Sample Questions for QUIZ-3

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.

Ans: 1NF

3NF:

R1(AuID, AuAddress)

R2(Title, PubId, AuId, Price)

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: 1NF.

3NF:

R1(City, CityPopulation)

R2(City, Street, HouseNumber, HouseColor)

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. 1NF.

Q. Is the relation is 2NF? Why?

Ans. No. because (BC 🡪D)

Q. Transform it into 2NF.

Ans. R1(B, C, D)

R2(A, B, C, E, F,G, H)

Q. Is R1 and R2 are in 3NF? Why?

Ans. R1 is in 3NF. But R2 is not 3NF. Because (E 🡪 F, E 🡪 G)

Q. Transform R2 into 3NF.

Ans. R2-1(E, F, G)

R2-2(A, B, C, E, H)

**Q4:** The relation schema and the functional dependencies are given as follows:

R(A, B, C, D, E, F, G, H, I)

ABC → DEFGHI

B → G, B → H, D → E, D → I

Normalize up to 3NF.

Ans.

R1(B, G, H)

R2-1(D, E, I)

R2-2(A, B, C, D, F)

User Management

Q5:

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

1. Granting individual privileges
2. Creating and granting role (st for student, tch for teacher and hd for head).
3. Give a comparison of these two methods.
4. GRANT INSERT , UPDATE ON Student TO S1, S2, S3;

GRANT INSERT, UPDATE ON Enroll TO S1, S2, S3;

GRANT INSERT, UPDATE ON Teacher TO T1, T2;

GRANT INSERT, UPDATE ON Offers TO T1, T2;

GRANT INSERT, UPDATE, DELETE ON Student TO “Dr. Sajjad”;

GRANT INSERT, UPDATE, DELETE ON Enroll TO “Dr. Sajjad”;

GRANT INSERT, UPDATE, DELETE ON Course TO “Dr. Sajjad”;

GRANT INSERT, UPDATE, DELETE ON Teacher TO “Dr. Sajjad”;

GRANT INSERT, UPDATE, DELETE ON Offer TO “Dr. Sajjad”;

1. CREATE ROLE st;

CREATE ROLE tch;

CREATE ROLE hd;

GRANT st TO S1, S2, S3;

GRANT tch TO T1, T2;

GRANT hd TO “Dr. Sajjad”;

GRANT INSERT, UPDATE ON Student TO st;

GRANT INSERT, UPDATE ON Enroll TO st;

GRANT INSERT, UPDATE ON Teacher TO tch;

GRANT INSERT, UPDATE ON Offers TO tch;

GRANT INSERT, UPDATE, DELETE Student TO hd;

GRANT INSERT, UPDATE, DELETE Course TO hd;

GRANT INSERT, UPDATE, DELETE Teacher TO hd;

GRANT INSERT, UPDATE, DELETE Offer TO hd;

1. Comparison:

Individual Privileges:

Pros:

- **Fine-grained control:** Each user gets specific privileges tailored to their role.

- **Flexibility**: Privileges can be easily adjusted for each individual user.

Cons:

- **Management overhead**: Each user's privileges need to be managed separately.

- **Increased risk of errors**: Misconfiguration might occur when setting up privileges for each user individually.

Role-based Privileges:

Pros:

- Simplified management: Users are grouped into roles, making it easier to manage privileges for similar users.

- Consistency: Ensures that users with the same role have the same set of privileges, reducing the risk of misconfiguration.

Cons:

- Less flexibility: Users within the same role have identical privileges, which might not always align perfectly with individual needs.

- Limited granularity: Roles might not cover all possible combinations of privileges, leading to the creation of additional roles or resorting to individual privilege assignments.

In summary, individual privileges offer more fine-grained control but require more management effort, while role-based privileges provide simplicity and consistency at the cost of some flexibility. The choice between the two depends on the specific requirements and complexity of the system.

Q6:

New head is Dr. Rashed. How can you authorize Dr. rashed by

1. Revoking and Granting individual privileges
2. Revoking and granting role.
3. REVOKE INSERT, DELETE, UPDATE ON Student FROM “Dr. Sajjad”;

REVOKE INSERT, DELETE, UPDATE ON Enrolls FROM “Dr. Sajjad”;

REVOKE INSERT, DELETE, UPDATE ON Course FROM “Dr. Sajjad”;

REVOKE INSERT, DELETE, UPDATE ON Teacher FROM “Dr. Sajjad”;

REVOKE INSERT, DELETE, UPDATE ON Offer FROM “Dr. Sajjad”;

GRANT INSERT, DELETE, UPDATE ON Student TO “Dr. Rashed”;

GRANT INSERT, DELETE, UPDATE ON Enrolls TO “Dr. Rashed”;

GRANT INSERT, DELETE, UPDATE ON Course TO “Dr. Rashed”;

GRANT INSERT, DELETE, UPDATE ON Teacher TO “Dr. Rashed”;

GRANT INSERT, DELETE, UPDATE ON Offer To “Dr. Rashed”;

1. REVOKE hd FROM “Dr. Sajjad”;

GRANT hd TO “Dr. Rashed”;

Integrity Constraint

Q7:

Given the schema

Student (Sid, name, street, city. Mobile, email, CGPA, age, tot-cred)

Enroll (course-id, Sid, semester, year, grade)

Define DDL to have the following integrity constraints:

1. The composite primary key will be (course-id, Sid, semester, year).
2. Semester can be only ‘Spring’, ’Summer’ or ’Fall’.
3. Year must be higher than 2017.
4. Grade must be 0 to 4
5. Sid not present in student is not allowed in the enroll table.

Answer:

Create Table Student

(

Sid INT PRIMARY KEY,

name VARCHAR(50),

street VARCHAR(50),

city VARCHAR(50),

Mobile INT,

email VARCHAR(50),

CGPA INT,

Age INT,

tot-cred INT

)

Create Table Enroll

(

course\_id INT,

Sid INT,

semester VARCHAR(50) CHECK (semester IN (‘Spring’, ‘Summer’, ‘Fall’)),

year INT CHECK (year > 2017),

grade INT CHECK(grade BETWEEN 0 AND 4),

PRIMARY KEY(course\_id, Sid, semester, year),

FOREIGN KEY(Sid) REFERENCES Student(Sid)

)

**View**

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)

Q8: create a view st-grade-22 with Sid, name, course-id, grade for year 2022

create view st-grade-22 as

Select s.Sid, s.name, e.course\_id, e.grade

From Student s Join Enroll e on s.Sid =e.Sid

Where e.year = 2022;

Q9: create a view st-list-22 with Sid, name for year 2022 using the view st-grade-22.

create view st-list-22 as

Select Distinct Si

d, name

From st-grade-22;