Assignment - 01.

Kasi Sai Yaswanth

192373061.

CSA0593 [DBMS]

Name: Kallelaiyarwardh ✱ASSIGNMENT-1✱ RegNo: 192373061

Course Name: DBMS   CourseCode: CSA0593

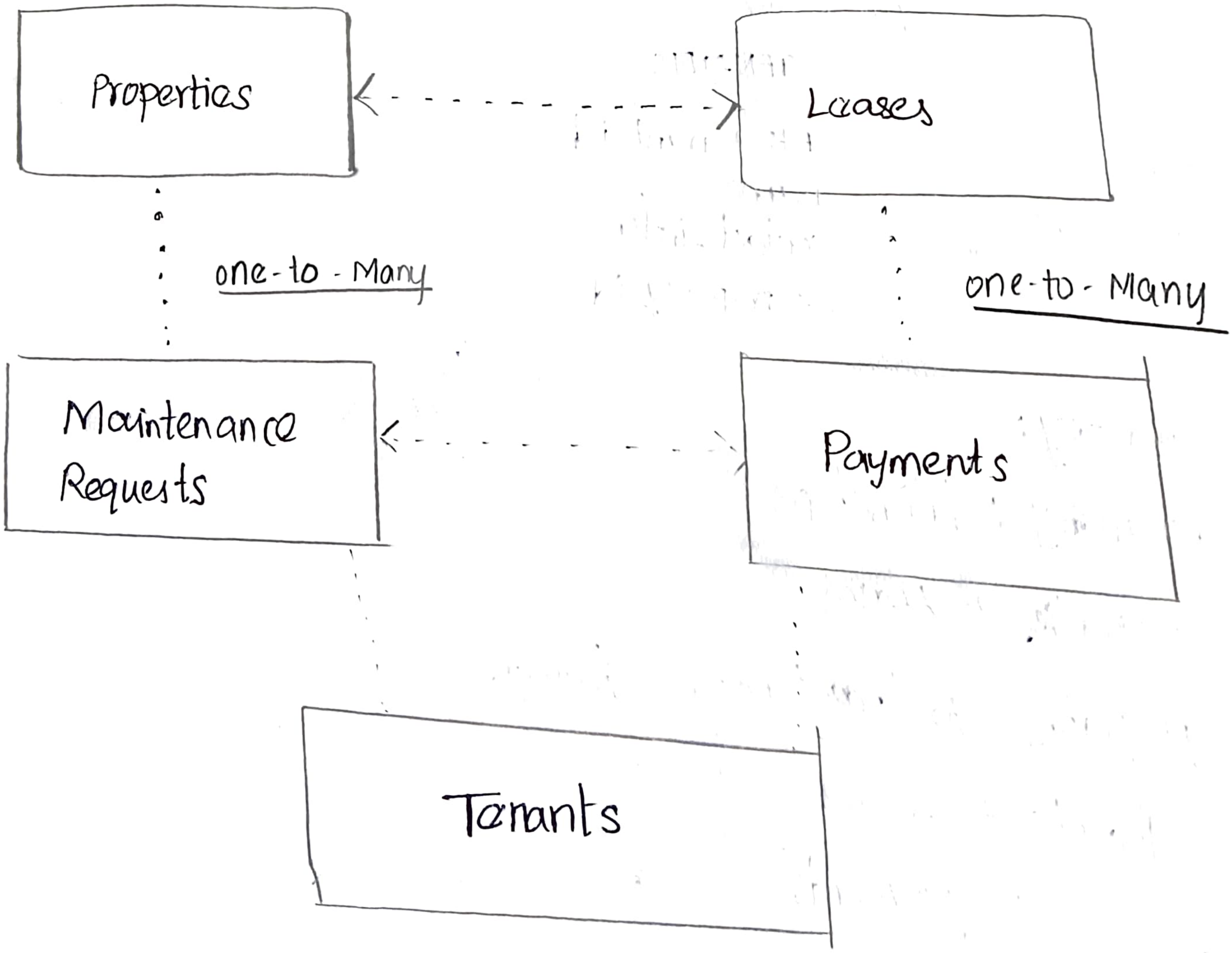
Real Estate Property Management

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

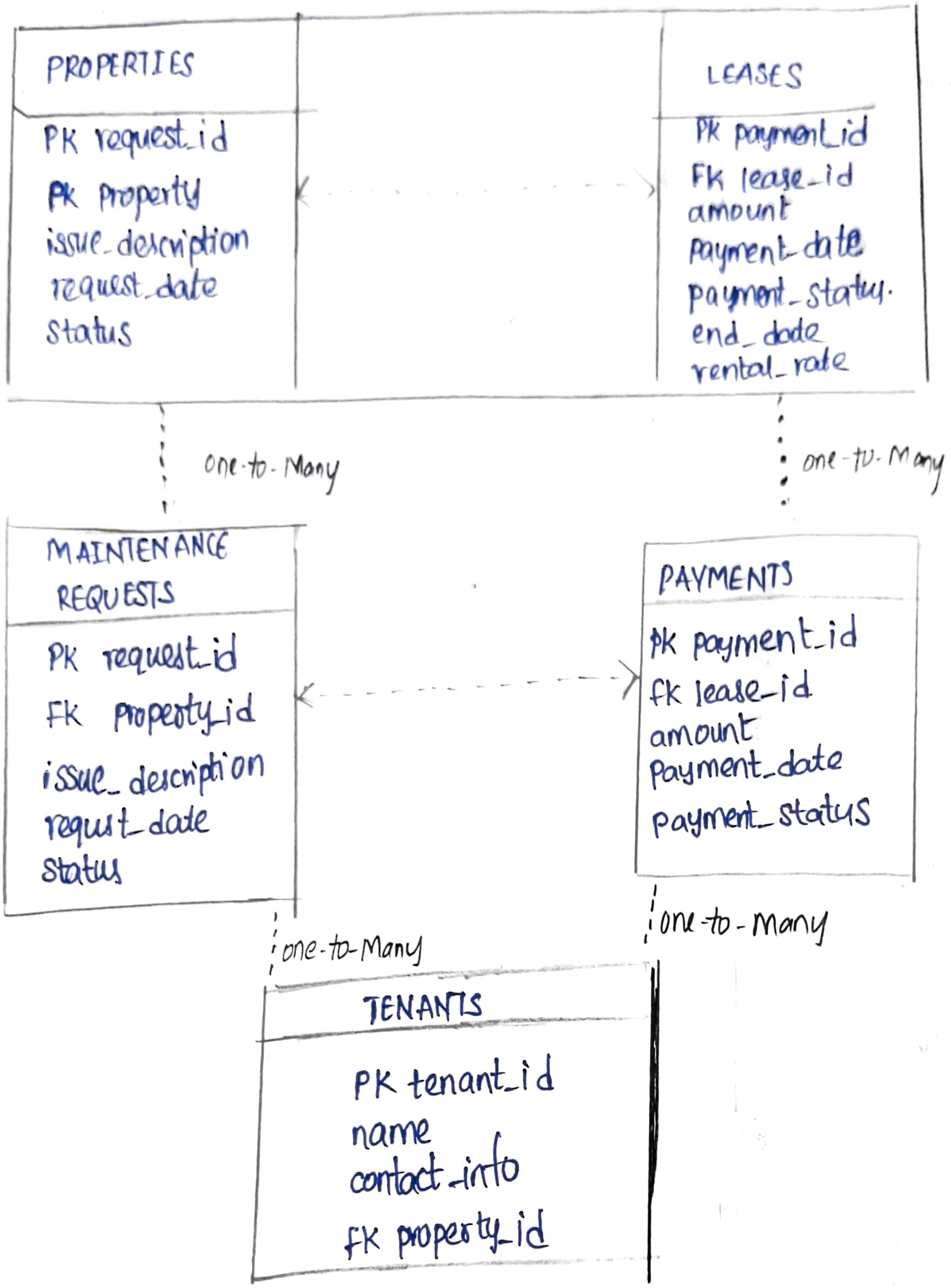
Requirements:

* 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 maintainance requests by property.
* Implement triggers to send notifications for lease expiration and overdue payments.

Conceptual ERD:



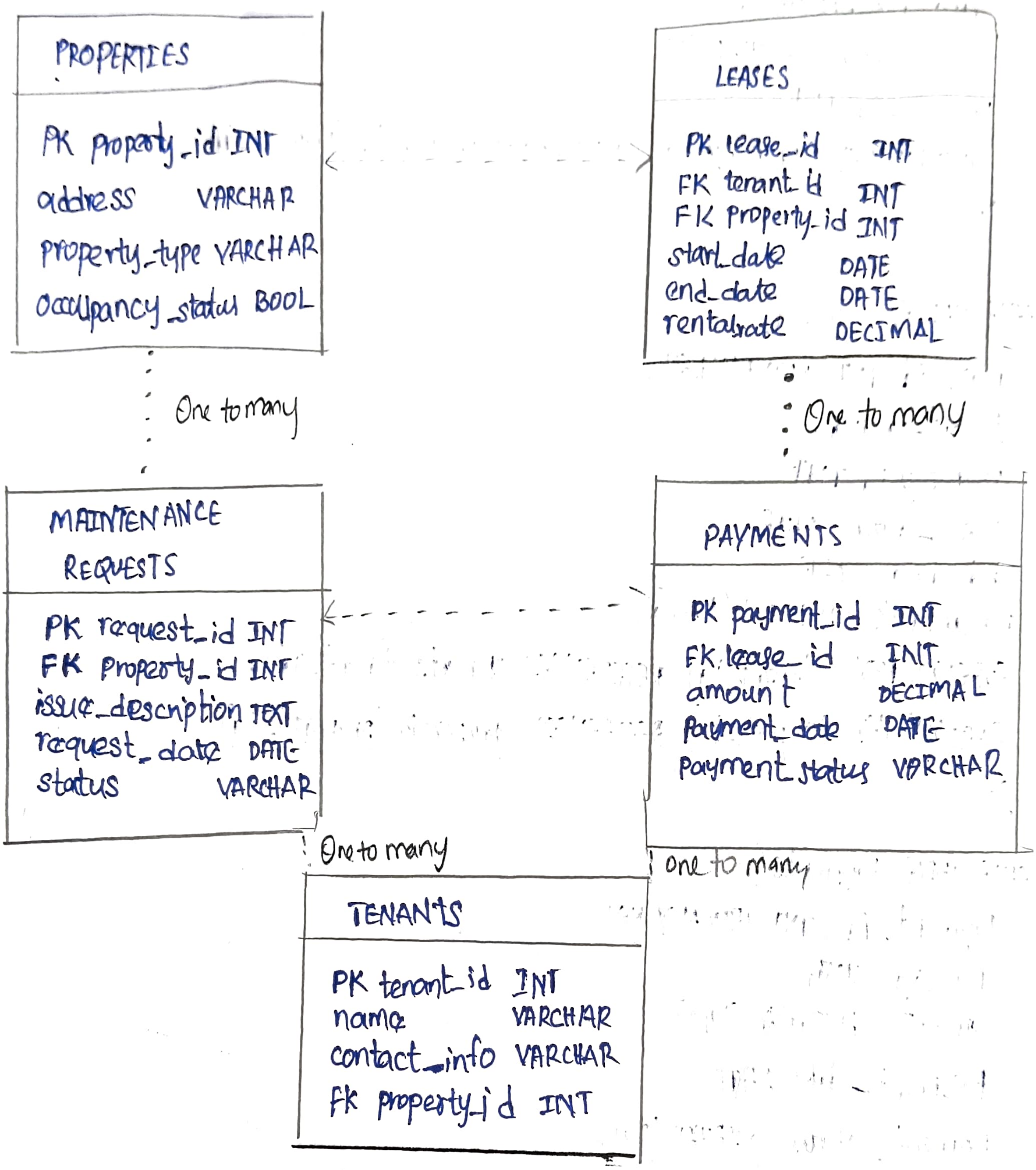
Logical ERD :-



Dotted Lines Key:

- one-to Many Relationships :
  - properties to Leases
  - properties to Maintenance Requsts.
  - Tenants to Leases
  - Leases to payments.

Physical ERD :-



PROPERTIES	
PK property_id	INT
address	VARCHAR
property_type	VARCHAR
occupancy_status	BOOL

LEASES	
PK lease_id	INT
FK tenant_id	INT
FK property_id	INT
start_date	DATE
end_date	DATE
rentalrate	DECIMAL

MAINTENANCE REQUESTS	
PK request_id	INT
FK Property_id	INT
issue_description	TEXT
request_date	DATE
status	VARCHAR

PAYMENTS	
PK payment_id	INT
FK lease_id	INIT
amount	DECIMAL
Payment_date	DATE
Payment_status	VARCHAR

TENANTS	
PK tenant_id	INT
name	VARCHAR
contact_info	VARCHAR
FK property_id	INT

scheme 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 (
    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 (
    lease_id INT PRIMARY KEY,
    tenant-id INT,
    Property_id INT,
    start_data DATE,
    and_data DATE,
    rental_rate DECIMAL(10,2),
    FOREIGN KEY (tenant_id) REFERENCES Tenants (tenants_id),
    FOREIGN KEY (Property_id) REFERENCES propertieg (property_id)
    );

CREATE TABLE Payments(
    Payment_id INT PRIMARY KEY,
    lease_id INT,
    amount DECIMAL (10,2),
    Payment-data DATE,
    Payment_status VARCHAR(20),
    FOREIGN KEY (lease_id) REFERENCES Leases (lease_id)
);
CREATE TABLE MaintenanceRequests (
  requst_id INT PRIMARY KEY
  Property_id INT
  issue_description TEXT
  request_data DATE,
  status VARCHAR(20),
  FOREIGN KEY (property_id) REFERENCES properties (property_id)
  );
```

Nov 14, 2024, 08:0

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 data
     WHERE lease_id = lease_id;
END;
```

2) calculate Monthly Rental Income :

```
CREATE PROCEDURE CalculateMonthlyIncome (IN month INT, IN Year INT; OUT total_income
                                                              DECIMAL (10,20))
BEGIN
     SELECT SUM (amount)
     INTO total_income
     FROM payments
    WHERE MONTH (payment data) =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 data DATE)
 BEGIN
     INSERT INTO Payments (lease id, amount, payment, data, payment_status)
     VALUES (lease_id, payment_data, 'completed');
 END;
```

REPORTS :

1) Occupancy Rates 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
                                                             total Income
FROM Payments
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 maintenane 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_data <= 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;
```

Conclusion

These SQl Statements create the database for RealEstate Data Base Managements. This setup provides robust foundation for managing a real estate property database with core operations automated by stored procedures and triggers. These examples can be extended further to cover additional business requirements.