Real Estate Management System

Group Members

Ngugi Isaac Kinyanjui - 1049049

Jepngetich Abigael Kibet - 1049517

Mwangi Brayan Waweru - 1049399

Nicole Gesare - 1049527

Serena Rolloh - 1048831

Submission Date: 21/11/2024

Introduction

Overview - Real-Estate-Management-system
Contains the database format for a real estate management system built using my sql

Rationale - Managing real estate data is complex and requires a structured system to handle multiple entities and relationships efficiently.

Objectives:

1.To design a normalized database schema for managing real estate data.
2. To implement CRUD operations and advanced queries for data manipulation and reporting.
3. To validate the system with test data and generate meaningful reports.

System Design

ER Diagram

The ER Diagram below illustrates the relationships between entities in the system:

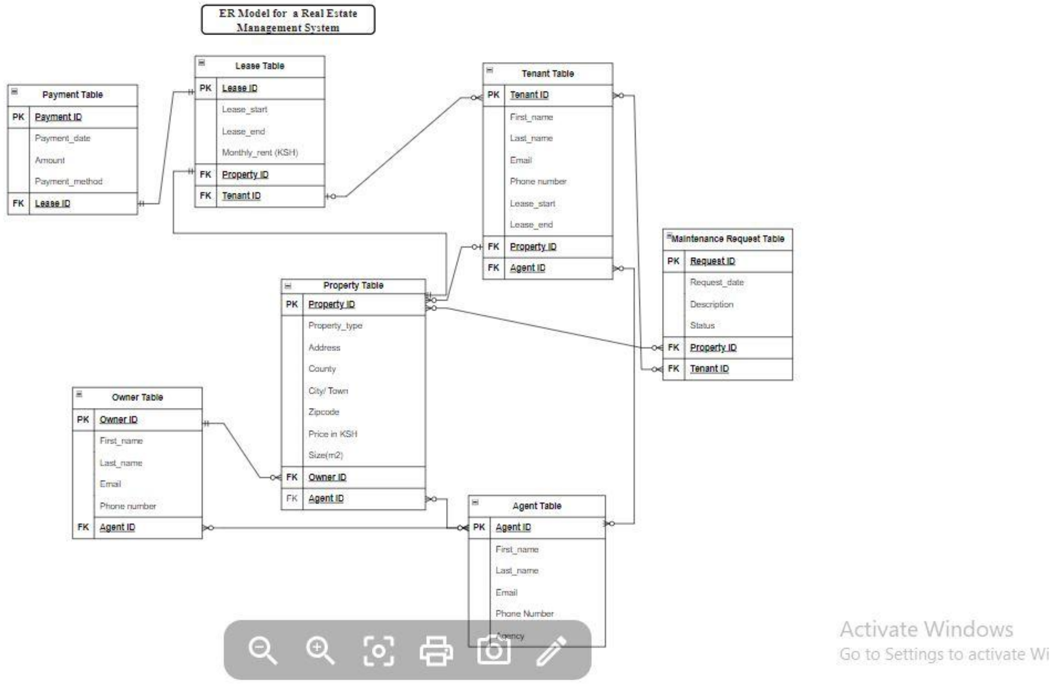

Table Structures

The database schema includes the following tables, defined using SQL scripts:Real-estate payment Script

```
DROP TABLE IF EXISTS `payment`;

CREATE TABLE `payment` (

  `Payment_id` int NOT NULL,

  `Payment_date` date DEFAULT NULL,

  `Amount_paid` decimal(10,2) DEFAULT NULL,

  `Payment_method` varchar(50) DEFAULT NULL,

  `Lease_id` int DEFAULT NULL,

  PRIMARY KEY (`Payment_id`),

  KEY `Lease_id` (`Lease_id`),
  CONSTRAINT `payment_ibfk_1` FOREIGN KEY (`Lease_id`) REFERENCES `lease` (`Lease_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Realestate property Script

```
DROP TABLE IF EXISTS `property`;
CREATE TABLE `property` (
  `Property_id` int NOT NULL,
  `Property_type` varchar(50) DEFAULT NULL,
  `Address` varchar(255) NOT NULL,
  `County` varchar(100) NOT NULL,
  `City` varchar(100) NOT NULL,
  `Price_ksh` decimal(10,2) DEFAULT NULL,
  `Owner_id` int DEFAULT NULL,
  `Agent_id` int DEFAULT NULL,
  `Zipcode` varchar(100) DEFAULT NULL,
  `Size_m2` int DEFAULT NULL,
  PRIMARY KEY (`Property_id`),
  KEY `Owner_id` (`Owner_id`),
  KEY `Agent_id` (`Agent_id`),
  CONSTRAINT `Agent_id` FOREIGN KEY (`Agent_id`) REFERENCES `agent` (`Agent_id`),
  CONSTRAINT `property_ibfk_1` FOREIGN KEY (`Owner_id`) REFERENCES `owner` (`Owner_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Realestate tenant Script

```
CREATE TABLE `tenant` (
  `Tenant_id` INT NOT NULL AUTO_INCREMENT,
  `Firstname` VARCHAR(100) NOT NULL,
  `Lastname` VARCHAR(100) NOT NULL,
  `Email` VARCHAR(100) NOT NULL,
  `Phone_number` VARCHAR(15) DEFAULT NULL,
  `Lease_start` DATE DEFAULT NULL,
  `Lease_end` DATE DEFAULT NULL,
  `Property_id` INT DEFAULT NULL,
  `Agent_id` INT DEFAULT NULL,
  PRIMARY KEY (`Tenant_id`),
  INDEX `Property_id` (`Property_id`),
  INDEX `Agent_id` (`Agent_id`),
  CONSTRAINT `tenant_ibfk_1` FOREIGN KEY (`Property_id`) REFERENCES
`property` (`Property_id`),
  CONSTRAINT `tenant_ibfk_2` FOREIGN KEY (`Agent_id`) REFERENCES `agent`
(`Agent_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Realestate agent Script

```
-- Table structure for table `agent`

DROP TABLE IF EXISTS `agent`;

/*!40101 SET @saved_cs_client     = @@character_set_client */;

/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `agent` (

  `Agent_id` int NOT NULL AUTO_INCREMENT,

  `Firstname` varchar(100) NOT NULL,

  `Lastname` varchar(100) NOT NULL,

  `Email` varchar(100) NOT NULL,

  `Phone` varchar(15) DEFAULT NULL,

  `Agency` varchar(100) DEFAULT NULL,

  PRIMARY KEY (`Agent_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Real-estate lease Script

```
DROP TABLE IF EXISTS `lease`;
CREATE TABLE `lease` (
  `Lease_id` int NOT NULL,
  `Lease_start` date DEFAULT NULL,
  `Lease_end` date DEFAULT NULL,
  `Monthly_rent` decimal(10,2) DEFAULT NULL,
  `Property_id` int DEFAULT NULL,
  `Tenant_id` int DEFAULT NULL,
  PRIMARY KEY (`Lease_id`),
  KEY `Property_id` (`Property_id`),
  KEY `Tenant_id` (`Tenant_id`),
  CONSTRAINT `lease_ibfk_1` FOREIGN KEY (`Property_id`) REFERENCES `property` (`Property_id`),
  CONSTRAINT `lease_ibfk_2` FOREIGN KEY (`Tenant_id`) REFERENCES `tenant` (`Tenant_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Realestate maintenance request Script

```
DROP TABLE IF EXISTS `maintenance_request`;

CREATE TABLE `maintenance_request` (

  `Request_id` int NOT NULL,

  `Request_date` date DEFAULT NULL,

  `Description` varchar(255) NOT NULL,

  `Status` varchar(100) DEFAULT NULL,

  `Property_id` int DEFAULT NULL,

  `Tenant_id` int DEFAULT NULL,

  PRIMARY KEY (`Request_id`),

  KEY `Property_id` (`Property_id`),

  KEY `Tenant_id` (`Tenant_id`),

  CONSTRAINT `maintenance_request_ibfk_1` FOREIGN KEY (`Property_id`) REFERENCES `property` (`Property_id`),

  CONSTRAINT `maintenance_request_ibfk_2` FOREIGN KEY (`Tenant_id`) REFERENCES `tenant` (`Tenant_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Real-estate owner Script

```
DROP TABLE IF EXISTS `owner`;

CREATE TABLE `owner` (

  `Owner_id` int NOT NULL,

  `Firstname` varchar(100) NOT NULL,

  `Lastname` varchar(100) NOT NULL,

  `Email` varchar(100) NOT NULL,

  `Phone_number` varchar(15) DEFAULT NULL,
  `Agent_id` int DEFAULT NULL,
  PRIMARY KEY (`Owner_id`),
  KEY `fk_agent` (`Agent_id`),
  CONSTRAINT `fk_agent` FOREIGN KEY (`Agent_id`) REFERENCES `agent` (`Agent_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Implementation

CRUD Operations

Examples of Create, Read, Update, and Delete operations:

Create Operations

-- To add data to the agents table

```
INSERT INTO agent (Agent_id, Firstname, Lastname, Email, Phone, Agency) VALUES

(1, 'David', 'Kamau', 'David.Kamau@gmail.com', '0712345678', 'Kikuyu Realty Ltd'),

(2, 'John', 'Mwangi', 'John.Mwangi@gmail.com', '0723456789', 'Mwangi Properties'),

(3, 'Grace', 'Njeri', 'Grace.Njeri@gmail.com', '0734567890', 'Grace Estates'),

(4, 'James', 'Odinga', 'James.Odinga@gmail.com', '0745678901', 'Odinga Real Estate'),

(5, 'Rose', 'Atieno', 'Rose.Atiens@gmail.com', '0756789012', 'Luo Heritage Realty'),

(6, 'Peter', 'Ngugi', 'Peter.Ngugi@gmail.com', '0767890123', 'Ngugi & Sons Realty'),

(7, 'Mary', 'Ochieng', 'Mary.Ochieng@gmail.com', '0778901234', 'Ochieng Premier Realty'),

(8, 'Charles', 'Kiptoo', 'Charles.Kiptoo@gmail.com', '0789012345', 'Kalenjin Investments'),

(9, 'Faith', 'Salma', 'Faith.Salma@gmail.com', '0790123456', 'Coastline Properties'),

(10, 'Charity', 'Nabwire', 'Charity.Nabwire@gmail.com', '0701234567', 'Bunyasi Estates');
```

-- To add data to the maintenance requests table

```
INSERT INTO maintenance_request (Request_id, Request_date, Description, Status, Property_id, Tenant_id) VALUES

(1, '2024-01-15', 'Leaking faucet in the bathroom', 'Pending', 101, 1),

(2, '2024-02-20', 'Broken window in the living room', 'Completed', 102, 2),
```

```sql
(3, '2024-03-10', 'Clogged sink in the kitchen', 'Pending', 103, 3),
(4, '2024-04-01', 'Air conditioning not working', 'In Progress', 104, 4),
(5, '2024-05-05', 'Electricity outage in the building', 'Pending', 105, 5),
(6, '2024-06-10', 'Pest infestation in the kitchen', 'Completed', 106, 6),
(7, '2024-07-15', 'Broken door lock', 'In Progress', 107, 7),
(8, '2024-08-01', 'Water heater malfunctioning', 'Completed', 108, 8),
(9, '2024-09-10', 'Faulty plumbing in the bathroom', 'Pending', 109, 9),
(10, '2024-10-05', 'Damaged floor tiles', 'In Progress', 110, 10);

-- To add data to the lease table
INSERT INTO lease (Lease_id, Lease_start, Lease_end, Monthly_rent, Property_id, Tenant_id) VALUES
(1, '2024-01-01', '2025-01-01', 50000.00, 101, 1),
(2, '2024-02-15', '2025-02-15', 70000.00, 102, 2),
(3, '2024-03-01', '2025-03-01', 45000.00, 103, 3),
(4, '2024-04-01', '2025-04-01', 60000.00, 104, 4),
(5, '2024-05-10', '2025-05-10', 85000.00, 105, 5),
(6, '2024-06-15', '2025-06-15', 30000.00, 106, 6),
(7, '2024-07-01', '2025-07-01', 45000.00, 107, 7),
(8, '2024-08-20', '2025-08-20', 75000.00, 108, 8),
(9, '2024-09-05', '2025-09-05', 120000.00, 109, 9),
(10, '2024-10-01', '2025-10-01', 65000.00, 110, 10);

-- To add data to the owners table
```

```
INSERT INTO owner (Owner_id, Firstname, Lastname, Email, Phone_number,
Agent_id) VALUES

(1, 'James', 'Karanja', 'james.karanja@gmail.com', '0712345678', 1),

(2, 'Wangari', 'Kamau', 'wangari.kamau@gmail.com', '0723456789', 2),

(3, 'David', 'Ochieng', 'david.ochieng@gmail.com', '0734567890', 3),

(4, 'Mary', 'Mwaniki', 'mary.mwaniki@gmail.com', '0745678901', 4),

(5, 'Grace', 'Atieno', 'grace.atieno@gmail.com', '0756789012', 5),

(6, 'Joseph', 'Mwangi', 'joseph.mwangi@gmail.com', '0767890123', 6),

(7, 'Peter', 'Njiru', 'peter.njiru@gmail.com', '0778901234', 7),

(8, 'Elizabeth', 'Mutiso', 'elizabeth.mutiso@gmail.com', '0789012345', 8),

(9, 'Njeri', 'Gikonyo', 'njeri.gikonyo@gmail.com', '0790123456', 9),

(10, 'John', 'Omondi', 'john.omondi@gmail.com', '0701234567', 10);
```

-- To add data to the property table

```
INSERT INTO property (Property_id, Property_type, Address, County, City, Price_ksh,
Owner_id, Agent_id, Zipcode, Size_m2) VALUES

(101, 'Apartment', '123 Riverside Drive', 'Nairobi', 'Nairobi', 50000.00, 1, 1, '00100', 120),

(102, 'House', '456 Mountain Road', 'Nakuru', 'Nakuru', 70000.00, 2, 2, '20100', 250),

(103, 'Apartment', '789 Lakeside Avenue', 'Kisumu', 'Kisumu', 45000.00, 3, 3, '40100',
100),

(104, 'Townhouse', '101 Coastal Lane', 'Mombasa', 'Mombasa', 60000.00, 4, 4, '80100',
200),

(105, 'Villa', '202 Beach Road', 'Kilifi', 'Kilifi', 85000.00, 5, 5, '90200', 350),

(106, 'Studio', '303 High Street', 'Meru', 'Nanyuki', 30000.00, 6, 6, '60000', 50),

(107, 'Condo', '404 Parkview Crescent', 'Machakos', 'Kangundo', 45000.00, 7, 7, '90100',
80),
(108, 'Apartment', '505 Downtown Street', 'Nairobi', 'Nairobi', 75000.00, 8, 8, '11000', 150),

(109, 'Mansion', '606 Forest View', 'Embu', 'Kiritiri', 120000.00, 9, 9, '70000', 500),

(110, 'Bungalow', '707 Greenfield Road', 'Nyeri', 'Nyeri', 65000.00, 10, 10, '20000', 200);
```

-- To add data to the tenants table

```
INSERT INTO tenant (Tenant_id, Firstname, Lastname, Email, Phone_number, Lease_start, Lease_end, Property_id, Agent_id) VALUES

(1, 'Peter', 'Njiru', 'peter.njiru@gmail.com', '0712345678', '2024-01-01', '2025-01-01', 101, 1),

(2, 'Susan', 'Karanja', 'susan.karanja@gmail.com', '0723456789', '2024-02-15', '2025-02-15', 102, 2),

(3, 'John', 'Ochieng', 'john.ochieng@gmail.com', '0734567890', '2024-03-01', '2025-03-01', 103, 3),

(4, 'Mary', 'Omondi', 'mary.omondi@gmail.com', '0745678901', '2024-04-01', '2025-04-01', 104, 4),

(5, 'Michael', 'Achieng', 'michael.achieng@gmail.com', '0756789012', '2024-05-10', '2025-05-10', 105, 5),

(6, 'Elizabeth', 'Mutiso', 'elizabeth.mutiso@gmail.com', '0767890123', '2024-06-15', '2025-06-15', 106, 6),

(7, 'Njeri', 'Gikonyo', 'njeri.gikonyo@gmail.com', '0778901234', '2024-07-01', '2025-07-01', 107, 7),

(8, 'Robert', 'Kipchirchir', 'robert.kipchirchir@gmail.com', '0789012345', '2024-08-20', '2025-08-20', 108, 8),

(9, 'Wangari', 'Nabwire', 'wangari.nabwire@gmail.com', '0790123456', '2024-09-05', '2025-09-05', 109, 9),

(10, 'James', 'Wekesa', 'james.wekesa@gmail.com', '0701234567', '2024-10-01', '2025-10-01', 110, 10);
```

-- To add data to the payments table

```
INSERT INTO payment (Payment_id, Payment_date, Amount_paid, Payment_method, Lease_id) VALUES

(1, '2024-01-10', 50000.00, 'Bank Transfer', 1),

(2, '2024-02-18', 70000.00, 'Mobile Payment', 2),

(3, '2024-03-05', 45000.00, 'Cash', 3),

(4, '2024-04-10', 60000.00, 'Cheque', 4),

(5, '2024-05-15', 85000.00, 'Bank Transfer', 5),

(6, '2024-06-20', 30000.00, 'Mobile Payment', 6),

(7, '2024-07-05', 45000.00, 'Cash', 7),

(8, '2024-08-15', 75000.00, 'Cheque', 8),

(9, '2024-09-12', 120000.00, 'Bank Transfer', 9),

(10, '2024-10-01', 65000.00, 'Mobile Payment', 10);
```

Read Operations

--- To select all properties that belong to a specific owner

```
SELECT p.Property_id, p.Property_type, p.Address, p.County, p.City, p.Price_ksh, p.Size_m2

FROM property p

JOIN owner o ON p.Owner_id = o.Owner_id

WHERE o.Owner_id = 3;
```

-- To fetch all the paymets that have been made towards a specific lease

```
SELECT p.Payment_id, p.Payment_date, p.Amount_paid, p.Payment_method

FROM payment p
```

```
JOIN lease l ON p.Lease_id = l.Lease_id

WHERE l.Lease_id = 6;
```

```
-- To get all the maintentance requests for a particular property

SELECT mr.Request_id, mr.Request_date, mr.Description, mr.Status

FROM maintenance_request mr

JOIN property p ON mr.Property_id = p.Property_id

WHERE p.Property_id = 107;
```

```
-- To show the agents and the tenants each caters to

SELECT a.Agent_id, a.Firstname AS Agent_Firstname, a.Lastname AS Agent_Lastname,

    t.Tenant_id, t.Firstname AS Tenant_Firstname, t.Lastname AS Tenant_Lastname,
t.Email AS Tenant_Email

FROM agent a

JOIN property p ON a.Agent_id = p.Agent_id

JOIN lease l ON p.Property_id = l.Property_id

JOIN tenant t ON l.Tenant_id = t.Tenant_id;
```

Update Operations

```
-- To update a tenants information

UPDATE `tenant` SET `Phone_number` = '0723456789' WHERE `Tenant_id` = 1;
```

```
-- To update the status of a maintenance request

UPDATE maintenance_request

SET Status = 'Completed'

WHERE Request_id = 3;
```

-- To update the monthly rent on a lease

```
UPDATE lease

SET Monthly_rent = 70000.00

WHERE Lease_id = 4;
```

-- To update a property if an agent is reassigned

```
UPDATE property

SET Agent_id = 2

WHERE Property_id = 101;
```

Delete Operations

```
DELETE FROM `tenant` WHERE `Tenant_id` = 1;
```

Advanced SQL Queries

Examples of advanced queries include joins, subqueries, and aggregations:

Joins Query Examples

-- To get the details of a tenant, the property they're renting and the amount they paid for the lease

```
SELECT tenant.Firstname, tenant.Lastname, property.Address, payment.Amount_paid
FROM tenant
JOIN lease ON tenant.Tenant_id = lease.Tenant_id
JOIN payment ON lease.Lease_id = payment.Lease_id
JOIN property ON lease.Property_id = property.Property_id
WHERE tenant.Tenant_id = 1;
```

-- To fetch all properties, their owners and the agents that manage those properties using INNER JOIN

```
SELECT p.Property_id, p.Property_type, p.Address, p.City, o.Owner_id, o.Firstname AS Owner_Firstname,
       o.Lastname AS Owner_Lastname, a.Agent_id, a.Firstname AS Agent_Firstname, a.Lastname AS Agent_Lastname
FROM property p
INNER JOIN owner o ON p.Owner_id = o.Owner_id
INNER JOIN agent a ON p.Agent_id = a.Agent_id;
```

-- To fetch all properties, their owners and the agents that manage those properties using OUTER JOIN

```
SELECT p.Property_id, p.Property_type, p.Address, p.City, o.Owner_id, o.Firstname AS Owner_Firstname,
       o.Lastname AS Owner_Lastname, a.Agent_id, a.Firstname AS Agent_Firstname, a.Lastname AS Agent_Lastname
```

```
FROM property p

LEFT JOIN owner o ON p.Owner_id = o.Owner_id

LEFT JOIN agent a ON p.Agent_id = a.Agent_id;
```

-- To fetch data for tenants, their leases and respective payments

```
SELECT l.Lease_id, t.Tenant_id, t.Firstname AS Tenant_Firstname, t.Lastname AS
Tenant_Lastname,

       l.Lease_start, l.Lease_end, p.Payment_id, p.Payment_date, p.Amount_paid

FROM lease l

JOIN tenant t ON l.Tenant_id = t.Tenant_id

JOIN payment p ON l.Lease_id = p.Lease_id;
```

Subqueries Query Examples

```
SELECT Firstname, Lastname FROM tenant WHERE Tenant_id IN (SELECT
Tenant_id FROM lease WHERE Property_id = 1);
```

Aggregations Query Examples

-- To calculate the total number of leases each agent manages

```
SELECT Agent_id, COUNT(*) AS Total_Leases FROM lease GROUP BY Agent_id;
```

-- To show the total revenue generated by an agent

```
SELECT a.Agent_id, a.Firstname AS Agent_Firstname, a.Lastname AS Agent_Lastname,

 SUM(p.Amount_paid) AS Total_Revenue
```

```sql
FROM agent a

JOIN property p ON a.Agent_id = p.Agent_id

JOIN lease l ON p.Property_id = l.Property_id

JOIN payment p ON l.Lease_id = p.Lease_id

GROUP BY a.Agent_id;
```

-- To show the average monthly rent for the properties in a specific area

```sql
SELECT AVG(p.Price_ksh) AS Avg_Monthly_Rent

FROM property p

WHERE p.City = 'Nairobi';
```

Testing and Validation

Testing involved running SQL scripts to verify CRUD operations and advanced queries. Sample data was inserted using the following script:

```
INSERT INTO agent (Agent_id, Firstname, Lastname, Email, Phone, Agency) VALUES

(1, 'David', 'Kamau', 'David.Kamau@gmail.com', '0712345678', 'Kikuyu Realty Ltd'),

(2, 'John', 'Mwangi', 'John.Mwangi@gmail.com', '0723456789', 'Mwangi Properties'),

(3, 'Grace', 'Njeri', 'Grace.Njeri@gmail.com', '0734567890', 'Grace Estates'),

(4, 'James', 'Odinga', 'James.Odinga@gmail.com', '0745678901', 'Odinga Real Estate'),

(5, 'Rose', 'Atieno', 'Rose.Atiens@gmail.com', '0756789012', 'Luo Heritage Realty'),
```

Reports were generated using the following script:

```
-- To generate a report on overdue payments
SELECT p.Payment_id, p.Payment_date, p.Amount_paid, l.Lease_id, t.Tenant_id, t.Firstname AS Tenant_Firstname,

    t.Lastname AS Tenant_Lastname

FROM payment p

JOIN lease l ON p.Lease_id = l.Lease_id

JOIN tenant t ON l.Tenant_id = t.Tenant_id

WHERE p.Payment_date < CURDATE() AND p.Amount_paid IS NULL;
```

```
-- To generate a report on active leases in a particular city

SELECT p.City, COUNT(l.Lease_id) AS Active_Leases

FROM lease l
```

```
JOIN property p ON l.Property_id = p.Property_id

WHERE l.Lease_end > CURDATE()

GROUP BY p.City;


-- To generate a report on tenants with overdue maintenance requests

SELECT t.Tenant_id, t.Firstname AS Tenant_Firstname, t.Lastname AS Tenant_Lastname,

    m.Request_id, m.Description, m.Request_date

FROM tenant t

JOIN maintenance_request m ON t.Tenant_id = m.Tenant_id

WHERE m.Request_date < CURDATE() AND m.Status != 'Completed';


-- To generate a report of all the properties and the total revenue from payments

SELECT property.Address, SUM(payment.Amount_paid) AS Total_Revenue

FROM property

JOIN lease ON property.Property_id = lease.Property_id

JOIN payment ON lease.Lease_id = payment.Lease_id

GROUP BY property.Address;
```

Conclusion and Recommendations

Conclusion

The Real Estate Management System meets its objectives by providing a structured and efficient database solution. It ensures data integrity, supports advanced queries, and simplifies reporting.

Recommendations

Future improvements could include:

1. 1. Developing a user-friendly interface (web or mobile) for better accessibility.
2. 2. Implementing advanced analytics for better insights into property trends.

References

1. MySQL Documentation: https://dev.mysql.com/doc/

2. ER Diagram Design Tools: Lucidchart, Draw.io

3. SQL Tutorials: TutorialsPoint SQL, W3Schools SQL

Appendices

The following SQL scripts are included in the project:

Aggregations Script

```sql
-- To calculate the total number of leases each agent manages

SELECT Agent_id, COUNT(*) AS Total_Leases FROM lease GROUP BY Agent_id;


-- To show the total revenue generated by an agent

SELECT a.Agent_id, a.Firstname AS Agent_Firstname, a.Lastname AS Agent_Lastname,

    SUM(p.Amount_paid) AS Total_Revenue

FROM agent a

JOIN property p ON a.Agent_id = p.Agent_id

JOIN lease l ON p.Property_id = l.Property_id

JOIN payment p ON l.Lease_id = p.Lease_id

GROUP BY a.Agent_id;


-- To show the average monthly rent for the properties in a specific area

SELECT AVG(p.Price_ksh) AS Avg_Monthly_Rent

FROM property p

WHERE p.City = 'Nairobi';
```

Joins Script

-- To get the details of a tenant, the property they're renting and the amount they paid for the lease

```
SELECT tenant.Firstname, tenant.Lastname, property.Address, payment.Amount_paid

FROM tenant

JOIN lease ON tenant.Tenant_id = lease.Tenant_id

JOIN payment ON lease.Lease_id = payment.Lease_id

JOIN property ON lease.Property_id = property.Property_id

WHERE tenant.Tenant_id = 1;
```

-- To fetch all properties, their owners and the agents that manage those properties using INNER JOIN

```
SELECT p.Property_id, p.Property_type, p.Address, p.City, o.Owner_id, o.Firstname AS Owner_Firstname,

       o.Lastname AS Owner_Lastname, a.Agent_id, a.Firstname AS Agent_Firstname, a.Lastname AS Agent_Lastname

FROM property p

INNER JOIN owner o ON p.Owner_id = o.Owner_id

INNER JOIN agent a ON p.Agent_id = a.Agent_id;
```

-- To fetch all properties, their owners and the agents that manage those properties using OUTER JOIN

```
SELECT p.Property_id, p.Property_type, p.Address, p.City, o.Owner_id, o.Firstname AS Owner_Firstname,
    o.Lastname AS Owner_Lastname, a.Agent_id, a.Firstname AS Agent_Firstname, a.Lastname AS Agent_Lastname
FROM property p
LEFT JOIN owner o ON p.Owner_id = o.Owner_id
LEFT JOIN agent a ON p.Agent_id = a.Agent_id;
```

Subqueries Script

```
SELECT Firstname, Lastname FROM tenant WHERE Tenant_id IN (SELECT Tenant_id FROM lease WHERE Property_id = 1);
```

Create Script

Realestate payment Script

```
DROP TABLE IF EXISTS `payment`;
CREATE TABLE `payment` (
  `Payment_id` int NOT NULL,
  `Payment_date` date DEFAULT NULL,
  `Amount_paid` decimal(10,2) DEFAULT NULL,
  `Payment_method` varchar(50) DEFAULT NULL,
  `Lease_id` int DEFAULT NULL,
  PRIMARY KEY (`Payment_id`),
  KEY `Lease_id` (`Lease_id`),
  CONSTRAINT `payment_ibfk_1` FOREIGN KEY (`Lease_id`) REFERENCES `lease` (`Lease_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Realestate property Script

```
DROP TABLE IF EXISTS `property`;
CREATE TABLE `property` (
  `Property_id` int NOT NULL,
  `Property_type` varchar(50) DEFAULT NULL,
  `Address` varchar(255) NOT NULL,
  `County` varchar(100) NOT NULL,
  `City` varchar(100) NOT NULL,
  `Price_ksh` decimal(10,2) DEFAULT NULL,
  `Owner_id` int DEFAULT NULL,
  `Agent_id` int DEFAULT NULL,
  `Zipcode` varchar(100) DEFAULT NULL,
  `Size_m2` int DEFAULT NULL,
  PRIMARY KEY (`Property_id`),
  KEY `Owner_id` (`Owner_id`),
  KEY `Agent_id` (`Agent_id`),
  CONSTRAINT `Agent_id` FOREIGN KEY (`Agent_id`) REFERENCES `agent`
(`Agent_id`),
  CONSTRAINT `property_ibfk_1` FOREIGN KEY (`Owner_id`) REFERENCES
`owner` (`Owner_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Delete Script

```
DELETE FROM `tenant` WHERE `Tenant_id` = 1;
```

Read Script

-- To select all properties that belong to a specific owner

```
SELECT p.Property_id, p.Property_type, p.Address, p.County, p.City, p.Price_ksh,
p.Size_m2
```

```
FROM property p

JOIN owner o ON p.Owner_id = o.Owner_id

WHERE o.Owner_id = 3;
```

-- To fetch all the paymets that have been made towards a specific lease

```
SELECT p.Payment_id, p.Payment_date, p.Amount_paid, p.Payment_method

FROM payment p

JOIN lease l ON p.Lease_id = l.Lease_id

WHERE l.Lease_id = 6;
```

-- To get all the maintentance requests for a particular property

```
SELECT mr.Request_id, mr.Request_date, mr.Description, mr.Status

FROM maintenance_request mr

JOIN property p ON mr.Property_id = p.Property_id

WHERE p.Property_id = 107;
```

```
-- To show the agents and the tenants each caters to

SELECT a.Agent_id, a.Firstname AS Agent_Firstname, a.Lastname AS Agent_Lastname,

    t.Tenant_id, t.Firstname AS Tenant_Firstname, t.Lastname AS Tenant_Lastname, t.Email AS Tenant_Email

FROM agent a

JOIN property p ON a.Agent_id = p.Agent_id

JOIN lease l ON p.Property_id = l.Property_id

JOIN tenant t ON l.Tenant_id = t.Tenant_id;
```

Update Script

```
-- To update a tenants information

UPDATE `tenant` SET `Phone_number` = '0723456789' WHERE `Tenant_id` = 1;
```

```
-- To update the status of a maintenance request

UPDATE maintenance_request

SET Status = 'Completed'

WHERE Request_id = 3;
```

```
-- To update the monthly rent on a lease

UPDATE lease

SET Monthly_rent = 70000.00

WHERE Lease_id = 4;
```

```sql
-- To update a property if an agent is reassigned
UPDATE property
SET Agent_id = 2
WHERE Property_id = 101;
```

Data Script

```
INSERT INTO agent (Agent_id, Firstname, Lastname, Email, Phone, Agency) VALUES

(1, 'David', 'Kamau', 'David.Kamau@gmail.com', '0712345678', 'Kikuyu Realty Ltd'),

(2, 'John', 'Mwangi', 'John.Mwangi@gmail.com', '0723456789', 'Mwangi Properties'),

(3, 'Grace', 'Njeri', 'Grace.Njeri@gmail.com', '0734567890', 'Grace Estates'),

(4, 'James', 'Odinga', 'James.Odinga@gmail.com', '0745678901', 'Odinga Real Estate'),

(5, 'Rose', 'Atieno', 'Rose.Atiens@gmail.com', '0756789012', 'Luo Heritage Realty'),

(6, 'Peter', 'Ngugi', 'Peter.Ngugi@gmail.com', '0767890123', 'Ngugi & Sons Realty'),

(7, 'Mary', 'Ochieng', 'Mary.Ochieng@gmail.com', '0778901234', 'Ochieng Premier Realty'),

(8, 'Charles', 'Kiptoo', 'Charles.Kiptoo@gmail.com', '0789012345', 'Kalenjin Investments'),

(9, 'Faith', 'Salma', 'Faith.Salma@gmail.com', '0790123456', 'Coastline Properties'),

(10, 'Charity', 'Nabwire', 'Charity.Nabwire@gmail.com', '0701234567', 'Bunyasi Estates');
```

```
INSERT INTO maintenance_request (Request_id, Request_date, Description, Status, Property_id, Tenant_id) VALUES

(1, '2024-01-15', 'Leaking faucet in the bathroom', 'Pending', 101, 1),

(2, '2024-02-20', 'Broken window in the living room', 'Completed', 102, 2),

(3, '2024-03-10', 'Clogged sink in the kitchen', 'Pending', 103, 3),

(4, '2024-04-01', 'Air conditioning not working', 'In Progress', 104, 4),

(5, '2024-05-05', 'Electricity outage in the building', 'Pending', 105, 5),

(6, '2024-06-10', 'Pest infestation in the kitchen', 'Completed', 106, 6),

(7, '2024-07-15', 'Broken door lock', 'In Progress', 107, 7),

(8, '2024-08-01', 'Water heater malfunctioning', 'Completed', 108, 8),

(9, '2024-09-10', 'Faulty plumbing in the bathroom', 'Pending', 109, 9),
(10, '2024-10-05', 'Damaged floor tiles', 'In Progress', 110, 10);
```

Reports Script
-- To generate a report on overdue payments

```
SELECT p.Payment_id, p.Payment_date, p.Amount_paid, l.Lease_id, t.Tenant_id,
t.Firstname AS Tenant_Firstname,

       t.Lastname AS Tenant_Lastname

FROM payment p

JOIN lease l ON p.Lease_id = l.Lease_id

JOIN tenant t ON l.Tenant_id = t.Tenant_id

WHERE p.Payment_date < CURDATE() AND p.Amount_paid IS NULL;
```

-- To generate a report on active leases in a particular city

```
SELECT p.City, COUNT(l.Lease_id) AS Active_Leases

FROM lease l

JOIN property p ON l.Property_id = p.Property_id

WHERE l.Lease_end > CURDATE()

GROUP BY p.City;
```

-- To generate a report on tenants with overdue maintenance requests

```
SELECT t.Tenant_id, t.Firstname AS Tenant_Firstname, t.Lastname AS
Tenant_Lastname,

       m.Request_id, m.Description, m.Request_date
```

```
FROM tenant t
JOIN maintenance_request m ON t.Tenant_id = m.Tenant_id
WHERE m.Request_date < CURDATE() AND m.Status != 'Completed';
```