ASSIGNMENT -2

Name:A.rakcha

Reg no:192311421

1. Create a database for managing properties, agents, clients, and transactions.

- Model tables for properties, agents, clients, and transactions.

- Write stored procedures for listing properties and closing deals.

- Implement triggers to update property availability and transaction status.

- Write SQL queries to analyze market trends and client preferences.

Answer:

**Listing Properties**

sql

Copy code

CREATE PROCEDURE ListProperty(

IN p\_address VARCHAR(255),

IN p\_price DECIMAL(10, 2),

IN p\_type VARCHAR(50),

IN p\_status VARCHAR(20),

IN p\_agent\_id INT,

IN p\_description TEXT

)

BEGIN

INSERT INTO Properties (address, price, type, status, agent\_id, listed\_date, description)

VALUES (p\_address, p\_price, p\_type, p\_status, p\_agent\_id, CURDATE(), p\_description);

END //

DELIMITER ;

**2. Closing a Deal**

sql

Copy code

CREATE PROCEDURE CloseDeal(

IN p\_transaction\_id INT,

IN p\_sale\_price DECIMAL(10, 2),

IN p\_status VARCHAR(20)

)

BEGIN

UPDATE Transactions

SET sale\_price = p\_sale\_price, status = p\_status

WHERE transaction\_id = p\_transaction\_id;

UPDATE Properties

SET status = 'Sold'

WHERE property\_id = (SELECT property\_id FROM Transactions WHERE transaction\_id = p\_transaction\_id);

**3. Triggers**

**3.1 Update Property Availability**

sql

Copy code

CREATE TRIGGER UpdatePropertyAvailability

AFTER UPDATE ON Transactions

FOR EACH ROW

BEGIN

IF NEW.status = 'Completed' THEN

UPDATE Properties

SET status = 'Sold'

WHERE property\_id = NEW.property\_id;

END IF;

**3.2 Track Transaction Status**

sql

Copy code

CREATE TRIGGER TrackTransactionStatus

AFTER UPDATE ON Properties

FOR EACH ROW

BEGIN

IF NEW.status = 'Sold' THEN

UPDATE Transactions

SET status = 'Completed'

WHERE property\_id = NEW.property\_id AND status != 'Completed';

END IF;

**4. SQL Queries for Analysis**

**4.1 Market Trends (Avg Price by Type)**

sql

Copy code

SELECT type, AVG(price) AS avg\_price

FROM Properties

WHERE status = 'Available'

GROUP BY type;

**4.2 Client Preferences (Preferred Property Types)**

sql

Copy code

SELECT c.first\_name, c.last\_name, COUNT(t.transaction\_id) AS number\_of\_transactions, p.type AS preferred\_type

FROM Clients c

JOIN Transactions t ON c.client\_id = t.client\_id

JOIN Properties p ON t.property\_id = p.property\_id

WHERE t.status = 'Completed'

GROUP BY c.client\_id, p.type

ORDER BY number\_of\_transactions DESC;

**4.3 Agent Performance (Total Sales by Agent)**

sql

Copy code

SELECT a.first\_name, a.last\_name, COUNT(t.transaction\_id) AS total\_sales, SUM(t.sale\_price) AS total\_revenue

FROM Agents a

JOIN Transactions t ON a.agent\_id = t.agent\_id

WHERE t.status = 'Completed'

GROUP BY a.agent\_id

ORDER BY total\_revenue DESC;

CONCLUSION:

This real estate management system helps efficiently handle properties, agents, clients, and transactions.