1. (20 points) The following incomplete E-R diagram tries to show a database for the course selection system. A student has student-id, name, major and email address. A course has course-id, instructor-name and credits. A student must select at least one course each term.

a. Please complete this E-R diagram based on above description. (Note: If there is any total participation or weak entity set, please also specify it by using double lines in this diagram.)

b. Students can be undergraduate students and graduate students, which have different score standard. We may just name their scores as U-score and G-score. Please extend this E-R diagram to model it.

Answer: A sample E-R diagram:
2. (20 points) Please use SQL to create relational tables based on E-R diagram you complete in Question 1. You should include the tables related to undergraduate students and graduate students.

Answer  We represent the following tables as SQL:

```sql
create table graduate(
    studentid integer,
    gscore numeric,
    primary key (studentid)
) ;
create table undergraduate(
    studentid integer,
    uscore numeric,
    primary key (studentid)
) ;
create table selectcourse(
    studentid integer,
    courseid integer,
    primary key (studentid,courseid)
) ;
create table course(
    courseid integer,
    instructorname char(30),
    credits integer,
    primary key (courseid)
) ;
create table student(
    studentid integer,
    name char(30),
    major char(20),
    email char(30),
    primary key (studentid)
) ;
```
3. (30 points) Consider the following relations:

Employee(eid: Integer, name: String, salary: real)
Works(eid: Integer, dname: String, title: String)
Department(dname: String, location: String, managerid: Integer)

Note: A manager is also an employee.

a. Write both relation algebra and SQL expressions for the following Query: “Find the names of the departments, such that the total salary of their employees is larger than the total salary of ‘Human Resource’ department”

Answer

Relational algebra:

\[ t_1 \leftarrow \sigma_{\text{department, sum(salary)}}(\text{employee} \bowtie \text{works}) \]
\[ t_2 \leftarrow \sigma_{\text{totalsalary as marketingsalary, department}= '\text{Human Resources}'}(t_1) \]
\[ \Pi_{\text{department, totalsalary }}> t_2.\text{marketingsalary}(t_1) \]

SQL:

create view departmentsalary as
select dname, sum(salary) as totalsalary
from employee natural join works group by dname ;

select dname
from departmentsalary
where totalsalary > (select totalsalary
from departmentsalary
where dname = 'Human Resources') ;

another solution:

select dname from works, employee
where works.eid = employee.eid
having sum (salary) > all(
select sum(salary)
from works, employee
where works.eid = employee.eid and
works.dname = 'Human Resources') ;

b. Consider the following datalog program:

\[ \text{Query}(x, y, d, l) : \neg P(x, y, d, l) \]
\[ P(x, y, d, l) : \neg \text{Employee}(n, x, z), z > 50000.00, \text{Works}(u, d, y), \text{Department}(d, l, m) \]
Please write down both relational algebra and the SQL expression for this query. (You can get 5 points if you only can write down this query by using natural language, i.e. English.)

**Answer**  Relational algebra:

\[ t_1 \leftarrow \sigma_{\text{employee.eid}=\text{work.eid} \land \text{employee.salary}>50000 \land \text{works.dname}=\text{department.dname}} \]

\[ \Pi_{\text{name}, \text{title}, \text{dname}, \text{location}}(t_1) \]

**SQL:**

```sql
select name, title, dname, location
from employee, works, department
where employee.eid=works.eid and employee.salary> 50000
    and works.dname=department.dname ;
```

or:

```sql
select name, title, dname, location
from (works natural join employee natural join department)
where salary > 50000 ;
```
4. (30 points) Suppose you are designing a database for a cellphone company. If we put all attributes in one relation, it looks like: `cellphones (customer_id (A), phone_number (B), plan (C), call_id (D), tonumber (E), call_start (F), call_minutes (G))`. To be simple, we represent it as \( R(A,B,C,D,E,F,G) \) and a set of functional dependencies \( F \) that hold for \( R \): \( \{ AEF \rightarrow D, B \rightarrow AC, BF \rightarrow D, D \rightarrow BEFG \} \).

a. List as many candidate key(s) for \( R \).

Answer BF, AEF, D

b. Represent as many FD in \( F^+ \) based on candidate keys (Hint: you can use “all of R” means that any combination of the attributes of R).

Answer Here is the start of an answer:

\[
F^+ = \{ \\
ACF \rightarrow \text{all} \\
AEF \rightarrow \text{all} \\
BF \rightarrow \text{all} \\
D \rightarrow \text{all} \\
B \rightarrow AC \\
\ldots \\
\}
\]

c. Please verify \( R \) is not in BCNF.

Answer \( B \rightarrow AC \) is nontrivial and \( B \) not a superkey, so therefore \( R \) is not in BCNF.

d. Can you get a BCNF from \( R \)? If yes, what’s your decomposition?

Answer Since \( B \rightarrow AC \) is not a candidate key, then remove \( AC \) and we get: \((BDEFG)(BAC)\). In \((BAC)\), \( B \rightarrow AC \) applies and \( B \) is a candidate key. In \((BDEFG)\), \( BF \rightarrow D \) and \( D \rightarrow BEFG \) both apply and \( BF \) and \( D \) are candidate keys, respectively.

Also, if a solution is in 3NF, as verified below, then it will be in BCNF, as well, if it is in 3NF and all the left-hand side functional dependencies are candidate keys. So \( (AEFD), (BAC), (BFD), (DBEFG) \) is another valid solution, as explained
below, though there are other valid solutions, as well.

e. Please verify (AEFD), (BAC), (BFD), (DBEFG) is in 3NF.

**Answer** We see that $F_c = F$ (i.e., no extraneous FD’s), so we can use $F$ in to compare. In (AEFD), $AEF \rightarrow D$, $AEF$ is the candidate key. In (BAC), $B \rightarrow AC$ applies and $B$ is the candidate key. In (BFD), $BF \rightarrow D$ applies and $BF$ is the candidate key. In (DBEFG), $D \rightarrow BEFG$ applies and $D$ is a candidate key.