Naveenya Gunasekaran

Impact training

PL/SQL

1)ADVANTAGES AND DISADVANTAGES OF PROCEDURES:

Advantages of Procedures

Performance improvement of the application.

If a procedure is being called frequently in an application in a single connection then the compiled version of the procedure is delivered.

They reduce the traffic between the database and the application

Disadvantages of Procedures

Procedures can cause a lot of memory usage

MySQL does not provide the functionality of debugging the stored procedures.

2)ADVANTAGES AND DISADVANTAGES OF FUNCTIONS:

Advantages of Functions

PL/SQL functions provide a multitude of benefits for developers working oracle database environment

Functions are designed to encapsulate logic into modular units that can be reused throughout the database.

Functions can enhance performance by minimizing network traffic.

Disadvantages of Functions

Cannot return multiple values

Memory and time overhead due to stack frame allocation and transfer of program control.

3)ADVANTAGES AND DISADVANTAGES OF VIEW

Advantages of Views

It acts like a table , so most things you'd do with a table will work with a view

It allows to keep the logic centralized , rather than repeating in code

It allow for massive performance improvements

Disadvanatages of views

If done wrong , it can result in performance issues

A view is one more moving piece that has to be maintained

4)TRIGGERS AND ITS TYPES:

It is a predefined programs that will be automatically invoked. It is stored in database and invoked repeatedly in particular scenario. Two states of triggers ENABLED and DISABLED.

CREATE TRIGGER statement creates a trigger

BEFORE and AFTER are the trigger timing points

```
create database aspire;
use aspire;
CREATE TABLE Employeeabt (
   eid INT PRIMARY KEY,
   ename VARCHAR(100),
   dob DATE,
   doj DATE,
   blood VARCHAR(5),
   address VARCHAR(255),
   designation ENUM('EEE', 'ECE', 'CIV', 'MECH'),
```

```
    location ENUM('SALEM', 'KERALA', 'KOLKATA'),
    Mnum VARCHAR(15),
    salary DECIMAL(10, 2)
);
drop table employeeabt;
INSERT INTO Employeeabt (eid, ename, dob, doj, blood, address, designation, location, Mnum, salary) VALUES
(1, 'John', '1989-02-11', '2020-01-23', 'O+', 'MCDC,Salem', 'ECE', 'Chennai', '9856745328', 60000.00),
(2, 'Naveenya', '1988-09-20', '2019-10-22', 'O+', 'kakapalayam', 'EEE', 'Hyderabad', '8756309321', 65000.00),
(3, 'Sri', '1995-06-11', '2020-07-22', 'O-', 'Trichy', 'CIVIL', 'Kochi', '7854234890', 55000.00),
(4, 'Uma', '1989-06-20', '2020-08-11', 'A+', 'Madurai', 'CSE', 'Kolkata', '6754362890', 45000.00),
(5, 'Guna', '1993-10-15', '2024-08-17', 'O+', 'Coimbatore', 'Mech', 'Hyderabad', '7893526289', 67000.00);
select * from Employeeabt;

CREATE TABLE RowCount (
    table_name VARCHAR(100) PRIMARY KEY,
    row_count INT
);

INSERT INTO RowCount (table_name, row_count) VALUES ('Employees', 0);

select * from RowCount;

DELIMITER $$
```

```
CREATE TRIGGER after_employee_insert
AFTER INSERT ON Employees
FOR EACH ROW
BEGIN
   UPDATE RowCount
   SET row_count = row_count + 1
   WHERE table_name = 'Employeeabt';
END $$

DELIMITER ;

INSERT INTO Employeeabt (eid, ename, dob, doj, blood, address, designation, location, Mnum, salary)
VALUES (6, 'Venkatesh', '1994-10-15', '2023-12-12', 'B+', 'Dindugal', 'MECH', 'Mumbai', '8512878199', 90000.00);

INSERT INTO Employeeabt (eid, ename, dob, doj, blood, address, designation, location, Mnum, salary)
VALUES (7, 'Rathinam', '1994-11-16', '2024-09-14', 'O+', 'Vellore', 'HR', 'West Bengal', '9678267813', 70000.00);

INSERT INTO Employeeabt (employee_id, employee_name, date_of_birth, date_of_join, blood_group, address, designation, location, mobile_number, salary)
VALUES (8, 'Arumugam', '1984-11-15', '2023-07-14', 'AB+', 'Bangalore', 'Testing', 'Kochi', '8726271690', 55000.00);

CREATE VIEW Kochi_Testing_Employees AS
```

```
SELECT eid, ename, dob, doj, blood, address, designation, location, Mnum, salary
FROM Employeeabt
WHERE location = 'Kochi' AND designation = 'Testing';
SELECT * FROM Kochi_Testing_Employeeabt;



-- Stored Procedure
DELIMITER $$
CREATE PROCEDURE GetKochiTestingEmployees()
BEGIN
   SELECT eid, ename, dob, doj, blood, address, designation, location, Mnum, salary
   FROM Employeeabt
   WHERE location = 'Kochi' AND designation = 'Testing';
END $$
DELIMITER ;

CALL GetKochiTestingEmployees();
-- function
DELIMITER $$
CREATE FUNCTION CheckKochiTestingEmployees()
RETURNS INT
DETERMINISTIC
BEGIN
   DECLARE emp_count INT;
   SELECT COUNT(*) INTO emp_count
   FROM Employees
```

```
    WHERE location = 'Kochi' AND designation = 'Testing';
    RETURN emp_count;
END $$
DELIMITER ;
SELECT CheckKochiTestingEmployees();
```