# **­­­­­­**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | |  |  |  |  | **Assignment Cover Sheet** | | | | |  |
|  | |  | |  |  |  |  |  |  |  |  |  |  |
|  | |  | | **Qualification** | |  |  |  | **Module Number and Title** | | | |  |
|  | |  | |  |  |  |  | |  |  | |  |  |
|  | |  | | HD in Software Engineering | | | | | CSE4005/Database Design Development | | | |  |
|  | |  | |  |  |  |  | |  |  | |  |  |
|  | |  | | **Student Name & No.** | | | | | **Assessor** | | | |  |
|  | |  | |  |  |  |  | |  |  | |  |  |
|  | |  | | Umesh Sathsarana Madadeniya  KD/HDCSE/CMU/64/28 | | |  | | M.r Induranga De Silva | | | |  |
|  | |  | |  |  |  |  |  |  |  | |  |  |
|  | |  | | **Hand out date** | |  |  |  |  | **Submission Date** | | |  |
|  | |  | |  |  |  |  |  |  |  | |  |  |
|  | |  | | 21/08/2023 |  |  |  |  |  | 07/09/2023 | |  |  |
|  | |  | |  | |  | | |  |  | | |  |
|  | |  | | **Assessment type** | | **Duration/Length of** | | |  | **Weighting of Assessment** | | |  |
|  | |  | | Reports 3000 Words | | **Assessment Type** | | | 100% | |  |  |  |
|  | |  | |  |  | End of the Module | | |  |  |  |  |  |
|  | |  | |  |  |  |  |  |  |  |  |  |  |
|  | |  | |  |  |  |  |  |  |  |  |  |  |
| **Learner declaration** | | | | | | | | | | | |
| I, ……… Umesh Sathsarana Madadeniya (KD/HDCSE/CMU/64/28)…………………………………., certify that the work submitted for this assignment is my own and research sources are fully acknowledged. | | | | | | | | | | | |
| |  |  |  |  | | --- | --- | --- | --- | | **Marks Awarded** | | | | | First assessor | |  | | | IV marks | |  | | | Agreed grade | |  | | | Signature of the assessor |  | Date |  | | | | | | | | | | |

**Feedback Form**

**International College of Business & Technology**

**Module :**  Database Design and Development /Design a Database Management System for ABC car rentals

**Student :**

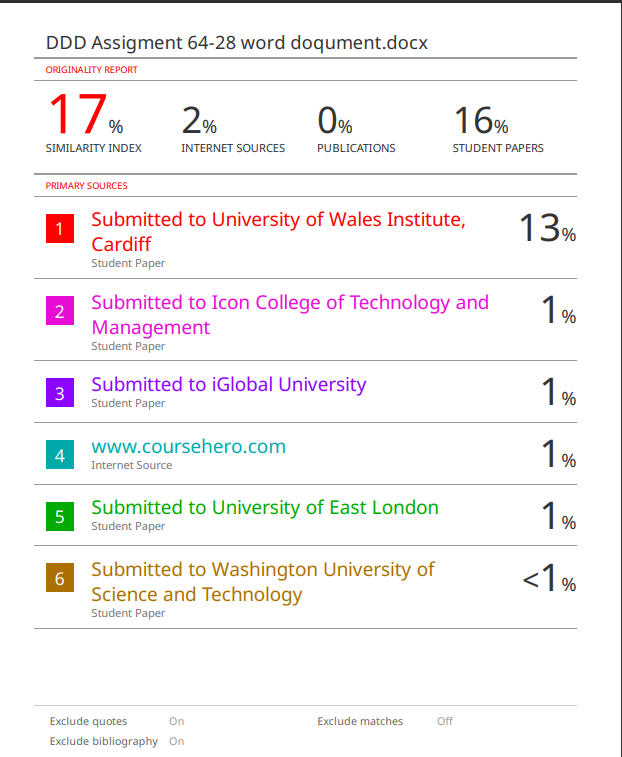
**Assessor :**

**Assignment :**

**Strong features of your work:**

**Areas for improvement:**

**Marks Awarded:**



Contents

[Introduction 6](#_Toc145010847)

[1.WHAT IS A DATA MODEL? 6](#_Toc145010848)

[TYPE OF DATA MODELS 6](#_Toc145010849)

[2 12](#_Toc145010850)

[3 ABC CAR RENT Company ER Diagram 16](#_Toc145010851)

[4. Relationship Schema 16](#_Toc145010852)

[6 SQL Queries 17](#_Toc145010853)

[7.Test case and test plan 17](#_Toc145010854)

[TEST CASE 18](#_Toc145010855)

[TEST Plan 23](#_Toc145010856)

[8 23](#_Toc145010857)

[Data quality control 24](#_Toc145010858)

[References 25](#_Toc145010859)

# Introduction

First of all, what is a data model, the types of data models and why older data models replaced to new data models were explained Approaches in task 2. Task 3 drew the ER table. Relation schema and mapping in task 3,4. Task 5 and 6, add the coding and SQL queries. At 7and 8 tasks add test plan and test case.

## 1.WHAT IS A DATA MODEL?

We must define data clearly before getting into data model. Data is made up of information that is collected saved, tested, and use to inform future decisions. We generate enormous amounts of data in today’s digital environment, and if it is to have any use, it must be processed effectively. In this situation, a data model is useful.

An information system or database’s usage of data structures to manage and organize data is referred to as having data model. It outlines the connection between the data elements, which stand in for actual physical objects, as how they are arranged, saved, and retrieved. It can be useful to picture the data model as the foundation upon which the entire system will be built.

Database and data warehouses are built using data models, which are also used to organize data for analytical processing and build application that give consumers meaningful access to information. (Suszterova, 2023)

There are various sorts of data models, including hierarchical, network, relational, object-oriented, and entity relationship models, each with its own strengths and weaknesses. Picking the right data model depends on the specific needs and requirements of the organization or project.

### TYPE OF DATA MODELS

* Logical data model;
* Physical data model;
* Hierarchical Model
* Network Model
* Entity-Relationship (ER) model
* Dimensional Data Mode
* Object-oriented data model

There are several types of data models, each with its own characteristics and use

* Logical data model:

The second descriptive data model is represented by the logical data model. The logical model will be used to further construct these entities and will also specify their relationships and attributes. The logical model improves the conceptual model's overall structure while avoiding database specifics because the model may be used to a variety of database systems and products.

* Physical data model

Before building a database, the physical data model is the last level of modeling; other models are thought of as entities, relationships, and attributes. In this data model, columns, keys, and tables are important. It must to be comprehensive enough to enable software developers and hardware engineers to construct the real database architecture required to support the utilizing applications.

* Hierarchical Model

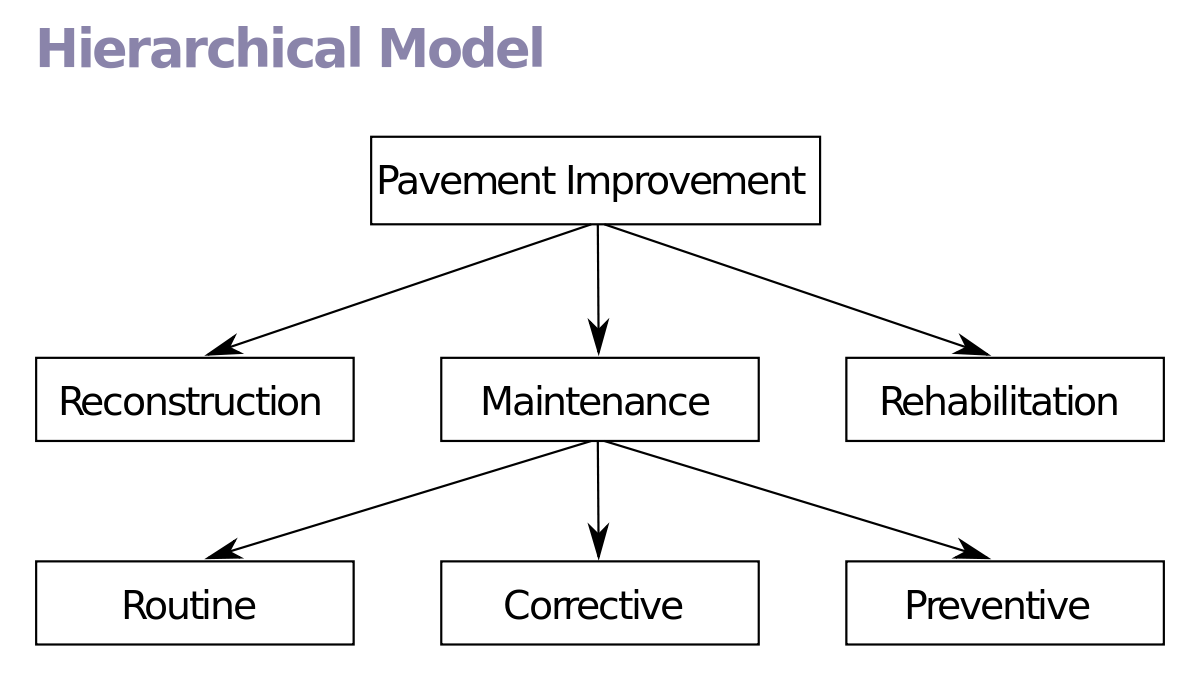
Each recode in the hierarchical model has a single parent record or base, which gives the data a tree-like shape. Records from siblings are arranged in a specific structure. This arrangement serves as the database’s physical storage order. Many relationships in real life can be accurately described by this approach.

Figure Hierarchical Model

* Network Model

The hierarchical approach is comparable, but it allows for more complex connections among records. A record can have more than one parent record and child record in this approach.

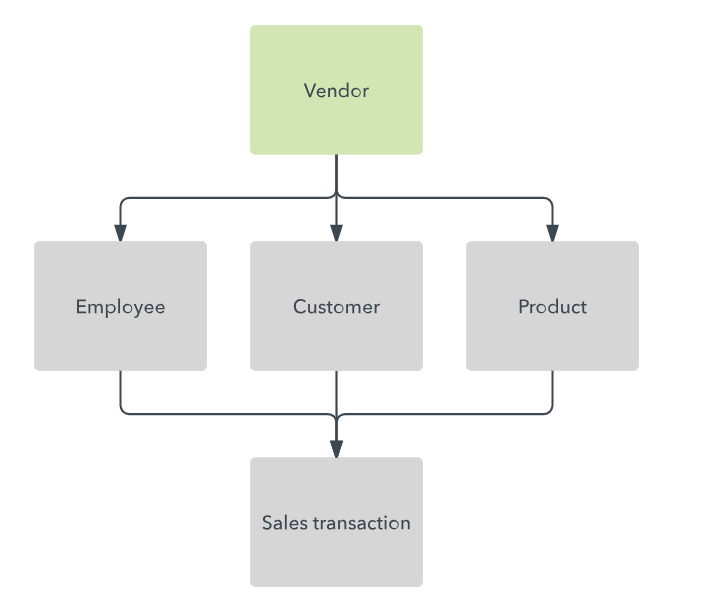


Figure network Model

* Entity-Relationship (ER) model

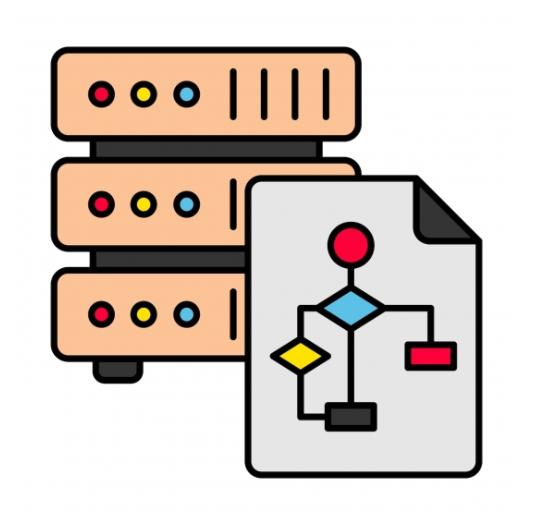
The Structure of a company can be illustrated by the entity-relationship model. To represent activities, functions or “entities” and relationships, dependencies, or “relationships”, consequently it has boxes with a wide range of forms and patterns. (Kimachia, 2022)

Figure Entity-Relationship (ER) model

* Dimensional Data Model

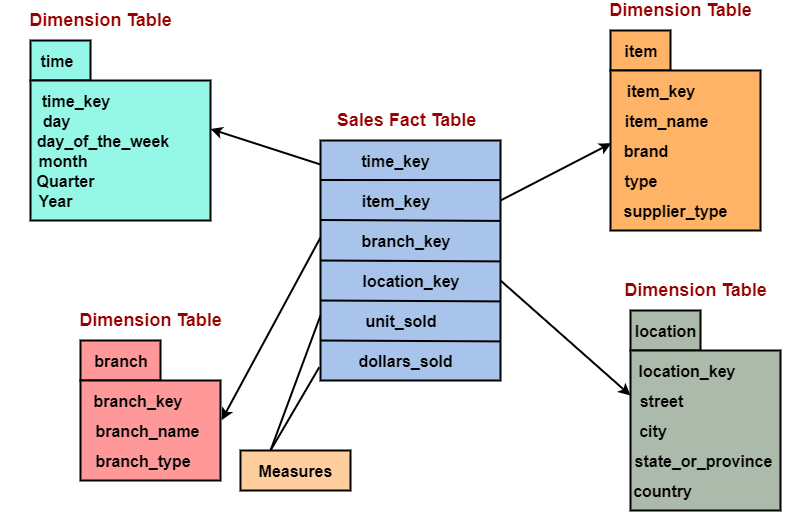
Application for data warehouse and business intelligence frequently use the dimensional data model. Fact tables and dimensional tables are used to organize the data. Dimension tables give context and descriptive features for the facts while fact tables contain numerical measurements. This model is enhanced for reporting and analytical queries. (sap.com, n.d.)

Figure Dimensional Data Mode

* Object-oriented data model

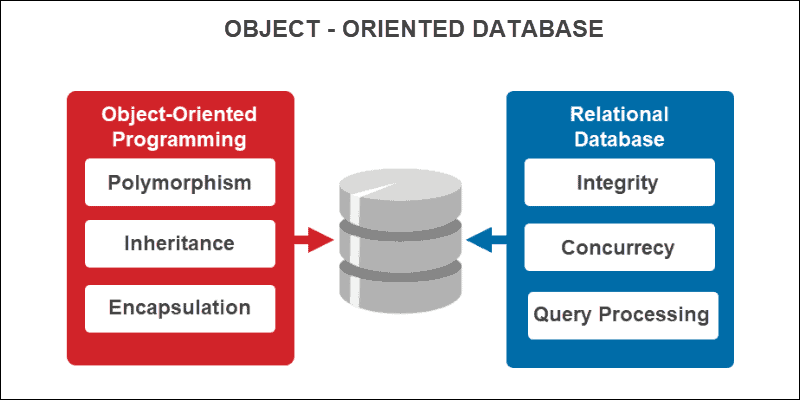
The object-oriented data as objects with attributes and methods that have attributes and behaviors. It is frequently used in object-oriented programming languages like Java and C++ to simulate complex data structures. When dealing with complicated data linkages and transmission, this model is quite helpful. (sap.com, n.d.)

Figure Dimensional Data Mode

Advantage and disadvantage in the data models

|  |  |  |
| --- | --- | --- |
| Data Model | Advantage | disadvantage |
| Hierarchical Model | * Effective at displaying hierarchical * Good for file systems and specific XML data types * Simple and clear for particular usage situations | * Constraints on how complicated data relationships can be represented are a drawback * The handling of connected or graph-like data is insufficient |
| Network Model | * Conceptional simplicity * Capability to handle more relationship types * Ease of data access | * System difficulty * Operational Errors * Structural independence is lacking |
| Entity-Relationship (ER) model | * Widely accepted and used * Provides data normalization and integrity * A powerful query language is provided by SQL | * Handling of unstructured or semi-structured material with limited flexibility * Not the best choice for intricate hierarchical data structures * Huge dataset performance problems. |
| Dimensional Data Mode | * Data Structure that is Simple * Analysis Effectiveness * Friendliness of Analysis * Scalability and flexibility (Suszterova, 2022) | * Redundancy in Data * Small-scale transaction processing: * Detailed Updates * Having trouble integrating data (Suszterova, 2022) |
| Object-oriented data model | * increased output from software development * Representation of complicated data structures efficiently * increased modularity and reuse of code * Being adaptable and flexible | * A learning curve and complexity * Limited SQL support * Limited vendor assistance * Possibility of vendor lock-in |

Let’s discuss the causes for the replacement of older data models by more modern ones.

1. Complexity

Older data models, such as hierarchical and network models, were frequently hard and complex, which made them difficult to use, particularly for modern, dynamic data needs.

1. Lack of flexibility

Older models found It difficult to respond to shifting data structures.

1. Limited Querying Capabilities

Finding information and analysis were made inefficient by the absence of strong query languages and indexing techniques in many models.

1. Normalization and data integrity

Relational models’ use of normalization techniques enhanced data integrity and eliminated redundancy which were frequently lacking in earlier models.

1. User-Friendliness

Modern data models, such as the relational model and dimensional modeling, are more understandable and user-friendly enabling non-technical individuals to efficiently deal with data.

1. Scalability

Newer models are frequently managing massive amounts of data and perform complicated queries.

1. Compatibility

Older data models might not work well with modern technology, which makes them less appropriate for the networked systems and data sharing needs of toda

# 2

There are numerous different methods for designing databases. There are tow different ways for creating database designs among them. Which are

* Top-down approach
* Bottom-up approach
* Centralized database design
* De-centralized database design
* Top-down approach

This strategy moves from the general to the specific. The system’s overall concept is presented first, followed by the data that must be stored. It necessitates a thorough comprehension of the system. In some professions, such as project management and engineering, the top -down method is necessary. This method is employed in high-level projects that use ERDs (Entity Relationship Diagrams) and lack precise information. Next, try from top to bottom. It is possible for the analyst and end users to overlook something crucial for the system. For instance, before beginning to prepare the curry, consider the type of curry you want to prepare. If you discovered the term, you must include more precise information about the appropriate curry ingredients. (studocu.com, n.d.)

* Bottom-up approach

It starts with the particular and moves up to the general. To decide what information should be kept in a database, the system operates backwards. This provides an overview of all the system’s interfaces, including screens, reports, and forms. We need people who can work from the bottom up, like doctors and statisticians. This strategy is used particularly in little initiatives that we can effectively manage. OR instance, if you visit a hospital or clinic because you are ill or sick. He will also identify the sickness that is causing your symptoms. This indicates that he is employing a bottom-up strategy to determine “what disease you are infected with”. If the doctor has the information, he will describe your symptoms and disease. (studocu.com, n.d.)



Figure Process diagram of Top-Down and Bottom-Up Database Design

* Centralized database design

The database may contain a little amount of information for a small business with a narrow range of operations. As a result, the database design may be rather simple to understand and readily completed by a single designer or small team. This is referred to as centralized database design. The users can validate the conceptual schema to make sure the database satisfies their needs and processing requirements. The designer can analyze the system operations, determine the restrictions, and develop it

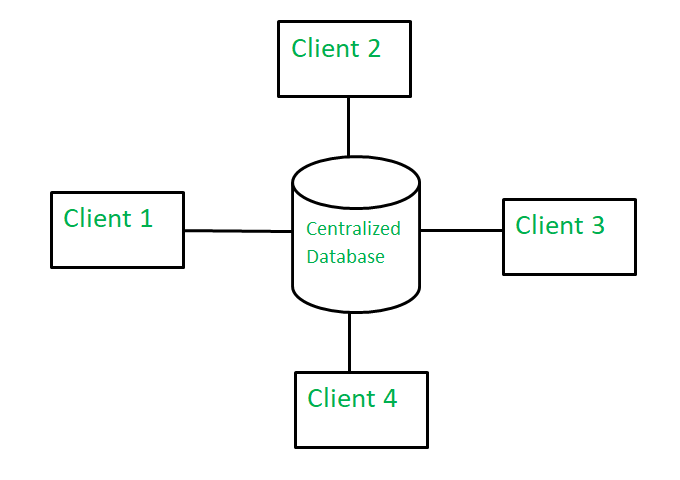


Figure Centralized database

* De-centralized database design

The database may contain a little amount of information for a small business with a narrow range of operations. Thus, the database design might be relatively simple. The database design may be handled by a division of labor when the database study indicates that the resulting database is for the “whole organizations” and that it contains a high number of entities and complicated relations on which highly complex operations are conducted. The research and creation of conceptual schema for each department or function for each department or functions for which the database is to be developed may be appropriate in this situation. The database design project broken into smaller modules, each of which is created by a different team of designer.



Figure Decentralized System

# 3 ABC CAR RENT Company ER Diagram

Figure ER Diagram

# 4. Relationship Schema

Figure Relationship

# 6 SQL Queries

* List of Borrowers of a particular car type

SELECT Cars\_Make, Cars\_Model, COUNT(Rentals.Customer\_ID) AS BorrowerCount

FROM Cars

LEFT JOIN Rental ON Car.Car\_ID = Rentals.Car\_ID

WHERE Car.Car\_Make = 'YourCarMake' AND Cars.Model = 'YourCarModel'

GROUP BY Cars.Make, Cars.Model;

* Sum of payments received with in a given duration

ELECT SUM(Bill\_Amount) AS TotalPayments

FROM Billing

INNER JOIN Rental ON Billing.Rental\_ID = Rental.Rental\_ID

WHERE Rent\_Date BETWEEN 'Rent\_Date' AND 'Return\_Date';

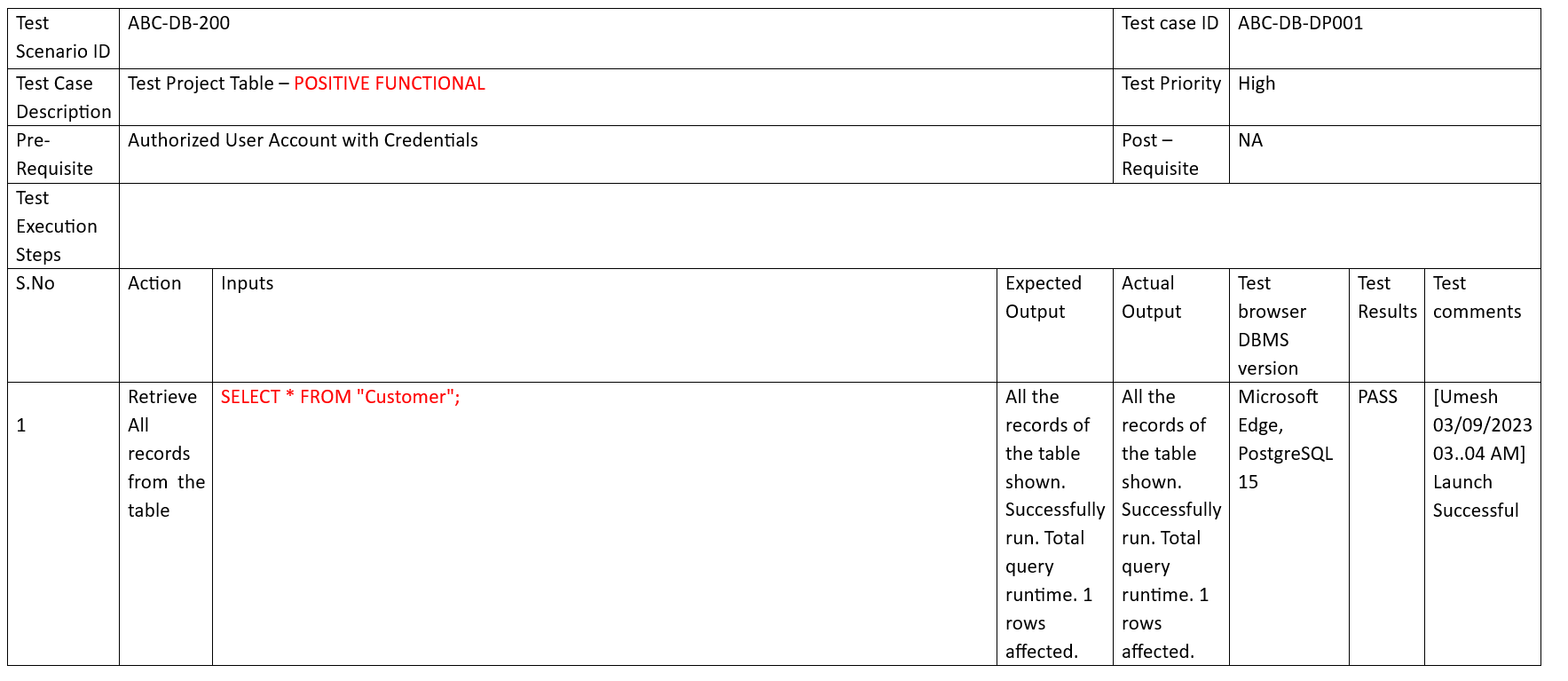
* List of car details that is in the fleet

SELECT Car\_ID, Car\_Make, Car\_Model, Car\_YearMade, Colour, CarLicense\_Plate FROM Car;

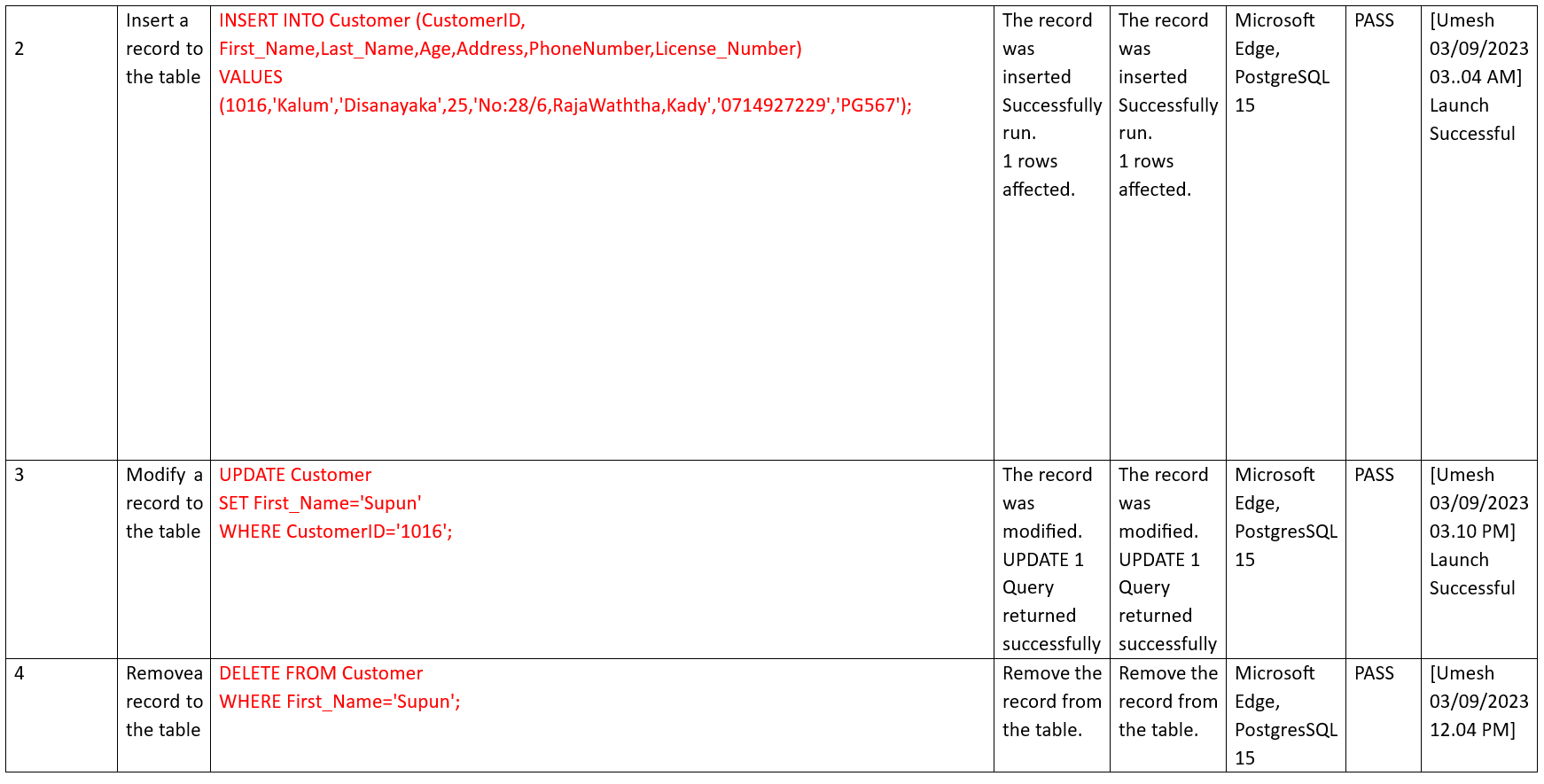
7.Test case and test plan

Here I created the database of ABC\_CAR\_COMPANY. Then I will check the test cases in this database

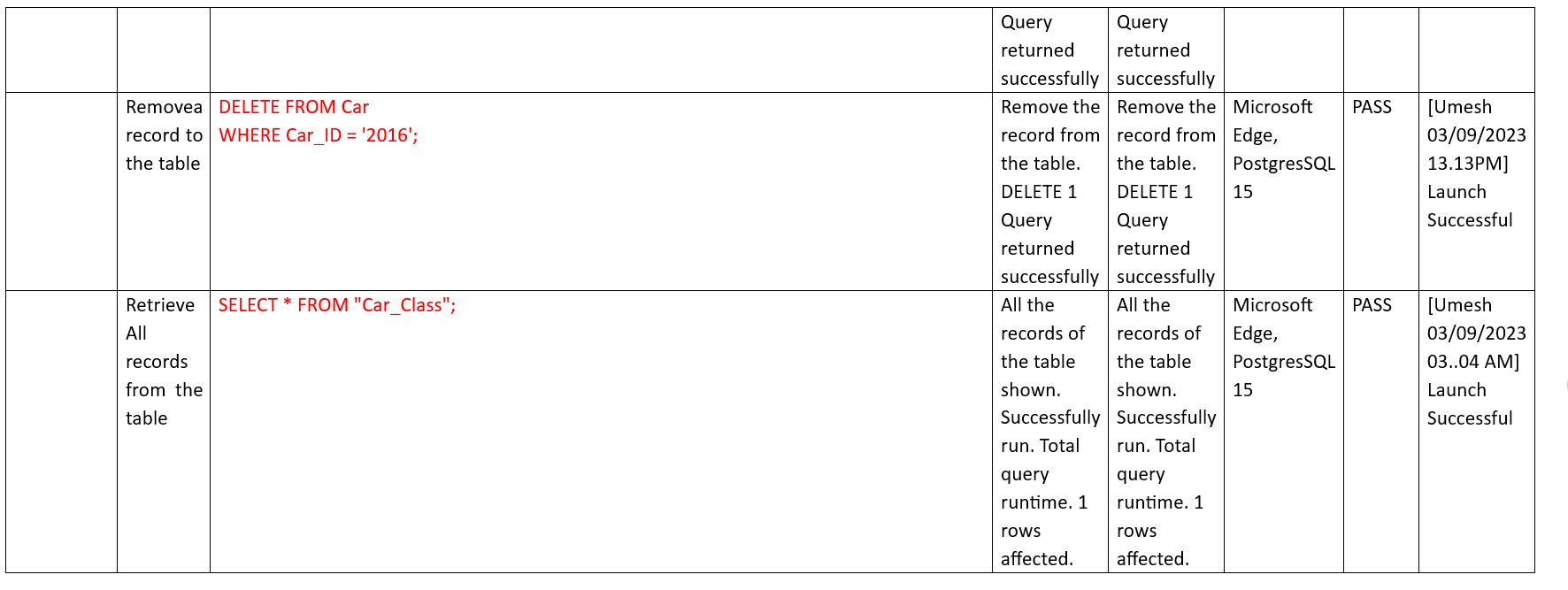
# TEST CASE



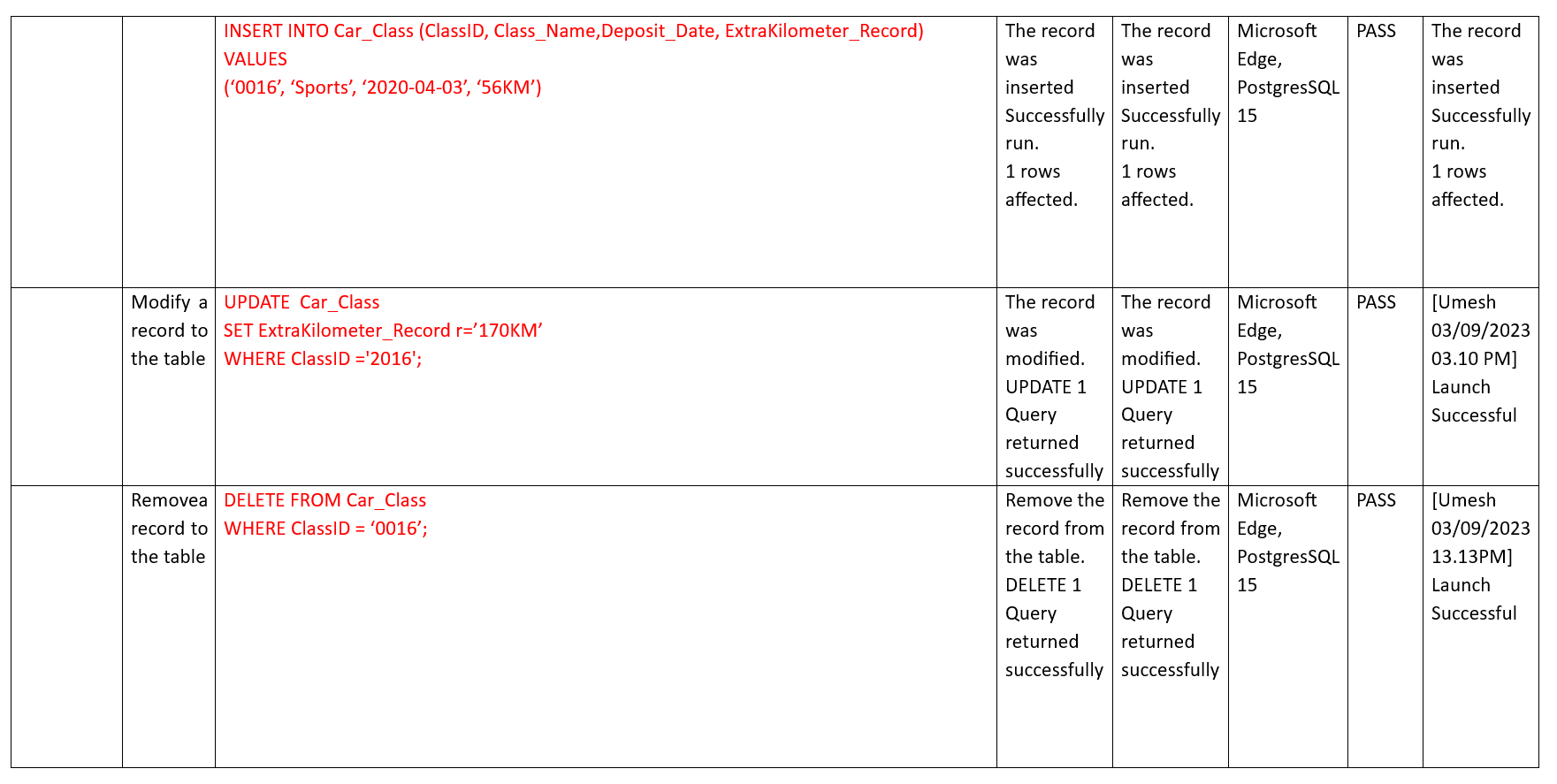
Figure



Figure



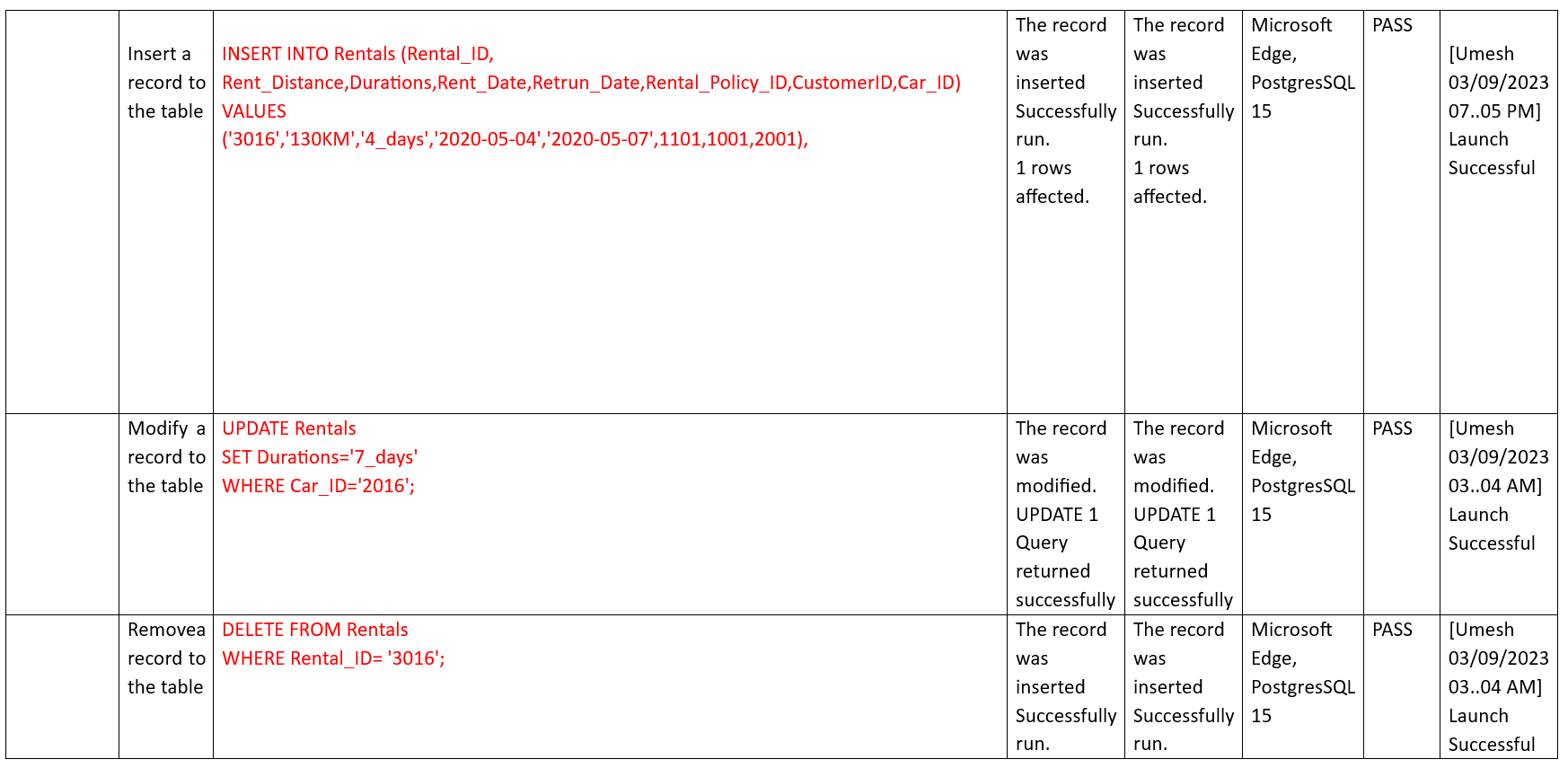
Figure



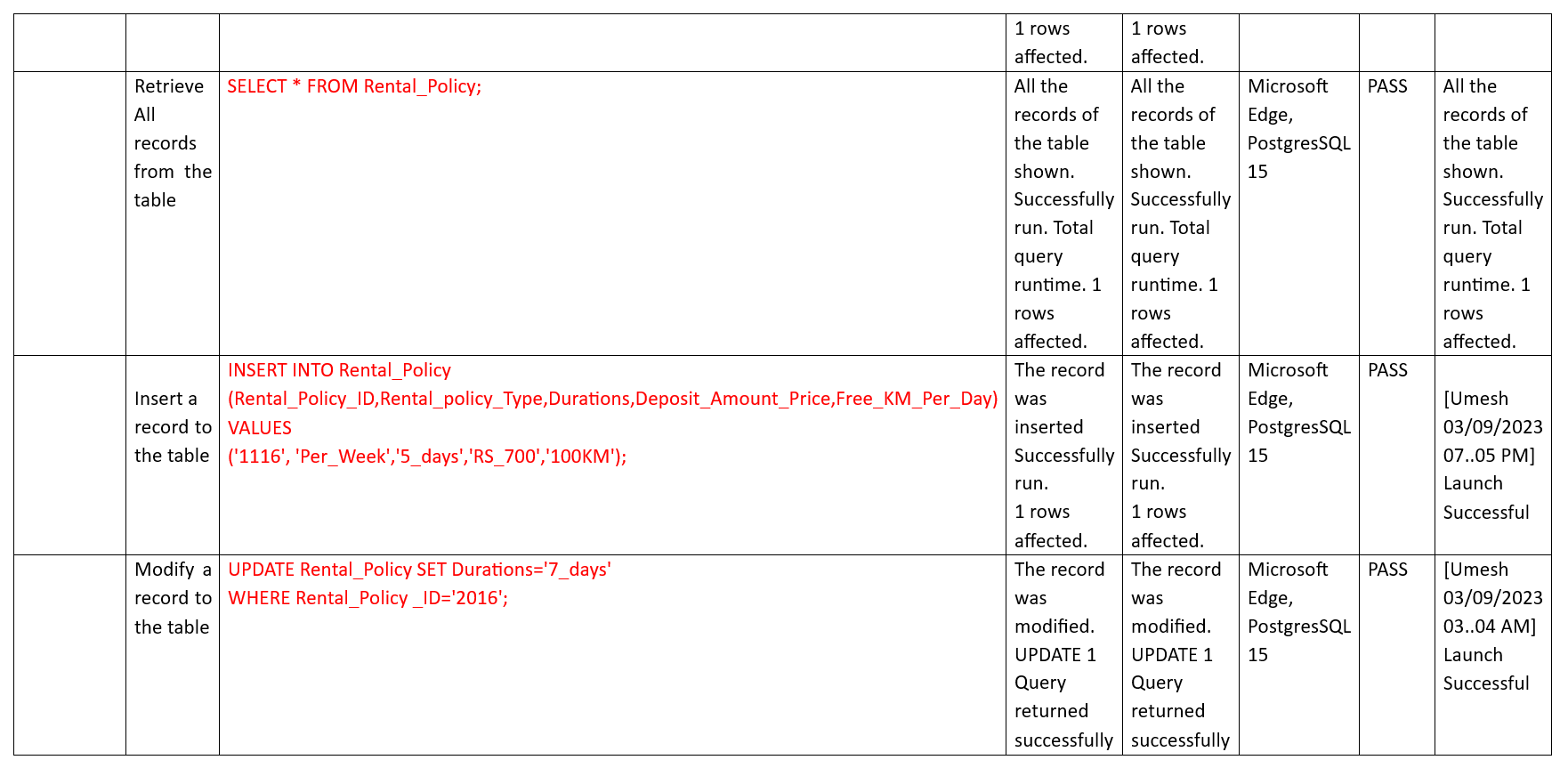
Figure



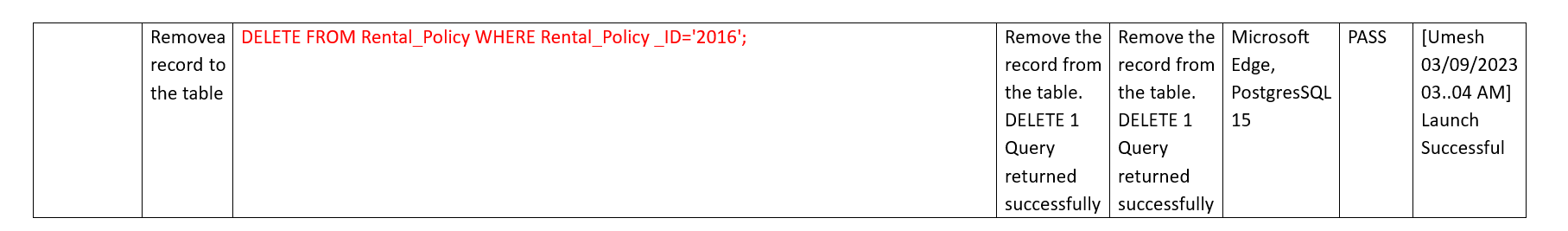
Figure



Figure



Figure



Figure

Test Case output

CREATE customer table

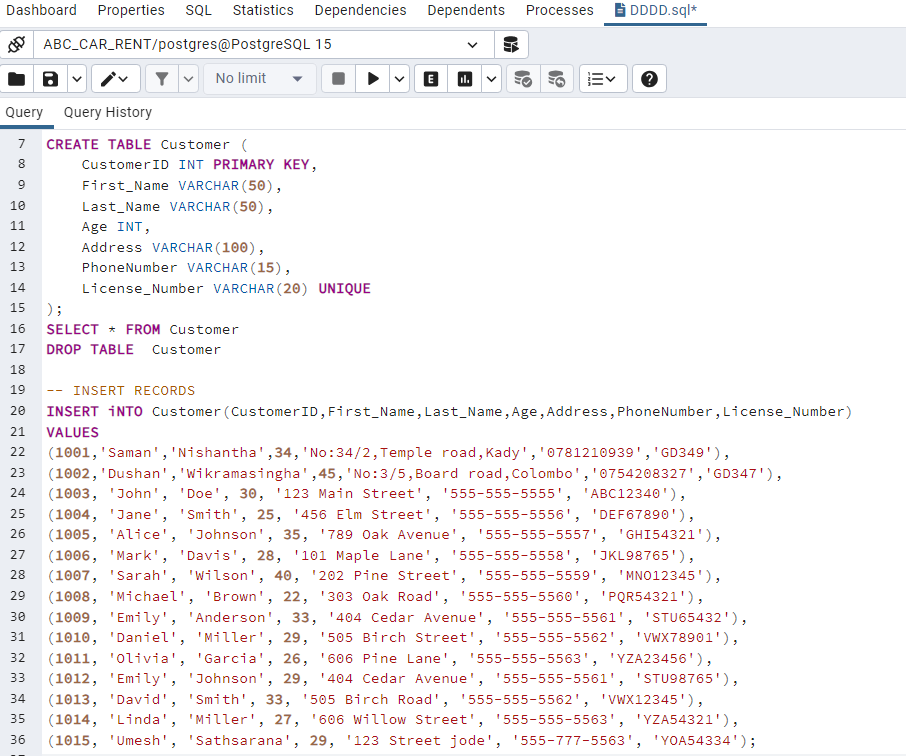


Figure Customer output

Insert Into Record

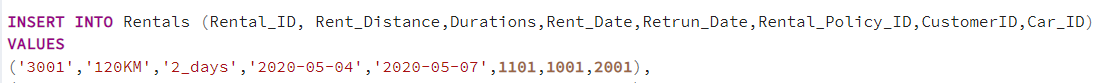


Figure Inset into record

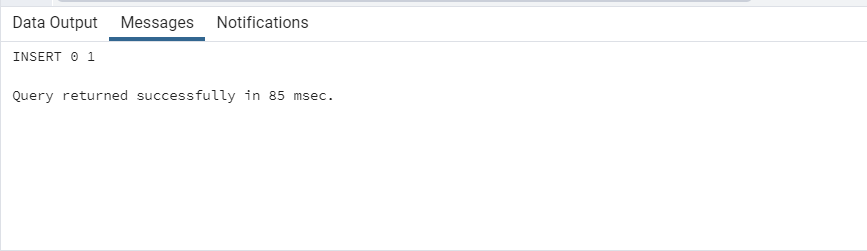
Successful insert record

Figure Successful insert record

Delete record

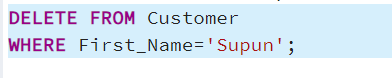


Figure Delete Record

Successful Delete record

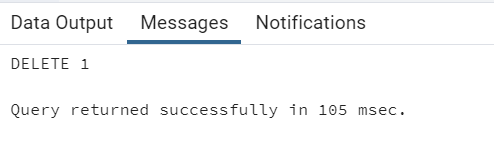


Figure Successful Delete Record

# TEST Plan

You may see the database for the ABC Car Rental Company here. This is the test strategy for a database table I created specifically for this company. I successfully ran the following queries against each table.

• SELECT \* FROM Function for Positive Functional

• Insert "positive functional" into the function

• UPDATE Function - Positive Functional

• DELETE Function - Positive Functional

# 8

A database schema and tables are established for a car rental system in this SQL code. The verification and validation processes for the data in the database are not explicitly described, even after the database structure is constructed and the data is entered. Through various tests and limitations, verification and validation often entail assuring data consistency and integrity. This can be accomplished using SQL constraints, triggers, and appropriate data validation in application code. A full description of how you can handle authentication and validation in your database, with different users having access rights, is provided below:

Data verification

* To guarantee that the data entered is in the right format, use the proper data types for the columns.
* To check data against particular rules, use check restrictions. You can make sure, for instance, that age ranges are appropriate or that phone numbers have a particular format.
* Use foreign key constraints to enforce referential integrity and make sure that data in one table matches data in another (for instance, CustomerID in Rentals matches CustomerID in the Customer table).

## Data quality control

In fields where uniqueness is required, such as License\_Number in the Customer record or CarLicense\_Plate in the Car table, implement unique constraints.To guarantee that relevant fields are filled up, use default values for columns or use triggers to set default values.

Access Management:

* Create user roles and permissions to manage database access.Based on user roles, assign SELECT, INSERT, UPDATE, and DELETE access to particular database objects (tables, views).
* When performing routine database tasks, avoid utilizing the root or superuser account; instead, make distinct accounts for each user or application that have fewer privileges.

# References

Kimachia, K., 2022. *techrepublic..* [Online]   
Available at: https://www.techrepublic.com/article/types-of-data-models/

sap.com, n.d. *sap.* [Online]   
Available at: https://www.sap.com/products/technology-platform/datasphere/what-is-data-modeling.html

studocu.com, n.d. *studocu.* [Online]   
Available at: https://www.studocu.com/en-gb/document/university-of-greenwich/introduction-to-database-systems/db-design-approach/40966039  
[Accessed 02 septhamber 2023].

Suszterova, S., 2022. *gooddata.* [Online]   
Available at: https://www.gooddata.com/blog/relational-dimensional-data-models/  
[Accessed 2 septhamber 2023].

Suszterova, S., 2023. *gooddata.* [Online]   
Available at: https://www.gooddata.com/blog/what-a-data-model/  
[Accessed 29 agust 2023].