DBMS LAB 6

NAME : NAGAVENI L G

SRN : PES2UG21CS315

SEC : 5F

TASK1

1) In the thriving world of art commerce, you've been assigned to enhance the system for tracking art availability. The gallery management wants to automate the process of updating art availability to "No" when a payment is made for an order.

   To address this, you've to implement a trigger.
   INSERT INTO art (art_id, art_description, art_type, artist_id, availability ,gallery_id,art_tittle,art_price,likes,created_date,age) VALUES ('ART016', 'Abstract sculpture', 'sculpture', 'A001', 'Yes','G002','Ancient Ruins',850,20,'2022-02-18',1);

   INSERT INTO purchase (order_id, cust_id, art_id) VALUES ('ORD011', 'C002', 'ART016');

   INSERT INTO art_order (order_id, amount,order_description,order_time) VALUES ('ORD011', 850,'Order for sculpture','2023-09-10 10:00:00');

```
PES2UG21CS315>-- Create the trigger for updating art availability when a purchase is made
PES2UG21CS315>DELIMITER //
PES2UG21CS315>CREATE TRIGGER update_art_availability_purchase
    -> AFTER INSERT ON purchase
    -> FOR EACH ROW
    -> BEGIN
    ->     -- Update art availability to 'No' when a purchase is made
    ->     UPDATE art
    ->     SET availability = 'No'
    ->     WHERE art_id = NEW.art_id;
    -> END;
    -> //
Query OK, 0 rows affected (0.01 sec)
```

```
PES2UG21CS315>-- Create the trigger for updating art availability when an order is placed
PES2UG21CS315>DELIMITER //
PES2UG21CS315>CREATE TRIGGER update_art_availability_order
    -> AFTER INSERT ON art_order
    -> FOR EACH ROW
    -> BEGIN
    ->     -- Update art availability to 'No' when an order is placed
    ->     UPDATE art
    ->     SET availability = 'No'
    ->     WHERE art_id = (SELECT art_id FROM purchase WHERE order_id = NEW.order_id);
    -> END;
    -> //
Query OK, 0 rows affected (0.01 sec)

PES2UG21CS315>DELIMITER ;
PES2UG21CS315>
```

```
PES2UG21CS315> INSERT INTO art (art_id, art_description, art_type, artist_id, availability ,gallery_id,art_tittle,art_price,likes,created_date,age) VALUES (
'ART018', 'Abstract sculpture', 'sculpture', 'A005', 'Yes','G002','Ancient Ruins',850,20,'2022-02-18',1);
Query OK, 1 row affected (0.01 sec)
```

```
PES2UG21CS315>INSERT INTO art_order (order_id, amount,order_description,order_time) VALUES ('ORD019', 850,'Order for sculpture','2023-09-10 10:00:00');
Query OK, 1 row affected (0.01 sec)

PES2UG21CS315>INSERT INTO purchase (order_id, cust_id, art_id) VALUES ('ORD019', 'C002', 'ART018');
Query OK, 1 row affected (0.00 sec)
```

```
PES2UG21CS315>SELECT * FROM art where art_id="ART018";
+--------+-----------+------------+---------------+-----------+--------------+-----------+-------+--------------------+--------------+------+
| art_id | artist_id | gallery_id | art_tittle    | art_type  | availability | art_price | likes | art_description    | created_date | age  |
+--------+-----------+------------+---------------+-----------+--------------+-----------+-------+--------------------+--------------+------+
| ART018 | A005      | G002       | Ancient Ruins | sculpture | No           |       850 |    20 | Abstract sculpture | 2022-02-18   |    1 |
+--------+-----------+------------+---------------+-----------+--------------+-----------+-------+--------------------+--------------+------+
1 row in set (0.00 sec)
```

2) The management has embraced an inventive strategy to honor and applaud the local artists within their community. recalibrating an artist's rating whenever one of their masterpieces is exhibited. The update is computed based on the cumulative likes received by the artist across all their creations to the total number of artworks they've contributed. Could you kindly give a trigger to address this?
   INSERT INTO exhibited (ex_id, art_id) VALUES ('EX010', 'ART017');
   INSERT INTO exhibited (ex_id, art_id) VALUES ('EX010', 'ART016');

```
PES2UG21CS315>DELIMITER //
PES2UG21CS315>CREATE TRIGGER update_artist_rating
    -> AFTER INSERT ON exhibited
    -> FOR EACH ROW
    -> BEGIN
    ->     -- Calculate the updated rating based on cumulative likes and total artworks
    ->     DECLARE artist_likes INT;
    ->     DECLARE total_artworks INT;
    ->     DECLARE new_rating DECIMAL(5,2);
    ->
    ->     -- Calculate the artist's cumulative likes
    ->     SELECT SUM(likes) INTO artist_likes
    ->     FROM art
    ->     WHERE artist_id = (SELECT artist_id FROM art WHERE art_id = NEW.art_id);
    ->
    ->     -- Calculate the artist's total number of artworks
    ->     SELECT COUNT(*) INTO total_artworks
    ->     FROM art
    ->     WHERE artist_id = (SELECT artist_id FROM art WHERE art_id = NEW.art_id);
    ->
    ->     -- Avoid division by zero
    ->     IF total_artworks = 0 THEN
    ->         SET new_rating = 0;
    ->     ELSE
    ->         -- Calculate the new artist rating
    ->         SET new_rating = artist_likes / total_artworks;
    ->     END IF;
    ->
    ->     -- Update the artist's rating
    ->     UPDATE artist
    ->     SET ratings = new_rating
    ->     WHERE artist_id = (SELECT artist_id FROM art WHERE art_id = NEW.art_id);
    -> END;
    -> //
Query OK, 0 rows affected (0.04 sec)
```

```
PES2UG21CS315>INSERT INTO exhibited (ex_id, art_id)
    -> VALUES ('EX010', 'ART018');
Query OK, 1 row affected (0.01 sec)
```

```
PES2UG21CS315>select * from art where art_id="ART018";
+--------+-----------+------------+---------------+-----------+--------------+-----------+-------+-------------------+--------------+------+
| art_id | artist_id | gallery_id | art_tittle    | art_type  | availability | art_price | likes | art_description   | created_date | age  |
+--------+-----------+------------+---------------+-----------+--------------+-----------+-------+-------------------+--------------+------+
| ART018 | A005      | G002       | Ancient Ruins | sculpture | No           |       850 |    20 | Abstract sculpture | 2022-02-18   |    1 |
+--------+-----------+------------+---------------+-----------+--------------+-----------+-------+-------------------+--------------+------+
1 row in set (0.00 sec)
```

```
PES2UG21CS315>select * from artist where artist_id="A005";
+------------+--------+---------+------------+---------+-----------+-----------+----------+--------+
| artist_id | f_name | l_name  | contact_no | ratings | username | password | location | gender |
+------------+--------+---------+------------+---------+-----------+-----------+----------+--------+
| A005      | Jaya   | Lalitha | 7477877927 |      65 | NULL      | NULL      | Mysuru   | F      |
+------------+--------+---------+------------+---------+-----------+-----------+----------+--------+
1 row in set (0.00 sec)
```

3)The management has embraced an inventive strategy to honor and applaud the local artists within their community. Update the ratings of artist The update is computed based on the cumulative likes received by the artist across all their creations to the total number of artworks they've contributed. Could you give a procedure using cursor to address this?

```
PES2UG21CS315>
PES2UG21CS315>CREATE PROCEDURE UpdateArtistRatings()
    -> BEGIN
    ->     DECLARE done INT DEFAULT 0;
    ->     DECLARE artist_id VARCHAR(10);
    ->     DECLARE artist_likes INT;
    ->     DECLARE total_artworks INT;
    ->     DECLARE new_rating DECIMAL(5,2);
    ->
    ->     -- Declare a cursor to iterate through artists
    ->     DECLARE artists_cursor CURSOR FOR
    ->     SELECT artist_id
    ->     FROM artist;
    ->
    ->     -- Declare continue handler to exit the loop
    ->     DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
    ->
    ->     OPEN artists_cursor;
    ->
    ->     -- Start looping through artists
    ->     artist_loop: LOOP
    ->         FETCH artists_cursor INTO artist_id;
    ->
    ->         IF done = 1 THEN
    ->             LEAVE artist_loop;
    ->         END IF;
    ->
    ->         -- Calculate the artist's cumulative likes
    ->         SELECT SUM(likes) INTO artist_likes
    ->         FROM art
    ->         WHERE artist_id = artist_id;
    ->
    ->         -- Calculate the artist's total number of artworks
    ->         SELECT COUNT(*) INTO total_artworks
    ->         FROM art
    ->         WHERE artist_id = artist_id;
    ->
    ->         -- Avoid division by zero
    ->         IF total_artworks = 0 THEN
    ->             SET new_rating = 0;
    ->         ELSE
    ->             -- Calculate the new artist rating
    ->             SET new_rating = artist_likes / total_artworks;
    ->         END IF;
    ->
    ->         -- Update the artist's rating
    ->         UPDATE artist
    ->         SET ratings = new_rating
    ->         WHERE artist_id = artist_id;
    ->     END LOOP artist_loop;
    ->
    ->     CLOSE artists_cursor;
    -> END;
    ->
    -> //
Query OK, 0 rows affected (0.05 sec)
```

```
PES2UG21CS315>
PES2UG21CS315>INSERT INTO exhibited (ex_id, art_id)
    -> VALUES ('EX010', 'ART016');
Query OK, 1 row affected (0.00 sec)
```

```
PES2UG21CS315>select * from art where art_id="ART016";
+--------+-----------+------------+---------------+----------+--------------+-----------+-------+--------------------+--------------+------+
| art_id | artist_id | gallery_id | art_tittle    | art_type | availability | art_price | likes | art_description    | created_date | age  |
+--------+-----------+------------+---------------+----------+--------------+-----------+-------+--------------------+--------------+------+
| ART016 | A001      | G002       | Ancient Ruins | sculpture | No          |       850 |    20 | Abstract sculpture | 2022-02-18   |    1 |
+--------+-----------+------------+---------------+----------+--------------+-----------+-------+--------------------+--------------+------+
1 row in set (0.00 sec)

PES2UG21CS315>
PES2UG21CS315>SELECT * FROM artist where artist_id="A001";
+-----------+--------+--------+------------+---------+----------+----------+----------+--------+
| artist_id | f_name | l_name | contact_no | ratings | username | password | location | gender |
+-----------+--------+--------+------------+---------+----------+----------+----------+--------+
| A001      | Guru   | Dev    | 2973234833 |      60 | NULL     | NULL     | Bengaluru | M      |
+-----------+--------+--------+------------+---------+----------+----------+----------+--------+
1 row in set (0.00 sec)
```

4)You are tasked with understanding the mentorship relationships within the political organization. Starting with 'Politician A,' who holds the position of 'Senator,' list all politicians in the mentorship chain, and provide the depth of their connection to 'Politician A.' This information will help reveal the hierarchy of mentoring within the organization and identify the individuals being mentored by 'Politician A.' output should contain ID, name, position, depth.

```
PES2UG21CS315>show tables;
+------------------------+
| Tables_in_politics     |
+------------------------+
| politicalrelationships |
| politicians            |
+------------------------+
2 rows in set (0.01 sec)
```

```
PES2UG21CS315>desc politicalrelationships;
+--------------------+-------------+------+-----+---------+----------------+
| Field              | Type        | Null | Key | Default | Extra          |
+--------------------+-------------+------+-----+---------+----------------+
| RelationshipID     | int         | NO   | PRI | NULL    | auto_increment |
| ParentPoliticianID | int         | NO   | MUL | NULL    |                |
| ChildPoliticianID  | int         | NO   | MUL | NULL    |                |
| RelationshipType   | varchar(50) | NO   |     | NULL    |                |
+--------------------+-------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)

PES2UG21CS315>desc politicians;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| PoliticianID | int          | NO   | PRI | NULL    | auto_increment |
| Name         | varchar(255) | NO   |     | NULL    |                |
| Party        | varchar(50)  | YES  |     | NULL    |                |
| Age          | int          | YES  |     | NULL    |                |
| Position     | varchar(100) | YES  |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

```
PES2UG21CS315>select * from politicians;
+--------------+--------------+---------+------+-------------+
| PoliticianID | Name         | Party   | Age  | Position    |
+--------------+--------------+---------+------+-------------+
|            1 | Politician A | Party 1 |   45 | Senator     |
|            2 | Politician B | Party 2 |   38 | Congressman |
|            3 | Politician C | Party 1 |   52 | Governor    |
|            4 | Politician D | Party 2 |   43 | Mayor       |
+--------------+--------------+---------+------+-------------+
4 rows in set (0.00 sec)

PES2UG21CS315>select * from politicalrelationships;
+----------------+--------------------+-------------------+------------------+
| RelationshipID | ParentPoliticianID | ChildPoliticianID | RelationshipType |
+----------------+--------------------+-------------------+------------------+
|              1 |                  1 |                 2 | Colleague        |
|              2 |                  1 |                 3 | Mentor           |
|              3 |                  3 |                 4 | Subordinate      |
+----------------+--------------------+-------------------+------------------+
3 rows in set (0.00 sec)
```

```
PES2UG21CS315>WITH RECURSIVE MentorshipChain AS (
    ->   SELECT PoliticianID, Name, Position, 0 AS Depth
    ->   FROM Politicians
    ->   WHERE Name = 'Politician A' AND Position = 'Senator'
    ->   UNION ALL
    ->   SELECT P.PoliticianID, P.Name, P.Position, M.Depth + 1
    ->   FROM Politicians P
    ->   INNER JOIN PoliticalRelationships R ON P.PoliticianID = R.ChildPoliticianID
    ->   INNER JOIN MentorshipChain M ON R.ParentPoliticianID = M.PoliticianID
    -> )
    ->
    -> SELECT PoliticianID, Name, Position, Depth
    -> FROM MentorshipChain
    -> ORDER BY Depth;
+--------------+--------------+-------------+-------+
| PoliticianID | Name         | Position    | Depth |
+--------------+--------------+-------------+-------+
|            1 | Politician A | Senator     |     0 |
|            2 | Politician B | Congressman |     1 |
|            3 | Politician C | Governor    |     1 |
|            4 | Politician D | Mayor       |     2 |
+--------------+--------------+-------------+-------+
4 rows in set (0.01 sec)
```

Task 2:

5) Charles, a customer service representative, is tasked with assisting customers and managing their feedback. Write an SQL query to grant Charles the necessary privileges for his role without creating a new CustomerServiceRepresentative role.

```
PES2UG21CS315>CREATE TABLE feedback (
    ->     feedback_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     customer_id VARCHAR(10),
    ->     feedback_text TEXT,
    ->     timestamp TIMESTAMP,
    ->     FOREIGN KEY (customer_id) REFERENCES customer(cust_id)
    -> );
Query OK, 0 rows affected (0.05 sec)

PES2UG21CS315>
```

```
PES2UG21CS315>desc feedback;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| feedback_id   | int         | NO   | PRI | NULL    | auto_increment |
| customer_id   | varchar(10) | YES  | MUL | NULL    |                |
| feedback_text | text        | YES  |     | NULL    |                |
| timestamp     | timestamp   | YES  |     | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)
```

```
PES2UG21CS315>CREATE USER 'Charles'@'localhost' IDENTIFIED BY '12345678;
    '>
    '> ^C
PES2UG21CS315>CREATE USER 'Charles'@'localhost' IDENTIFIED BY '12345678';
Query OK, 0 rows affected (0.03 sec)

PES2UG21CS315>GRANT GRANT OPTION ON *.* TO 'root'@'localhost';
Query OK, 0 rows affected (0.00 sec)

PES2UG21CS315>GRANT SELECT ON ART_GALLERY.feedback TO 'Charles'@'localhost';
Query OK, 0 rows affected (0.00 sec)

PES2UG21CS315>GRANT INSERT ON ART_GALLERY.feedback TO 'Charles'@'localhost';
Query OK, 0 rows affected (0.00 sec)

PES2UG21CS315>GRANT UPDATE ON ART_GALLERY.feedback TO 'Charles'@'localhost';
Query OK, 0 rows affected (0.00 sec)

PES2UG21CS315>GRANT DELETE ON ART_GALLERY.feedback TO 'Charles'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

6) John, in the role of an exhibition manager, is tasked with organizing upcoming exhibitions and overseeing displayed artworks. Write an SQL query to grant John the necessary privileges associated with the ExhibitionManager role without the need to create a new role named ExhibitionManager.

```
PES2UG21CS315>
PES2UG21CS315>CREATE USER 'John'@'localhost' IDENTIFIED BY '12345678';
Query OK, 0 rows affected (0.01 sec)

PES2UG21CS315>-- Grant SELECT, INSERT, UPDATE, and DELETE privileges on exhibition-related tables
PES2UG21CS315>GRANT SELECT, INSERT, UPDATE, DELETE ON ART_GALLERY.exhibited TO 'John'@'localhost';
Query OK, 0 rows affected (0.01 sec)

PES2UG21CS315>GRANT SELECT, INSERT, UPDATE, DELETE ON ART_GALLERY.exhibition TO 'John'@'localhost';
Query OK, 0 rows affected (0.00 sec)

PES2UG21CS315>
PES2UG21CS315>-- Grant SELECT, INSERT, UPDATE, and DELETE privileges on artwork-related tables
PES2UG21CS315>GRANT SELECT, INSERT, UPDATE, DELETE ON ART_GALLERY.art TO 'John'@'localhost';
Query OK, 0 rows affected (0.00 sec)

PES2UG21CS315>GRANT SELECT, INSERT, UPDATE, DELETE ON ART_GALLERY.art_order TO 'John'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

7) Imagine a scenario in an art gallery database (art_gallery) where there are three distinct roles: admin, artist, and customer. Define their privileges as follows: Admin: This role possesses complete access rights to all tables within the database. Artist: Artists have both read and write access specifically to the 'art' table. Customer: Customers are granted read access to the 'art' table and read access to the 'gallery' table. Now, provide the SQL commands to revoke the aforementioned privileges from these roles.

CREATING USERS:

```
PES2UG21CS315>
PES2UG21CS315>CREATE USER 'Admin'@'localhost';
Query OK, 0 rows affected (0.01 sec)

PES2UG21CS315>CREATE USER 'Artist'@'localhost';
Query OK, 0 rows affected (0.00 sec)

PES2UG21CS315>CREATE USER 'Customer'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

GRANTING PREVILAGES:

```
PES2UG21CS315>-- Grant full access to the 'art_gallery' database for Admin
PES2UG21CS315>GRANT ALL PRIVILEGES ON art_gallery.* TO 'Admin'@'localhost';
Query OK, 0 rows affected (0.01 sec)

PES2UG21CS315>
PES2UG21CS315>-- Grant read and write access to the 'art' table for Artist
PES2UG21CS315>GRANT SELECT, INSERT, UPDATE, DELETE ON art_gallery.art TO 'Artist'@'localhost';
Query OK, 0 rows affected (0.00 sec)

PES2UG21CS315>
PES2UG21CS315>-- Grant read access to the 'art' and 'gallery' tables for Customer
PES2UG21CS315>GRANT SELECT ON art_gallery.art TO 'Customer'@'localhost';
Query OK, 0 rows affected (0.00 sec)

PES2UG21CS315>GRANT SELECT ON art_gallery.gallery TO 'Customer'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

REVOKING PREVILAGES:

```
PES2UG21CS315>REVOKE ALL PRIVILEGES ON art_gallery.* FROM 'Admin'@'localhost';
Query OK, 0 rows affected (0.00 sec)

PES2UG21CS315>REVOKE SELECT, INSERT, UPDATE, DELETE ON art_gallery.art FROM 'Artist'@'localhost';
Query OK, 0 rows affected (0.00 sec)

PES2UG21CS315>REVOKE SELECT ON art_gallery.art FROM 'Customer'@'localhost';
Query OK, 0 rows affected (0.00 sec)

PES2UG21CS315>REVOKE SELECT ON art_gallery.gallery FROM 'Customer'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

8) A customer identified by the customer ID 'C011' has requested specialized read access to the 'comments' table, specifically to view comments related to specific artworks. Create a user profile 'C011' and grant read access to the 'comments' table for this customer. Additionally, provide the SQL commands to revoke the previously granted read access from user C011 when necessary.

```
PES2UG21CS315>
PES2UG21CS315>CREATE USER 'C011'@'localhost' IDENTIFIED BY '12345678';
Query OK, 0 rows affected (0.01 sec)
```

```
PES2UG21CS315>-- Grant SELECT access to the 'comments' table for user 'C011'@'localhost'
PES2UG21CS315>GRANT SELECT ON art_gallery.comments TO 'C011'@'localhost';
Query OK, 0 rows affected (0.00 sec)

PES2UG21CS315>-- Revoke SELECT access to the 'comments' table for user 'C011'@'localhost'
PES2UG21CS315>REVOKE SELECT ON art_gallery.comments FROM 'C011'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

9) Establish a role named 'admin' and assign it full privileges within the database. Additionally, provide the SQL commands to drop or remove this 'admin' role when necessary.

```
PES2UG21CS315>
PES2UG21CS315>CREATE USER 'admin'@'localhost' IDENTIFIED BY '12345678';
Query OK, 0 rows affected (0.01 sec)

PES2UG21CS315>GRANT ALL PRIVILEGES ON *.* TO 'admin'@'localhost' WITH GRANT OPTION;
Query OK, 0 rows affected (0.01 sec)

PES2UG21CS315>REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'admin'@'localhost';
Query OK, 0 rows affected (0.01 sec)

PES2UG21CS315>-- Delete the 'admin' user
PES2UG21CS315>DROP USER 'admin'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

THANK YOU 😊