANY database

You will create the ER diagram for the COMPANY database (see Figure 5.6, Figure 5.7, and Figure 8.2 in your textbook) using Erwin. In this exercise you will demonstrate the skills obtained from tutorial 1, 2, 3 and 4.

- Create all entities.
- Change entity names.
- Create all attributes.
- Modify appropriate data types, primary key, foreign key, and not null constraints.
- Define the relationships in the COMPANY database.
- Create relations, and give the names.
- Please use the crow's foot notation for your logical model.

Figure 5.6

One possible database state for the COMPANY relational database schema.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

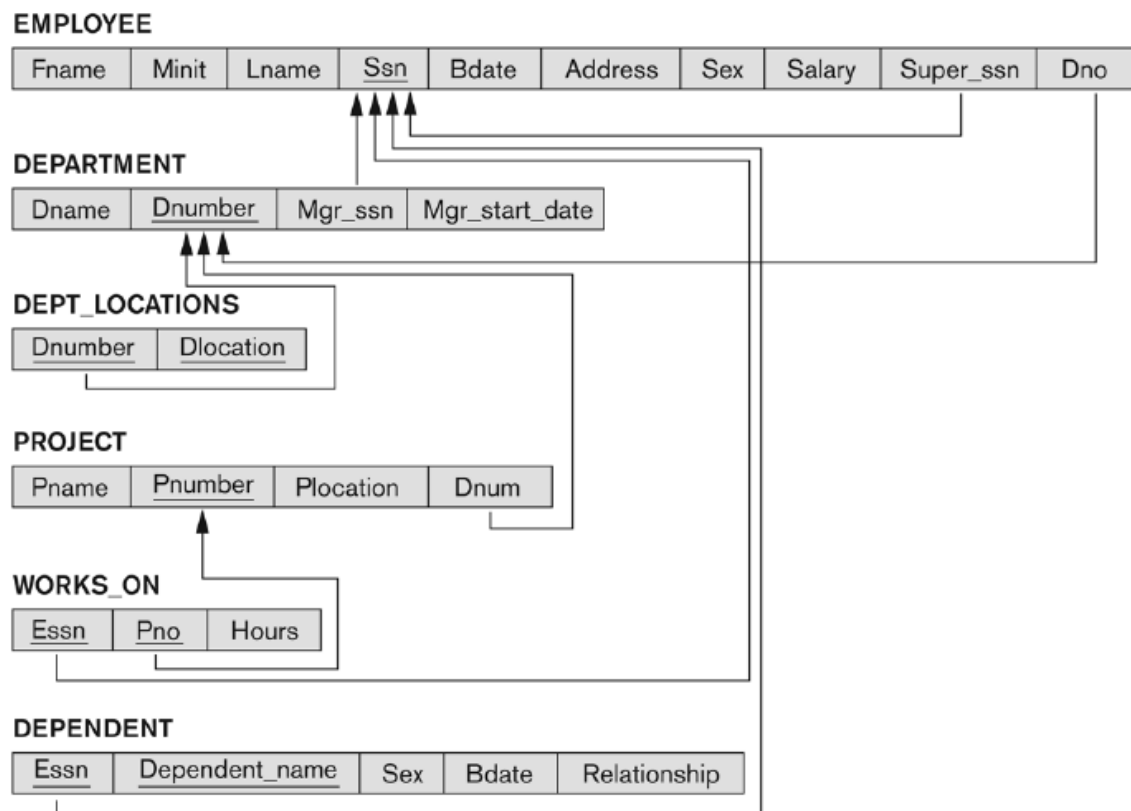
Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

Figure 5.7

Referential integrity constraints displayed on the COMPANY relational database schema.



```

CREATE TABLE EMPLOYEE
( ...,
  Dno          INT          NOT NULL          DEFAULT 1,
  CONSTRAINT EMPCHK
    PRIMARY KEY(Ssn),
  CONSTRAINT EMPSUPERFK
    FOREIGN KEY(Super_ssn) REFERENCES EMPLOYEE(Ssn)
      ON DELETE SET NULL      ON UPDATE CASCADE,
  CONSTRAINT EMPDEPTFK
    FOREIGN KEY(Dno) REFERENCES DEPARTMENT(Dnumber)
      ON DELETE SET DEFAULT  ON UPDATE CASCADE );

CREATE TABLE DEPARTMENT
( ...,
  Mgr_ssn      CHAR(9)     NOT NULL          DEFAULT '888665555',
  ...,
  CONSTRAINT DEPTPK
    PRIMARY KEY(Dnumber),
  CONSTRAINT DEPTSK
    UNIQUE(Dname),
  CONSTRAINT DEPTMGRFK
    FOREIGN KEY(Mgr_ssn) REFERENCES EMPLOYEE(Ssn)
      ON DELETE SET DEFAULT  ON UPDATE CASCADE );

CREATE TABLE DEPT_LOCATIONS
( ...,
  PRIMARY KEY(Dnumber, Dlocation),
  FOREIGN KEY(Dnumber) REFERENCES DEPARTMENT(Dnumber)
    ON DELETE CASCADE      ON UPDATE CASCADE );

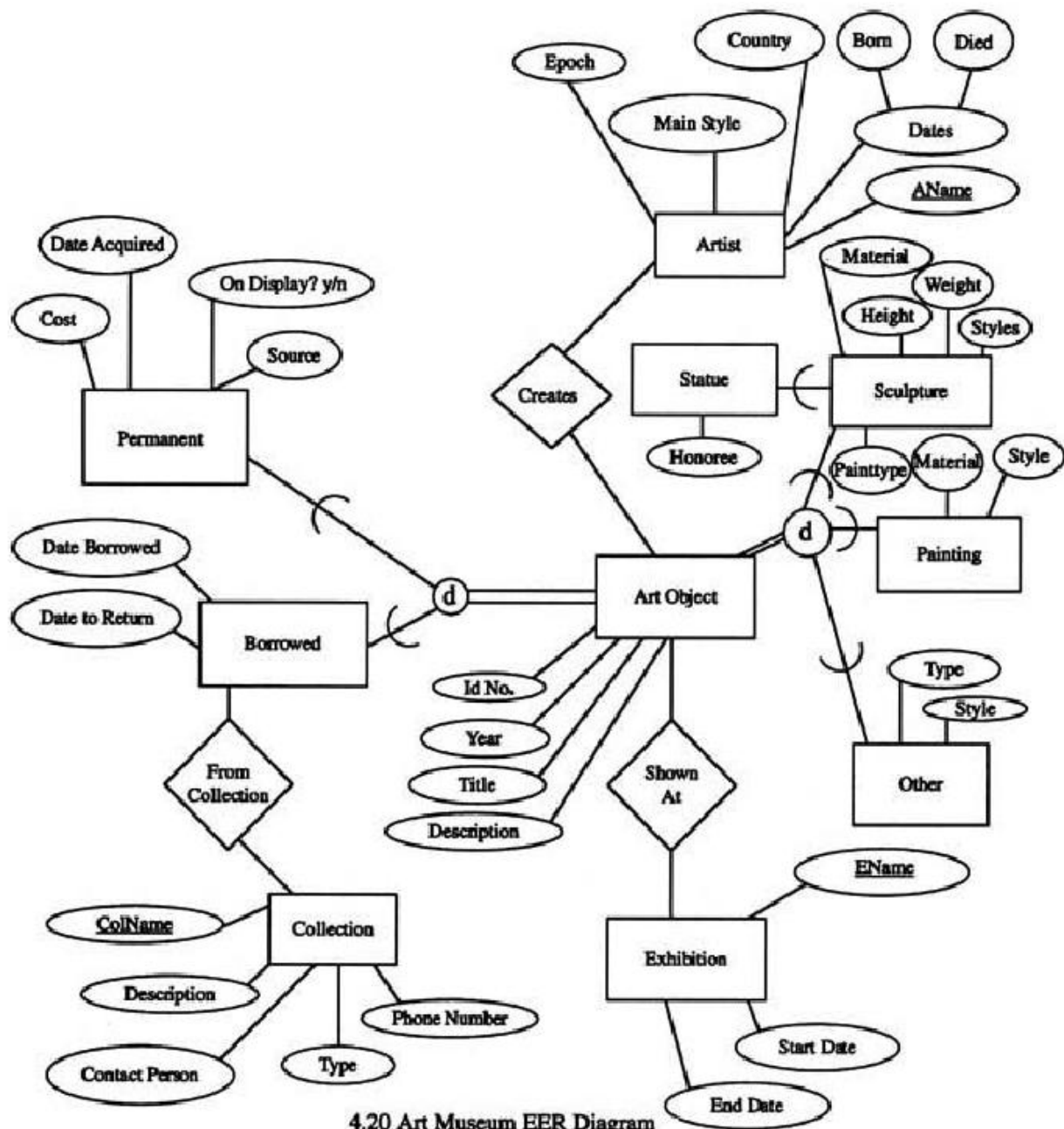
```

Figure 8.2

Example illustrating how default attribute values and referential integrity triggered actions are specified in SQL.

Exercise 2: EER Diagram for Art Museum database

This is the EER diagram for Art Museum database (see Chapter 4, Exercise 4.20 for complete requirement). From the diagram, create EER diagram using ERwin. You need to study by yourself how to generate EER diagram using Erwin. In this exercise you will demonstrate the skills obtained from tutorial 1, 2, 3, 4 and 6.



4.20 Art Museum EER Diagram

Exercise 3: ER modeling

Design an ER schema for keeping track of information about vote taken in the U.S. House of Representatives during the current two-year congressional session. The database needs to keep track of each U.S. **STATE**'s Name (e.g., 'Texas', 'New York', 'Connecticut') and include the **Region** of the state (whose domain is {'Northeast', 'Midwest', 'Southeast', 'Southwest', 'West'}). Each **CONGRESS_PERSON** in the House of Representatives is described by his or her **Name**, plus the **District** represented, the **Start_date** when the congressperson was first elected, and the political **Party** to which he or she belongs (whose domain is {"Republican", 'Democrat', 'Independent', 'Other'}). The database keeps track of each **BILL** (i.e., proposed law), including the **Bill_name**, the **Date_of_vote** on the bill, whether the bill **Passed_or_failed** (whose domain is {'Yes', 'No'}, and the **Sponsor** (the congressperson(s) who sponsored-that is, proposed-the bill). The database keeps track of how each congressperson voted on each bill (domain of Vote attribute is {'Yes', 'No', 'Abstain', 'Absent'}). In this exercise you will demonstrate the skills obtained from tutorial 1, 2, 3 and 4.

- Find Entities.
- Find Relationships.
- Draw an ER schema diagram using ERwin with appropriate options.

State clearly any assumptions you make.

How to submit Lab1 (Exercise 1 to 3)

Exercise 1

Save your model into ERwin type (file name: "Your Last Name"_"UB ID"_e1.erwin). For example, in case of "John Smith" and "1234567", the file names are smith_1234567_e1.erwin.

Exercise 2

Save your file into ERwin type (file name: "Your Last Name"_"UB ID"_e2.erwin). For example, in case of "John Smith" and "1234567", the file names are smith_1234567_e2.erwin.

Exercise 3

Save your file into ERwin type (file name: "Your Last Name"_"UB ID"_e3.erwin). For example, in case of "John Smith" and "1234567", the file names are smith_1234567_e3.erwin.

Lab assignment is due in 1 week. After you create three files, you must make one zip file, "Your Last Name"_"UB ID"_LAB1.zip. Then, you will submit your program using blackboard digital drop-box (<http://bb.ctdlc.org/>) by 11:59:59 PM. If you have any trouble to use blackboard, you can contact GA or instructor.