CS 377: Database Systems

Homework #2 Solutions

1. **Emory College University Database** (35 points): Map the ER diagram from homework 1 for the Emory College database into a relational database. Underline the primary key in each relation table and draw an arrow from a foreign key to the relation that the foreign key is pointing to (i.e., its home relation). The ER diagram is shown below (Figure 1):

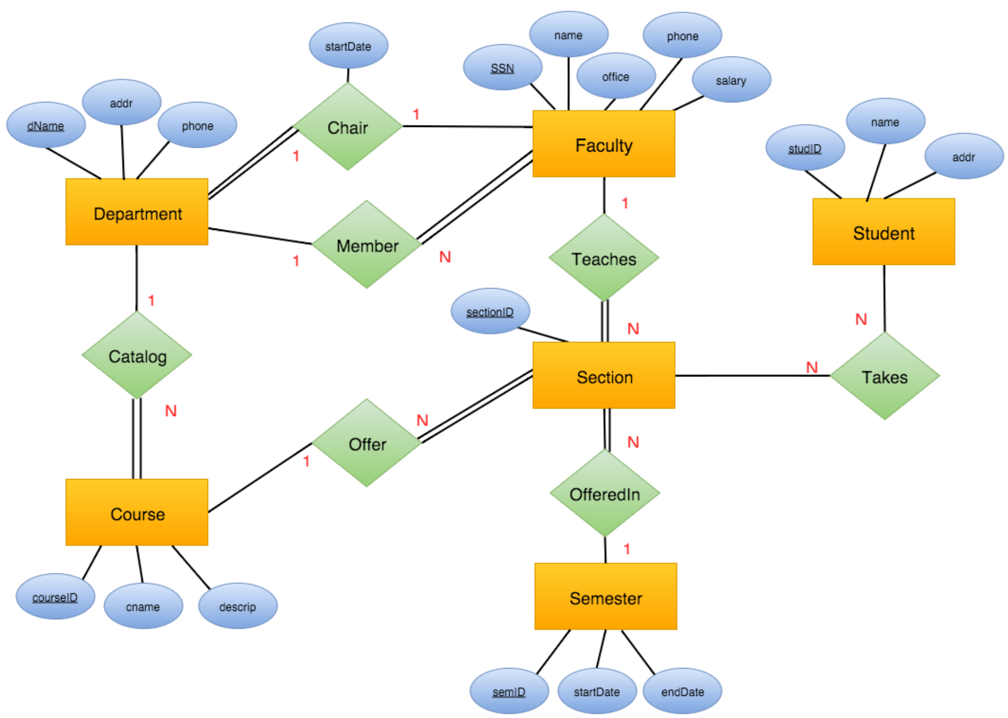

Figure 1: ER diagram for Emory College Database

(**ANSWER**) The previous homework solutions provide a template for the relational model. From the diagram, we convert the entities and their associated attributes to a relation:

- Department(<u>dName</u>, address, phone)
- Faculty(<u>SSN</u>, name, office, phone, salary)
- Course(<u>courseID</u>, cName, description)
- Semester(<u>semID</u>, startDate, endDate)
- Section(<u>secID</u>)
- Student(<u>studID</u>, name, addr)

The next step is to take care of the relationships. We will favor expanding existing relations over creating new relations

- Member(Department, Faculty) introduces a new column, dName, in the Faculty relation that is a foreign key to the primary key in Department.
- Chair(Department, Faculty) introduces two new columns in the Department relation, chairSSN and startDate, with chairSSN a foreign key that points to the primary key in Faculty. Note that this makes more sense because every department must have a chair. Adding the relationship to the Faculty relation will introduce too many NULL values.
- Member(Department, Faculty) introduces a new column, dName, in the Faculty relation that is a foreign key to the primary key in Department.
- Catalog(Department, Course) introduces a new column, offeringDept, in the course relation that is a foreign key to dName in Department.
- Offer(Section, Course) introduces a new column to section, courseID, which is a foreign key to courseID in the Course relation.
- OfferedIn(Section, Semester) introduces another foreign key semID which points to semID in the Semester relation.
- Teaches(Section, Faculty) expands the Section relation to have an attribute with the faculty's SSN (foreign key to Faculty relation)
- Takes(Student, Section) can not be easily captured by expanding existing relations. Thus a new relation is created that is Takes(StudID, secID).

Putting everything together yields the following relational data model.

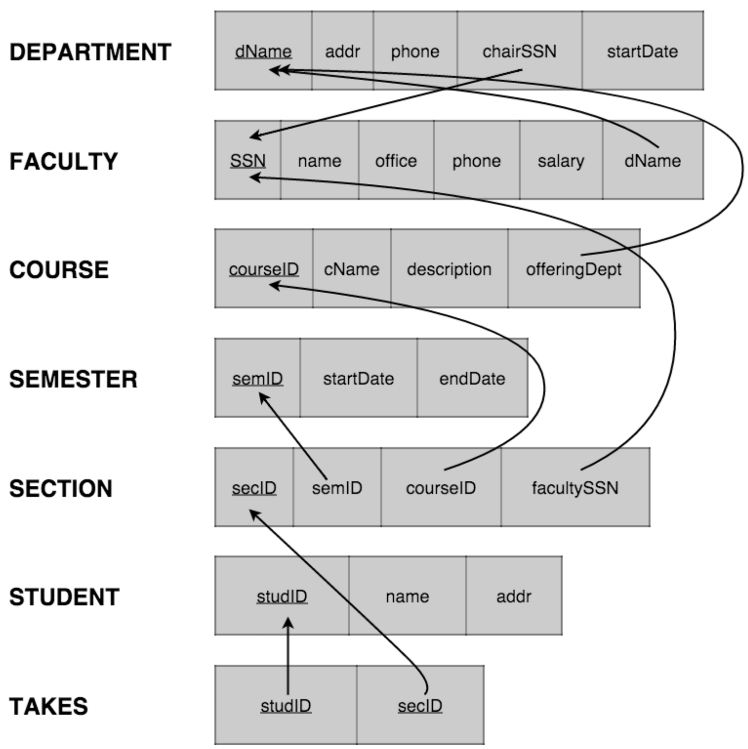

Figure 2: Relational Data Model for Emory College Database

2. **Library** (20 points): Consider the following relational schema for a library:

```
member(member_no, name, dob)
books(isbn, title, authors, publisher)
borrowed(memb_no, isbn, date)
```

Express the following queries in Relational Algebra:

(a) Find the names of members who have borrowed any book published by "McGraw-Hill".

(b) Find the name of members who have borrowed all books published by "McGraw-Hill".

(c) Find the name and membership number of members who have borrowed more than five different books published by "McGraw-Hill".

(d) For each publisher, find the name and membership number of members who have borrowed more than five books of that publisher.

(**ANSWER**)

(a)

$$\text{MGH\_BOOKS} = \sigma_{\text{publisher='McGraw-Hill'}}(\text{BOOKS}) \qquad \text{(books published by McGraw-Hill)}$$
$$\text{R2} = \text{BORROWED} * \text{MGH\_BOOKS}$$
(since both relations have isbn we can use natural join)
$$\text{ANSWER} = \pi_{\text{name}}(\text{MEMBER} \bowtie_{\text{member\_no=memb\_no}} \text{R2})$$

(b) Count each members number of books borrowed that was published by "McGraw-Hill" and join it with the total number of books published by McGraw-Hill.

$$\text{MGH\_BOOKS} = \sigma_{\text{publisher='McGraw-Hill'}}(\text{BOOKS})$$
$$\text{MEM\_BOOKS} = {}_{\text{memb\_no}}\mathcal{F}_{\text{count(isbn)}}(\text{BORROWED} * \text{MGH\_BOOKS})$$
$$\text{MGH\_COUNT} = \mathcal{F}_{\text{count(isbn)}}(\text{MGH\_BOOKS}) \qquad \text{(number of MGH books)}$$
$$\text{ANSWER} = \pi_{\text{name}}(\text{MEMBER} \bowtie_{\text{member\_no=memb\_no}} \text{MEM\_BOOKS} \bowtie_{\text{count=count}} \text{MGH\_COUNT})$$

(c)

$$\text{MGH\_BOOKS} = \sigma_{\text{publisher='McGraw-Hill'}}(\text{BOOKS})$$
$$\text{MEM\_BOOKS} = {}_{\text{memb\_no}}\mathcal{F}_{\text{count(isbn)}}(\text{BORROWED} * \text{MGH\_BOOKS})$$
$$\text{ANSWER} = \pi_{\text{name,memb\_no}}(\text{MEMBER} \bowtie_{\text{member\_no=memb\_no}} (\sigma_{\text{count}>5}(\text{MEM\_BOOKS})))$$

(d) This is an extension of (c), with the change that now we want to group on publisher and member number to find the number of books per member per publisher.

$$\text{MPB} = {}_{\text{publisher,memb\_no}}\mathcal{F}_{\text{count(isbn)}}(\text{BORROWED} * (\pi_{\text{isbn,publisher}}(\text{BOOKS})))$$
$$\text{ANSWER} = \pi_{\text{publisher,name,memb\_no}}(\text{MEMBER} \bowtie_{\text{member\_no=memb\_no}} (\sigma_{\text{count}>5}(\text{MPB})))$$

3. **Company Database Queries** (45 points): Consider the company database relational data model discussed in class and shown below. Formulate the following queries in Relational Algebra:

(a) Find the name of the projects in Atlanta that have been worked on at least a total of 100 person hours.

(b) Find the name of the department(s) that pay the highest salary.

(c) Find the fname & lname of the employee(s) who work the highest total number of hours.

(d) For each department, find the department name and the total salary paid to employees in the department.

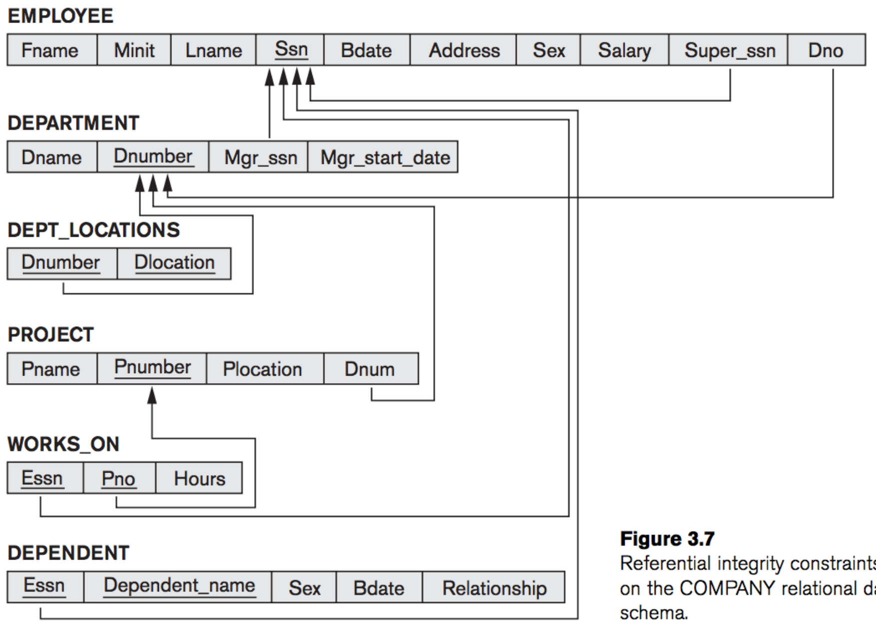

Figure 3.7
Referential integrity constraints displayed on the COMPANY relational database schema.

Figure 3: Company Database Relational Model

(e) Find the department(s) in which *all* employees in the department have at least one dependent.

(**ANSWER**)

(a)

$$\begin{aligned} \text{PH} &= {}_{\text{Pname,Pno}}\mathcal{F}_{\text{sum(Hours)}}(\sigma_{\text{Plocation='Atlanta'}}(\text{WORKS\_ON} \bowtie_{\text{Pno=Pnumber}} \text{PROJECT})) \\ \text{ANSWER} &= \pi_{\text{Pname}}(\sigma_{\text{sum}\geq 100}(\text{PH})) \end{aligned}$$

(b)

$$\begin{aligned} \text{MS} &= \mathcal{F}_{\text{max(Salary)}}(\text{EMPLOYEE}) \\ \text{MSE} &= \text{EMPLOYEE} \bowtie_{\text{Salary=max}} (\text{MS}) \\ \text{ANSWER} &= \pi_{\text{Dname}}(\text{DEPARTMENT} \bowtie_{\text{Dnumber=Dno}} \text{MSE}) \end{aligned}$$

(c)

$$\begin{aligned} \text{TEH(Essn, TotalHours)} &= {}_{\text{Essn}}\mathcal{F}_{\text{sum(Hours)}}(\text{WORKS\_ON}) \\ \text{MT(max)} &= \mathcal{F}_{\text{max(TotalHours)}}(TEH) \\ \text{EWMT} &= \text{TEH} \bowtie_{\text{TotalHours=max}} (\text{MT}) \\ \text{ANSWER} &= \pi_{\text{Fname,Lanme}}(\text{EMPLOYEE} \bowtie_{\text{Ssn=Essn}} \text{EWMT}) \end{aligned}$$

(d)

$$\text{ANSWER} = {}_{\text{Dname}}\mathcal{F}_{\text{sum(Salary)}}(\text{DEPARTMENT} \bowtie_{\text{Dnumber=Dno}} \text{EMPLOYEE})$$

(e) Solution #1: Find the departments in which no employees have zero dependents

$$R1 = \pi_{\text{Ssn}}(\text{EMPLOYEE} \bowtie_{\text{Ssn}=\text{Essn}} \text{DEPENDENT}) \quad \text{(employees with some dependent)}$$

$$R2 = \pi_{\text{Ssn, Dno}}(\text{EMPLOYEE}) - \text{R1} \quad \text{(employees with zero dependents)}$$

$$R3 = \pi_{\text{Dno}}(\text{DEPARTMENT} \bowtie_{\text{Dnumber}=\text{Dno}} \text{R2})$$

(departments with some employees that have zero dependents)

$$\text{ANSWER} = \pi_{\text{Dno}}(\text{DEPARTMENT}) - \text{R3}$$

Solution # 2: Count the number of employees with dependents per employee and the number of employees per department. If they are equal, then that department has at least one dependent.

$$R1 = {}_{\text{Dnumber}}\mathcal{F}_{\text{count(Ssn)}}(\text{DEPARTMENT} \bowtie_{\text{Dnumber}=\text{Dno}} \text{EMPLOYEE} \bowtie_{\text{Ssn}=\text{Essn}} \text{DEPENDENT})$$

$$R2 = {}_{\text{Dnumber}}\mathcal{F}_{\text{count(Ssn)}}(\text{DEPARTMENT} \bowtie_{\text{Dnumber}=\text{Dno}} \text{EMPLOYEE})$$

$$\text{ANSWER} = \pi_{\text{Dno}}(\text{DEPARTMENT} \bowtie_{\text{Dnumber}=\text{Dnumber}} \text{R1} \bowtie_{\text{Dnumber}=\text{Dnumber} \wedge \text{count}=\text{count}} \text{R2})$$