Team 2
Database Design Document

Team Members :

1. **Daksh Patel : 002920190**
2. **Dhruv Patel : 002928881**
3. **Pavan Soma Shekar : 002737038**
4. **Shreyansi Patel : 002728928**
5. **Vidhi Sejpal : 002750429**

Database Purpose:

Our Purpose for the Project is to provide a robust and efficient database management system in order to make the car rental process smooth and streamlined. The primary objective is to provide a dependable and efficient system for managing the car rental business by offering a large selection of automobiles at reasonable prices, recording customer service and car maintenance, and utilising technology to make the car rental process as simple and hassle-free as possible.

Business Problems Addressed:

Through this database management system we planned to address the following business problems:

- The customers can rent a car by selecting the cars depending upon their preferences like date, time, location, car category, etc.
- The customers can opt for their preferred insurance and also avail discounts.
- The customers can get a bill according to their usage depending on time, car selected, insurance type, discounts, penalties and taxes.
- The system keeps a track of the payment details including information such as payment status, payment type and transaction details.
- Customer support services are provided to assist the customers with their queries by the employees.
- The customers can raise complaints to provide their feedback on their car rental experience.
- Car maintenance can be tracked to manage the service and repair history of rental cars.

Business Rules:

1. Each person can be either a customer or employee.
2. Each person may have one location.
3. Each customer can have zero or more Complaints.
4. Each customer can have zero or more Reservations.
5. Many employees can provide customer support to multiple customers.
6. Each car may have zero or more car reservations.
7. Each car may have one or more car maintenance.
8. Each car category may have one or many cars.
9. Each car reservation may have one or zero discount.
10. Each discount may be applied to zero or many car reservations.
11. Each car reservation generates a bill.
12. Each bill may have zero or more payments.
13. Each payment may have one bill.
14. Each bill may have zero or more penalties.
15. Each car reservation may have one and only one car insurance.
16. Each insurance may be applied to zero or many car reservations.
17. Each cancelled reservation may be linked with one and only one car reservation.
18. Each car reservation may or may not be cancelled.

Key Design Decisions:

All entities in this ERD are connected

Sr No	Entity Name	Why is Entity Included	Relationship to others
1	Person	Person entity is the most basic entity of this database Management system. The Person table contains PersonID as primary ID and LocationID as foreign key. It includes person details first name, last name, address, contact,email, gender and date of birth details.	This entity is connected to Customers, Employee and Location entities. **<u>The relations are:</u>** - Person is connected to Customer with One-to-One relationship as Person can be either a customer or an

			employee. - Person is connected with Many-to-One with Location as one location ID is associated with a many person. - Person is connected with a One-to-One relationship to the Employee entity as a person can either be an employee or a customer.
2	Customers	Customers inherit from the Person table. The Customers table contains CustomerID as a primary key and PersonID as foreign key. DrivingLicense is also associated with Customers.	Customers entity is connected with Car_Reservation, Person Customer_Support and Complaint entities. **<u>The relations are:</u>** - Customers entity has One to Zero/Many relationships with Car_Reservation. -Customers entity has One to Zero/Many relationships with Customer_Support. -Customers entity has One to Zero/Many relationship with Complaint. - Customers and Person entities have Zero/One to One relationship.
3	Employee	The Employee table inherits the Person table. The Employee table contains EmployeeID as a primary key and PersonID as foreign key.	**<u>The relations are:</u>** - Employee entity has One-to-One/Many relationship with Customer_Support Entity. - Employee and Person can have One-to-One relationship

4.	Location	Location entity is used to store all the information of all the locations including with the address, city, state, country and zip code.	Location entity is connected with Person and Car_Reservation entities. **<u>The relations are:</u>** - Location has a One-to-Zero/Many relationship with Car_Reservation as one location can be given to multiple reservations. - Location has a One-to-Zero/Many relationship with Person as many people can share the same address.
5	Cars	The Car entity is included because it is a fundamental component of the business process. It is required to track and manage the inventory of cars that are available for rental. The business is able to effectively manage its fleet of vehicles, keep track of which cars are available, and keep tabs on how they are being used. Data driven decisions can be made by the company using this information.	Cars entity is connected with Car_Reservation, Car_Maintenance and Car_Category entities. **<u>The relations are:</u>** - Cars has a One-to-Zero/Many relationship with Car_Reservation as many reservations can be made for one car. - Cars has a One-to-Many relationship with Car_Maintenance as one car can require a lot of maintenance. - Cars has a Many-to-One relationship with Car_Category as there can be so many cars of the same category.
6	Car_Category	This entity is included to help classify and organize the vehicles available for rent. Car rental companies typically	Car_Category is connected with Cars entity. **<u>The relations are:</u>**

		offer different types or categories of vehicles, each with its own unique features and price points.This allows the company to offer a variety of options to customers and helps customers choose the right car for their needs. The database system can then use this information to calculate the total cost of a rental based on the car category selected by the customer.	- Car_Category has a One-to-Many relationship with Cars as one category of car can have many cars.
7	Car_Maintenance	This entity is included to help track and manage the maintenance and repair history of the rental cars. Car rental companies rely on their vehicles to generate revenue, and it is crucial that the cars are well-maintained and in good working order to ensure customer safety and satisfaction. The database system can store information about the maintenance schedule for each car, including oil changes, tire rotations, and other routine maintenance tasks.	Car_Maintenance is connected with Cars entity **The relations are:** - Car_Maintenance has a Many-to-One relationship with Cars as multiple maintenance can be required for a car.
8	Car_Reservation	The Car_Reservation entity represents a record of a customer's request to rent a car from a car rental system. Each reservation is associated with a specific car, customer, billing record, insurance coverage, discount or promotional offer, pickup location, drop-off location, and expected drop-off time. The reservation serves as a reference point for the rental system to track the rental process from start to finish, from the moment the customer requests the rental until the car is returned. This entity is an integral part of the rental system as it provides a central record of all the rental-related information, helping	Car_Reservation is connected with Billing, Discount, Cancelled_Reservation, Insurance, Cars, Customers and Location entity **The relations are:** - Car_Reservation has a Zero/Many-to-Zero/One relationship with Discount as multiple reservations can have only one or no Discount. - Car_Reservation has a Zero/Many-to-One relationship with Cars as one Car can be associated with

		the rental company manage its resources and customer interactions effectively.	multiple or no reservations. - Car_Reservation has a One-to-One relationship with Billing as only one table is connected with a reservation. - Car_Reservation has a Zero/Many-to-One relationship with Insurance as multiple or no reservations can have only one Discount. - Car_Reservation has a Zero/Many-to-One relationship with Customers as each Customer can have many or no Reservation. - Car_Reservation has a Zero/Many-to-One relationship with Location as multiple reservations can have only one location. - Car_Reservation has a One-to-Zero/One relationship with Cancelled_Reservatin as each cancelled reservation can have only one reservation.
9	Cancelled_Reservation	Cancelled Reservation entity will maintain a record of all the cancelled reservations, the reasons for which they were cancelled.	Cancelled_Reservation is connected with Car_Reservation entity **<u>The relations are:</u>** - Cancelled_Reservation has a Zero/One-to-One relationship with Car_Reservation as only one or no cancellation can be done for a reservation.
10	Insurance	This entity handles various types of	Insurance is connected with

		car insurance which will be selected by the customer according to their will.	Car_Reservation entity **<u>The relations are:</u>** - Insurance has a One-to-Zero/Many relationship with Car_Reservation as only one type of insurance can be linked with many other reservations.
11	Discount	It contains all the discounts and promotional offers and its validity time.	Discount is connected with Car_Reservation entity **<u>The relations are:</u>** - Discount has a Zero/One-to-Zero/Many relationship with Car_Reservation as only one type of discount or no discount can be linked with many or no other reservations.
12	Billing	The billing entity generates a comprehensive bill that includes details of the car reservation, such as the cost of the car, any applicable discounts, insurance fees, taxes, and penalties. The bill breaks down the costs into fixed and variable components, providing you with a clear understanding of what the customer is paying for. With our smart billing system, the customers can easily track their expenses and manage the budget effectively. The system will provide a transparent and hassle-free rental experience	Billing entity is connected to Car Reservation, Payment and Billing_Penalty entity. **<u>The relations are:</u>** - One bill is generated for a particular reservation. The Billing table has a One-to-One relationship with the Car Reservation table. - The generated bill will be paid by the customer via the respective payment method. The Billing table has a One-to-Zero/Many relationship with the Payment table. - Penalty costs are added to the bill depending on the

			penalty reasons. The Billing table has a One-to-Zero/Many relationship with the Billing_Penalty table.
13	Payment	Payment entity is used to keep track of all the payment information such as payment type, payment status, transaction id and timestamp	Many payments may be associated with each generated bill. **<u>The relations are:</u>** - The Payment entity has a Zero/Many-to-One relationship with the billing entity.
14	Penalty	Penalty entity is used to manage penalties and fines that may be incurred by customers during the rental period. It stores information such as: penalty reason and cost	Penalty entity is connected to the Billing_Penalty entity. **<u>The relations are:</u>** - The Penalty entity has a One-to-Zero/Many relationship with the billing entity.
15	Billing_Penalty	Billing_Penalty is used to manage penalties associated with the bills.	Billing_Penalty is connected to the Billing and Penalty entities. **<u>The relations are:</u>** - The Billing_Penalty entity has a Zero/Many-to-One relationship with the billing entity. - The Billing_Penalty entity has a Zero/Many-to-One relationship with the penalty entity.
16	Customer_Support	Customer_Support is associated with Customers and Employee in order to assist Customers with their queries. CustomerSupportRequestTimeStamp contains the time when particular ticket was raised	Customer_Support entity is connected to the Customers and Employee. **<u>The relations are:</u>** - Customer_Support has

			Zero/Many to one relationship with Customers and Employee.
17	Complaint	Complaint is associated with a customer's complaint which is an essential part of any business. It consists of ComplaintID as primary key and CustomerID as foreign key. We are using ComplaintDate to keep the track along with ComplaintType and ComplaintDescription.	Complaint entity is connected to the Customers. **<u>The relations are:</u>** - Complaint has Zero/Many to One relationship with Customers

Entity Relationship Diagram