

Name: Neha Vithob Nayak

USN: 2SD22CS129

Branch: CSE (A)

Course: Database Management System

Scenario:

It is required to design the database for a Property Management System that allows a real estate company to efficiently manage properties, tenants, lease contracts, and rental payments. Additionally, the system must track maintenance requests made by tenants for specific properties they are renting. Currently, the company manages information manually using spreadsheets, which leads to frequent errors, duplication of data, and inefficiency in tracking rental contracts and payments. This results in loss of revenue, poor customer service, and compliance issues. The company seeks to implement a digital system that will:

- Track properties and associate them with owners.
- Manage tenants and their corresponding rental contracts.
- Record payments made by tenants under each contract.
- Improve data consistency, reduce manual errors, and streamline business operations.
- Enhance data management with no redundancies.
- Easier tracking of contract renewals, expirations, and payments.
- Better financial management with payment histories tied to tenants and properties.

Q2> Schema:

Property			
Pid#	OAdhar	PType	PRegDate

Owner		
Oid#	OName	OAdhar

Tenant			
Tid#	TName	TAdhar	Contract

Contract				
Tid#	Pid#	Start_Date	End_Date	Amount

Payment			
Tid#	Pid#	PaymentDate	Amount

Payment			
Tid#	Pid#	PaymentDate	Amount

Maintenance_Request			
Tid#	Pid#	PaymentDate	ContractNo

Q3 (a)> Create Tables for all attributes

Query>

1. Create Property table

```
CREATE TABLE Property (  
    Pid INT PRIMARY KEY,  
    OAdhar VARCHAR(20),  
    PType VARCHAR(20),  
    PRegDate DATE  
);
```

Sql output> Table created.

2. Create Owner table

```
CREATE TABLE Owner (  
    Oid INT PRIMARY KEY,  
    OName VARCHAR(20),  
    OAdhar VARCHAR(20),  
    FOREIGN KEY (OAdhar) REFERENCES Property(OAdhar)  
);
```

Sql output> Table created.

3. Create Tenant table

```
CREATE TABLE Tenant (  
    Tid INT PRIMARY KEY,  
    TName VARCHAR(100),  
    TAdhar VARCHAR(255),  
    Contract INT  
);
```

Sql output> Table created.

4. Create Contract table

```
CREATE TABLE Contract (  
    Tid INT,  
    Pid INT,  
    Start_Date DATE,  
    End_Date DATE,  
    Amount INT,  
    FOREIGN KEY (Pid) REFERENCES Property(Pid),  
    FOREIGN KEY (Tid) REFERENCES Tenant(Tid)  
);
```

Sql output> Table created.

5. Create Payment table

```
CREATE TABLE Payment (  
    Tid INT,  
    Pid INT,  
    PaymentDate DATE,  
    Amount INT,  
    FOREIGN KEY (Tid) REFERENCES Tenant (Tid),  
    FOREIGN KEY (Pid) REFERENCES Property (Pid)  
);
```

Sql output > Table created.

6. Create Maintenance_Request table

```
CREATE TABLE Maintenance_Request (  
    Tid INT,  
    Pid INT,  
    PaymentDate DATE,  
    ContractNo INT,  
    FOREIGN KEY (Pid) REFERENCES Property(Pid),  
    FOREIGN KEY (Tid) REFERENCES Tenant(Tid),  
    FOREIGN KEY (PaymentDate) REFERENCES Payment (PaymentDate)  
);
```

Sql output > Table created.

Q3 (b)> Insert values to tables

Query>

1. Insert into Owner table

```
INSERT INTO Owner VALUES (1, 'Rajesh Kumar', 'AD1234567890');  
INSERT INTO Owner VALUES (2, 'Meena Patel', 'AD9876543210');  
INSERT INTO Owner VALUES (3, 'Pooja Pai', 'AD9776437210');  
INSERT INTO Owner VALUES (4, 'Amit Naik', 'AD9858573970');
```

Sql output > 1 row created.

2. Insert into Property table

```
INSERT INTO Property VALUES (1, 'AD1234567890', 'Residential', '2022-01-15');  
INSERT INTO Property VALUES (2, 'AD9876543210', 'Industrial', '2023-03-25');  
INSERT INTO Property VALUES (3, 'AD9776437210', 'Residential', '2023-03-25');  
INSERT INTO Property VALUES (4, 'AD9858573970', 'Industrial', '2023-03-25');
```

Sql output > 1 row created.

3. Insert into Tenant table

```
INSERT INTO Tenant VALUES (1, 'Amit Singh', 'AD2345678901', 1001);  
INSERT INTO Tenant VALUES (2, 'Priya Sharma', 'AD3456789012', 1002);
```

Sql output > 1 row created.

4. Insert into Contract table

```
INSERT INTO Contract VALUES (1, 1, '2022-02-01', '2023-02-01', 15000);  
INSERT INTO Contract VALUES (2, 2, '2023-04-01', '2024-04-01', 25000);  
INSERT INTO Contract VALUES (1, 4, '2021-08-01', '2025-04-01', 30000);
```

Sql output > 1 row created.

5. Insert into Payment table

```
INSERT INTO Payment VALUES (1, 1, '2022-02-10', 15000);  
INSERT INTO Payment VALUES (2, 2, '2023-04-10', 25000);  
INSERT INTO Payment VALUES (1, 1, '2023-03-10', 25000);
```

Sql output > 1 row created.

6. Insert into Maintenance_Request table

```
INSERT INTO Maintenance_Request VALUES (1, 1, '2022-02-10', 1001);  
INSERT INTO Maintenance_Request VALUES (2, 2, '2023-04-10', 1002);
```

Sql output > 1 row created.

Q3 (c)> Display each tables

Query>

1. Display the Property Table

SELECT * FROM Property;

Pid	OAdhar	PType	PRegDate
1	AD1234567890	Residential	2022-01-15
2	AD9876543210	Industrial	2023-03-25
3	AD9776437210	Residential	2023-03-25
4	AD9858573970	Industrial	2023-03-25

2. Display the Owner Table

SELECT * FROM Owner;

Oid	OName	OAdhar
1	Rajesh Kumar	AD1234567890
2	Meena Patel	AD9876543210
3	Pooja Pai	AD9776437210
4	Amit Naik	AD9858573970

3. Display the Tenant Table

SELECT * FROM Tenant;

Tid	TName	TAdhar	Contract
1	Amit Singh	AD2345678901	1001
2	Priya Sharma	AD3456789012	1002

4. Display the Contract Table

SELECT * FROM Contract;

Tid	Pid	Start_Date	End_Date	Amount
1	1	2022-02-01	2023-02-01	15000
2	2	2023-04-01	2024-04-01	25000
1	4	2021-08-01	2025-04-01	30000

5. Display the Payment Table

SELECT * FROM Payment;

Tid	Pid	PaymentDate	Amount
1	1	2022-02-10	15000
2	2	2023-04-10	25000
1	1	2023-03-10	25000

6. Display the Maintenance_Request Table

```
SELECT * FROM Maintenance_Request;
```

Tid	Pid	PaymentDate	ContractNo
1	1	2022-02-10	1001
2	2	2023-04-10	1002

Q3 (d)> Executing Queries

Query>

1. Find all payments made by a specific tenant (tenant id = 1)

```
SELECT * from Payment  
where Tid=1;
```

Sql output >

Tid	Pid	PaymentDate	Amount
1	1	2022-02-10	15000
1	1	2023-03-10	25000

Relational Algebraic expression>

$$\pi (\sigma_{Tid = 1} (Payment))$$

2. Find tenants who have rented residential properties

```
SELECT T.TName, T.TAdhar, P.PType  
FROM Tenant T  
JOIN Contract C ON T.Tid = C.Tid  
JOIN Property P ON C.Pid = P.Pid  
WHERE P.PType = 'Residential';
```

Sql output >

TName	TAdhar	PType
Amit Singh	AD2345678901	Residential

Relational Algebraic expression>

$$\Pi_{TName, TAdhar, PType} (\sigma_{PType = 'Residential'} (Property) \bowtie Contract \bowtie Tenant)$$

3. Find properties that have not been rented

```
SELECT P.Pid, P.OAdhar, P.PType, P.PRegDate  
FROM Property P  
WHERE P.Pid NOT IN (SELECT C.Pid FROM Contract C);
```

Sql output >

Pid	OAdhar	PType	PRegDate
3	AD9776437210	Residential	2023-03-25

Relational Algebraic expression>

$\pi_{Pid, OAdhar, PType, PRegDate}(\text{Property}) - \pi_{Pid}(\text{Contract})$

4. List all owners and their tenants

```
SELECT O.Oid AS Owner_ID, O.OName AS Owner_Name, T.Tid AS Tenant_ID, T.TName  
AS Tenant_Name  
FROM Owner O  
JOIN Property P ON O.OAdhar = P.OAdhar  
JOIN Contract C ON P.Pid = C.Pid  
JOIN Tenant T ON C.Tid = T.Tid;
```

Sql output >

Owner_ID	Owner_Name	Tenant_ID	Tenant_Name
1	Rajesh Kumar	1	Amit Singh
4	Amit Naik	1	Amit Singh
2	Meena Patel	2	Priya Sharma

Relational Algebraic expression>

$\pi_{Oid, Owner.OName, Tid, Tenant.TName}(\text{Owner} \bowtie \text{Property} \bowtie \text{Contract} \bowtie \text{Tenant})$

5. List all contracts and their associated properties for a specific tenant.

```
SELECT C.Tid, P.Pid, P.OAdhar  
FROM Contract C  
JOIN Property P ON C.Pid = P.Pid  
WHERE C.Tid = 1;
```

Sql output >

Tid	Pid	OAdhar
1	1	AD1234567890
1	4	AD9858573970

Relational Algebraic expression>

$\pi_{Tid, Pid, OAdhar}(\sigma_{Tenant_ID=1}(\text{Contract} \bowtie \text{Property}))$