Broadband Billing System Database

Mini Project Report -Database Lab (DSE 2260)

Department of Data Science & Computer Applications



B. Tech Data Science

4th Semester – Batch: B1

Submitted By

|  |  |
| --- | --- |
| Shreyas Allani | 210968022 |
| Ritesh v Kamath | 210968026 |
| Tanisha Agarwal | 210968020 |
| Sankarshana S Sagaram | 210968023 |
| Rishaan Rathore | 210968029 |

**Mentored By**

|  |  |
| --- | --- |
| Vinayak M | Archana H/ Dr. Natesh |
| Assistant Professor-Senior | Assistant Professor-Senior |
| DSCA, MIT | DSCA, MIT |

Text

Description automatically generated

**CERTIFICATE**

This is to certify that the Shreyas Allani (210968022), Tanisha Agarwal (210968020),Sankarshana S Sagaram(210968023), Ritesh V Kamath (2100968026), Rishaan Rathore (210968029) have successfully executed a mini project titled “Broadband Billing System Database” rightly brining for the competencies and skill sets they have gained during the course- Database Lab (DSE 2262 & DSE), thereby resulting in the culmination of this project.

|  |  |
| --- | --- |
| **Vinayak M** | **Archana H / Dr. Natesh** |
| **Assistant Professor-Senior** | **Assistant Professor-Senior** |
| **DSCA, MIT** | **DSCA, MIT** |

**ABSTRACT**

A broadband billing system is an essential part of any Internet service provider's operations. It enables ISPs to manage their billing and subscription services effectively, ensuring that subscribers are billed accurately and in a timely manner. This project on a broadband billing system involves creating a database schema, implementing the database, developing a user interface, testing the system, deploying it in a production environment, and maintaining it over time. The project requires a thorough understanding of database management, as well as programming languages such as SQL, to create a functional and efficient system.

Contents

**1. Introduction**

**2. Synopsis**

**2.1 Proposed System**

**2.2 Objectives**

**3. Data Requirements**

**4. Functional Requirements**

**5. Detailed Design**

**5.1 ER Diagram**

**5.2 Schema Diagram**

**5.3 Data Dictionary**

**5.4 Relational Model Implementation**

**5.5 Queries**

**5.7 Triggers**

**5.8 Stored Procedures .**

**5.9 Stored Functions**

**6. Roles and Responsibilities**

**7. Result**

**8. Conclusion and Future Work**

Chapter 1

Introduction

The system's database schema should be designed to accommodate various subscription plans, billing cycles, and payment methods. The user interface will allow the ISP to add, modify, and delete subscribers, plans, bills, and payments quickly and easily. Reports on subscriber usage and billing will also be available. The system will be thoroughly tested to ensure that it meets the requirements and is free of errors. Once the system has been tested, it can be deployed in a production environment, where users can be trained on how to use the system. Maintenance of the system is crucial to ensure that it continues to meet the ISP's changing business requirements over time

Chapter 2

Synopsis

**2.1 Proposed System**

This project involves the development of a broadband billing system for an Internet service provider, with the main objectives of handling different billing cycles, subscription plans, and payment methods, and providing an easy-to-use user interface for managing subscribers, plans, bills, and payments. The project requires the creation of a database schema, implementation of the database using SQL, development of a user interface, testing of the system, deployment in a production environment, and ongoing maintenance. Chapter 1 provides an introduction to the project, while Chapter 2 describes the proposed system and its objectives. Chapter 3 outlines the data requirements, including the entities and data relevant to the transactions. Chapter 4 provides a brief description of the User Registering/Login module, including its functionalities such as new user registration, login, and forgot password, as well as money withdrawal.Chapter 5 provides you an insight into our approach with respect to how we are implementing this idea with the help of ER model representation.

**2.2 Objectives**

The primary objectives are as follows:

* Handle different billing cycles, subscription plans, and payment methods.
* Easy to add, modify, and delete subscribers, and the system should be able to track subscriber usage and billing history.
* To process different payment methods, such as credit cards and online payments, and generate receipts automatically.

**Chapter 3**

**Data Requirements**

[ for your understanding only

Ensures that data produced and consumed satisfies business objectives, is understood by all relevant stakeholders, and meets the needs of the business processes that create and use the data.]

**3.1** **Purpose,** **scope and overview**

The Data Requirement Document (DRD) is a central document of the project, in which all information relating to data is gathered for agreement by the key stakeholders and then for guidance and information for those involved in the project. The data requirements are listed in the following sections and describe the essential data requirements for the *Banking* application.

The application requires data describing the Customers, Login, Subscriptions ,Transaction and Billing Details managing different operations such as Bill Generation, Payment operations. Following are main entities about which data requirement is to be documented.

1. Customers
2. Login Details
3. Subscriptions of Customers
4. Data relevant to the transactions done by customers.

**3.2 Requirements**

* **SQLPlus**
* **Oracle Database**

**3.2.1 CustomerDetails**

|  |  |  |
| --- | --- | --- |
| Following Data related to customer purchasing a Broadband Connection need to be captured. | | |
| **Data Name** | **Description** | **Example** |
| Customer\_Id | An unique alphanumeric value to identify each customer. | SIB12345 |
| Name | Name of the Customer |  |
| Address |  |  |
| Mobile | An alphanumeric value of minimum 8 characters containing at least one capital, number, special character | Abc$ef29 |

**3.2.2 Subscription**

|  |  |  |
| --- | --- | --- |
| Following Data is related to different accounts owned by the customer. Each Customer can have one or more different accounts such as SB, RD, DEPOSIT etc. Also need to record Branch in which account is opened and date of opening. | | |
| **Data Name** | **Description** | **Example** |
| SubId | A unique 10-digit numeric value to identify Subscription Availed by every customer. | 127839989234 |
| S\_Details | Numeric data which tells the no of months in the Subscription | 6 months |
| S\_Speed | Numeric data representing the Speed of Internet connection in Mbps | 100 mbps |
| S\_cost | Numeric data represents Subsciption Cost of account per month. | 650 |

**3.2.3 Bills**

|  |  |  |
| --- | --- | --- |
| Following Data is used to Generate bill for the broadband connection every month. | | |
| **Data Name** | **Description** | **Example** |
| Bill\_no | A unique 5-digit numeric value to identify the bill generated by the system. | 00212 |
| Cost | Numeric Data representing the bill amount generated. | 1003.2 |
| Status | Represents the payment of the bill Status. | P/F/PR |
| Month | Numeric data representing the particular month | 05 |

**Harry1305@**

**3.2.4 Transaction\_details**

|  |  |  |
| --- | --- | --- |
| Following Data is used to Generate bill for the broadband connection every month. | | |
| **Data Name** | **Description** | **Example** |
| Tran\_id | A unique 6-digit numeric value to identify the bill generated by the system. | 422465 |
| Paid\_amt | Numeric Data representing the bill amount paid. | 1903.2 |
| Method | Represents the Method of the payment. | CARD/UPI/N.BANKING/ |
| Date of Issue | Date attribute representing the date of the transaction | 12-04-2023 |

**Chapter 4**

**Functional Requirements**

**4.1 User Registering/Login module**

**4.1.1 New User Registration**

The user must be able to create user id and password by supplying appropriate details.

|  |  |
| --- | --- |
| INPUT | New username, Password, phone |
| Processing | The system must check availability of entered user name.  Password must follow criteria- minimum 9 char, at least one capital, one number and one special character.  Check for validity of phone number by prompting to enter OTP |
| OUTPUT | User created Successfully message / highlight the information entered which is wrong and allow to renter. |

**4.1.2 Login**

The existing user must be able to login upon entering proper user name and password.

|  |  |
| --- | --- |
| INPUT | username, Password |
| Processing | Check the user name and password against information stored in data storage |
| OUTPUT | If user entered correct user name & Password  Login successful and open main application menu  Else  Display Login not successful, retry logging in |

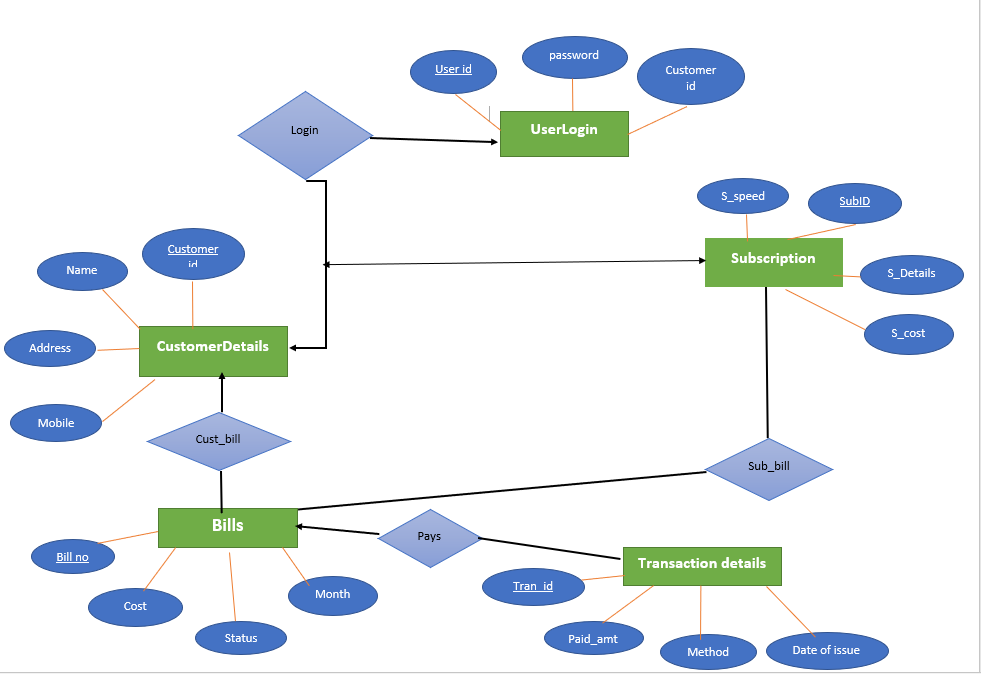
**4.1.3 Bill Generation**

|  |  |
| --- | --- |
| INPUT | Prompt user to enter SubId,UserID |
| Processing | Find the Bill related to User in Bill table  else  Cancel withdrawal & Show message “Not Found” |
| OUTPUT | Billno/Not Found |

**Chapter 5**

**Detailed Design**

**5.1 ER Diagram**

****

**5.2 Schema Diagram**

Bills (Billno , Cost, Status, Month,Customer\_id);

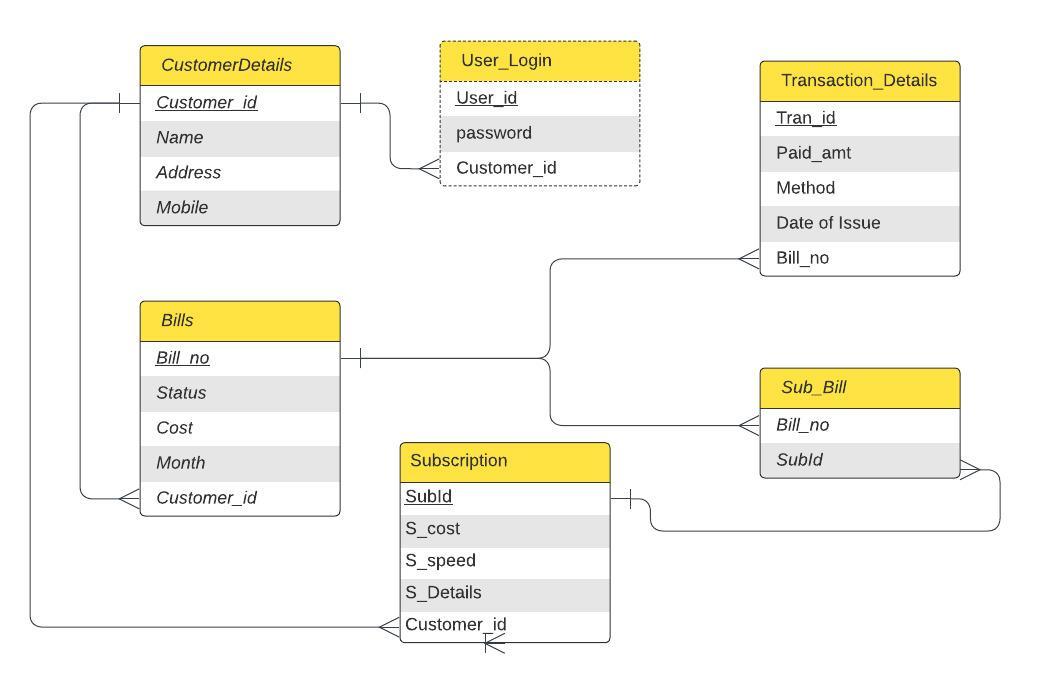
Tansaction\_details(Tran\_id,Paid\_amt,Method,Date\_of\_issue,Bill\_no);

Customer\_details (Customer\_id, Mobile, Name, Address,);

User\_login(User\_id,password,Customer\_id);

Subscription (SubId, S\_cost, S\_Details, S\_speed);

Sub\_bill(Billno, SubId);

****

**5.3 Data Dictionary**

**CustomerDetails**

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data type (size) | Constraint | Constraint Name |
| Customer\_id | int(5) | Primary Key | Customer\_id\_PKey |
| Name | Varchar2(100) | - | - |
| Address | Varchar (255) | - | - |
| Mobile | Number(10) | Starts with +91-,… | NOT NULL |

**Subscription**

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data type (size) | Constraint | Constraint Name |
| SubId | Varchar2(3) | Primary Key | Sub\_PKey |
| S\_cost | Number(10) | Not Null | - |
| S\_speed | Number(5) | >10 | - |
| S\_Details | Varchar2(20) | >1000 | - |
|  |  |  |  |

**Bills**

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data type (size) | Constraint | Constraint Name |
| Bill\_no | Int(5) | Primary Key | Bill\_PKey |
| Status | Varchar2(10) | Pro,Fail,Suc | NOT NULL |
| Cost | Number(10,5) | - | - |
| Month | Varchar(10) | Starts with Jan,Feb… | - |
|  |  |  |  |

**Transaction\_Details**

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data type (size) | Constraint | Constraint Name |
| Tran\_id | Int(5) | Primary Key | NOT NULL |
| Paid\_amt | Varchar2(10) | Foreign Key | NOT NULL |
| Method |  |  | - |
| Date of Issue | Number (5) | >1000 | - |
|  |  |  |  |

**User\_Login**

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data type (size) | Constraint | Constraint Name |
| User\_id | Int(4) | Primary Key | User\_PKey |
| Password | Varchar2(50) | Capital Letter,Special Char,Number | NOT NULL |
|  |  |  |  |

**5.4 Relational Model Implementation**

Create Table CustomerDetails (

Customer\_ID integer NOT NULL, Mobile varchar(50) NOT NULL, Name varchar2(255), Address varchar2(255), UserID int NOT NULL, SubID int , CONSTRAINT PK\_Cust\_Details\_CustomerID PRIMARY KEY(Customer\_ID), CONSTRAINT FK\_Cust\_Details\_UserID FOREIGN KEY (UserID) REFERENCES User\_login(UserID), CONSTRAINT FK\_Cust\_Details\_SubscriptionID FOREIGN KEY (SubID) REFERENCES Subscription(Sub\_ID)

);

Create Table Transaction\_Details(

Tran\_id integer NOT NULL, DateOfIssue Date , Paid\_Amt int, Method Varchar(50), BillNo int, CONSTRAINT PK\_Transaction\_Details\_TransactionID PRIMARY KEY(Tran\_id), CONSTRAINT FK\_Transaction\_Details\_BillNo FOREIGN KEY (BillNo) REFERENCES Bills(Bill\_no)

);

Create Table Bills(

Bill\_no Number(5) NOT NULL, Month Varchar(255), Cost int, Status Varchar(50) default 'Pending, CustomerID int, SubscriptionID int, CONSTRAINT PK\_Bills\_BillNo PRIMARY KEY(Bill\_no)

, CONSTRAINT FK\_Bills\_CustomerID FOREIGN KEY (CustomerID) REFERENCES CustomerDetails(Customer\_ID)

, CONSTRAINT FK\_Bills\_SubscriptionID FOREIGN KEY (SubscriptionID) REFERENCES Subscription(Sub\_ID)

);

Create Table Subscription(

Sub\_ID integer NOT NULL, SCost int, S\_Details varchar2(20), S\_Speed varchar2(50), CONSTRAINT PK\_Subscription\_SubscriptionID PRIMARY KEY(Sub\_ID)

);

Create table User\_login(

UserId number(4) constraints PK\_User\_UserID PRIMARY KEY,Password Nvarchar(50) NOT NULL);

**5.5 Queries**

1. Get the details of all the customers in the "CustomerDetails", ”Subscription”, ”Bills”, ”Transaction\_Details” tables:

SELECT \* FROM CustomerDetails;

SELECT \* FROM Subscription;

SELECT \* FROM Bills;

SELECT \* FROM Transaction\_Details;

2. Get the details of the customers who have subscribed to a particular subscription plan:

SELECT \* FROM CustomerDetails WHERE SubID = 2;

OUTPUT: CUSTOMER\_ID MOBILE NAME ADDRESS USERID SUBID

---------- ----------

26 9876543210 Aarav Patel 123 Main Street, Bangalore 1 2

3. Get the total amount paid by a particular customer:

SELECT SUM(Paid\_Amt) FROM Transaction\_Details WHERE BillNo IN (SELECT Bill\_no FROM Bills WHERE CustomerID = 26);

OUTPUT:

SUM(PAID\_AMT)

-------------

300

4. Get the number of customers who have subscribed to a particular s ubscription plan:

**SELECT COUNT(\*) FROM CustomerDetails WHERE SubID = 3;**

**OUTPUT:**

**COUNT(\*)**

**----------**

**2**

**5.Query to fetch all the customer names and subscription details**

**DECLARE**

**CURSOR c\_customer\_sub IS**

**SELECT c.Name, s.S\_Details, s.S\_Speed, s.SCost**

**FROM CustomerDetails c**

**JOIN Subscription s ON c.SubID = s.Sub\_ID;**

**BEGIN**

**FOR customer\_sub IN c\_customer\_sub**

**LOOP**

**DBMS\_OUTPUT.PUT\_LINE(customer\_sub.Name || ' is subscribed to ' || customer\_sub.S\_Details || ' with a speed of ' || customer\_sub.S\_Speed || ' and a cost of ' || customer\_sub.SCost);**

**END LOOP;**

**END;**

**OUTPUT:**

**Aarav Patel is subscribed to Standard with a speed of 20 and a cost of 30**

**Aarushi Desai is subscribed to Premium with a speed of 50 and a cost of 40**

**Aditi Sharma is subscribed to Premium with a speed of 50 and a cost of 40**

**Advait Gupta is subscribed to Basic with a speed of 15 and a cost of 25**

**Aishwarya Singh is subscribed to Standard with a speed of 25 and a cost of 35**

**Anjali Mehra is subscribed to Premium with a speed of 60 and a cost of 45**

**Arnav Dubey is subscribed to Premium Plus with a speed of 100 and a cost of 50**

**Dhruv Gupta is subscribed to Ultimate with a speed of 200 and a cost of 60**

**Esha Khanna is subscribed to Deluxe with a speed of 150 and a cost of 70**

**Ishaan Sharma is subscribed to Elite with a speed of 300 and a cost of 80**

**PL/SQL procedure successfully completed.**

**5.6 PROCEDURES**

**1)**

**CREATE OR REPLACE PROCEDURE InsertUserLogin(**

**p\_user\_id IN User\_Login.UserID%TYPE,**

**p\_password IN User\_Login.Password%TYPE)**

**IS**

**BEGIN**

**INSERT INTO User\_Login (UserID, Password)**

**VALUES (p\_user\_id, p\_password);**

**COMMIT;**

**dbms\_output.put\_line('New user with UserID ' || p\_user\_id || ' has been inserted.');**

**EXCEPTION**

**WHEN OTHERS THEN**

**dbms\_output.put\_line('Error: ' || SQLCODE || ' - ' || SQLERRM);**

**END InsertUserLogin;**

**/**

**2)**

**CREATE OR REPLACE PROCEDURE UpdateUserPassword(**

**p\_user\_id IN User\_Login.UserID%TYPE,**

**p\_password IN User\_Login.Password%TYPE**

**)**

**IS**

**BEGIN**

**UPDATE User\_Login**

**SET Password = p\_password**

**WHERE UserID = p\_user\_id;**

**COMMIT;**

**dbms\_output.put\_line('Password for UserID ' || p\_user\_id || ' has been updated.');**

**EXCEPTION**

**WHEN NO\_DATA\_FOUND THEN**

**dbms\_output.put\_line('User with UserID ' || p\_user\_id || ' not found.');**

**WHEN OTHERS THEN**

**dbms\_output.put\_line('Error: ' || SQLCODE || ' - ' || SQLERRM);**

**END UpdateUserPassword;**

**3)**

**CREATE OR REPLACE PROCEDURE InsertCustomerDetails(**

**p\_customer\_id IN CustomerDetails.Customer\_ID%TYPE,**

**p\_mobile IN CustomerDetails.Mobile%TYPE,**

**p\_name IN CustomerDetails.Name%TYPE,**

**p\_address IN CustomerDetails.Address%TYPE,**

**p\_user\_id IN CustomerDetails.UserID%TYPE,**

**p\_sub\_id IN CustomerDetails.SubID%TYPE**

**)**

**IS**

**BEGIN**

**INSERT INTO CustomerDetails (Customer\_ID, Mobile, Name, Address, UserID, SubID)**

**VALUES (p\_customer\_id, p\_mobile, p\_name, p\_address, p\_user\_id, p\_sub\_id);**

**COMMIT;**

**dbms\_output.put\_line('New customer with CustomerID ' || p\_customer\_id || ' has been inserted.');**

**EXCEPTION**

**WHEN OTHERS THEN**

**dbms\_output.put\_line('Error: ' || SQLCODE || ' - ' || SQLERRM);**

**END InsertCustomerDetails;**

**4)**

**CREATE OR REPLACE PROCEDURE InsertNewBill(**

**billNo IN Bills.Bill\_no%TYPE,**

**month IN Bills.Month%TYPE,**

**cost IN Bills.Cost%TYPE,**

**status IN Bills.Status%TYPE,**

**customerId IN Bills.CustomerID%TYPE,**

**subId IN Bills.SubscriptionID%TYPE**

**)**

**IS**

**BEGIN**

**INSERT INTO Bills(Bill\_no, Month, Cost, Status, CustomerID, SubscriptionID)**

**VALUES(billNo, month, cost, status, customerId, subId);**

**dbms\_output.put\_line('Bill added successfully');**

**END;**

**/**

**5)**

**CREATE OR REPLACE PROCEDURE InsertNewTransaction(**

**tranId IN Transaction\_Details.Tran\_id%TYPE,**

**dateOfIssue IN Transaction\_Details.DateOfIssue%TYPE,**

**paidAmt IN Transaction\_Details.Paid\_Amt%TYPE,**

**method IN Transaction\_Details.Method%TYPE,**

**billNo IN Transaction\_Details.BillNo%TYPE**

**)**

**IS**

**BEGIN**

**INSERT INTO Transaction\_Details(Tran\_id, DateOfIssue, Paid\_Amt, Method, BillNo)**

**VALUES(tranId, dateOfIssue, paidAmt, method, billNo);**

**dbms\_output.put\_line('Transaction added successfully');**

**END;**

**/**

**6)**

**CREATE OR REPLACE PROCEDURE InsertNewSubscription(**

**subId IN Subscription.Sub\_ID%TYPE,**

**sCost IN Subscription.SCost%TYPE,**

**sDetails IN Subscription.S\_Details%TYPE,**

**sSpeed IN Subscription.S\_Speed%TYPE**

**)**

**IS**

**BEGIN**

**INSERT INTO Subscription(Sub\_ID, SCost, S\_Details, S\_Speed)**

**VALUES(subId, sCost, sDetails, sSpeed);**

**dbms\_output.put\_line('Subscription added successfully');**

**END;**

**/**

**5.7 ) TRIGGERS**

**1)** **Trigger to update the subscription cost in the bills table when the subscription cost is updated in the subscription table:**

**CREATE OR REPLACE TRIGGER update\_subscription\_cost**

**AFTER UPDATE OF SCost ON Subscription**

**FOR EACH ROW**

**BEGIN**

**UPDATE Bills**

**SET Cost = :new.SCost**

**WHERE SubscriptionID = :old.Sub\_ID;**

**END;**

**/**

**2)** **Trigger to automatically insert a new bill for a customer when a new subscription is added for that customer:**

**CREATE OR REPLACE TRIGGER add\_new\_subscription\_bill**

**AFTER INSERT ON CustomerDetails**

**FOR EACH ROW**

**BEGIN**

**INSERT INTO Bills (Bill\_no, Month, Cost, CustomerID, SubscriptionID)**

**VALUES ((SELECT MAX(Bill\_no) FROM Bills) + 1, TO\_CHAR(SYSDATE, 'Month'), :new.SCost, :new.Customer\_ID, :new.SubID);**

**END;**

**/**

**3)** **Trigger to automatically update a bill's status to "Paid" when a transaction is recorded:**

**CREATE OR REPLACE TRIGGER update\_bill\_status\_to\_paid**

**AFTER INSERT ON Transaction\_Details**

**FOR EACH ROW**

**BEGIN**

**UPDATE Bills**

**SET Status = 'Paid'**

**WHERE Bill\_no = :new.BillNo;**

**END;**

**/**

**4)** **Trigger to update customer's subscription details when a new bill is added**

**CREATE OR REPLACE TRIGGER update\_subscription\_details**

**AFTER INSERT ON Bills**

**FOR EACH ROW**

**BEGIN**

**UPDATE CustomerDetails**

**SET SubID = :new.SubscriptionID**

**WHERE Customer\_ID = :new.CustomerID;**

**END;**

**6) ROLES AND RESPONSIBILITIES**

Shreyas Allani – ER diagram , Schema Diagram,Procedures

Tanisha Agarwal – Creation of tables

Ritesh v Kamath-Triggers, Relational Schema

Sankarshana s Sagaram- Data Dictionary, Data Requirements,Synopsis

Rishaan Rathore- Editing the pdf,Abstract