Problem 1 (Relational algebra)

Following are relational schemas of a project management (PM) database:
project(pid, name, budget, tid)
employee(eid, name, age, gender)
team(tid, title, size, leader)
member(eid, tid, date)

Note:
project is identified by **pid**, and described by **name** ,**budget and tid, w**here **tid** is the primary key of the team that takes on the project.
employee is identified by **eid**, and described with **name**, **age** and **gender**
team is identified by **tid**, and described by **title** , **size,** and **leader**, where **leader** is the **eid** of an employee who leads the team.
member records the **date** when an employee(**eid**) participates in the a team(tid).

Please answer following questions.
(1) Transform following query to a SQL statement without nested subquery.

```
select eid, name
from employee as E
where age <20 and gender = 'F' and eid =some
        (select eid from member as M
         where M.eid = E.eid and date >= '2017-01-01' and tid in
        ( select tid from team as T where title='Cloud Migration'));
```

(2) Transform above query to an equivalent relational algebra expression.
(3) Write a relational algebra expression to find out the name of employees who is the member of all teams.

Problem 2 (SQL Query)

For the PM database in **problem 1,** please write SQL statements to fulfill following query requests.

(1) Find out the projects with max size of project team.
(2) Find out employees who is the member of only one team.
(3) Find out teams that takes on max number of projects.

Problem 3 (SQL View)

,

(1) For the PM database in **problem 1**, please define a view named **project_statistics** to capture the information of the total number and average age of all team members for each project.

(2) Please write a SQL statement to grant the select priviledge on the view above to all database users.

Problem 4 (SQL Transaction)

For the PM database in **problem 1**, please write a SQL transaction that moves **employee** “e001” from **team** “t001” to **team** “t002”. Note: when an employee becomes a member of a team, the **size** of the team should be increased by 1, and vise versa.

Problem 5 (SQL trigger)

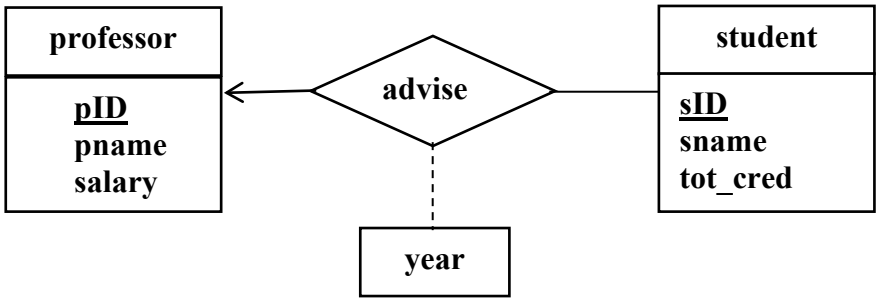
For the PM database in **problem 1**, please write a SQL trigger to implement following functionality: if an employee becomes a member of a team (i.*e.,* inserting a record into **member** table), then the **size** of the **team** is automatically increased by 1.

Problem 6 (E-R Model)

Meeting Preservation System has a database to store information about **meeting rooms**, **telephone lines**, **employees**, and **meetings**, etc. A meeting room is identified by name, and described by capacity, condition, and state. A telephones line is identified by no#, and described by area code, phone number and passcode. An employee is identified by eid, and described by name, age, gender, department, email, and a couple of phones. Every meeting has a unique name, and is described by start time, end time. A meeting preserves a meeting room, and has a host who is also an employee. A meeting may also preserve a telephone line for offsite participants to call in. An employee may participate in several meetings.

(1) Draw an ER diagram that captures this information.
(2) Transform the ER diagram into relational schemas.

Problem 7 (Relational Database Design)



Above is an E-R diagram that depicts one-to-many relationship **advise** between two entity sets **professor** and **student**. The candidate key of the entity sets are underlined in the diagram. If the E-R diagram is transform to following relation schema:

advisor(pID, pname, salary, year , sID, sname, tot_cred).

Please answer following questions:

(1) Give out the functional dependency set F that holds on relation **advisor**
(2) List the candidate key(s) of relation **advisor**.
(3) Explain that relation **advisor** is not in BCNF.
(4) Explain the pitfalls of relation **advisor**.
(5) Prove that decomposing of relation **advisor** into two relation schemas **professor(pID, pname, salary)** and **student(sID, sname, tot_cred, pID, year)** is lossless join decomposition.

Problem 8(Relational Formalization)

For the relation schema R(A, B, C, D, E) with the functional dependencies set F={A→B,A→C, C→D,C→E, A→D,D→E},

(1) Compute the Canonical Cover of F.
(2) Compute the Closure of attribute set {B,D} .
(3) Decompose the relation R into a collection of BCNF relations, and give out the functional dependency set for each relation.
(4) Explain whether above decomposition is dependency preserving or not.