**SVKM’s NMIMS**

**School of Technology Management & Engineering, Chandigarh**

A.Y. 2023 - 24

**Course: Database Management Systems**

**Project Report**

|  |  |  |
| --- | --- | --- |
| Program | B.tech Computer Engineering | |
| Semester | 4 | |
| Name of the Project: | car rental database management | |
|  | | |
| Details of Project Members |  |  |
| Batch | Roll No. | Name |
| B2 | A122 &. A127 | Chiraag Chaudhary , MD Rehan Shafiq |
| Date of Submission: 02-04-2024 | | |

**Contribution of each project Members:**

|  |  |  |
| --- | --- | --- |
| Roll No. | Name: | Contribution |
| A127 | Rehan Shafiq | Code , sql |
| A122 | Chiraag Chaudhary | Er and relational model |

**Github link of your project:**

**Note:**

1. Create a readme file if you have multiple files
2. All files must be properly named (Example:R004\_DBMSProject)
3. Submit all relevant files of your work ( Report, all SQL files, Any other files)
4. **Plagiarism is highly discouraged (Your report will be checked for plagiarism)**

**Rubrics for the Project evaluation:**

|  |  |
| --- | --- |
| First phase of evaluation:  Innovative Ideas (5 Marks)  Design and Partial implementation (5 Marks) | 10 marks |
| Final phase of evaluation  Implementation, presentation and viva, Self-Learning and Learning Beyond classroom | 10 marks |

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sr no.** | **Topic** | **Page no.** |
| **1** | Storyline |  |
| **2** | Components of Database Design |  |
| **3** | Entity Relationship Diagram |  |
| **4** | Relational Model |  |
| **5** | Normalization |  |
| **6** | SQL Queries |  |
| **7** | Learning from the Project |  |
| **8** | Project Demonstration |  |
| **9** | Self-learning beyond classroom |  |
| **10** | Learning from the project |  |
| **8** | Challenges faced |  |
| **9** | Conclusion |  |

1. **Storyline**

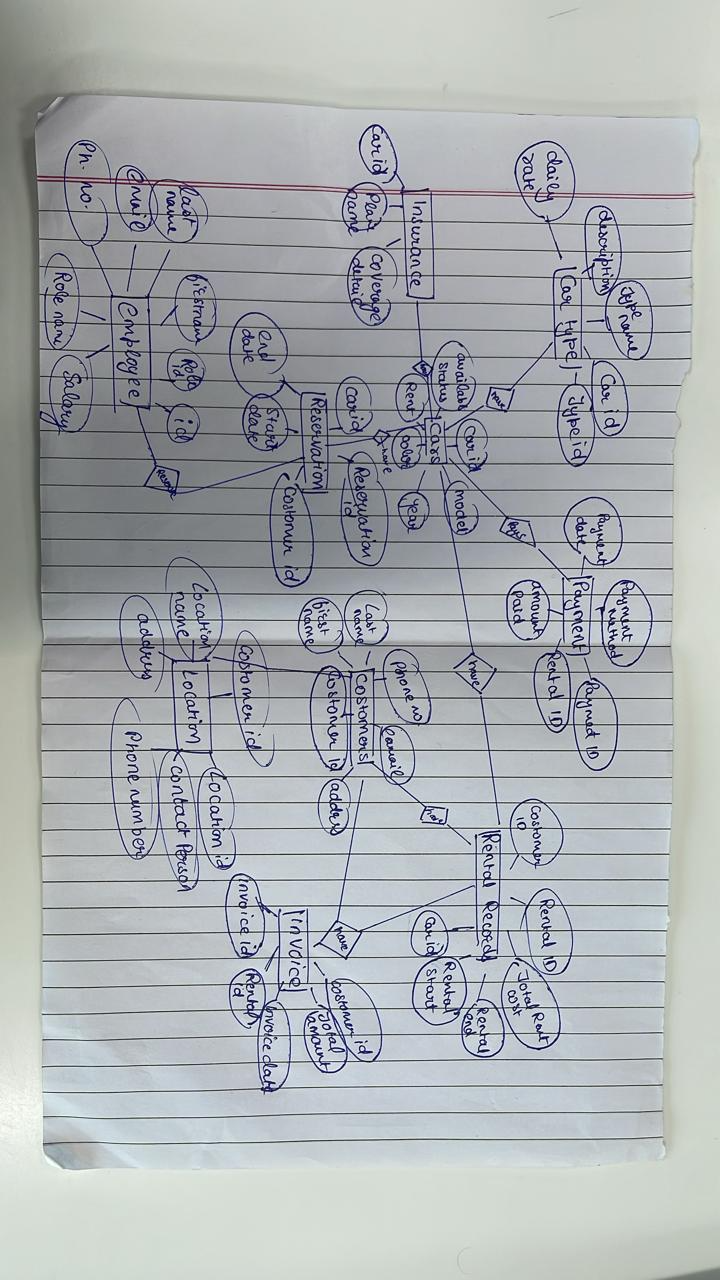
Welcome to WHEEL on WINGS, your premier destination for hassle-free car rentals! At our service,

we pride ourselves on providing top-notch service and a wide selection of vehicles to meet your every need. Whether you're looking for a sleek sedan for a business trip, a spacious SUV for a family vacation, or a luxurious convertible for a weekend getaway, we have the perfect ride for you. With competitive rates, convenient booking options, and a team of friendly professionals ready to assist you every step of the way, renting a car with us is a breeze. Experience the freedom and flexibility of exploring your destination on your own terms with [Shop Name]. Book your rental today and let the adventure begin. The system, name WHEEL on Wings Management System, will efficiently handle rental plans, vehicle records,employees , and various administrative tasks.

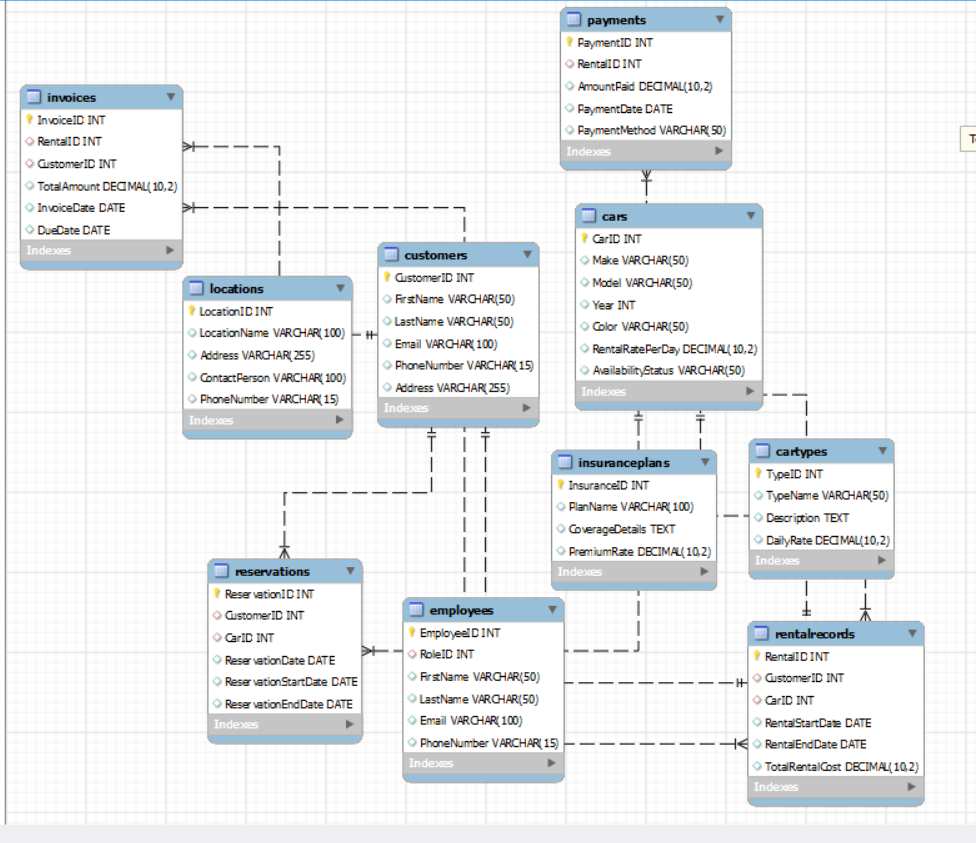
**II. Components of Database Design**

**(entities, attributes, relationships and cardinality)**

**III. Entity Relationship Diagram**



**IV. Relational Model**



**V. Normalization**

Perform normalization (1NF, 2NF, 3NF, BCNF) as applicable for the entire database.

**VI. SQL Queries**

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Email VARCHAR(100),

PhoneNumber VARCHAR(15),

Address VARCHAR(255)

);

CREATE TABLE Cars (

CarID INT PRIMARY KEY,

Make VARCHAR(50),

Model VARCHAR(50),

Year INT,

Color VARCHAR(50),

RentalRatePerDay DECIMAL(10, 2),

AvailabilityStatus VARCHAR(50)

);

CREATE TABLE RentalRecords (

RentalID INT PRIMARY KEY,

CustomerID INT,

CarID INT,

RentalStartDate DATE,

RentalEndDate DATE,

TotalRentalCost DECIMAL(10, 2),

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),

FOREIGN KEY (CarID) REFERENCES Cars(CarID)

);

CREATE TABLE Reservations (

ReservationID INT PRIMARY KEY,

CustomerID INT,

CarID INT,

ReservationDate DATE,

ReservationStartDate DATE,

ReservationEndDate DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),

FOREIGN KEY (CarID) REFERENCES Cars(CarID)

);

CREATE TABLE Payments (

PaymentID INT PRIMARY KEY,

RentalID INT,

AmountPaid DECIMAL(10, 2),

PaymentDate DATE,

PaymentMethod VARCHAR(50),

FOREIGN KEY (RentalID) REFERENCES RentalRecords(RentalID)

);

CREATE TABLE CarTypes (

TypeID INT PRIMARY KEY,

TypeName VARCHAR(50),

Description TEXT,

DailyRate DECIMAL(10, 2)

);

CREATE TABLE Locations (

LocationID INT PRIMARY KEY,

LocationName VARCHAR(100),

Address VARCHAR(255),

ContactPerson VARCHAR(100),

PhoneNumber VARCHAR(15)

);

CREATE TABLE InsurancePlans (

InsuranceID INT PRIMARY KEY,

PlanName VARCHAR(100),

CoverageDetails TEXT,

PremiumRate DECIMAL(10, 2)

);

CREATE TABLE EmployeeRoles (

RoleID INT PRIMARY KEY,

RoleName VARCHAR(100),

Responsibilities TEXT,

Salary DECIMAL(10, 2)

);

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

RoleID INT,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Email VARCHAR(100),

PhoneNumber VARCHAR(15),

FOREIGN KEY (RoleID) REFERENCES EmployeeRoles(RoleID)

);

CREATE TABLE Invoices (

InvoiceID INT PRIMARY KEY,

RentalID INT,

CustomerID INT,

TotalAmount DECIMAL(10, 2),

InvoiceDate DATE,

DueDate DATE,

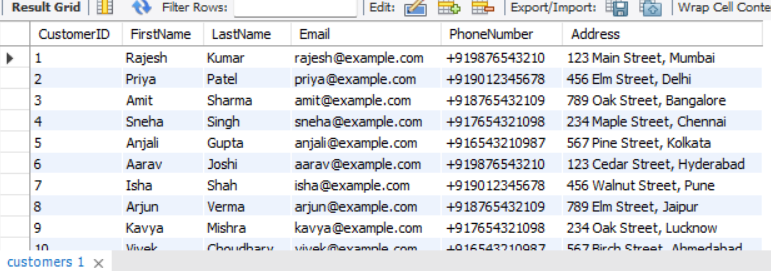
FOREIGN KEY (RentalID) REFERENCES RentalRecords(RentalID),

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

**Output of Tables:**

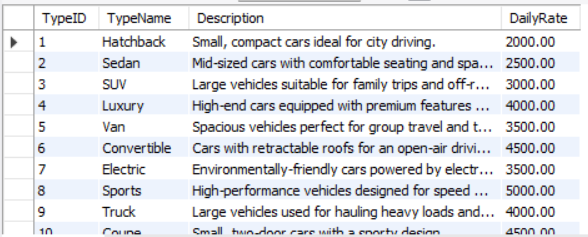
Customers:



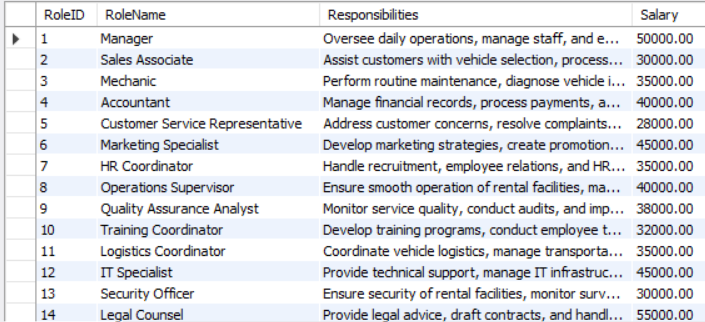
Cars:



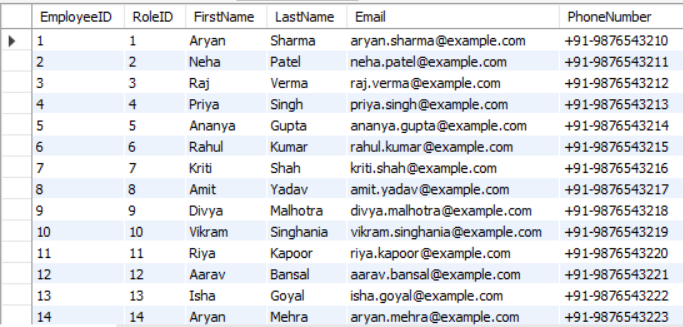
Cartypes:



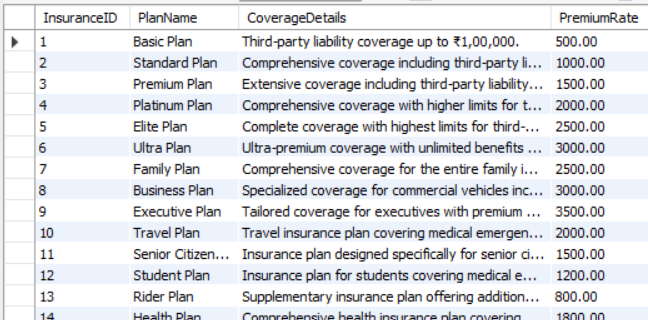
Employee roles:



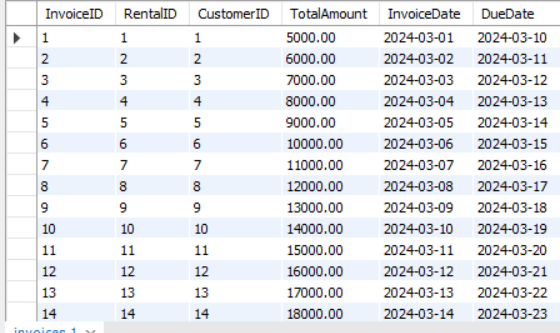
Employees:



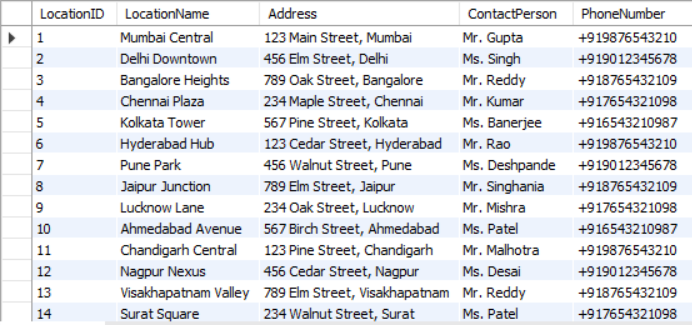
Insuranceplans:



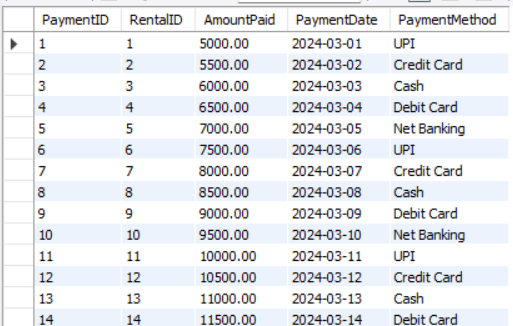
Invoices:



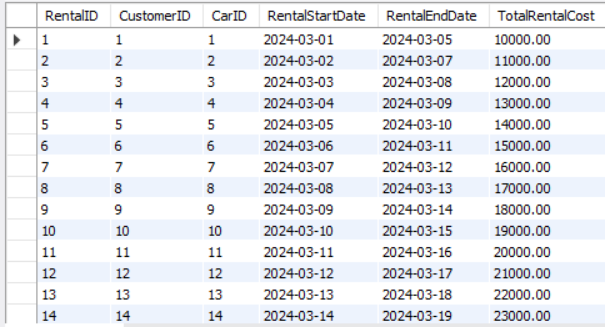
Locations:



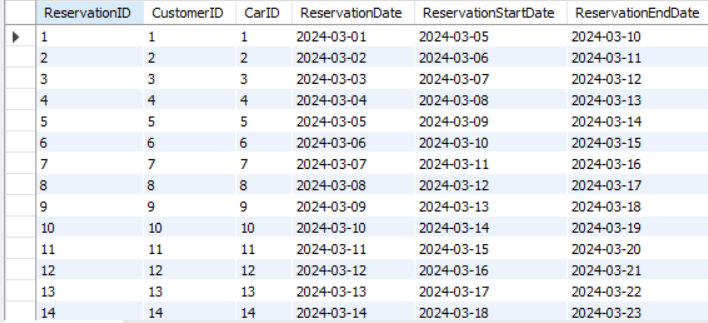
Payments:



Rentalrecords:



Reservations:



**SQL Queries:minimum 20 sql queries and outputs**

#Retrieve rental records along with customer details:

SELECT R.RentalID, R.RentalStartDate, R.RentalEndDate, R.TotalRentalCost, C.FirstName, C.LastName, C.Email, C.PhoneNumber

FROM RentalRecords R

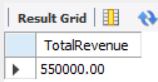
JOIN Customers C ON R.CustomerID = C.CustomerID;



#Calculate the total revenue generated from rentals:

SELECT SUM(TotalRentalCost) AS TotalRevenue

FROM RentalRecords;

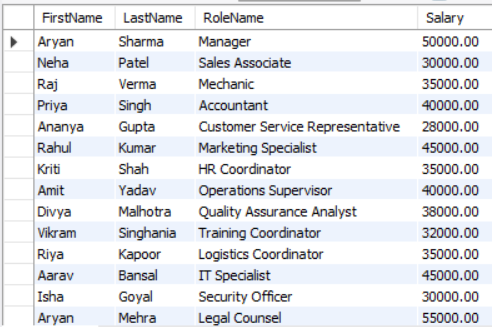


#Retrieve employees with their roles and salaries:

SELECT E.FirstName, E.LastName, ER.RoleName, ER.Salary

FROM Employees E

JOIN EmployeeRoles ER ON E.RoleID = ER.RoleID;

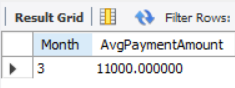


#Retrieve the average payment amount made per month:

SELECT EXTRACT(MONTH FROM PaymentDate) AS Month, AVG(AmountPaid) AS AvgPaymentAmount

FROM Payments

GROUP BY EXTRACT(MONTH FROM PaymentDate);

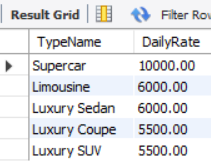
  
#Retrieve the top 5 car types with the highest daily rates:

SELECT TypeName, DailyRate

FROM CarTypes

ORDER BY DailyRate DESC

LIMIT 5;



#Retrieve the employee who made the highest payment:

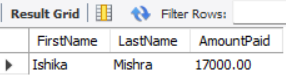
SELECT E.FirstName, E.LastName, P.AmountPaid

FROM Employees E

JOIN Payments P ON E.EmployeeID = P.PaymentID

ORDER BY P.AmountPaid DESC

LIMIT 1;



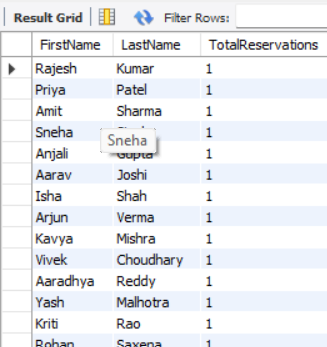
#Retrieve the number of reservations made by each customer:

SELECT C.FirstName, C.LastName, COUNT(\*) AS TotalReservations

FROM Customers C

JOIN Reservations R ON C.CustomerID = R.CustomerID

GROUP BY C.FirstName, C.LastName;

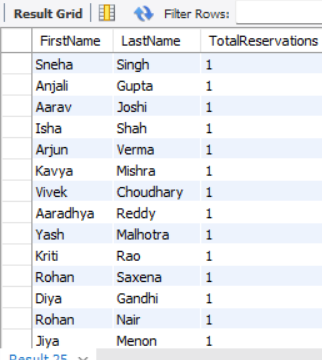


#Retrieve the average rental rate per day for each car make:

SELECT Make, AVG(RentalRatePerDay) AS AvgRentalRate

FROM Cars

GROUP BY Make;



#Retrieve the cars that are currently available for rent:

SELECT \*

FROM Cars

WHERE AvailabilityStatus = 'Available';

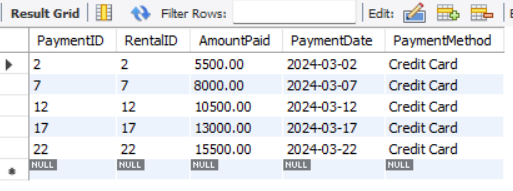


#Retrieve the payments made using a specific payment method:

SELECT \*

FROM Payments

WHERE PaymentMethod = 'Credit Card';

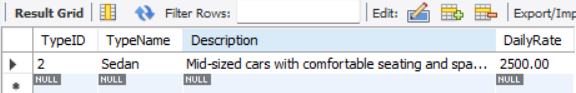


#Retrieve the details of a specific car type:

SELECT \*

FROM CarTypes

WHERE TypeName = 'Sedan';



**VI. Project demonstration**

* I have used MySQL for this Project.
* This project is done of the website we created about car rentals
* Because of sql we created backend of the details which were essential for the website

**VII. Self -Learning beyond classroom**:

* What new aspects did you learn on your own ? You have to mention learning beyond the classroom
* We learned the importance of backend and how it works in a website which we use on a very daily basis , we got to know that sql. Requires a very technical mindset important for engineering students and will play a very important part in the coming future.

**VIII. Learning from the Project**

Include learning from the project:

* How this project helped you?
* We learned to design and implement a comprehensive database system for a grocery store,

including tables for customers, products, orders, and payments. I also explored creating relationships

between tables using foreign keys and learned how to insert data into these tables. This hands-on

experience expanded my understanding of SQL and database design principles beyond the classroom.

**IX. Challenges Faced**

**1. Designing the Database Schema: Creating a schema that accurately represented the**

**relationships between different entities while ensuring data integrity and**

**efficiency was challenging.**

**2. Managing Foreign Keys and Relationships: Establishing and maintaining the correct**

**relationships between tables, especially when dealing with foreign keys, was complex and required careful**

**planning to avoid data inconsistencies.**

**3.Data Normalization: Ensuring that the database was normalized to reduce redundancy and**

**improve data integrity was a challenge. It required a good understanding of normalization rules and the**

**ability to identify and eliminate redundancies.**

**4. Security and Data Protection: Implementing security measures to protect sensitive data, such**

**as customer information and payment details, was crucial. This included understanding and applying**

**encryption techniques and access controls.**

**5. Performance Optimization: Ensuring that the database could handle the expected load and**

**perform efficiently, especially with complex queries and large datasets, was a challenge. This required**

**optimizing queries and indexing strategies.**

**6. User Interface Design: Designing a user-friendly interface that allowed users to easily interact**

**with the database was challenging. It required a good understanding of user experience (UX) principles**

**and the ability to create intuitive and responsive designs.**

**X. Conclusion**

* What are the key takeaways from the project?

1. Database Design and Management: Gained proficiency in designing and managing complex

databases, including creating tables, relationships, and ensuring data integrity.

* 2. SQL Proficiency: Enhanced SQL skills, including writing efficient queries, managing data, and

optimizing database performance.

* 3. Software Development Practices: Learned the importance of planning, designing, and testing

software applications to ensure they meet user needs and are robust and secure.

4. Problem-Solving and Critical Thinking: Developed strong problem-solving and critical

thinking skills, essential for tackling complex technical challenges and finding innovative solutions.

1. Communication and team work : there was. A good communication and we learned that work when divided makes job much easier.