DBMS

Hitarth Patel
150096724046
Jensen Huang
4 March 2025

3) Consider the following Entities and Relationships

Project (pno, pname, start_date, budget, status)
Department (dno, dname, HOD)
Relation between Project and Department is **Many to One**
Constraint: Primary key.
Project Status Constraints: C – completed,
P-Progressive, I-Incomplete
Create a Database in 3NF & write queries for following.

- List the project name and department details worked in projects that are 'Complete'.
- Display total budget of each department.
- Display incomplete project of each department
- Find the names of departments that have budget greater than 50000 .
- Display all project working under 'Mr.Desai'.

```
mysql> create database Practical;
Query OK, 1 row affected (0.01 sec)

mysql> use Practical;
Database changed
mysql> create table department (
    ->     dno int primary key,
    ->     dname varchar(255),
    ->     hod varchar(255)
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> create table project (
    ->     pno int primary key,
    ->     pname varchar(255),
    ->     start_date date,
    ->     budget decimal(10,2),
    ->     status char(1),
    ->     dno int,
    ->     foreign key (dno) references department(dno)
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> insert into department values
    -> (1, 'IT', 'Mr.Desai'),
    -> (2, 'HR', 'Ms.Sharma'),
    -> (3, 'Finance', 'Mr.Kumar');
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>
mysql> insert into project values
    -> (101, 'Alpha', '2024-01-01', 60000, 'C', 1),
    -> (102, 'Beta', '2024-02-01', 40000, 'P', 2),
    -> (103, 'Gamma', '2024-03-01', 70000, 'I', 1),
    -> (104, 'Delta', '2024-04-01', 80000, 'C', 3);
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

```
mysql> select pname, dname, hod from project
    -> join department on project.dno = department.dno
    -> where status = 'C';
+-------+---------+----------+
| pname | dname   | hod      |
+-------+---------+----------+
| Alpha | IT      | Mr.Desai |
| Delta | Finance | Mr.Kumar |
+-------+---------+----------+
2 rows in set (0.00 sec)

mysql> select dname, sum(budget) as total_budget from project
    -> join department on project.dno = department.dno
    -> group by dname;
+---------+--------------+
| dname   | total_budget |
+---------+--------------+
| IT      |    130000.00 |
| HR      |     40000.00 |
| Finance |     80000.00 |
+---------+--------------+
3 rows in set (0.01 sec)

mysql> select pname, dname from project
    -> join department on project.dno = department.dno
    -> where status = 'I';
+-------+-------+
| pname | dname |
+-------+-------+
| Gamma | IT    |
+-------+-------+
1 row in set (0.00 sec)

mysql> select distinct dname from project
    -> join department on project.dno = department.dno
    -> where budget > 50000;
+---------+
| dname   |
+---------+
| IT      |
| Finance |
+---------+
2 rows in set (0.00 sec)

mysql> select pname from project
    -> join department on project.dno = department.dno
    -> where hod = 'Mr.Desai';
+-------+
| pname |
+-------+
| Alpha |
| Gamma |
+-------+
2 rows in set (0.00 sec)
```

DBMS Practical List (Based on User, Indexing, Role, Trigger, and Stored Procedure)

This practical list covers key **database management concepts** related to **users, indexing, roles, triggers, and stored procedures**.

---

1. User Management in DBMS

Practical 1: Create and Manage Users

Objective: Create different database users (User1 and User2)with specific privileges.

- Create a new database user.
- Grant and revoke privileges (SELECT, INSERT, DELETE, UPDATE).
- Display user privileges.s

```
mysql> grant select, insert, update, delete on Practical.Project to 'user1'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> grant select, insert, update, delete on Practical.Department to 'user2'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

---

2. Indexing in DBMS

Practical 2: Implement Indexing on a Project Table

Objective: Improve query performance using indexing.

- Create Project table with a primary key.
- Insert multiple user records.
- Create an index on the Budget column.
- Display index.
- Remove an index and analyze the impact.
- **Composite Indexing** – Create an index on dame and HOD and analyze query performance.

```
mysql> create index idx_budget on project(budget);
Query OK, 0 rows affected, 1 warning (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 1

mysql> show index from project;
+---------+------------+------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table   | Non_unique | Key_name   | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+---------+------------+------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| project |          0 | PRIMARY    |            1 | pno         | A         |           4 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| project |          1 | dno        |            1 | dno         | A         |           3 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
| project |          1 | iBudget    |            1 | budget      | A         |           4 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
| project |          1 | idx_budget |            1 | budget      | A         |           4 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
+---------+------------+------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
4 rows in set (0.00 sec)

mysql> drop index idx_budget on project;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> create index idx_composite on project(name, hod);
ERROR 1072 (42000): Key column 'name' doesn't exist in table
mysql> show index from project;
+---------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table   | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+---------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| project |          0 | PRIMARY  |            1 | pno         | A         |           4 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| project |          1 | dno      |            1 | dno         | A         |           3 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
| project |          1 | iBudget  |            1 | budget      | A         |           4 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
+---------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
3 rows in set (0.00 sec)
```

3. Role Management in DBMS

Practical 3: Create and Assign Roles

Objective: Manage user roles and permissions efficiently.

- Create different roles (Admin, Editor, Viewer).
- Assign privileges to roles.(select,insert,update privileges)
- Assign roles to users and test access levels.

```
mysql> create role 'admin', 'editor', 'viewer';
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> set default role 'admin' to 'user1'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> grant all privileges on Practical.project to 'admin';
Query OK, 0 rows affected (0.00 sec)

mysql> grant select, insert, update on Practical.project to 'editor';
Query OK, 0 rows affected (0.01 sec)

mysql> grant select on Practical.project to 'viewer';
Query OK, 0 rows affected (0.00 sec)

mysql> grant 'admin' to 'user1'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

4. Triggers in DBMS

Create a Trigger to Prevent Invalid Salary Updates

```
mysql> DELIMITER $$
mysql> create trigger prevent_invalid_salary
    -> before update on employees
    -> for each row
    -> begin
    ->     if new.salary < 0 then
    ->         signal sqlstate '45000' set message_text = 'Invalid salary amount';
    ->     end if;
    -> end;
    -> DELIMITER ;
    -> END$$
Query OK, 0 rows affected (0.00 sec)
```

5. Stored Procedure in DBMS

Create a Stored Procedure to Insert a New Employee

```
mysql> DELIMITER $$
mysql> create procedure insert_employee(
    -> in emp_id int,in name varchar(255),in salary decimal(10,2))
    -> BEGIN
    -> insert into employees values (emp_id, name, salary);
    -> END $$
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql>
mysql> set @id=312;
Query OK, 0 rows affected (0.00 sec)

mysql> set @n ="hello";
Query OK, 0 rows affected (0.00 sec)

mysql> set @s=123.21;
Query OK, 0 rows affected (0.00 sec)

mysql> call insert_employee(@id,@n,@s);
Query OK, 1 row affected (0.00 sec)
```