Normalization

Q. 1: Given the relation
Book_Pub_Author(Title, PubId, AuId, Price, AuAddress)
The functional dependencies are given below.

1. Key □ {Title, PubId, AuId}
2. {Title, PubId, AuID} □ {Price}
3. {AuID} □ {AuAddress}

Which normal form is the relation? Normalize it up to 3NF.

Q. 2: Given the relation

City_Population(City, Street, HouseNumber, HouseColor, CityPopulation)
1. **key □ {City, Street, HouseNumber}**
2. **{City, Street, HouseNumber} □ {HouseColor}**
3. **{City} □ {CityPopulation}**

Which normal form is the relation? Normalize it up to 3NF.

Ans: Solved in the slide

Q.3. Given the relation R(A, B, C, D, E, F, G, H)

The functional dependencies are given as follows:

ABC □ DEFGH
So ABC is the primary key. The other functional dependencies are given as follows:
BC □D
ABC □ E
E □ F
E □ G

Q. Which normal form R is?

Ans: The relation is in 1NF.

Q. Is the relation is 2NF? Why?

The relation is not 2NF because BC □D is a partial dependency of non-key attribute. No other partial dependency.

Q. Transform it into 2NF.

Removing the partial dependency BC $\to$ D we get

R1(<u>B, C</u>, D)
The primary key and other attributes except D forms
R2(<u>A, B, C</u>, E, F, G, H)

Q. Is R1 and R2 are in 3NF? Why?
Ans. R1 is in 3NF because there is no transitive dependency.
R2 is not in 3NF because there are transitive dependency (non-key attribute have dependency to one another) for the following:

ABC $\to$ E, E $\to$ F So ABC $\to$ E (transitive dependency)
ABC $\to$ E, E $\to$ G, So ABC $\to$ G (transitive dependency)

Q. Transform R2 into 3NF.

Ans. Removing the attributes related to transitive dependencies (non-key attribute have dependency to one another) i.e., E and G and adding primary key for them as E, we get

R2-1(<u>E</u>, F, G)
With the remaining attributes of R2, we get
R2-2(<u>A, B, C</u>, E, H)

These two relations are in 3NF.

User Management

Q1:

There three students users S1, S2 and S3, two teacher users T1 and T2 and one departmental head (Dr. Sajjad) user.

The relational schema has been given as follows:

Student (Sid, name, street, city. Mobile, email, CGPA, age, tot-cred)
Enroll (course-id, Sid, semester, year, grade)
Course (course-id, title, credit-hour)
Teacher (Tid, name, designation, street, city, Mobile, email, salary, date-of-birth)
Offer (course-id, Tid, semester, year, remuneration)

S1, S2 and S3 can insert and update on student and enroll table, T1 and T2 can insert and update on teacher and offer table and head can insert, delete and update to all tables.

Manage these users by
   i.   Granting individual privileges
   ii.   Creating and granting role (st for student, tch for teacher and hd for head).
   iii.   Give a comparison of these two methods.

Solution:

   i.   Granting individual privileges

Grant insert. Update on student to S1, S2, S3
Grant insert. Update on enrol to S1, S2, S3
Grant insert. Update on teacher to T1, T2
Grant insert. Update on offer to T1, T2
Grant insert. Delete, Update on student to sajjad
Grant insert. Delete, Update on enrol to sajjad
Grant insert. Delete, Update on teacher to sajjad
Grant insert. Delete, Update on offer to sajjad
Grant insert. Delete, Update on course to sajjad

ii Creating and granting role (st for student, tch for teacher and hd for head).

Create role st;
Grant insert. Update on student to st;
Grant insert. Update on enrol to st;

Grant st to S1;
Grant st to S2;

Create role tch;
Grant insert. Update on teacher to tch;
Grant insert. Update on offer to tch;
Grant tch to T1;
Grant tch to T2;

Create role hd;
Grant insert. Delete, Update on student to hd
Grant insert. Delete, Update on enrol to hd
Grant insert. Delete, Update on teacher hd;
Grant insert. Delete, Update on offer to hd;
Grant insert. Delete, Update on course to hd;
Grant hd to sajjad;

Q2:

New head is Dr. Rashed. How can you authorize Dr. rashed by
   i.   Revoking and Granting individual privileges
  ii.   Revoking and granting role.

Ans.:

   i.   Revoking and Granting individual privileges

Revoke insert. Delete, Update on student from sajjad
Revoke insert. Delete, Update on enrol from sajjad
Revoke insert. Delete, Update on teacher from sajjad
Revoke insert. Delete, Update on offer from sajjad
Revoke insert. Delete, Update on course from sajjad

Grant insert. Delete, Update on student to rashed
Grant insert. Delete, Update on enrol to rashed
Grant insert. Delete, Update on teacher to rashed
Grant insert. Delete, Update on offer to rashed
Grant insert. Delete, Update on course to rashed

   i.   Revoking and granting role.

Revoke hd from sajjad;
Grant hd to rashed;

Function, Procedure

Q.1. Given the relation
student (id, name, dept_name, tot-credit).

Write a function named student-status that will take id as input parameter and return the status as follows:
If tot-credit is less than 50, the status will be 'Junior'
If tot-credit is less than 100 and greater than or equal to 50, the status is 'mid level'
Otherwise, the status is 'Senior'.

Write an SQL statement to find the id and status of all students.

Answer:

```
Create function student-status(st-id int) return varchar2
Begin
        declare st-stat varchar2;
        temp int;
        Select tot-credit into temp
        From student
        Where student.id = st-id
        If (temp <50) then
                st-stat = 'Junior';
        Elseif (temp < 100) then
                st-stat = 'mid level'
        Else
                st-stat = 'Senior'
        Endif
Return st-stat
End;
```

```
Select id, student-status(id)
From instructor
```

Procedure

Q4. Given the relation
Instructor(id, name, dept_name, salary).

Write a procedure named salary-update that will take two ids (i-id1 and i-id2) as input parameter and update the salaries as follows:
Consider i-id2 is always greater than i-id1.
For all instructors id less than or equal to i-id1, salary will be increased 20%.
For all instructors id less than or equal to i-id2 and greater than i-id1, salary will be increased 15%.
For all other instructors, salary will be increased 10%.

Call the procedure for updating salaries of instructors as above with i-di1 = 50 and i-id2 =80.

```
Create procedure salary-update (IN i-id1 int, IN i-id2 int)
Begin
        Update instructor
        Set salary = salary * 1.2
        Where id <= i-id1;
        Update instructor
        Set salary = salary * 1.15
        Where id <= i-id2 AND id > i-id1;
        Update instructor
        Set salary = salary * 1.1
        Where id > i-id2;
End


Call salary-update(50, 80);
```