

Task 1: Project Proposal

Software as a Service (SaaS)

Project Overview:

This project is a web-based SaaS platform that allows users to subscribe to various software plans, receive applicable discounts, and manage their subscription data securely. Its Built using the Microsoft .NET Framework with VB.NET and backed by MS SQL Server, the platform offers role-based access for both customers and administrators.

2. Project Objectives:

- Deliver a data-driven web application using MS SQL Server.
- Ensure state management for consistent user sessions and data handling.
- Implement form and input validation on both client and server sides.
- Secure sensitive data through authentication, authorization, and encryption.
- Provide user roles (Admin/Customer) with tailored functionalities.

3. Key Features:

- User Registration & Login
- View and Subscribe to Plans
- Apply Discounts
- View Subscription History
- Admin: Add/Edit/Delete Plans, Manage Users
- Validation on all input fields
- Secure user authentication with hashed passwords
- Role-based access control

Task 2: Software Requirements Specification (SRS)

1. Introduction

- **Purpose:** To outline the software requirements for the SaaS platform.
- **Scope:** A web-based application allowing users to manage software subscriptions and discounts with role-specific access.

2. Functional Requirements

- **FR1:** Users can register and log in with role distinction.
- **FR2:** Customers can browse and subscribe to software plans.
- **FR3:** Discounts can be applied.
- **FR4:** Admins can manage users, plans, and discounts.
- **FR5:** All forms must include field validation.
- **FR6:** Users can view current and past subscriptions.

3. Non-Functional Requirements

- **NFR5:** Browser compatibility: Chrome, Edge, Firefox.
- **NFR2:** Passwords must be encrypted.
- **NFR3:** Only authenticated users can access the system.

- **NFR4:** Session timeout after 15 minutes of inactivity.

4. System Features

- Login/Logout
- Session-based user state
- User dashboard
- Admin control panel
- Report generation (PDF/Excel)

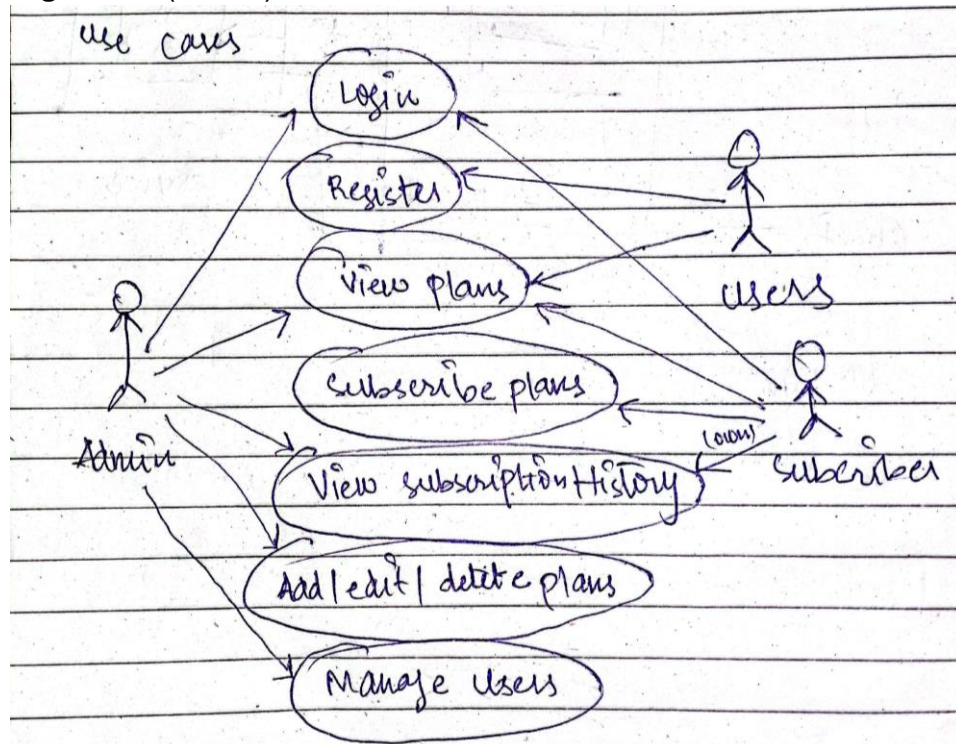
5. Assumptions and Constraints

- Internet connectivity is required.
- Only Admins can create or delete plans.
- Payments are not handled in this phase.

Task 3: Analysis and Design:

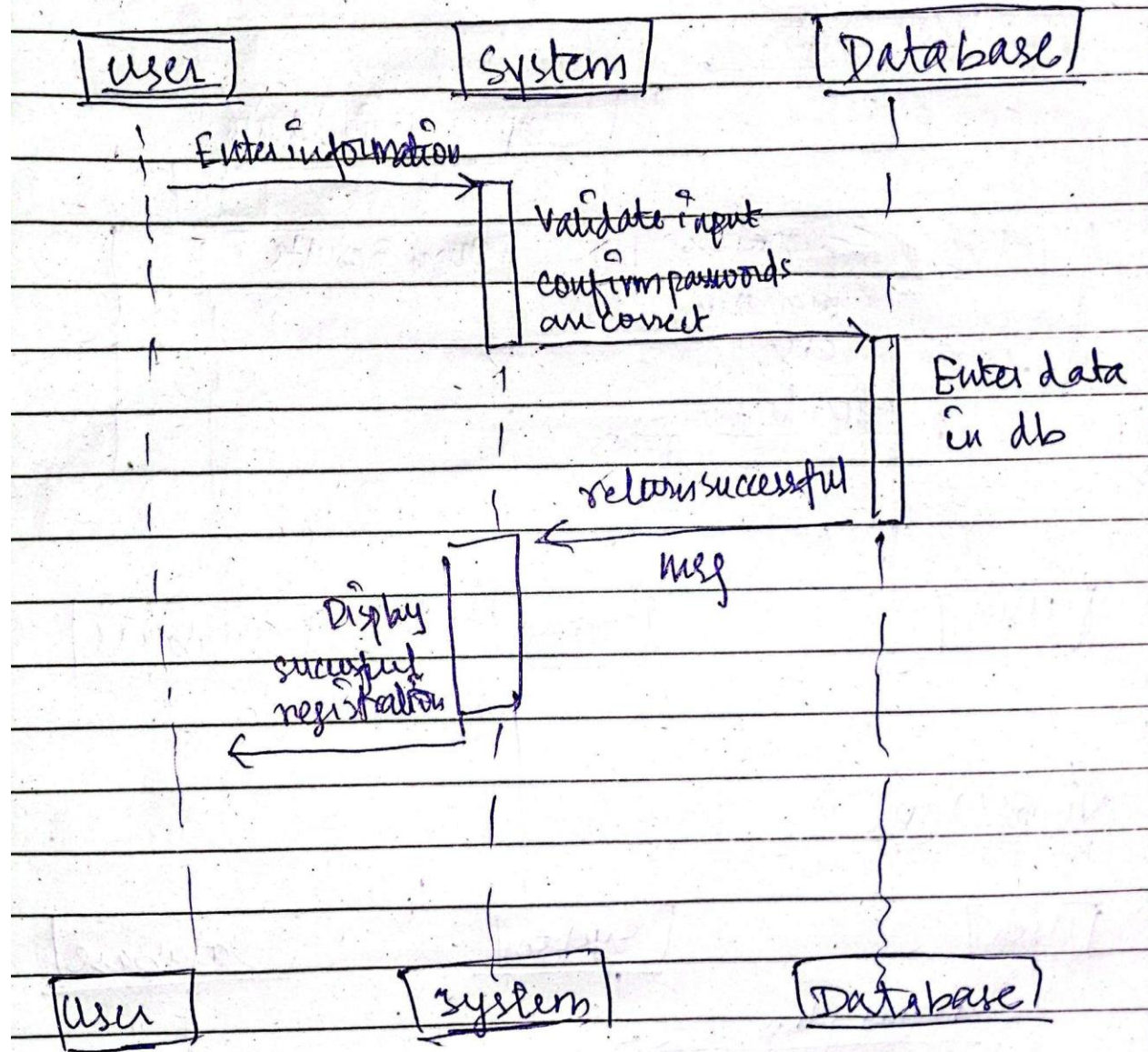
A. Use Case Model Actors:

- Customer Admin , Users
- **Use Cases:**
- Register/Login
- View Plans
- Subscribe to Plan
- Apply Discount
- View Subscription History
- Add/Edit/Delete Plans (Admin)
- Manage Users (Admin)

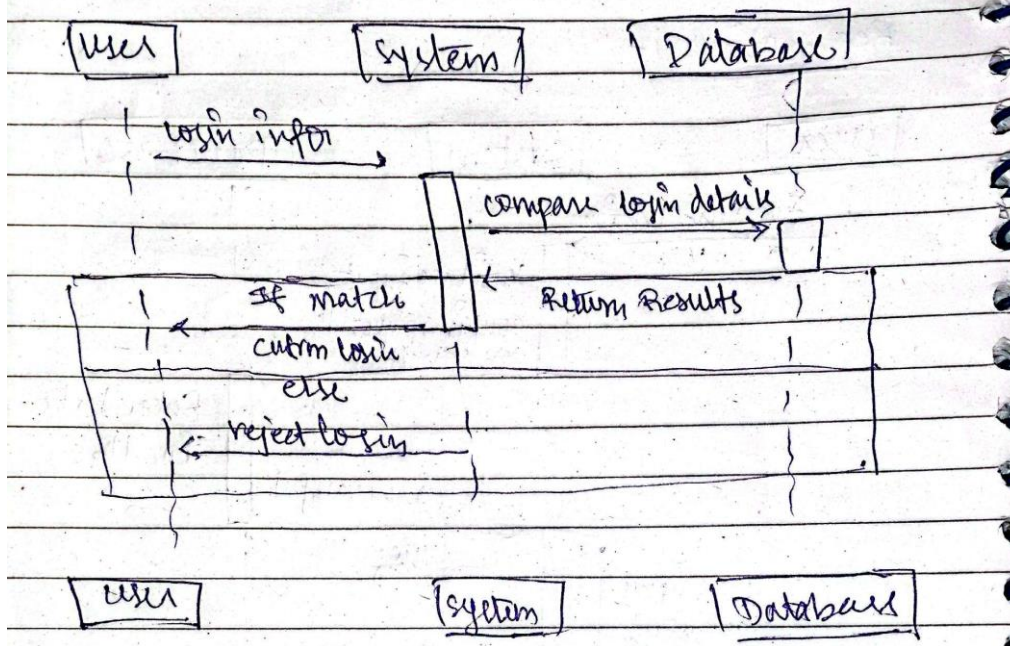


Sequence Diagrams

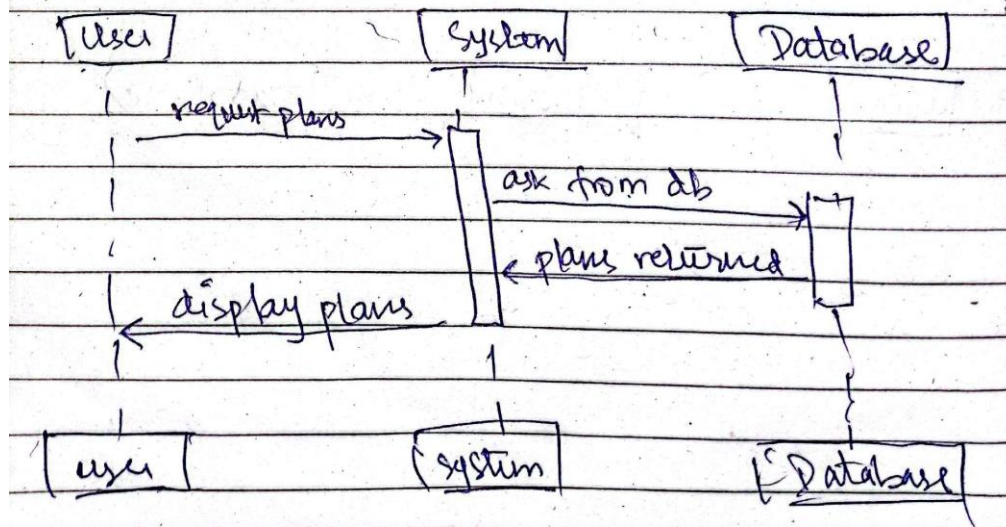
Registration 1 -



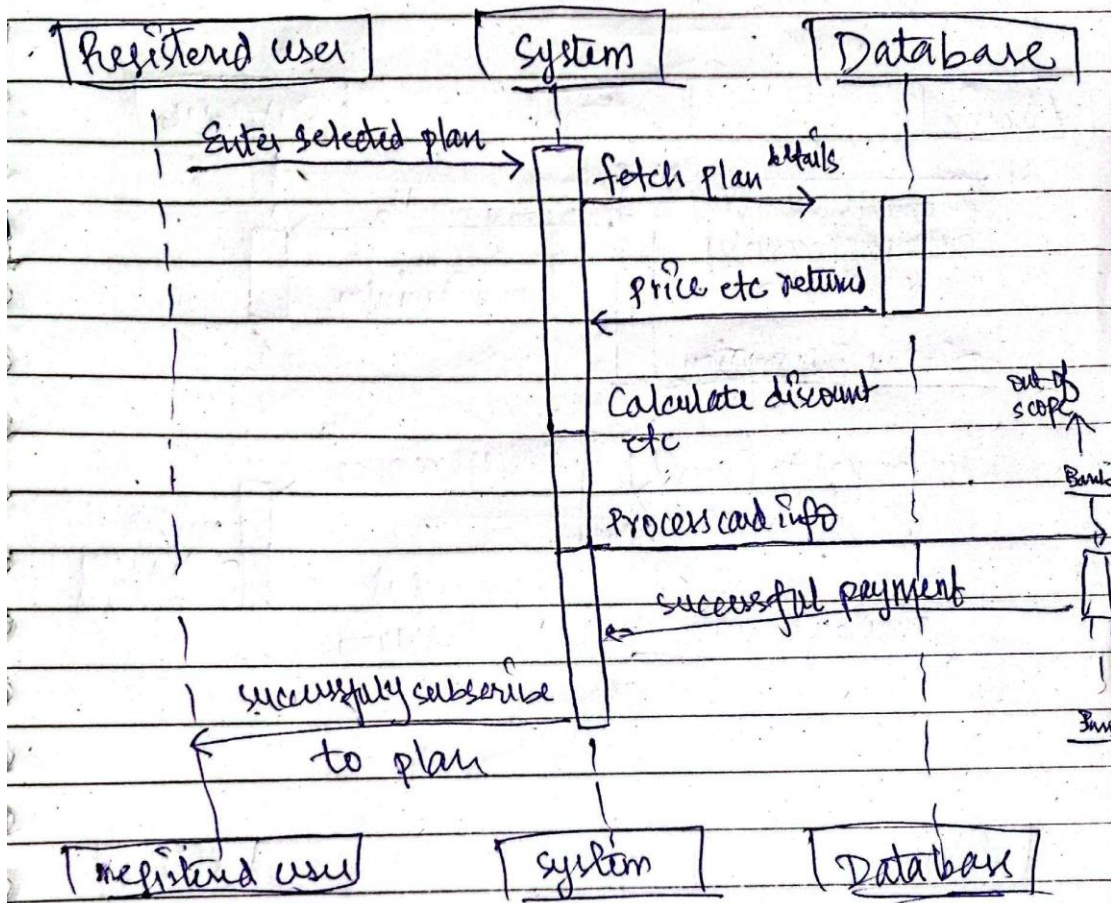
Login



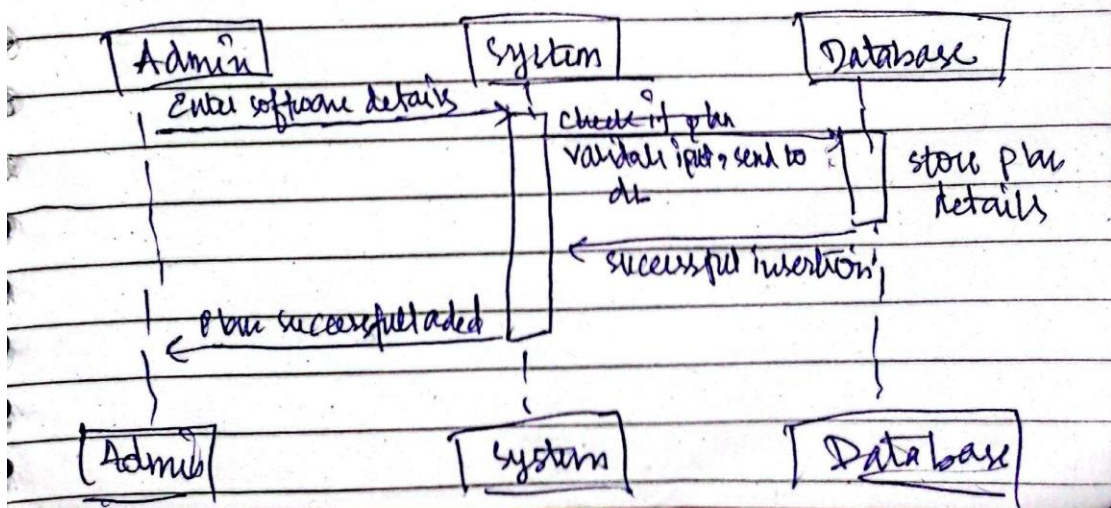
View Plans



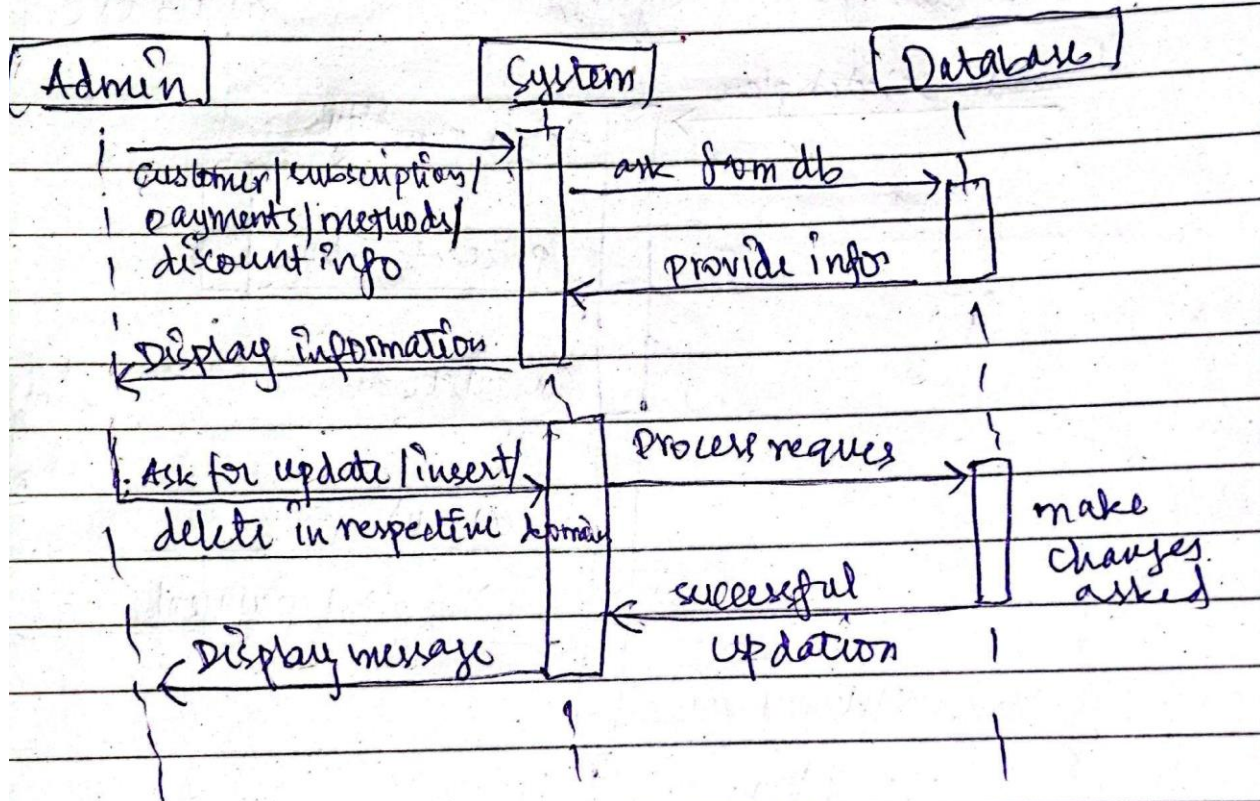
Subscribe plan



Add software Plans:-



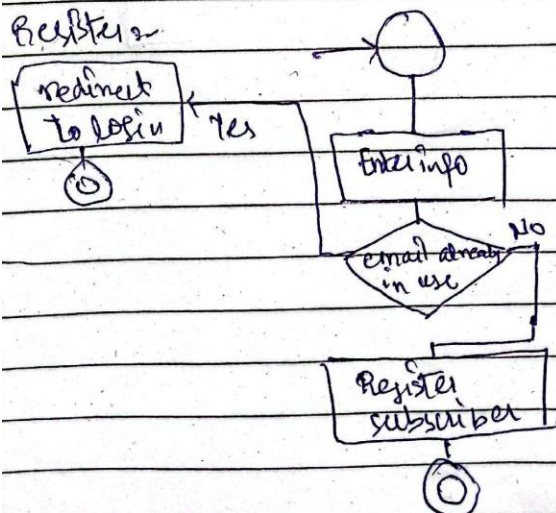
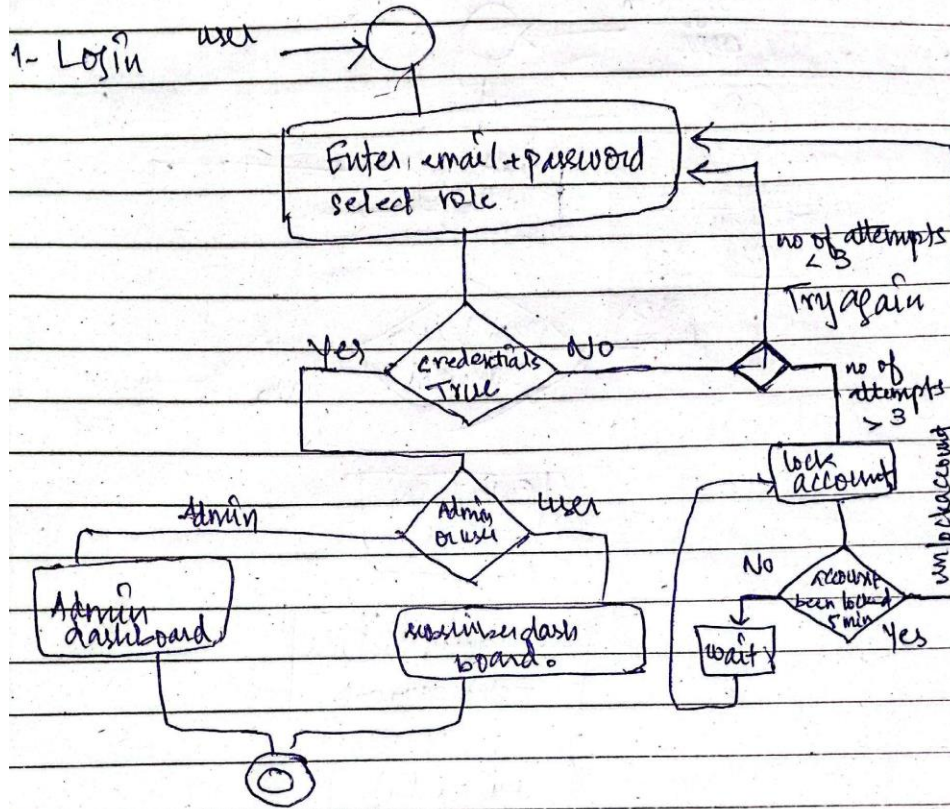
Manage customers, subscriptions, payments, methods and discounts



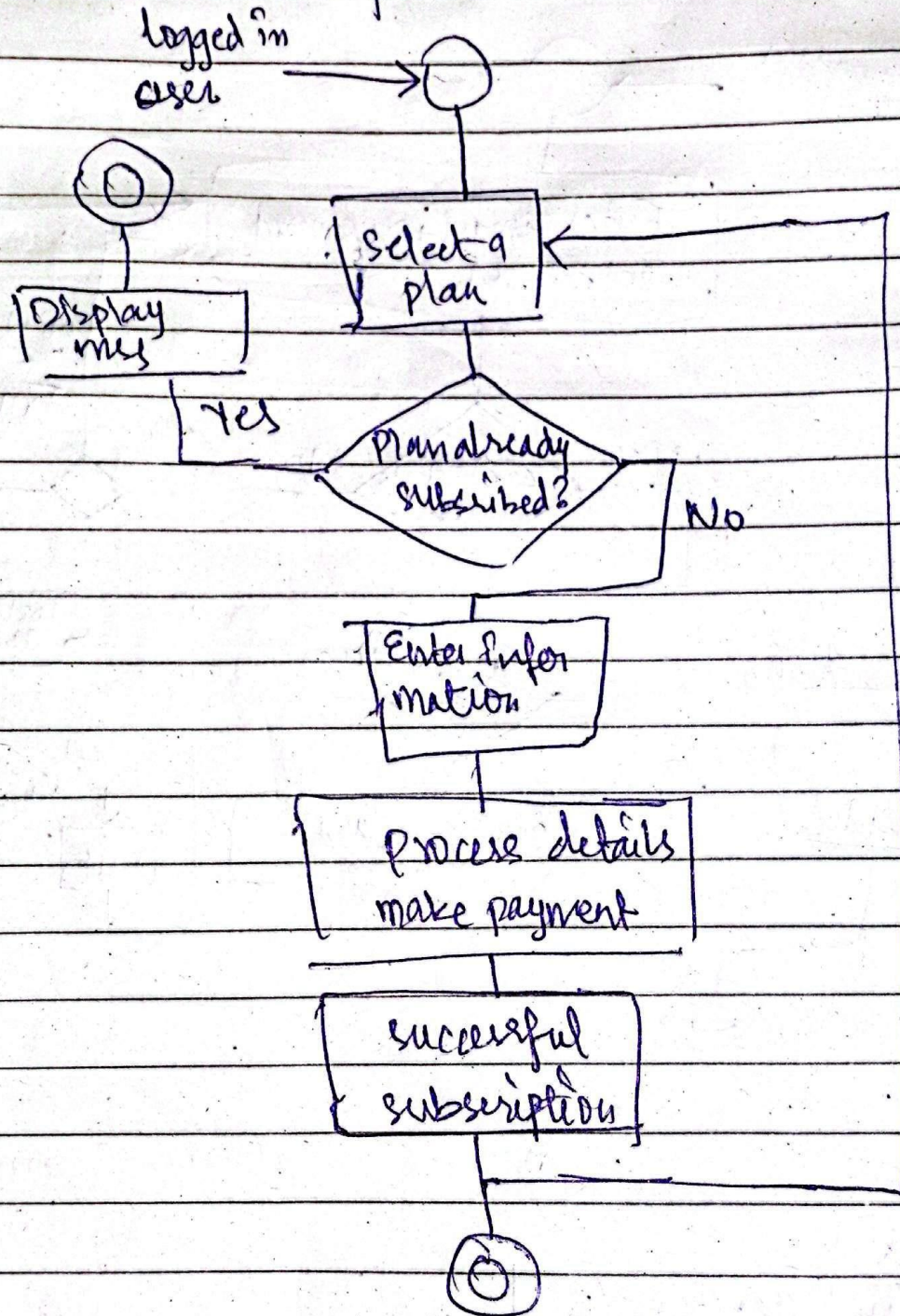
Admin System Database

Activity Diagrams

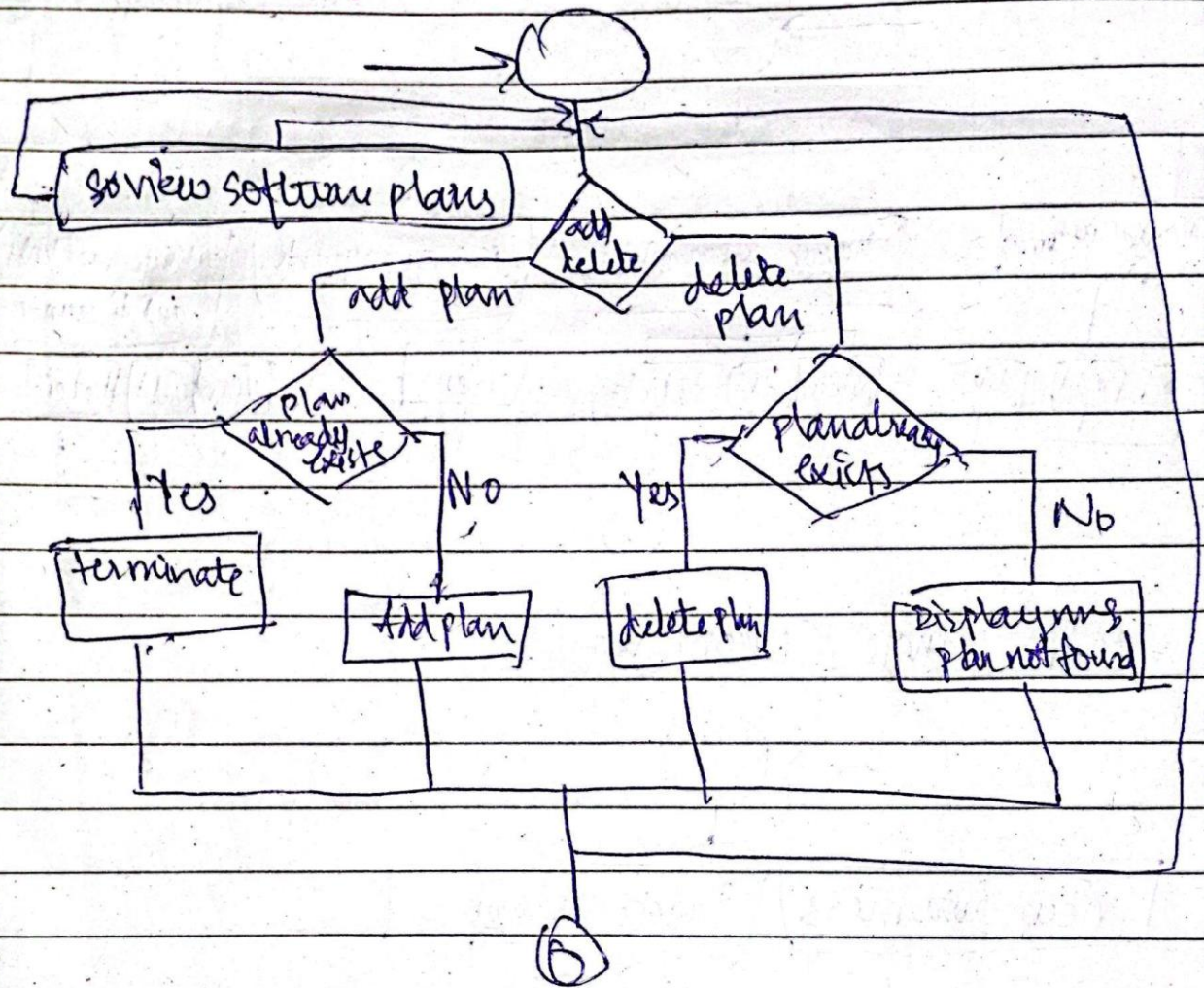
Activity Diagrams 2-



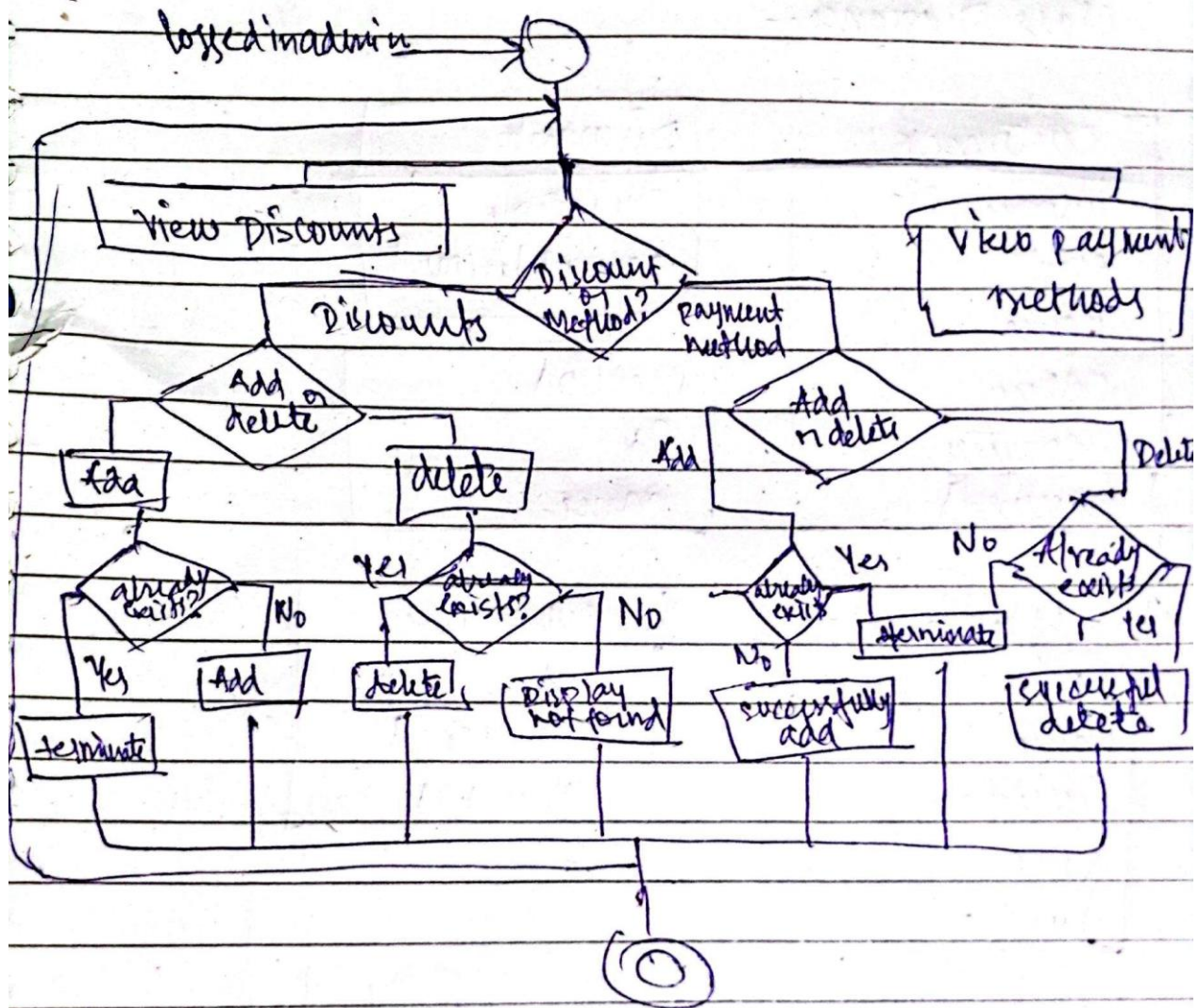
Subscribe to a plan



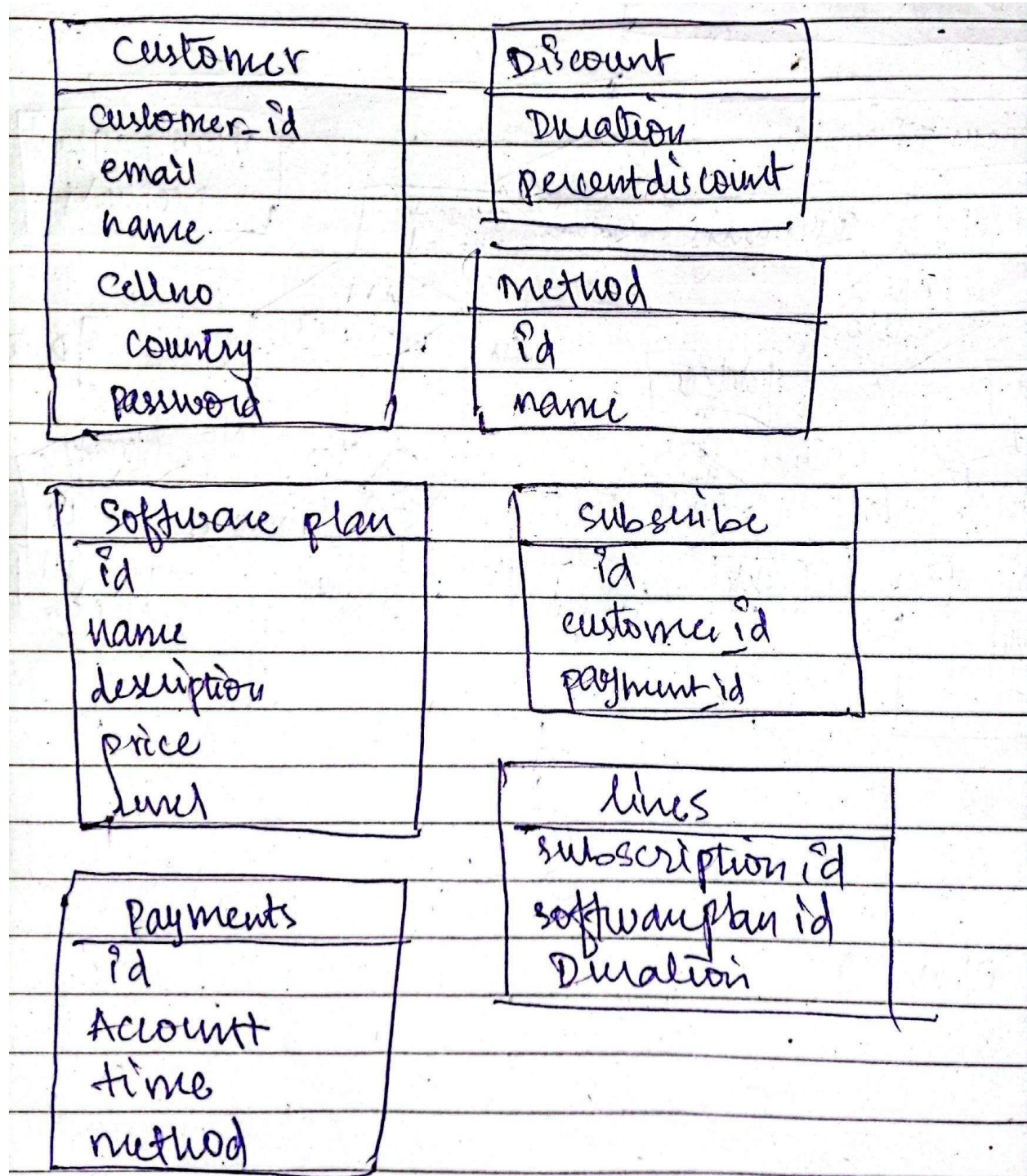
Admin Software Plans Manage



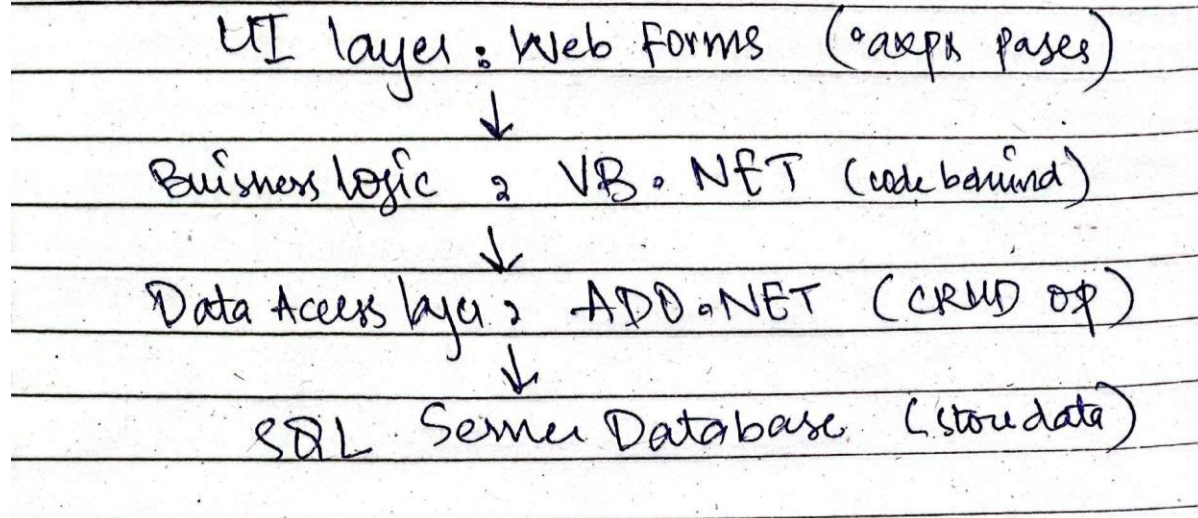
Admin Discount and Payments Method Manage



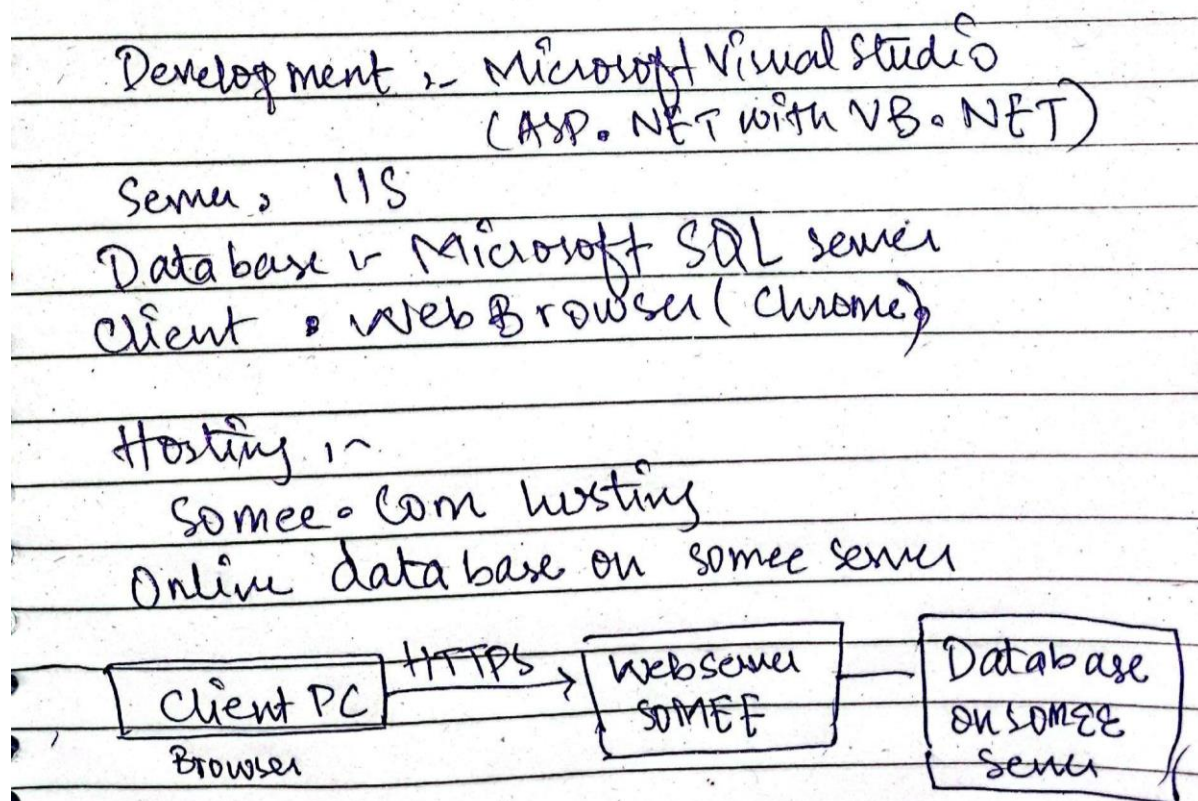
Class Diagrams



Component Model



Deployment Model



DBMS Project Report: Introduction:

The goal of this project is to create a comprehensive database system for efficiently managing software subscription plans. The database handles core aspects of subscription services, including customer records, subscription details, software plans, payments, and discounts. By automating and streamlining operations like billing, discount application, and subscription tracking, the system ensures efficiency, scalability, and security. It enables role-based access control to protect sensitive data and supports business analysis by providing insights into popular plans and revenue generation.

This solution simplifies subscription management for both customers and administrators, enhances customer service, and supports the growth of subscription-based businesses with a scalable and reliable database structure.

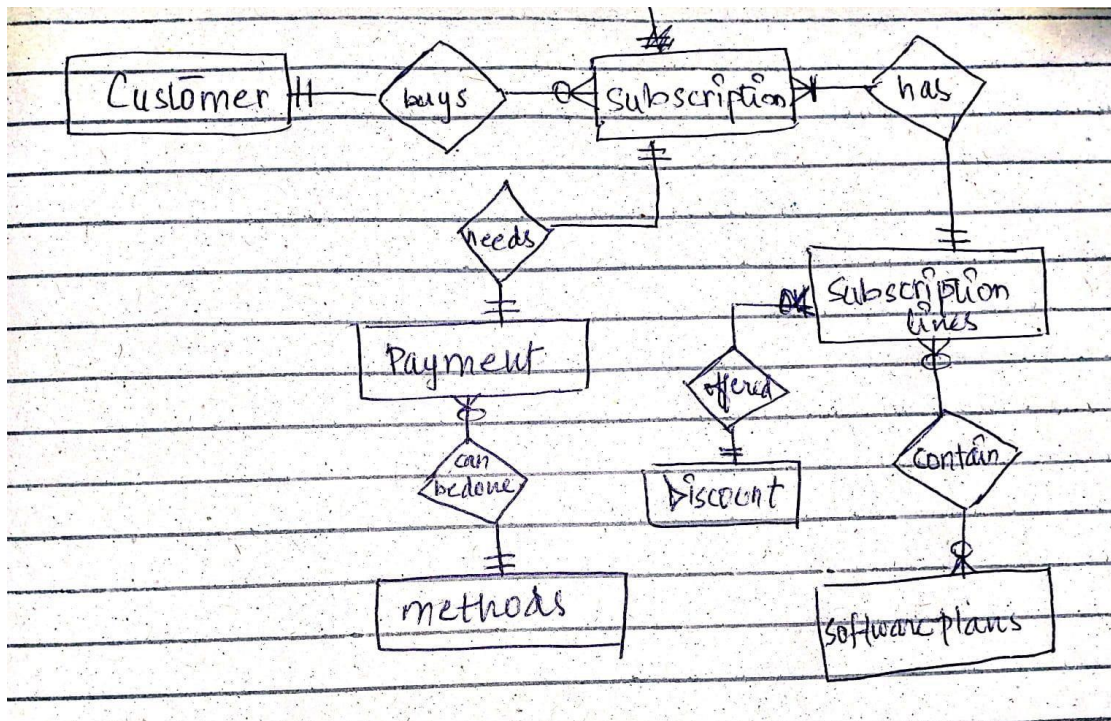
Entities in this model:

- **Customer:** Represents individuals or entities purchasing software subscriptions.
- **Software_Plan:** Represents the different software products available for subscription.
- **Method:** Defines the types of payment methods supported by the system.
- **Discount_On_Duration:** Details how discounts are applied based on subscription duration.
- **Payments:** Tracks payments made for subscriptions, including amounts and payment methods.
- **Subscriptions:** Represents a customer's subscription to one or more software plans, including payment details.
- **Subscription_Lines:** A junction table that connects each subscription to the corresponding software plans and associated discount.

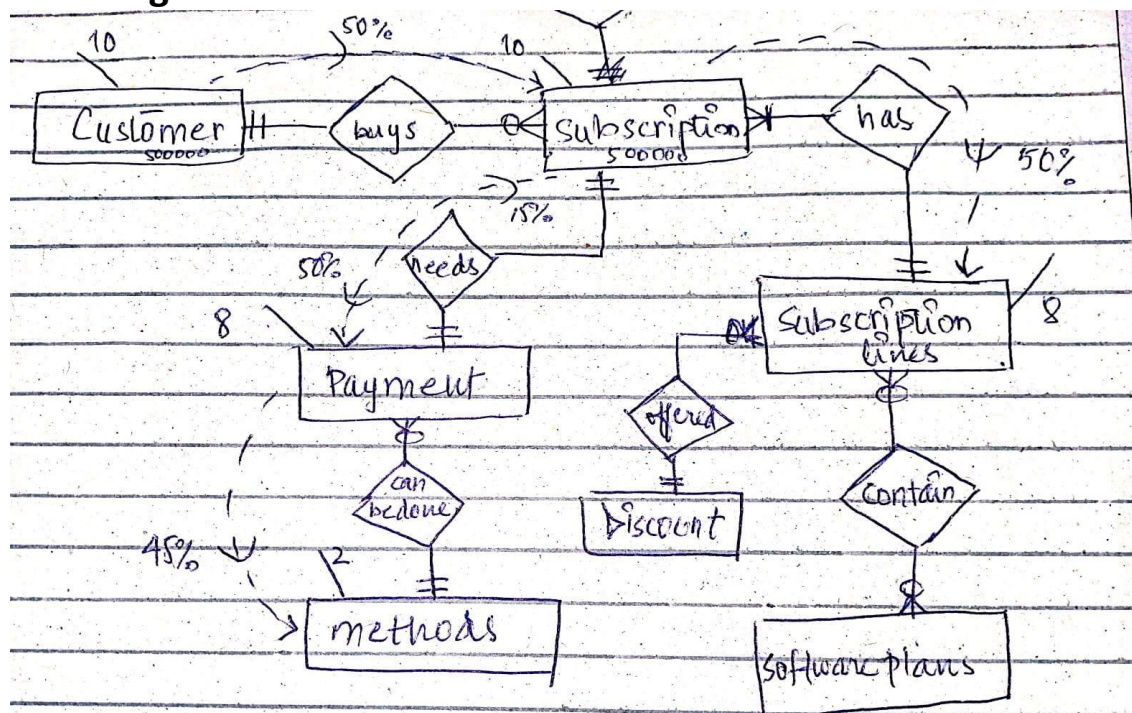
Business Rules:

1. **Customer Information:** A customer can subscribe to one or more software plans. Each customer will have a unique identifier (Customer_Id), along with personal details like email, name, and country.
2. **Software Plans:** A software plan represents the software package offered to customers. Each plan will have a unique identifier (Software_Plan_Id) and pricing information (Price_Per_Month).
3. **Subscriptions:** A customer can have multiple subscriptions, and each subscription will be associated with a specific payment. The subscription will include multiple software plans, each with a defined duration (Duration), which could influence the price due to available discounts.
4. **Discounts:** Discounts may apply based on the duration of the subscription, represented by the Discount_On_Duration table. This allows flexible pricing strategies for long-term commitments.
5. **Payments:** Payments for subscriptions are made through various methods, each identified by Method_Id. The payment details will be captured in the Payments table, including the payment timestamp and the method used.
6. **Relationships:**
 - Customers can have multiple subscriptions, and each subscription can have multiple software plans associated with it.
 - A software plan can be part of multiple subscriptions.
 - A subscription is associated with a specific payment, and the payment can be processed through different methods (e.g., credit card, PayPal).

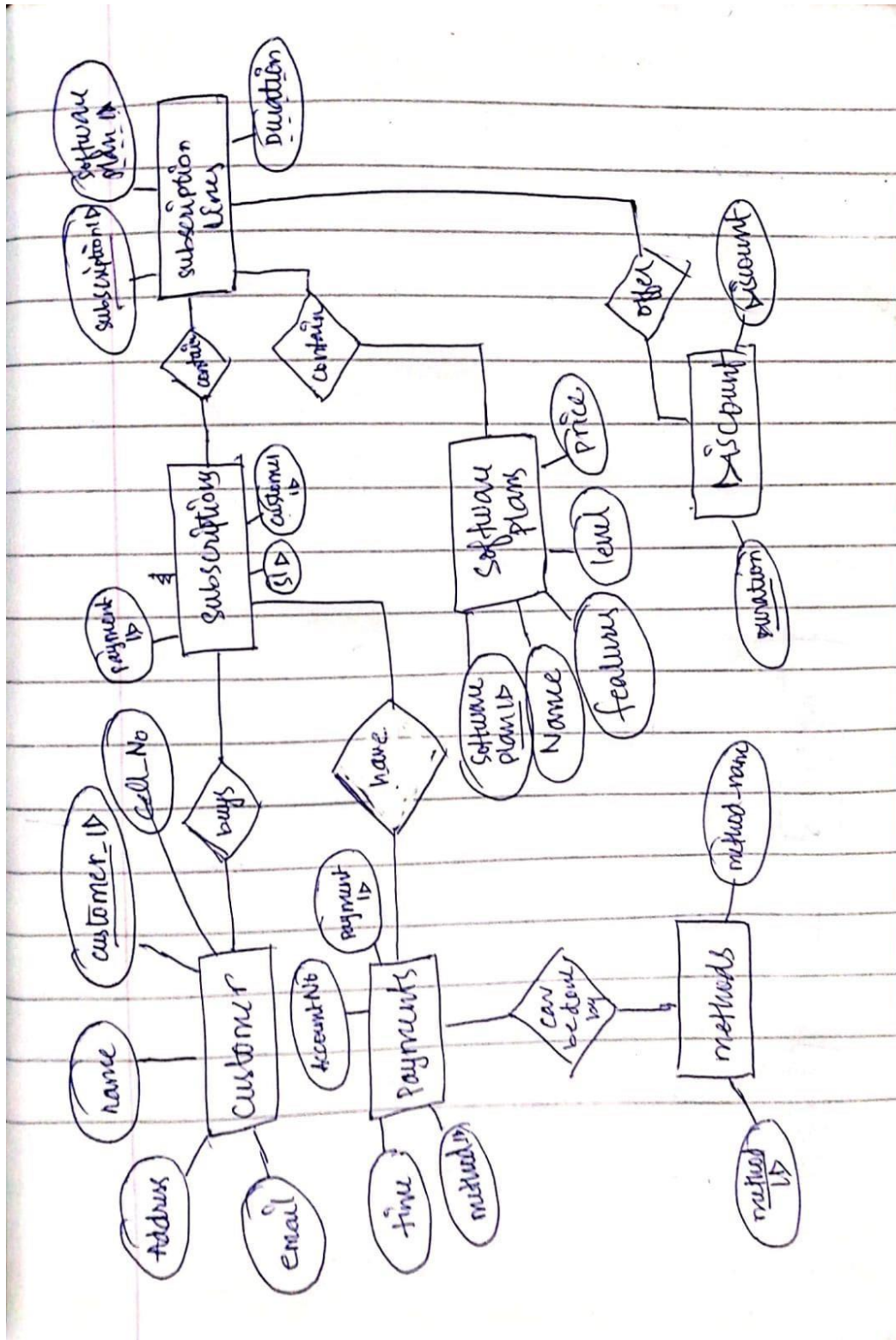
Enterprise Data Model



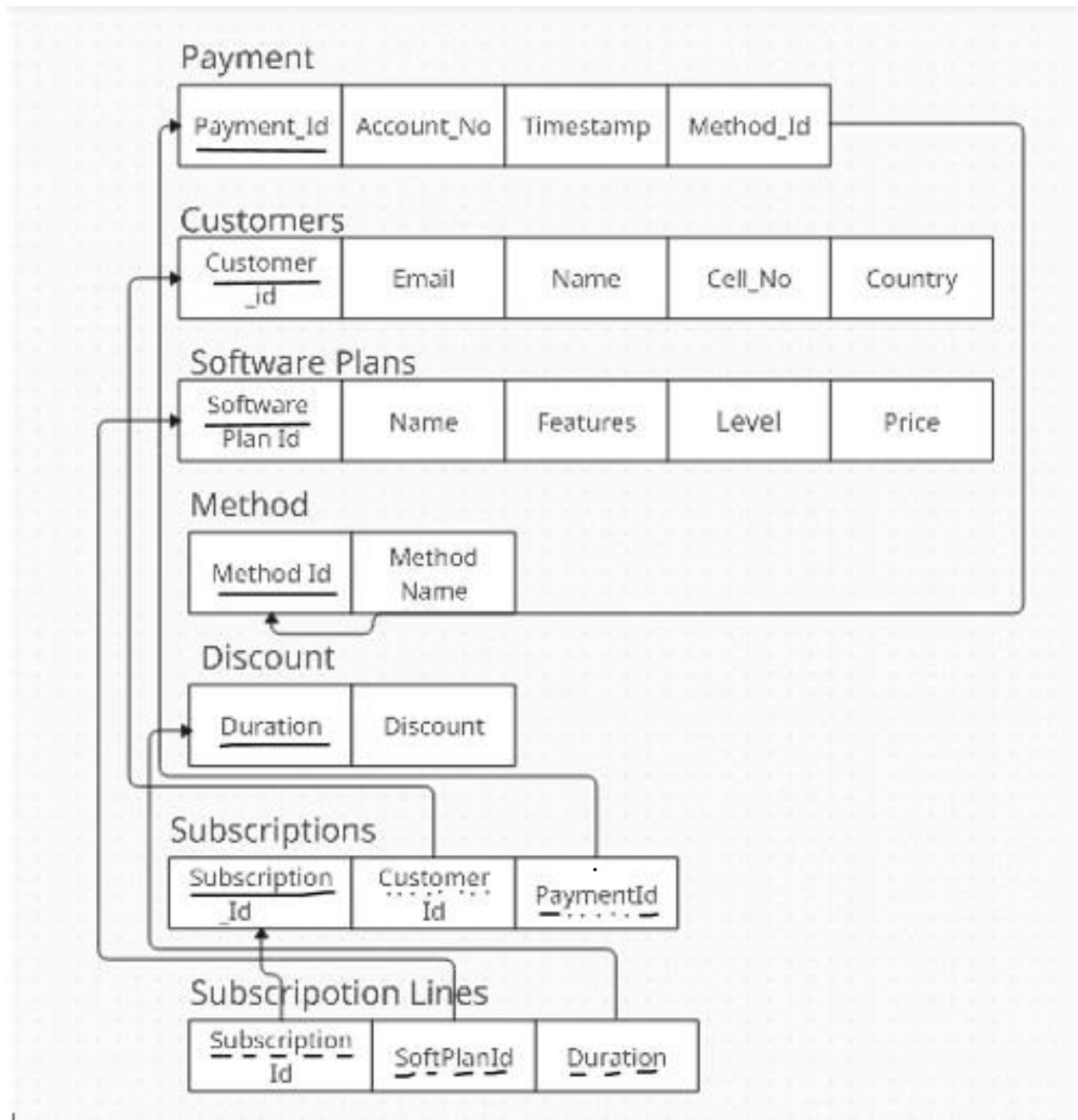
Composite Usage Model



Entity Relation Model



Relational Model:



Normalization:

There are no multiple values, redundancy or partial functional dependencies or transitive dependencies so the model is already in 3NF.

Physical DataBase Design:

MetaData :

1. Customer Table:

Column Name	Data Type	Description	Nullable	Length (in Bytes)
Customer_Id	VARCHAR(10)	Unique identifier for the customer. (Primary Key)	NO	10
Email	VARCHAR(50)	Email address of the customer.	NO	50
Name	VARCHAR(50)	Full name of the customer.	NO	50
Cell_NO	VARCHAR(11)	Contact number of the customer.	NO	11
Country	VARCHAR(2)	Country code (e.g., 'US' for the United States).	NO	2

2. Software_Plan Table:

Column Name	Data Type	Description	Nullable	Length (in Bytes)
Software_Plan_Id	VARCHAR(10)	Unique identifier for the software plan. (Primary Key)	NO	10
Software_Name	VARCHAR(20)	Name of the software plan.	NO	20
Software_Features	VARCHAR(120)	Features of the software plan.	NO	120
Software_Level	VARCHAR(20)	The level of the plan (e.g., Basic, Premium).	NO	20
Price_Per_Month	INTEGER	Monthly price of the software plan.	NO	4

3. Method Table:

Column Name	Data Type	Description	Nullable	Length (in Bytes)
Method_Id	INTEGER	Unique identifier for the payment method. (Primary Key)	NO	4
Method_Name	VARCHAR(20)	Name of the payment method (e.g., Credit Card, PayPal).	NO	20

4. Discount_On_Duration Table:

Column Name	Data Type	Description	Nullable	Length (in Bytes)
Duration	INTEGER	Duration in months or years for the discount eligibility. (Primary Key)	NO	4
Percent_Discount	INTEGER	Discount percentage applied based on the duration.	NO	4

5. Payments Table:

Column Name	Data Type	Description	Nullable	Length (in Bytes)
Payment_Id	INTEGER	Unique identifier for each payment. (Primary Key)	NO	4
Account_No	VARCHAR(20)	Account number associated with the payment.	NO	20
Payment_Timestamp	TIMESTAMP	Date and time when the payment was made.	NO	7 (variable)
Method_Id	INTEGER	Foreign Key linking to the Method table.	NO	4

6. Subscriptions Table:

Column Name	Data Type	Description	Nullable	Length (in Bytes)
Subscription_Id	INTEGER	Unique identifier for each subscription. (Primary Key)	NO	4
Customer_Id	VARCHAR(10)	Foreign Key linking to the Customer table.	NO	10
Payment_Id	INTEGER	Foreign Key linking to the Payments table.	NO	4

7. Subscription_Lines Table:

Column Name	Data Type	Description	Nullable	Length (in Bytes)
Subscription_Id	INTEGER	Foreign Key linking to the Subscriptions table.	NO	4
Software_Plan_Id	VARCHAR(10)	Foreign Key linking to the Software_Plan table.	NO	10
Duration	INTEGER	The duration of the subscription for this plan.	NO	4

Creating base tables:

```

CREATE TABLE Customer (
  Customer_Id VARCHAR(10) NOT NULL,
  Email VARCHAR(50) NOT NULL,
  Name VARCHAR(50) NOT NULL,
  Cell_NO VARCHAR(11) NOT NULL,
  Country VARCHAR(2) NOT NULL,
  CONSTRAINT Customer_PK PRIMARY KEY (Customer_Id)
);

CREATE TABLE Software_Plan (
  Software_Plan_Id VARCHAR(10) NOT NULL,
  Software_Name VARCHAR(20) NOT NULL,
  Software_Features VARCHAR(120) NOT NULL,
  Software_Level VARCHAR(20) NOT NULL,
  Price_Per_Month INTEGER NOT NULL,
  CONSTRAINT Software_PK PRIMARY KEY (Software_Plan_Id)
);

CREATE TABLE Method (
  Method_Id INTEGER NOT NULL,
  Method_Name VARCHAR(20) NOT NULL,
  CONSTRAINT Method_PK PRIMARY KEY (Method_Id)
);

CREATE TABLE Discount_On_Duration (

```



```

Duration          INTEGER    NOT NULL,
Percent_Discount  INTEGER    NOT NULL,
CONSTRAINT Discount_PK PRIMARY KEY (Duration)
);
CREATE TABLE Payments (
Payment_Id        INTEGER    NOT NULL,
Account_No        VARCHAR(20) NOT NULL, Payment_Timestamp  TIMESTAMP  NOT NULL,
Method_Id         INTEGER    NOT NULL,
CONSTRAINT Payment_PK PRIMARY KEY (Payment_Id),
CONSTRAINT Payment_FK2 FOREIGN KEY (Method_Id) REFERENCES Method(Method_Id)
);
CREATE TABLE Subscriptions (
Subscription_Id    INTEGER    NOT NULL,
Customer_Id        VARCHAR(10) NOT NULL,
Payment_Id         INTEGER    NOT NULL,
CONSTRAINT Subscription_PK PRIMARY KEY (Subscription_Id),
CONSTRAINT Subscription_FK1 FOREIGN KEY (Customer_Id) REFERENCES Customer(Customer_Id),
CONSTRAINT Subscription_FK2 FOREIGN KEY (Payment_Id) REFERENCES Payments(Payment_Id),  CONSTRAINT unique_payment UNIQUE
(Payment_Id)
);
CREATE TABLE Subscription_Lines(
Subscription_Id    INTEGER    NOT NULL,
Software_Plan_Id   VARCHAR(10) NOT NULL,
Duration           INTEGER    NOT NULL,
CONSTRAINT Subscription_Lines_PK PRIMARY KEY (Subscription_Id,Software_Plan_Id),
CONSTRAINT Subscription_Lines_FK1 FOREIGN KEY (Subscription_Id) REFERENCES Subscriptions(Subscription_Id),
CONSTRAINT Subscription_Lines_FK2 FOREIGN KEY (Software_Plan_Id) REFERENCES Software_Plan(Software_Plan_Id),
CONSTRAINT Subscription_Lines_FK3 FOREIGN KEY (Duration) REFERENCES Discount_On_Duration(Duration)
);
INSERT INTO Customer (Customer_Id, Email, Name, Cell_NO, Country) VALUES
('C001', 'john.doe@email.com', 'John Doe', '12345678901', 'US');
INSERT INTO Customer (Customer_Id, Email, Name, Cell_NO, Country) VALUES
('C002', 'jane.smith@email.com', 'Jane Smith', '12345678902', 'US');
INSERT INTO Customer (Customer_Id, Email, Name, Cell_NO, Country) VALUES
('C003', 'mike.jones@email.com', 'Mike Jones', '12345678903', 'CA');
INSERT INTO Customer (Customer_Id, Email, Name, Cell_NO, Country) VALUES
('C004', 'emily.davis@email.com', 'Emily Davis', '12345678904', 'CA');
INSERT INTO Customer (Customer_Id, Email, Name, Cell_NO, Country) VALUES
('C005', 'alex.brown@email.com', 'Alex Brown', '12345678905', 'UK');
INSERT INTO Customer (Customer_Id, Email, Name, Cell_NO, Country) VALUES
('C006', 'susan.miller@email.com', 'Susan Miller', '12345678906', 'UK');
INSERT INTO Customer (Customer_Id, Email, Name, Cell_NO, Country) VALUES
('C007', 'william.wilson@email.com', 'William Wilson', '12345678907', 'US');
INSERT INTO Customer (Customer_Id, Email, Name, Cell_NO, Country) VALUES
('C008', 'linda.moore@email.com', 'Linda Moore', '12345678908', 'US');
INSERT INTO Customer (Customer_Id, Email, Name, Cell_NO, Country) VALUES
('C009', 'patricia.johnson@email.com', 'Patricia Johnson', '12345678909', 'AU');
INSERT INTO Customer (Customer_Id, Email, Name, Cell_NO, Country) VALUES
('C010', 'james.white@email.com', 'James White', '12345678910', 'AU');
INSERT INTO Customer (Customer_Id, Email, Name, Cell_NO, Country) VALUES
('C011', 'robert.garcia@email.com', 'Robert Garcia', '12345678911', 'IN');
INSERT INTO Customer (Customer_Id, Email, Name, Cell_NO, Country) VALUES
('C012', 'mary.martin@email.com', 'Mary Martin', '12345678912', 'IN');
INSERT INTO Customer (Customer_Id, Email, Name, Cell_NO, Country) VALUES
('C013', 'charles.lewis@email.com', 'Charles Lewis', '12345678913', 'US');
INSERT INTO Customer (Customer_Id, Email, Name, Cell_NO, Country) VALUES
('C014', 'sophia.walker@email.com', 'Sophia Walker', '12345678914', 'US');
INSERT INTO Customer (Customer_Id, Email, Name, Cell_NO, Country) VALUES
('C015', 'george.hall@email.com', 'George Hall', '12345678915', 'CA');
INSERT INTO Customer (Customer_Id, Email, Name, Cell_NO, Country) VALUES
('C016', 'barbara.allen@email.com', 'Barbara Allen', '12345678916', 'CA');
INSERT INTO Customer (Customer_Id, Email, Name, Cell_NO, Country) VALUES
('C017', 'david.young@email.com', 'David Young', '12345678917', 'UK');
INSERT INTO Customer (Customer_Id, Email, Name, Cell_NO, Country) VALUES
('C018', 'elizabeth.king@email.com', 'Elizabeth King', '12345678918', 'UK');
INSERT INTO Customer (Customer_Id, Email, Name, Cell_NO, Country) VALUES

```

```
('C019', 'joseph.clark@email.com', 'Joseph Clark', '12345678919', 'US');
INSERT INTO Customer (Customer_Id, Email, Name, Cell_NO, Country) VALUES
('C020', 'nancy.harris@email.com', 'Nancy Harris', '12345678920', 'US');
```

```
INSERT INTO Software_Plan (Software_Plan_Id, Software_Name, Software_Features, Software_Level, Price_Per_Month) VALUES
('SP001', 'Microsoft Office 365', 'Word, Excel, PowerPoint, Outlook, OneDrive', 'Basic', 10);
INSERT INTO Software_Plan (Software_Plan_Id, Software_Name, Software_Features, Software_Level, Price_Per_Month) VALUES
('SP002', 'Microsoft Office 365', 'Word, Excel, PowerPoint, Outlook, OneDrive, Access, Teams', 'Intermediate', 20);
INSERT INTO Software_Plan (Software_Plan_Id, Software_Name, Software_Features, Software_Level, Price_Per_Month) VALUES
('SP003', 'Microsoft Office 365', 'Word, Excel, PowerPoint, Outlook, OneDrive, Access, Teams, Publisher, Skype', 'Premium', 30);
INSERT INTO Software_Plan (Software_Plan_Id, Software_Name, Software_Features, Software_Level, Price_Per_Month) VALUES
('SP004', 'Adobe Creative Cloud', 'Photoshop, Lightroom, Acrobat, Illustrator', 'Basic', 15);
INSERT INTO Software_Plan (Software_Plan_Id, Software_Name, Software_Features, Software_Level, Price_Per_Month) VALUES
('SP005', 'Adobe Creative Cloud', 'Photoshop, Lightroom, Acrobat, Illustrator, InDesign', 'Intermediate', 25);
INSERT INTO Software_Plan (Software_Plan_Id, Software_Name, Software_Features, Software_Level, Price_Per_Month) VALUES
('SP006', 'Adobe Creative Cloud', 'Photoshop, Lightroom, Acrobat, Illustrator, InDesign, After Effects, Premiere Pro', 'Premium', 35);
INSERT INTO Software_Plan (Software_Plan_Id, Software_Name, Software_Features, Software_Level, Price_Per_Month) VALUES
('SP007', 'Zoom', 'Video Conferencing, Screen Sharing, Virtual Backgrounds', 'Basic', 12);
INSERT INTO Software_Plan (Software_Plan_Id, Software_Name, Software_Features, Software_Level, Price_Per_Month) VALUES
('SP008', 'Zoom', 'Video Conferencing, Screen Sharing, Virtual Backgrounds, Webinar Hosting', 'Intermediate', 22);
INSERT INTO Software_Plan (Software_Plan_Id, Software_Name, Software_Features, Software_Level, Price_Per_Month) VALUES
('SP009', 'Zoom', 'Video Conferencing, Screen Sharing, Virtual Backgrounds, Webinar Hosting, Cloud Recording', 'Premium', 32);
INSERT INTO Software_Plan (Software_Plan_Id, Software_Name, Software_Features, Software_Level, Price_Per_Month) VALUES
('SP010', 'Slack', 'Messaging, File Sharing, Team Collaboration', 'Basic', 18);
INSERT INTO Software_Plan (Software_Plan_Id, Software_Name, Software_Features, Software_Level, Price_Per_Month) VALUES
('SP011', 'Slack', 'Messaging, File Sharing, Team Collaboration, Video Calls', 'Intermediate', 28);
INSERT INTO Software_Plan (Software_Plan_Id, Software_Name, Software_Features, Software_Level, Price_Per_Month) VALUES
('SP012', 'Slack', 'Messaging, File Sharing, Team Collaboration, Video Calls, Workflow Automation, Analytics', 'Premium', 38);
INSERT INTO Software_Plan (Software_Plan_Id, Software_Name, Software_Features, Software_Level, Price_Per_Month) VALUES
('SP013', 'Spotify', 'Music Streaming, Playlists, Offline Mode', 'Basic', 14);
INSERT INTO Software_Plan (Software_Plan_Id, Software_Name, Software_Features, Software_Level, Price_Per_Month) VALUES
('SP014', 'Spotify', 'Music Streaming, Playlists, Offline Mode, Podcasts', 'Intermediate', 24);
INSERT INTO Software_Plan (Software_Plan_Id, Software_Name, Software_Features, Software_Level, Price_Per_Month) VALUES
('SP015', 'Spotify', 'Music Streaming, Playlists, Offline Mode, Podcasts, High-Quality Audio, Family Plan', 'Premium', 34);
```

```
INSERT INTO Method (Method_Id, Method_Name) VALUES
(1, 'Credit Card');
INSERT INTO Method (Method_Id, Method_Name) VALUES
(2, 'Debit Card');
INSERT INTO Method (Method_Id, Method_Name) VALUES
(3, 'PayPal');
INSERT INTO Method (Method_Id, Method_Name) VALUES
(4, 'Bank Transfer');
```

```
INSERT INTO Discount_On_Duration (Duration, Percent_Discount) VALUES
(1, 0);
INSERT INTO Discount_On_Duration (Duration, Percent_Discount) VALUES
(3, 10);
INSERT INTO Discount_On_Duration (Duration, Percent_Discount) VALUES
(6, 15);
INSERT INTO Discount_On_Duration (Duration, Percent_Discount) VALUES
(12, 25);
```

```
INSERT INTO Payments (Payment_Id, Account_No, Payment_Timestamp, Method_Id) VALUES
(1, 'ACC001', TIMESTAMP '2024-12-01 10:00:00', 1);
INSERT INTO Payments (Payment_Id, Account_No, Payment_Timestamp, Method_Id) VALUES
(2, 'ACC002', TIMESTAMP '2024-12-01 11:00:00', 2);
INSERT INTO Payments (Payment_Id, Account_No, Payment_Timestamp, Method_Id) VALUES
(3, 'ACC003', TIMESTAMP '2024-12-01 12:00:00', 3);
INSERT INTO Payments (Payment_Id, Account_No, Payment_Timestamp, Method_Id) VALUES
(4, 'ACC004', TIMESTAMP '2024-12-01 13:00:00', 2);
INSERT INTO Payments (Payment_Id, Account_No, Payment_Timestamp, Method_Id) VALUES
(5, 'ACC005', TIMESTAMP '2024-12-01 14:00:00', 1);
INSERT INTO Payments (Payment_Id, Account_No, Payment_Timestamp, Method_Id) VALUES
(6, 'ACC006', TIMESTAMP '2024-12-02 10:00:00', 2);
INSERT INTO Payments (Payment_Id, Account_No, Payment_Timestamp, Method_Id) VALUES
```



```

(7, 'ACC007', TIMESTAMP '2024-12-02 11:00:00', 3);
INSERT INTO Payments (Payment_Id, Account_No, Payment_Timestamp, Method_Id) VALUES
(8, 'ACC008', TIMESTAMP '2024-12-02 12:00:00', 3);
INSERT INTO Payments (Payment_Id, Account_No, Payment_Timestamp, Method_Id) VALUES
(9, 'ACC009', TIMESTAMP '2024-12-02 13:00:00', 1);
INSERT INTO Payments (Payment_Id, Account_No, Payment_Timestamp, Method_Id) VALUES
(10, 'ACC010', TIMESTAMP '2024-12-02 14:00:00', 2);
INSERT INTO Payments (Payment_Id, Account_No, Payment_Timestamp, Method_Id) VALUES
(11, 'ACC011', TIMESTAMP '2024-12-03 10:00:00', 3);
INSERT INTO Payments (Payment_Id, Account_No, Payment_Timestamp, Method_Id) VALUES
(12, 'ACC012', TIMESTAMP '2024-12-03 11:00:00', 4);
INSERT INTO Payments (Payment_Id, Account_No, Payment_Timestamp, Method_Id) VALUES
(13, 'ACC013', TIMESTAMP '2024-12-03 12:00:00', 3);
INSERT INTO Payments (Payment_Id, Account_No, Payment_Timestamp, Method_Id) VALUES
(14, 'ACC014', TIMESTAMP '2024-12-03 13:00:00', 2);
INSERT INTO Payments (Payment_Id, Account_No, Payment_Timestamp, Method_Id) VALUES
(15, 'ACC015', TIMESTAMP '2024-12-03 14:00:00', 3);
INSERT INTO Payments (Payment_Id, Account_No, Payment_Timestamp, Method_Id) VALUES
(16, 'ACC016', TIMESTAMP '2024-12-04 10:00:00', 4);
INSERT INTO Payments (Payment_Id, Account_No, Payment_Timestamp, Method_Id) VALUES
(17, 'ACC017', TIMESTAMP '2024-12-04 11:00:00', 1);
INSERT INTO Payments (Payment_Id, Account_No, Payment_Timestamp, Method_Id) VALUES
(18, 'ACC018', TIMESTAMP '2024-12-04 12:00:00', 2);
INSERT INTO Payments (Payment_Id, Account_No, Payment_Timestamp, Method_Id) VALUES
(19, 'ACC019', TIMESTAMP '2024-12-04 13:00:00', 3);
INSERT INTO Payments (Payment_Id, Account_No, Payment_Timestamp, Method_Id) VALUES
(20, 'ACC020', TIMESTAMP '2024-12-04 14:00:00', 3);

```

INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Payment_Id) VALUES

```

(1, 'C001', 1);
INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Payment_Id) VALUES
(2, 'C002', 2);
INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Payment_Id) VALUES
(3, 'C003', 3);
INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Payment_Id) VALUES
(4, 'C001', 4);
INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Payment_Id) VALUES
(5, 'C009', 5);
INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Payment_Id) VALUES
(6, 'C006', 6);
INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Payment_Id) VALUES
(7, 'C007', 7);
INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Payment_Id) VALUES
(8, 'C008', 8);
INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Payment_Id) VALUES
(9, 'C009', 9);
INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Payment_Id) VALUES
(10, 'C010', 10);
INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Payment_Id) VALUES
(11, 'C016', 11);
INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Payment_Id) VALUES
(12, 'C012', 12);
INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Payment_Id) VALUES
(13, 'C012', 13);
INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Payment_Id) VALUES
(14, 'C014', 14);
INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Payment_Id) VALUES
(15, 'C015', 15);
INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Payment_Id) VALUES
(16, 'C016', 16);
INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Payment_Id) VALUES
(17, 'C017', 17);
INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Payment_Id) VALUES
(18, 'C018', 18);
INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Payment_Id) VALUES
(19, 'C019', 19);

```

```
INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Payment_Id) VALUES
(20, 'C016', 20);
```

INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES

```
(1, 'SP001', 1);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(1, 'SP004', 3);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(1, 'SP007', 6);
(1, 'SP010', 12);
(2, 'SP005', 1);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(2, 'SP010', 3);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(3, 'SP007', 6);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(3, 'SP011', 1);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(4, 'SP010', 3);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(4, 'SP001', 12);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(5, 'SP013', 1);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(5, 'SP010', 3);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(5, 'SP001', 6);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(6, 'SP001', 12);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(6, 'SP015', 1);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(6, 'SP011', 3);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(7, 'SP004', 6);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(7, 'SP015', 12);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(7, 'SP011', 1);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(8, 'SP007', 3);
(9, 'SP010', 1);
(10, 'SP013', 12);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(11, 'SP001', 6);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(12, 'SP004', 3);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(12, 'SP014', 6);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(13, 'SP007', 1);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(14, 'SP010', 12);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(15, 'SP013', 6);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(15, 'SP001', 12);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(15, 'SP014', 1);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(16, 'SP015', 3);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(16, 'SP001', 6);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(16, 'SP009', 12);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
```


(17, 'SP004', 1);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(17, 'SP010', 3);
INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration) VALUES
(17, 'SP014', 6)

Customer Table:

CUSTOMER_ID	EMAIL	NAME	CELL_NO	COUNTRY
C001	john.doe@email.com	John Doe	12345678901	US
C002	jane.smith@email.com	Jane Smith	12345678902	US
C003	mike.jones@email.com	Mike Jones	12345678903	CA
C004	emily.davis@email.com	Emily Davis	12345678904	CA
C005	alex.brown@email.com	Alex Brown	12345678905	UK
C006	susan.miller@email.com	Susan Miller	12345678906	UK
C007	william.wilson@email.com	William Wilson	12345678907	US
C008	linda.moore@email.com	Linda Moore	12345678908	US
C009	patricia.johnson@email.com	Patricia Johnson	12345678909	AU
C010	james.white@email.com	James White	12345678910	AU
C011	robert.garcia@email.com	Robert Garcia	12345678911	IN
C012	mary.martin@email.com	Mary Martin	12345678912	IN
C013	charles.lewis@email.com	Charles Lewis	12345678913	US
C014	sophia.walker@email.com	Sophia Walker	12345678914	US
C015	george.hall@email.com	George Hall	12345678915	CA
C016	barbara.allen@email.com	Barbara Allen	12345678916	CA
C017	david.young@email.com	David Young	12345678917	UK
C018	elizabeth.king@email.com	Elizabeth King	12345678918	UK
C019	joseph.clark@email.com	Joseph Clark	12345678919	US
C020	nancy.harris@email.com	Nancy Harris	12345678920	US

Software Plan Table:

SOFTWARE_PLAN_ID	SOFTWARE_NAME	SOFTWARE_FEATURES	SOFTWARE_LEVEL	PRICE_PER_MONTH
SP001	Microsoft Office 365	Word, Excel, PowerPoint, Outlook, OneDrive	Basic	10
SP002	Microsoft Office 365	Word, Excel, PowerPoint, Outlook, OneDrive, Access, Teams	Intermediate	20
SP003	Microsoft Office 365	Word, Excel, PowerPoint, Outlook, OneDrive, Access, Teams, Publisher, Skype	Premium	30
SP004	Adobe Creative Cloud	Photoshop, Lightroom, Acrobat, Illustrator	Basic	15
SP005	Adobe Creative Cloud	Photoshop, Lightroom, Acrobat, Illustrator, InDesign	Intermediate	25
SP006	Adobe Creative Cloud	Photoshop, Lightroom, Acrobat, Illustrator, InDesign, After Effects, Premiere Pro	Premium	35
SP007	Zoom	Video Conferencing, Screen Sharing, Virtual Backgrounds	Basic	12
SP008	Zoom	Video Conferencing, Screen Sharing, Virtual Backgrounds, Webinar Hosting	Intermediate	22
SP009	Zoom	Video Conferencing, Screen Sharing, Virtual Backgrounds, Webinar Hosting, Cloud Recording	Premium	32
SP010	Slack	Messaging, File Sharing, Team Collaboration	Basic	18
SP011	Slack	Messaging, File Sharing, Team Collaboration, Video Calls	Intermediate	28
SP012	Slack	Messaging, File Sharing, Team Collaboration, Video Calls, Workflow Automation, Analytics	Premium	38
SP013	Spotify	Music Streaming, Playlists, Offline Mode	Basic	14
SP014	Spotify	Music Streaming, Playlists, Offline Mode, Podcasts	Intermediate	24
SP015	Spotify	Music Streaming, Playlists, Offline Mode, Podcasts, High-Quality Audio, Family Plan	Premium	34

METHOD_ID	METHOD_NAME
1	Credit Card
2	Debit Card
3	PayPal
4	Bank Transfer

Method Table:

DURATION	PERCENT_DISCOUNT
1	0
3	10
6	15
12	25

Discount Table:

Subscriptions Table:

SUBSCRIPTION_ID	CUSTOMER_ID	PAYMENT_ID
1	C001	1
2	C002	2
3	C003	3
4	C001	4
5	C009	5
6	C006	6
7	C007	7
8	C008	8
9	C009	9
10	C010	10
11	C016	11

12	C012	12
13	C012	13
14	C014	14
15	C015	15
16	C016	16
17	C017	17
18	C018	18
19	C019	19
20	C016	20

Subscription_lines Table:

SUBSCRIPTION_ID	SOFTWARE_PLAN_ID	DURATION
1	SP001	1
1	SP004	3
1	SP007	6
1	SP010	12
2	SP005	1
2	SP010	3
3	SP007	6
3	SP011	1
4	SP010	3
4	SP001	12
5	SP013	1
5	SP010	3
5	SP001	6
6	SP001	12
6	SP015	1
6	SP011	3
7	SP004	6
7	SP015	12
7	SP011	1
8	SP007	3

Queries:

1_Revenue generated by each customer in Descending Order:

```
SELECT
    sl.subscription_id,
    ROUND(
        SUM(sp.price_per_month * sl.duration - (sp.price_per_month * sl.duration * COALESCE(d.percent_discount, 0) / 100))
        , 2) AS total_price
FROM
    subscription_lines sl
JOIN
    subscriptions sub
ON
    sl.subscription_id = sub.subscription_id
JOIN
    software_plan sp
ON
    sl.software_plan_id = sp.software_plan_id
LEFT JOIN
    discount_on_duration d
ON
    sl.duration = d.duration
GROUP BY
    sl.subscription_id
ORDER BY Total_Price DESC ;
```

SUBSCRIPTION_ID	TOTAL_PRICE
16	430.8
7	410.5
1	273.7
6	199.6
17	186
15	185.4
12	162.9
14	162
4	138.6
10	126
5	113.6
18	108
3	89.2
2	73.6
11	51
19	38
20	37.8
8	32.4
9	18
13	12

2_Most Popular Payment Methods:

METHOD_NAME	USAGE_COUNT
PayPal	8
Debit Card	6
Credit Card	4
Bank Transfer	2

```
SELECT
    m.method_name,
    COUNT(p.method_id) AS usage_count
FROM
    payments p
JOIN
    method m
ON
    p.method_id = m.method_id
GROUP BY
    m.method_name
ORDER BY
    usage_count DESC;
```

3_Customer who have not subscribed to any plan yet.

CUSTOMER_ID	EMAIL	COUNTRY	NAME
C020	nancy.harris@email.com	US	Nancy Harris
C013	charles.lewis@email.com	US	Charles Lewis
C004	emily.davis@email.com	CA	Emily Davis
C005	alex.brown@email.com	UK	Alex Brown
C011	robert.garcia@email.com	IN	Robert Garcia

```
SELECT
    c.customer_id,
    c.email,
    c.country,
    c.name
FROM
    customer c
LEFT JOIN
    subscriptions s
ON
    c.customer_id = s.customer_id
WHERE
    s.subscription_id IS NULL;
```

4_Finding Information about Customer with ID C001:

```
SELECT
  c.customer_id,c.name, c.email,s.subscription_id,sp.software_name,sp.price_per_month,sl.duration,
  ROUND(sp.price_per_month * sl.duration - (sp.price_per_month * sl.duration * COALESCE(d.percent_discount, 0) / 100), 2) AS total_price,
  p.payment_id,
  p.method_id,
  p.Account_No,
  p.Payment_Timestamp
FROM
  customer c
JOIN
  subscriptions s ON c.customer_id = s.customer_id
JOIN
  subscription_lines sl ON s.subscription_id = sl.subscription_id
JOIN
  software_plan sp ON sl.software_plan_id = sp.software_plan_id
LEFT JOIN
  discount_on_duration d ON sl.duration = d.duration
JOIN
  payments p ON s.payment_id = p.payment_id
WHERE S.CUSTOMER_ID = 'C001'
ORDER BY
  c.customer_id, s.subscription_id;
```

CUSTOMER_ID	NAME	EMAIL	SUBSCRIPTION_ID	SOFTWARE_NAME	PRICE_PER_MONTH	DURATION	TOTAL_PRICE	PAYMENT_ID	METHOD_ID	ACCOUNT_NO	PAYMENT_TIMESTAMP
C001	John Doe	john.doe@email.com	1	Adobe Creative Cloud	15	3	40.5	1	1	ACC001	01-DEC-24 10:00:00.000000 AM
C001	John Doe	john.doe@email.com	1	Slack	18	12	162	1	1	ACC001	01-DEC-24 10:00:00.000000 AM
C001	John Doe	john.doe@email.com	1	Microsoft Office 365	10	1	10	1	1	ACC001	01-DEC-24 10:00:00.000000 AM
C001	John Doe	john.doe@email.com	1	Zoom	12	6	61.2	1	1	ACC001	01-DEC-24 10:00:00.000000 AM
C001	John Doe	john.doe@email.com	4	Slack	18	3	48.6	4	2	ACC004	01-DEC-24 01:00:00.000000 PM
C001	John Doe	john.doe@email.com	4	Microsoft Office 365	10	12	90	4	2	ACC004	01-DEC-24 01:00:00.000000 PM

5_Information For All Customers:

```
SELECT
  c.customer_id,c.name, c.email,s.subscription_id,sp.software_name,sp.price_per_month,sl.duration,
  ROUND(sp.price_per_month * sl.duration - (sp.price_per_month * sl.duration * COALESCE(d.percent_discount, 0) / 100), 2) AS total_price,
  p.payment_id,
  p.method_id,
  p.Account_No,
  p.Payment_Timestamp
FROM
  customer c
JOIN
  subscriptions s ON c.customer_id = s.customer_id
JOIN
  subscription_lines sl ON s.subscription_id = sl.subscription_id
JOIN
  software_plan sp ON sl.software_plan_id = sp.software_plan_id
LEFT JOIN
  discount_on_duration d ON sl.duration = d.duration
JOIN
  payments p ON s.payment_id = p.payment_id
ORDER BY
  c.customer_id, s.subscription_id;
```

CUSTOMER_ID	NAME	EMAIL	SUBSCRIPTION_ID	SOFTWARE_NAME	PRICE_PER_MONTH	DURATION	TOTAL_PRICE	PAYMENT_ID	METHOD_ID	ACCOUNT_NO	PAYMENT_TIMESTAMP
C001	John Doe	john.doe@email.com	1	Zoom	12	6	61.2	1	1	ACC001	01-DEC-24 10:00:00.000000 AM
C001	John Doe	john.doe@email.com	1	Slack	18	12	162	1	1	ACC001	01-DEC-24 10:00:00.000000 AM
C001	John Doe	john.doe@email.com	1	Microsoft Office 365	10	1	10	1	1	ACC001	01-DEC-24 10:00:00.000000 AM
C001	John Doe	john.doe@email.com	1	Adobe Creative Cloud	15	3	40.5	1	1	ACC001	01-DEC-24 10:00:00.000000 AM
C001	John Doe	john.doe@email.com	4	Slack	18	3	48.6	4	2	ACC004	01-DEC-24 01:00:00.000000 PM
C001	John Doe	john.doe@email.com	4	Microsoft Office 365	10	12	90	4	2	ACC004	01-DEC-24 01:00:00.000000 PM
C002	Jane Smith	jane.smith@email.com	2	Slack	18	3	48.6	2	2	ACC002	01-DEC-24 11:00:00.000000 AM
C002	Jane Smith	jane.smith@email.com	2	Adobe Creative Cloud	25	1	25	2	2	ACC002	01-DEC-24 11:00:00.000000 AM
C003	Mike Jones	mike.jones@email.com	3	Slack	28	1	28	3	3	ACC003	01-DEC-24 12:00:00.000000 PM
C003	Mike Jones	mike.jones@email.com	3	Zoom	12	6	61.2	3	3	ACC003	01-DEC-24 12:00:00.000000 PM
C006	Susan Miller	susan.miller@email.com	6	Spotify	34	1	34	6	2	ACC006	02-DEC-24 10:00:00.000000 AM
C006	Susan Miller	susan.miller@email.com	6	Slack	28	3	75.6	6	2	ACC006	02-DEC-24 10:00:00.000000 AM
C006	Susan Miller	susan.miller@email.com	6	Microsoft Office 365	10	12	90	6	2	ACC006	02-DEC-24 10:00:00.000000 AM
C007	William Wilson	william.wilson@email.com	7	Adobe Creative Cloud	15	6	76.5	7	3	ACC007	02-DEC-24 11:00:00.000000 AM
C007	William Wilson	william.wilson@email.com	7	Slack	28	1	28	7	3	ACC007	02-DEC-24 11:00:00.000000 AM
C007	William Wilson	william.wilson@email.com	7	Spotify	34	12	306	7	3	ACC007	02-DEC-24 11:00:00.000000 AM
C008	Linda Moore	linda.moore@email.com	8	Zoom	12	3	32.4	8	3	ACC008	02-DEC-24 12:00:00.000000 PM
C009	Patricia Johnson	patricia.johnson@email.com	5	Spotify	14	1	14	5	1	ACC005	01-DEC-24 02:00:00.000000 PM
C009	Patricia Johnson	patricia.johnson@email.com	5	Microsoft Office 365	10	6	61	5	1	ACC005	01-DEC-24 02:00:00.000000 PM

6_Finding Popularity of Softwares:

```
SELECT
    sp.software_name,
    COUNT(sl.subscription_id) AS subscription_count
FROM
    subscription_lines sl
JOIN
    software_plan sp ON sl.software_plan_id = sp.software_plan_id
GROUP BY
    sp.software_name
ORDER BY
    subscription_count DESC;
```

SOFTWARE_NAME	SUBSCRIPTION_COUNT
Slack	11
Spotify	10
Microsoft Office 365	7
Zoom	6
Adobe Creative Cloud	5

7_Finding Most popular Software Level:

```
SELECT
    sp.software_level,
    COUNT(sl.subscription_id) AS subscription_count
FROM
    subscription_lines sl
JOIN
    software_plan sp ON sl.software_plan_id = sp.software_plan_id
GROUP BY
    sp.software_level
ORDER BY
    subscription_count DESC;
```

SOFTWARE_LEVEL	SUBSCRIPTION_COUNT
Basic	27
Intermediate	7
Premium	5

8_Most Popular Software programs by country:

COUNTRY	SOFTWARE_NAME	PLAN_COUNT
AU	Slack	2
AU	Spotify	2
AU	Microsoft Office 365	1
CA	Spotify	4
CA	Microsoft Office 365	3
CA	Zoom	2
CA	Slack	1
IN	Spotify	1
IN	Adobe Creative Cloud	1
IN	Zoom	1
UK	Slack	2
UK	Spotify	2
UK	Zoom	1
UK	Microsoft Office 365	1
UK	Adobe Creative Cloud	1
US	Slack	6
US	Adobe Creative Cloud	3
US	Microsoft Office 365	2
US	Zoom	2
US	Spotify	1

```

SELECT
    c.country,
    sp.software_name,
    COUNT(sl.subscription_id) AS plan_count
FROM
    customer c
JOIN
    subscriptions s ON c.customer_id = s.customer_id
JOIN
    subscription_lines sl ON s.subscription_id = sl.subscription_id
JOIN
    software_plan sp ON sl.software_plan_id = sp.software_plan_id
GROUP BY
    c.country, sp.software_name
ORDER BY
    c.country, plan_count DESC;

```

9_Total revenue generated by each Software:

```

SELECT
    sp.software_name,
    SUM(sp.price_per_month * sl.duration - (sp.price_per_month * sl.duration * COALESCE(d.percent_discount, 0) / 100)) AS total_revenue
FROM
    subscription_lines sl
JOIN
    software_plan sp ON sl.software_plan_id = sp.software_plan_id
LEFT JOIN
    discount_on_duration d ON sl.duration = d.duration
GROUP BY
    sp.software_name
ORDER BY
    total_revenue DESC;

```

SOFTWARE_NAME	TOTAL_REVENUE
Spotify	949.8
Slack	706
Zoom	562.8
Microsoft Office 365	433
Adobe Creative Cloud	197.5

10_No of Customers in Each State:

COUNTRY	NUMBER_OF_CUSTOMERS
US	8
CA	4
UK	4
AU	2
IN	2

```

SELECT
    country,
    COUNT(customer_id) AS number_of_customers
FROM
    customer
GROUP BY
    country
ORDER BY
    number_of_customers DESC;

```

Creating TIGGER:

```
CREATE OR REPLACE TRIGGER Prevent_Duplicate_Subscription
BEFORE INSERT ON Subscriptions
FOR EACH ROW
DECLARE
    v_count NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO v_count
    FROM Subscriptions
    WHERE Customer_Id = :NEW.Customer_Id AND Software_Plan_Id = :NEW.Software_Plan_Id;

    IF v_count > 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Duplicate subscription is not allowed.');
```

```
END IF;
END;
```

Results Explain Describe Saved SQL History

Trigger created.

```
(19, 16, 18, 6, 'ACC678902', TO_TIMESTAMP('2025-02-28 15:45:00', 'YYYY-MM-DD HH24:MI:SS'), 1);
INSERT INTO Subscriptions (Subscription_Id, Customer_Id, Software_Plan_Id, Duration, Account_No, Payment_Timestamp, Method_Id)
VALUES (
(21, 1, 1, 6, 'ACC678902', TO_TIMESTAMP('2025-02-28 15:45:00', 'YYYY-MM-DD HH24:MI:SS'), 2);
```

```
SELECT * FROM CUSTOMER;
SELECT * FROM SOFTWARE_PLAN ORDER BY SOFTWARE_PLAN_ID;
```

Results Explain Describe Saved SQL History

```
ORA-20001: Duplicate subscription is not allowed.
ORA-06512: at "SAAS.PREVENT_DUPLICATE_SUBSCRIPTION", line 10
ORA-04088: error during execution of trigger "SAAS.PREVENT_DUPLICATE_SUBSCRIPTION"
```

Transaction:

BEGIN;

-- 1. Insert a record into the Subscriptions table (customer subscribes)

INSERT INTO Subscriptions (Customer_Id, Payment_Id)

VALUES ('C001', 101);

-- 2. Insert a record into the Subscription_Lines table (linking the subscription to the software plan)

INSERT INTO Subscription_Lines (Subscription_Id, Software_Plan_Id, Duration)

VALUES (LAST_INSERT_ID(), 'SP001', 12);

-- 3. Insert a record into the Payments table (recording the payment made by the customer)

INSERT INTO Payments (Account_No, Payment_Timestamp, Method_Id)

VALUES ('ACC001', SYSDATE, 1);

-- If all steps succeed, commit the transaction

COMMIT;

-- If any error occurs during the transaction, rollback all changes

ROLLBACK;

Creating Users and Roles:

CREATE ROLE Admin;

CREATE ROLE Manager;

CREATE ROLE Customer;

CREATE ROLE Sales;

Grant All Permissions to Admin

Admins have full control over the database, including DDL and DML operations.

Grant All Privileges On All Tables To Admin;

Grant All Privileges On All Tables To Admin;

Grant All Privileges On All Sequences To Admin;

Grant Create, Drop, Alter On Schema To Admin; **Manager Role:**

Managers can manage subscriptions and view reports but cannot modify critical database structures.

Grant Select, Insert, Update On Table Subscriptions To Manager;

Grant Select On Table Payments, Customer, Software_Plan, Subscription_Lines To Manager;

Grant Usage, Select On Sequences Subscriptions_Subscription_Id_Seq To Manager;

Customer Role:

Customers can view their own subscriptions and payment details but cannot modify them.

Grant Select On Table Customer, Subscriptions, Subscription_Lines, Payments To Customer;

Sales Role:

Sales personnel can add and update customer information and create subscriptions but cannot delete data.

Grant Select, Insert, Update On Table Customer To Sales;

Grant Insert, Update On Table Subscriptions, Subscription_Lines To Sales;

Create Users and Assign Roles:

CREATE USER admin_user WITH PASSWORD 'AdminPass123';

CREATE USER manager_user WITH PASSWORD 'ManagerPass123';

CREATE USER customer_user WITH PASSWORD 'CustomerPass123';

CREATE USER sales_user WITH PASSWORD 'SalesPass123';

Revoke Role from a User:

REVOKE Manager FROM manager_user; **Size Estimation:**

Size per Table:

1. Customer Table:

- Customer_Id (VARCHAR(10)): 10 bytes
- Email (VARCHAR(50)): 50 bytes
- Name (VARCHAR(50)): 50 bytes
- Cell_NO (VARCHAR(11)): 11 bytes
- Country (VARCHAR(2)): 2 bytes

Total Size per Record:

10 + 50 + 50 + 11 + 2 = **123 bytes**

- Total records: 20 customers
- Total size: 123 bytes × 20 = 2,460 bytes

2. Software_Plan Table:

- Software_Plan_Id (VARCHAR(10)): 10 bytes

- Software_Name (VARCHAR(20)): 20 bytes
- Software_Features (VARCHAR(120)): 120 bytes
- Software_Level (VARCHAR(20)): 20 bytes
- Price_Per_Month (INTEGER): 4 bytes

Total Size per Record:

$10 + 20 + 120 + 20 + 4 = \mathbf{174 \text{ bytes}}$

- Total records: 20 software plans
- Total size: $174 \text{ bytes} \times 20 = 3,480 \text{ bytes}$

3. Method Table:

- Method_Id (INTEGER): 4 bytes
- Method_Name (VARCHAR(20)): 20 bytes

Total Size per Record:

$4 + 20 = \mathbf{24 \text{ bytes}}$

- Total records: 4 methods
- Total size: $24 \text{ bytes} \times 4 = 96 \text{ bytes}$

4. Discount_On_Duration Table:

- Duration (INTEGER): 4 bytes
- Percent_Discount (INTEGER): 4 bytes

Total Size per Record:

$4 + 4 = \mathbf{8 \text{ bytes}}$

- Total records: 4 discount durations
- Total size: $8 \text{ bytes} \times 4 = 32 \text{ bytes}$

5. Payments Table:

- Payment_Id (INTEGER): 4 bytes
- Account_No (VARCHAR(20)): 20 bytes
- Payment_Timestamp (TIMESTAMP): 7 bytes
- Method_Id (INTEGER): 4 bytes

Total Size per Record:

$4 + 20 + 7 + 4 = \mathbf{35 \text{ bytes}}$

- Total records: 20 payments
- Total size: $35 \text{ bytes} \times 20 = 700 \text{ bytes}$

6. Subscriptions Table:

- Subscription_Id (INTEGER): 4 bytes
- Customer_Id (VARCHAR(10)): 10 bytes
- Payment_Id (INTEGER): 4 bytes

Total Size per Record:

$4 + 10 + 4 = \mathbf{18 \text{ bytes}}$

- Total records: 20 subscriptions
- Total size: $18 \text{ bytes} \times 20 = 360 \text{ bytes}$

7. Subscription_Lines Table:

- Subscription_Id (INTEGER): 4 bytes
- Software_Plan_Id (VARCHAR(10)): 10 bytes
- Duration (INTEGER): 4 bytes

Total Size per Record:

$4 + 10 + 4 = 18$ bytes

- Total records: 40 subscription lines
- Total size: 18 bytes \times 40 = 720 bytes

Current Size of My Database:

Total Size = 2,460 + 3,480 + 96 + 32 + 700 + 360 + 720 = **7,848 bytes** (approximately **7.8 KB**) **Size of each table:**

- Customer Table (one record): 123 bytes
- Software_Plan Table (one record): 174 bytes
- Method Table (one record): 24 bytes
- Discount_On_Duration Table (one record): 8 bytes
- Payments Table (one record): 35 bytes
- Subscriptions Table (one record): 18 bytes
- Subscription_Lines Table (one record): 18 bytes

Estimating the size of DB after 1 year:

New Data for 1 Year Later:

- New Customers: 1000
- New Software Plans: 20
- New Methods: 3
- New Discounts: 3
- New Subscriptions: 500
- New Subscription Lines: 1000

Calculations:

1. Customer Table:

- Size per record: 123 bytes
- Total records after 1 year: 20 (existing) + 1000 (new) = 1020 customers
- Total size: 123 bytes \times 1020 = 125,460 bytes

2. Software_Plan Table:

- Size per record: 174 bytes
- Total records after 1 year: 20 (existing) + 20 (new) = 40 software plans
- Total size: 174 bytes \times 40 = 6,960 bytes

3. Method Table:

- Size per record: 24 bytes
- Total records after 1 year: 4 (existing) + 3 (new) = 7 methods
- Total size: 24 bytes \times 7 = 168 bytes

4. Discount_On_Duration Table:

- Size per record: 8 bytes
- Total records after 1 year: 4 (existing) + 3 (new) = 7 discounts
- Total size: 8 bytes \times 7 = 56 bytes

5. Payments Table:

- Size per record: 35 bytes
- Total records after 1 year: 20 (existing) + 500 (new subscriptions) = 520 payments
- Total size: 35 bytes × 520 = 18,200 bytes

6. Subscriptions Table:

- Size per record: 18 bytes
- Total records after 1 year: 20 (existing) + 500 (new) = 520 subscriptions
- Total size: 18 bytes × 520 = 9,360 bytes

7. Subscription_Lines Table:

- Size per record: 18 bytes
- Total records after 1 year: 40 (existing) + 1000 (new) = 1040 subscription lines
- Total size: 18 bytes × 1040 = 18,720 bytes

Now, sum up the sizes of all tables after 1 year:

- Customer Table: 125,460 bytes
- Software_Plan Table: 6,960 bytes
- Method Table: 168 bytes
- Discount_On_Duration Table: 56 bytes
- Payments Table: 18,200 bytes
- Subscriptions Table: 9,360 bytes
- Subscription_Lines Table: 18,720 bytes

Total Size After 1 year

= 125,460 + 6,960 + 168 + 56 + 18,200 + 9,360 + 18,720 = **178,924 bytes (approximately 178.92 KB)**