

Assignment-1

Name: Karan Jayant Wadh

Reg No: 192373061

Course Name: DBMS

Course Code: CSA0593

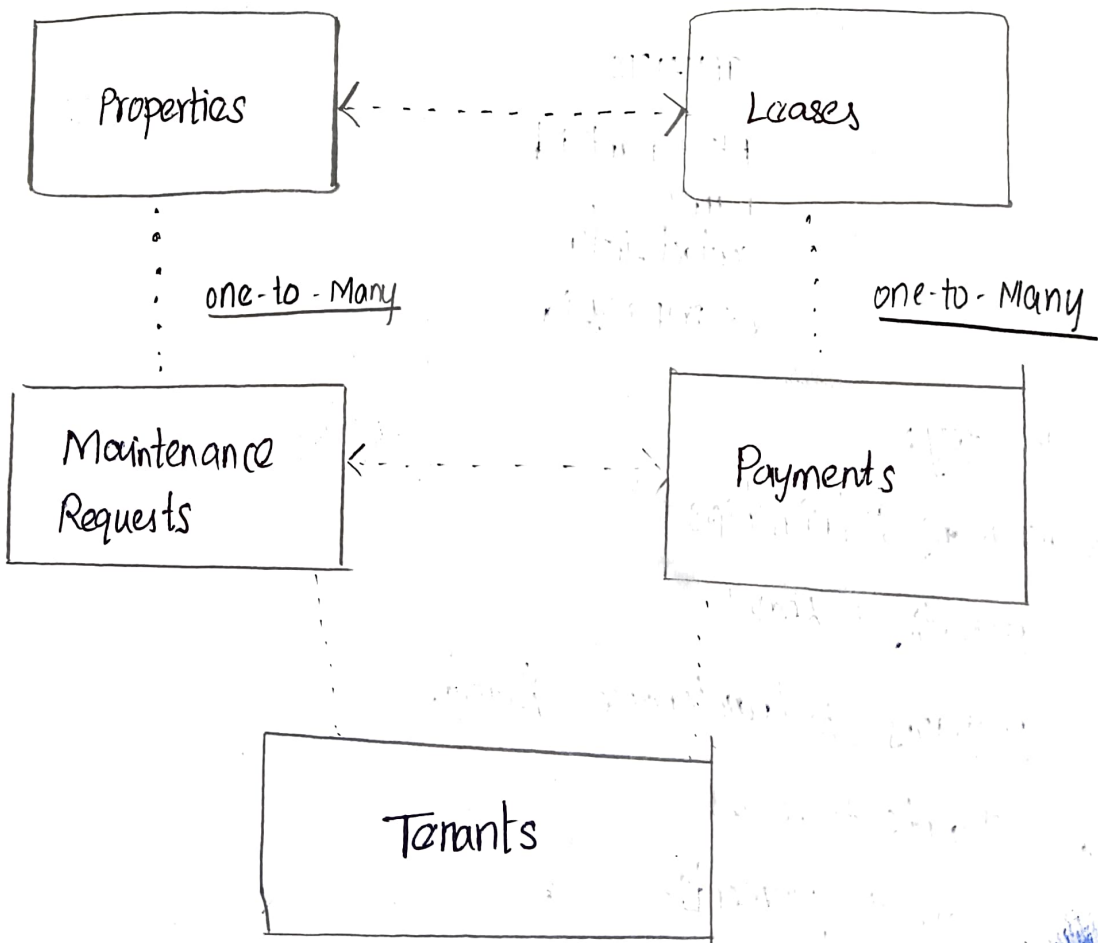
Real Estate Property Management

- Design a database for a real estate company to manage properties, tenants, leases and payments.

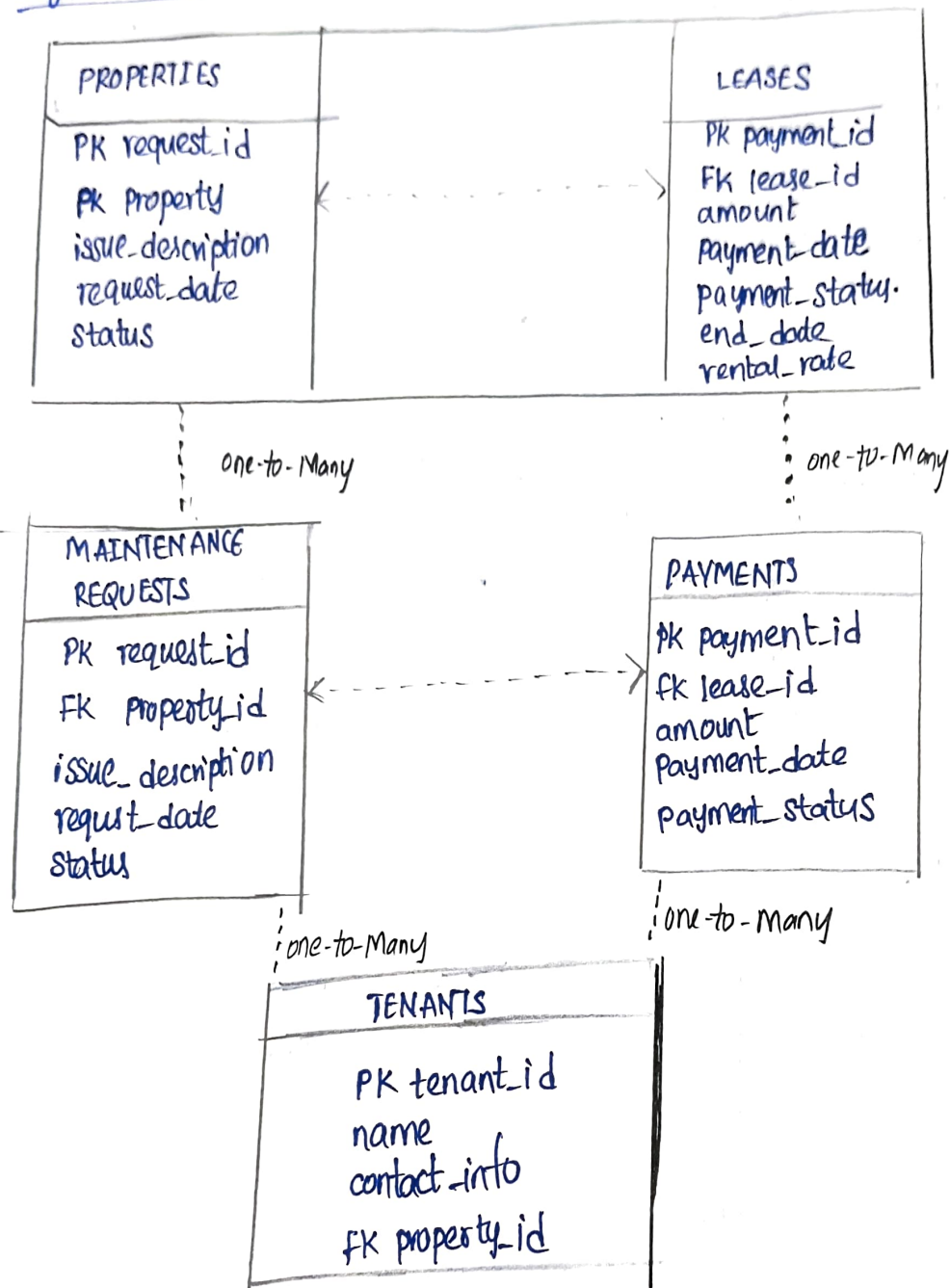
Requirements:

- * Model tables for properties, leases, tenants, payments and maintain requests.
- * Write stored procedures to automate lease renewals, calculate rental income and process payments.
- * Write queries to generate reports on occupancy rates, monthly rental income, and maintenance requests by property.
- * Implement triggers to send notifications for lease expiration and overdue payments.

Conceptual ERD:



Logical ERD :-

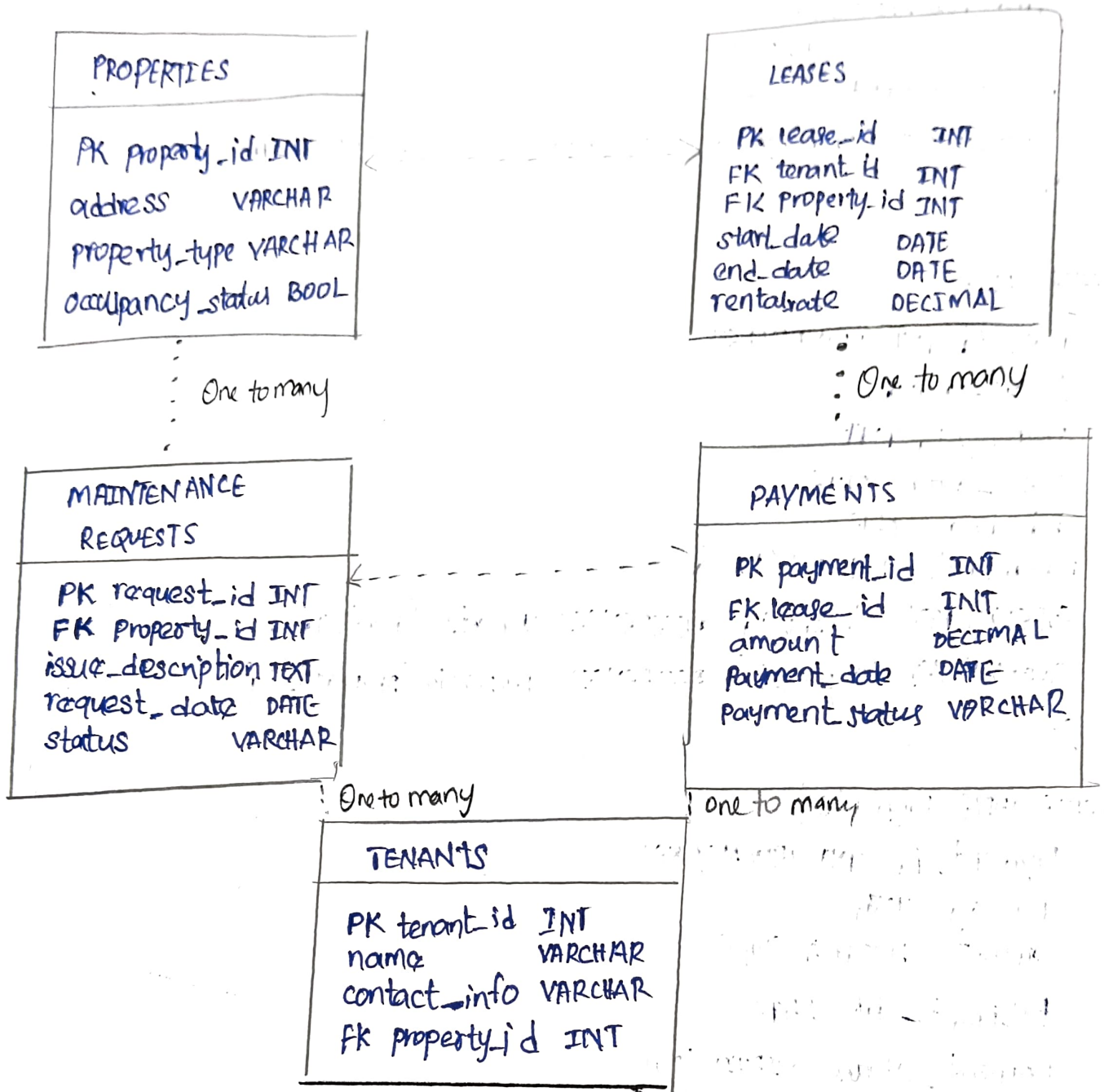


Dotted Lines Key:

• one-to-many Relationships:

- properties to leases
- properties to Maintenance Requests
- Tenants to leases
- leases to payments.

Physical ERD :-



Schema Design:-

CREATE TABLE properties (

property_id INT PRIMARY KEY,
address VARCHAR(255) NOT NULL,

property_id INT

occupancy_status BOOLEAN DEFAULT FALSE);

CREATE TABLE Tenants C

tenant_id INT PRIMARY KEY,
name VARCHAR(100),
contact_info VARCHAR(255),
property_id INT,
FOREIGN KEY (property_id) REFERENCES properties (property_id)
);

CREATE TABLE leases C

lease_id INT PRIMARY KEY,
tenant_id INT,
property_id INT,
start_date DATE,
end_date DATE,
rental_rate DECIMAL(10,2),
FOREIGN KEY (tenant_id) REFERENCES tenants (tenant_id),
FOREIGN KEY (property_id) REFERENCES properties (property_id)
);

CREATE TABLE Payments C

Payment_id INT PRIMARY KEY,
lease_id INT,
amount DECIMAL(10,2),
Payment_date DATE,
Payment_status VARCHAR(20),
FOREIGN KEY (lease_id) REFERENCES leases (lease_id)
);

CREATE TABLE MaintenanceRequests C

request_id INT PRIMARY KEY
Property_id INT
issue_description TEXT
request_date DATE,
status VARCHAR(20),
FOREIGN KEY (property_id) REFERENCES properties (property_id)
);

Stored Procedures :-

1) Lease Renewal Automation :

CREATE PROCEDURE RenewLease (IN lease_id INT, IN new_end_date DATE)

BEGIN

UPDATE Leases

SET end_date = new_end_date

WHERE lease_id = lease_id;

END;

2) Calculate Monthly Rental Income :

CREATE PROCEDURE Calculate Monthly Income (IN month INT, IN Year INT; OUT total_income
DECIMAL (10,2))

BEGIN

SELECT SUM (amount)

INTO total_income

FROM payments

WHERE MONTH (payment_date) = month AND YEAR (payment_date) = year;

END;

3) process payments :

CREATE PROCEDURE process Payment (IN lease_id INT, IN payment_amount DECIMAL (10,2),
IN payment_date DATE)

BEGIN

INSERT INTO Payments (lease_id, amount, payment_date, payment_status)

VALUES (lease_id, payment_amount, payment_date, 'completed');

END;

REPORTS :

1) Occupancy Rate by property

SELECT property_id COUNT (tenant_id) AS occupied_units

FROM payments

GROUP BY YEAR (payment_date), MONTH

GROUP BY property_id;

2) Monthly Rental Income Report :

```
SELECT MONTH(payment_date) AS month, YEAR(payment_date) AS YEAR, SUM(amount) AS  
FROM Payments total Income  
GROUP BY YEAR(payment_date), MONTH(payment_date)  
ORDER BY year, month;
```

3) Maintenance Requests by Property :

```
SELECT property_id, COUNT(request_id) AS total_requests  
FROM maintenance Requests  
GROUP BY property_id;
```

Triggers:-

1. Trigger for Lease Expiration Notification :

```
CREATE TRIGGER LeaseExpirationNotification  
BEFORE UPDATE ON leases  
FOR EACH ROW  
BEGIN  
    IF NEW.end_date <= CURDATE() THEN  
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'lease expiration';  
    END IF;  
END;
```

2. Trigger for Overdue Payments Notifications :

```
CREATE TRIGGER OverduePaymentNotification  
AFTER INSERT ON Payments  
FOR EACH ROW  
BEGIN  
    IF NEW.payment_status = 'overdue' THEN  
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'overdue payment';  
    END IF;  
END;
```