LAB-6 DBMS

AKHILESH. M. K

PES1UG21CS059

**TASK-1**

1)

DELIMITER //

CREATE TRIGGER update\_art\_availability

AFTER INSERT ON purchase

FOR EACH ROW

BEGIN

UPDATE art

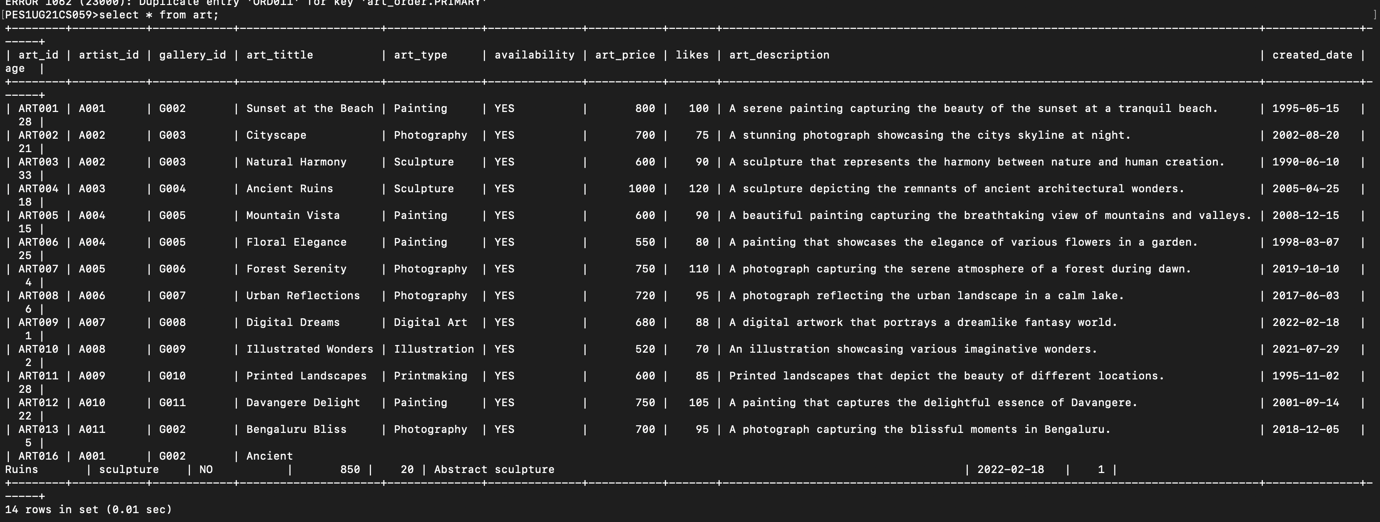
SET availability = 'NO'

WHERE art\_id = NEW.art\_id;

END;

//

DELIMITER ;



2)

DELIMITER //

CREATE TRIGGER recalibrate\_artist\_rating

AFTER INSERT ON exhibited

FOR EACH ROW

BEGIN

DECLARE artist\_id VARCHAR(10);

DECLARE total\_likes INT;

DECLARE total\_artworks INT;

SELECT artist\_id INTO artist\_id

FROM art

WHERE art\_id = NEW.art\_id;

SELECT SUM(likes) INTO total\_likes

FROM art

WHERE artist\_id = artist\_id;

SELECT COUNT(\*) INTO total\_artworks

FROM art

WHERE artist\_id = artist\_id;

IF total\_artworks > 0 THEN

UPDATE artist

SET ratings = total\_likes / total\_artworks

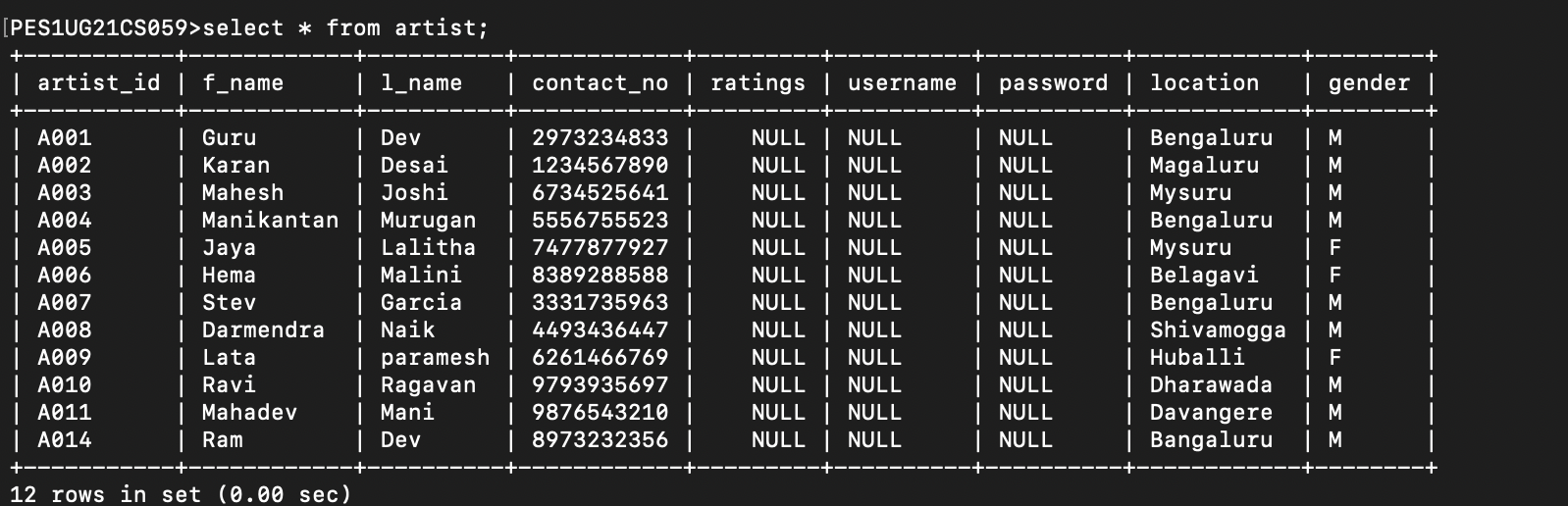
WHERE artist\_id = artist\_id;

END IF;

END;

//

DELIMITER ;



3)

DELIMITER //

PES1UG21CS059>CREATE PROCEDURE UpdateArtistRatings()

-> BEGIN

-> DECLARE done INT DEFAULT FALSE;

-> DECLARE artistIdVar CHAR(5);

-> DECLARE totalLikes INT;

-> DECLARE totalArtworks INT;

-> DECLARE artistRating DECIMAL(5, 2); -- Moved outside the loop

-> DECLARE artistCursor CURSOR FOR

-> SELECT artist\_id, SUM(likes), COUNT(\*)

-> FROM art

-> GROUP BY artist\_id;

-> DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

-> OPEN artistCursor;

-> read\_loop: LOOP

-> FETCH artistCursor INTO artistIdVar, totalLikes, totalArtworks;

-> IF done THEN

-> LEAVE read\_loop;

-> END IF;

-> IF totalArtworks > 0 THEN

-> SET artistRating = totalLikes / totalArtworks;

-> ELSE

-> SET artistRating = 0.0;

-> END IF;

-> UPDATE artist

-> SET rating = artistRating

-> WHERE artist\_id = artistIdVar;

-> END LOOP;

-> CLOSE artistCursor;

-> END;

-> //



4)

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, MC.Depth + 1

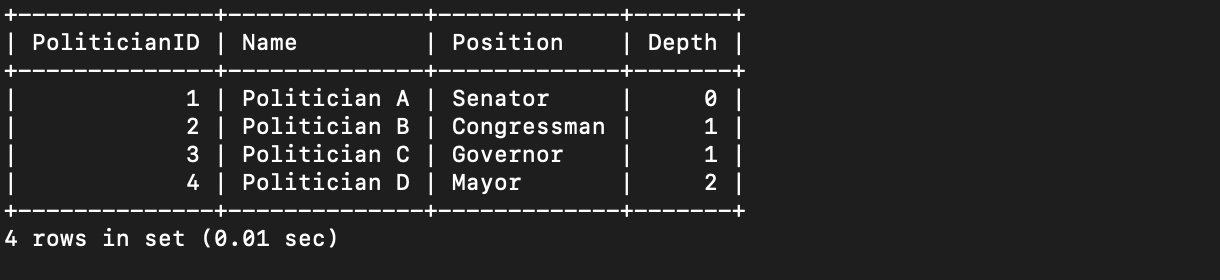
FROM Politicians AS P

INNER JOIN PoliticalRelationships AS PR ON P.PoliticianID = PR.ChildPoliticianID

INNER JOIN MentorshipChain AS MC ON PR.ParentPoliticianID = MC.PoliticianID

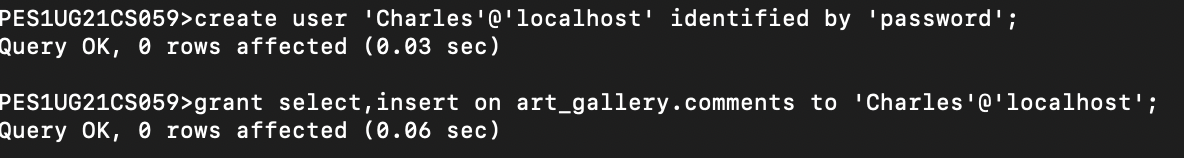
)

SELECT PoliticianID, Name, Position, Depth FROM MentorshipChain;

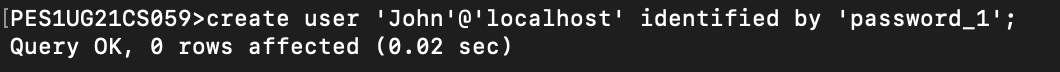


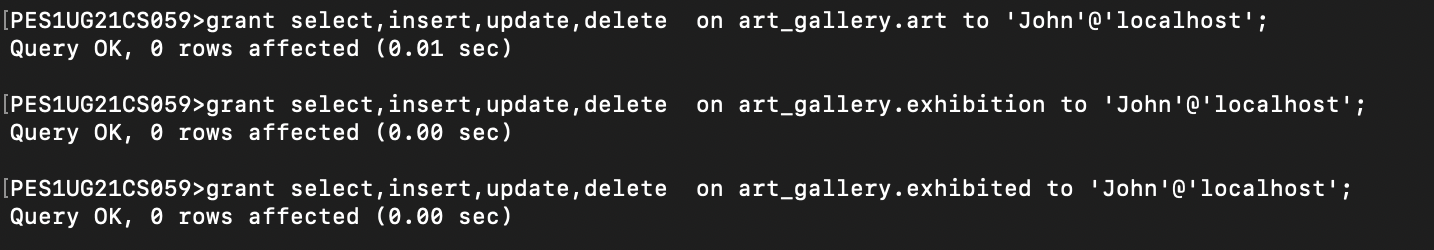
**TASK-2:**

**5)**

****

**6)**

****

****

7)

REVOKE ALL PRIVILEGES ON art\_gallery.\* FROM Admin;

REVOKE SELECT, INSERT, UPDATE, DELETE ON art FROM Artist;

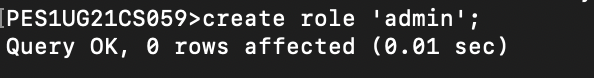
REVOKE SELECT, INSERT, UPDATE, DELETE ON art, gallery FROM Customer;

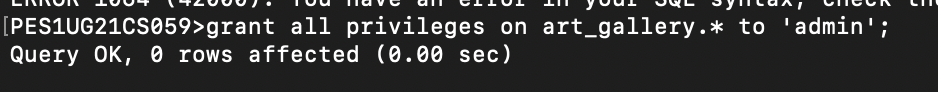
8)

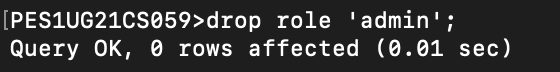
GRANT SELECT ON comments TO C011;

REVOKE SELECT ON comments FROM C011;

9)

****

****

****