Computer Science Department  
Michigan State University  
CSE480 Database Systems  
Lab Week #3

(40 MINUTES)

1. Requirements specification and corresponding EER diagram for lab 3 are given.

2. Add the following additional requirements to the EER diagram. Add new entity types and attribute types as needed.

   (a) Departments have joint degree programs (e.g., CSE and EE department jointly offer B.S. degree in computer Engineering. Note that this is not a situation at MSU because MSU does not allow a degree program to be administered by more than one Department).

   (b) Graduate students can teach courses.

   (c) Students can mentor other students (e.g., a senior student mentoring some junior students).

   (d) Each Laboratory has a faculty as director.

   (e) A faculty can supervise multiple Projects.

   (f) Students using certain labs for certain research projects.

   (g) Make changes in the EER diagram to indicate that every graduate student must have an advisor.

3. (a) An instructor may have office hours which

   i. May correspond to each course OR

   ii. May correspond each instructor (i.e, same office hours for all the courses the instructor is teaching)

How can each one of the above two cases be incorporated, independently, in the EER diagram?
(b) Consider now instructor as a subclass of the faculty? Make necessary modifications to the EER diagram to incorporate the changes.

(ONE HOUR)

4. In this part of the lab you will implement 1:1, 1:N and N:M relationship constraints in oracle. Examples given below may not include all the attributes in the tables created last week. Objective is to run the following and understand why they work.

(a) Implement 1:1 relationship chair between Department and Faculty entity type in the EER diagram:
Refer to the sql file that you created in last week’s lab containing the table definitions for the entity types. Let us call that file X.sql.
In X.sql file add a foreign key to the Department table by adding the column chair in the Department table. Note that we need to create the Faculty table first.

drop table Department;
drop table Faculty;
create table Faculty(
FacSSNo    varchar2(9) CONSTRAINT PK_Faculty PRIMARY KEY,
FacName    varchar2(30));
insert into Faculty values(’000405555’, ’John Doe’);
insert into Faculty values(’111506666’, ’Mark Moore’);
select * from faculty;

Open another window and get into sqlplus
start X.sql
Include a foreign key in the definition of Department table in the X.sql file.

create table Department(
Did     number(7) CONSTRAINT PK_Department PRIMARY KEY,
DeptName varchar2(30),
chair   varchar2(9),
CONSTRAINT FK_chair_Department FOREIGN KEY (chair)
REFERENCES Faculty(FacSSNo));
insert into Department values(01, 'CSE', '000405555');
select * from Department;
select * from Student;

start X.sql
insert the following into X.sql
insert into Department values(04, 'EE', '000405555');
select * from Department;

start X.sql

Should CSE and EE have the same faculty as chair in 1:1 constraint? no
System did not detect it. Following will prevent inserting the second tuple.
Change Department table to make the foreign key a unique attribute.
Go back to X.sql and make the chair column of the department table UNIQUE as follows:

create table Department(
    Did number(7) CONSTRAINT PK_Department PRIMARY KEY,
    DeptName varchar2(30),
    chair varchar2(9) UNIQUE,
    CONSTRAINT FK_chair_Department FOREIGN KEY (chair)
        REFERENCES Faculty(FacSSNo));

start X.sql;
Does it accept the second tuple in the Department table now?
Now create a table for DegreeProgram and then implement the coordinator 1:1 relationship.

(b) 1:N relationship constraint is implemented exactly the same way as 1:1 relationship except that UNIQUE is not needed. Foreign key should be defined in the table on N side.
Now you implement two 1:N relationships from the EER diagram.
(c) For N:M relationship constraint a new table is created as follows:

\textbf{offers} relationship between \textbf{Department} and \textbf{Course Description}:

```
Create table offers(
   did   number(7),
   cno   number(8),
   CONSTRAINT PK_offers PRIMARY KEY (did,cno),
   CONSTRAINT FK_did_offers FOREIGN KEY (did)
      REFERENCES Department(Did),
   CONSTRAINT FK_cno_offers FOREIGN KEY (cno)
      REFERENCES CourseDescription(Cno));
```

Insert a few example tuples into CourseDescription table. Now you have tuples in both CourseDescription and Department tables. Insert a few example tuples into the \textit{offers} table to create N:M relationship set.

Now you implement two other N:M relationships from the EER diagram.