



INTERNATIONAL SCHOOL OF MANAGEMENT AND TECHNOLOGY

FACULTY OF COMPUTING

ASSIGNMENT COVER SHEET

This form is to be completed by students submitting assignments of level 4 and level 5. Students are required to complete all sections and attach to your assignment.

STUDENT DETAILS	
STUDENT NAME	Krishna Prasad Bajgai
STUDENT ID	

UNIT AND ASSIGNMENT DETAILS			
UNIT TITLE	Database Design & Development		
UNIT NUMBER	A/618/7400		
ASSIGNMENT TITLE	United Limited IT Systems		
ISSUE DATE	12/05/2023	DUE DATE	20/07/2023
ASSESSOR NAME	Rupak Chaudhary		
ESTIMATED WORD LENGTH			

SUBMISSION	
HAND IN DATE	20/07/2023

DECLARATION AND ACKNOWLEDGEMENT

When submitting assignments, each student must sign a declaration confirming that the work is their own.

Plagiarism and Collusion

Plagiarism: to use or pass off as one's own, the writings or ideas of another without acknowledging or crediting the source from which the ideas are taken.

Collusion: submitting an assignment, project or report completed by another person and passing it off as one's.

In accordance with the Academic Integrity and Plagiarism Policy:

1. I declare that:

- a) this assignment is entirely my own work, except where I have included fully-documented references to the work of others,
- b) the material contained in this assignment has not previously been submitted for any other subject at the University or any other educational institution, except as otherwise permitted,
- c) no part of this assignment or product has been submitted by me in another (previous or current) assessment, except where appropriately referenced, and with prior permission from the Lecturer / Tutor / Unit Coordinator for this unit.

2. I acknowledge that:

- a) if required to do so, I will provide an electronic copy of this assignment to the assessor;
- b) the assessor of this assignment may, for the purpose of assessing this assignment:
 - I. reproduce this assignment and provide a copy to another member of academic staff;
 - II. communicate a copy of this assignment to a plagiarism checking service such as Plagiarism Check (which may then retain a copy of this assignment on its database for the purpose of future plagiarism checking).

I am aware of and understand that any breaches to the Academic Code of Conduct will be investigated and sanctioned in accordance with the College Policy.			
SIGNATURE	Krishna	DATE	

International School of Management and Technology Chitwan

Naurangi Chowk, Bharatpur-12, Chitwan



A report of Relational Database System of United Limited

Submitted By: Krishna Prasad Bajgai

First Semester

Submitted To: Rupak Chaudhary

Ismt College Chitwan, Faculty

.....**Error! Bookmark not defined.**

Introduction	10
.....	10
Task 1	10
Introduction	10
User Requirement	11
System Requirement	12

Relational Database System	13
Data Dictionary	14
Entity Relationship Diagram	19
Enhanced Entity Relational Diagram	20
NORMALIZATION	21
Advantages of Normalization	22
Disadvantages of Normalization	22
Types of Normalization.....	22
1 st Normal Form (1NF).....	22
2 nd Normal Form (2NF).....	23
3 rd Normal Form (3NF).....	25
Interface Diagram.....	26
Data Validation mechanism	30
Data Type Check	31
Length Check	31
Format Check	31
Conclusion	32
Task 2	32
DBMS	32
Implementation of query against help-desk system	33
Creating Database	33
Use Database	34
Creating Table Users	35

Creating Table Callers	36
Creating Table Operators	37
Creating Table specialists	38
Creating Table Calls	39
Creating Table Problem	40
Insert Data into Users table	41
Insert data into Callers Table	42
Inserting Data into Operators Table	43
Inserting data into Specialist table	44
Insert Data into Call Table	45
Insert Data into Problem Table	46
Retrieving Data	48
For Problem Table	48
For Calls table	49
For Specialist Table	49
For Operator Table	50
For Callers Table	51
For Users Table	51
Updating Data	53
Delete data	53
For Security and Maintenance	54
Sort the data	54
Joins	56

Types of Joins	57
Inner Joins	57
Left Joins	58
Right Joins	59
Full Outer Joins	60
Union	61
Trigger	62
Store Procedure	64
Store Procedure on save	65
Index	67
Group By	68
BackUp	69
Create and Implement a test plan with effective tests.	70
Test plan	70
Why test plan is necessary for United Limited ?	70
Testing	71
Unit Testing	71
Integration test	72
Stress test	72
Data Type Check	75
Data Length Testing	78
Not Null Testing	80
Recovery Testing	82

Primary Key testing	84
Performance Testing	85
Evaluate Database effectiveness according to user needs and make recommendations for improvement.....	87
Querying process of Selecting	87
Update callers Set CallerName	89
Update Specialist set SpecialistName	90
Effectiveness of testing with explanation of choice of the test data used	92
Effectiveness of unit testing:	93
Conclusion:	93
Introduction	94
User Documentation	95
Start SQL server management studio	95
Connecting SQL	95
Right Click on Database	97
Backup Database	102
Delete Database	104
Restore Database	106
References:	113

Introduction

United Limited is a renowned company in the real estate sector, specializing in residential, commercial, and industrial properties. To support the company's growth and expansion, an IT helpdesk has been established to address hardware and software issues related to IT systems. The IT department provides comprehensive support across various professional services, including property development, property management, brokerage, financing, legal, interior design, and construction. As part of my role, I work closely with the IT team to understand the specific requirements of the company. I then design and build database systems that align with these requirements. This involves creating an efficient and organized structure for storing and retrieving data. To optimize the performance efficiency of the database systems, I implement various techniques such as indexing, query optimization, and database normalization. This ensures that data can be accessed and manipulated quickly and accurately. In addition to designing and developing the databases, I also perform thorough testing to identify and resolve any issues or bugs. I work closely with the IT helpdesk to troubleshoot problems related to the database systems and provide timely resolutions. Overall, my role as a database developer at United Limited involves designing, developing, and implementing efficient database systems, optimizing their performance, conducting testing and troubleshooting, and resolving any issues that may arise.

Task 1

Introduction

I have been assigned the responsibility of crafting a thorough and customized business proposal specifically designed for United Limited as a crucial part of my assignment. This proposal aims to introduce a comprehensive and efficient relational database design that effectively caters to the IT help-desk needs of the organization, ensuring it is properly organized and operational. In order to achieve this goal, the proposal will delve into a diverse range of key areas that need to be covered extensively.

Firstly, the proposal will extensively discuss the various user and system requirements that are crucial for the successful implementation of the IT help-desk solution. This includes identifying the specific needs of both the end-users and the internal systems,

ensuring that all necessary functionalities and features are incorporated into the database design.

Next, the proposal will focus on the development of an entity-relationship diagram (ERD) that will serve as a blueprint for the database structure. The ERD will provide a detailed visual representation of the connections between various entities and attributes.

Furthermore, the proposal will delve into the importance of normalization strategies in ensuring the efficiency and integrity of the database. Different normalization techniques will be explored, aiming to eliminate data redundancy and anomalies while maintaining data consistency.

Additionally, the proposal will outline various data validation approaches to guarantee the accuracy and reliability of the information stored within the database.

Moreover, the proposal will address output design considerations, emphasizing the significance of designing intuitive and user-friendly interfaces for data retrieval and reporting. The proposal will explore different techniques to present data in a clear and organized manner, enhancing the overall user experience.

Lastly, the proposal will include interface diagrams that illustrate the interaction between the IT help-desk solution and other systems or modules within the organization. These diagrams will highlight the integration points and data flows, enabling a comprehensive understanding of the overall system architecture.

User Requirement

User requirements encompass the specific preferences and necessities articulated by stakeholders concerning the system. They encompass the expectations and viewpoints of users concerning the system's functionality and efficiency. I endeavored to comprehend the perspectives of users to gain a deeper understanding of the desired operation of the company's system. The ensuing list comprises the requests made by users regarding the requirements:

The ability to retrieve personal data and document specific details of future calls.

The system should facilitate the seamless retrieval of previous question types and enable the assignment of subject-matter experts to address them.

The ability to register device information and link it to specific issues.

User Requirements	
Requirements	Description
Login System	Administrator can login directly as a default user. Other users will be created for data entry only or both (owner & data entry) privileges.
Product	The system should be able to store product information
Supplier	The system should be able to store supplier information
Elements	The system should be able to store elements information and supplier can be one or more in element area.

Table 1 : User Requirement Table

System Requirement

Subsequently, we consolidated the system requirement into a comprehensive chart that outlines the attributes, functionalities, and operational constraints of the software system.

Components	Descriptions
RAM	The requirement of RAM is 4 GB or above to run smooth and perform better.
System Capability	64 bits operating system, (x64) based processor,
Operating System	Windows server 7, windows 10 12H2, windows11, window server 12
Hard Disk Drive	2 GB free space
Processor	Intel, <u>AMD-Ryzen</u> , corei9 etc. to perform better.
Full set up	.net framework 4.0 or above and 1.8 size (Zip or .exe) file.

Table 2 : System Requirement Table

Relational Database System

A particular type of database called a relational database was created with the intention of storing and making accessible connected data points. It adheres to the relational paradigm, which makes use of tables to present data in an understandable and structured way. Each table has rows of related data that have sound structures. The columns in the table correspond to different types of data, such as names or emails, and each element inside a column has the same data type, such as text or integer. The combination of rows and columns forms a table.

Relational databases have taken the place of older hierarchical and network databases due to their ease of use and comprehensibility. Every entry within the table has its own unique key. It is customary to assign a descriptive name to each table based on its contents.

(Secoda.co, 2023)

Data Dictionary

A data dictionary serves as a centralized information hub regarding the data within an organization, business, or enterprise. The primary objective of a data dictionary is to establish a uniform technical language that can be comprehended by both technical personnel and business professionals. Additionally, a data dictionary can serve as an "audit trail," allowing tracking of who accessed or modified data within the database. This aids in identifying any unauthorized alterations that may have taken place.

A noteworthy feature of a data dictionary is its ability to establish cross-references between various variables and data elements. In the context of a database, it enables users to acquire more information about a particular variable by utilizing the cross-referencing system. Organizing data effectively is fundamental to facilitate straightforward data exploration. A data dictionary functions as a compilation of essential terms and metrics, essentially a business glossary. Despite its apparent simplicity, aligning the definitions across different business departments can be challenging.

(Secoda.co, 2023)

The table below displays the elements, along with their corresponding attributes and explanations.

User Table				
Column Name (Attributes)	Data Type	Length	Constrains	Description
UserID	Int	10	Primary Key	Unique identification of user identity
UserName	Varchar	100	Not Null	Require 100 characters data in various format
LastName	Varchar	100	Not Null	Store LastName of user
FirstName	Varchar	100	Not Null	Store FirstName of user
Password	Varchar	50	Not Null	Need 100 character data in various format

Caller Table

Field Name / Column Name /Name of Field/Table Field	Data Type	Length/Width	Constrains	Description
Caller ID	Int	10	Primary Key	To Identify unique value of Callers.
Job Title	Varchar	50	Primary Key	Unique identification of user identity
Department	Varchar		Foreign Key	
Equipment ID	Int		Primary Key	To Identify unique value of Equipment.
Address	Varchar	50	Not Null	Need 50 character data in various format
Email	Varchar	50	Not Null	Need 50 character data in various format

Problem Table

Field Name /Column Name/Name of Field/Table Field	Data Type	Length/Width	Constrains	Description
Problem ID	Int	10	Primary Key	To Identify unique value of Problems.
Caller ID	Int	10	Foreign Key	Make relation to Call Info
Problem Type	Varchar	50	Not null	Need 50 character data in various formats
Problem Description	Varchar	300	Not Null	Need 300 character data in various formats
SpecialistId	Int	10	Foreign Key	Make relation to Call Info Table

Call Table

Field Name /Column Name/Name of Field/Table Field	Data Type	Length/Width	Constrains	Description
Call ID	Int	10	Primary Key	Unique identification of user identity
Caller ID	Int	10	Foreign Key	Make relation to Caller info table
Operator ID	Int	50	Foreign Key	Required 50 characters of data in various formats.
Time	Time		Not Null	Store the time of the call
Date	date		Not Null	Store the time of the call
Duration	Time		Not Null	Store the duration of another call

Operator Table

Field Name /Column Name/Name of Field/Table Field	Data Type	Length/Width	Constrains	Description
Operator ID	Int		Primary Key	Unique identification of user identity
Specialist Name	Varchar	50	Not null	Required 50 characters of data in various formats.

Equipment Table

Field Name /Column Name/Name of Field/Table Field	Data Type	Length/width	Constrains	Description
EquipmentID	Int	'	Primary Key	To write unique serial number of equipment
Equipment Name	Varchar	100	Foreign Key	To Store equipment name of system

Specialist Table

Field Name /Column Name/Name of Field/Table Field	Data Type	Length/Width	Constraints	Description
Specialist ID	Int	10	Primary Key	To Identify unique value of Specialists.
Specialist Name	Varchar	50	Foreign Key	To Store specialist name of system
Solution Info	Varchar	300		

Entity Relationship Diagram

An Entity Relationship Diagram (ERD), also referred to as an ER Diagram, is a graphical representation that visually presents the interconnectedness and associations between sets of entities stored within a database. Essentially, ER diagrams serve as an elucidation of the logical structure inherent in databases. These diagrams are constructed using three fundamental concepts, namely entities, attributes, and relationships, to provide a comprehensive portrayal of the database's organization and connectivity. Within an ER Diagram, distinct symbols are employed to symbolize various elements: rectangular shapes are utilized to depict entities, oval shapes are employed to define attributes, and diamond shapes are employed to represent relationships.

Upon initial observation, an ER diagram may bear resemblance to a flowchart; however, it possesses a distinctiveness characterized by the inclusion of numerous specialized symbols and their respective meanings. The primary objective of an ER Diagram is to visually represent the foundational framework of entities that comprise a particular system.

The provided ERD has been successfully generated based on the depicted scenario in the accompanying image.

This diagram represents the ERD of United Limited, displaying multiple entities along with their respective attributes.

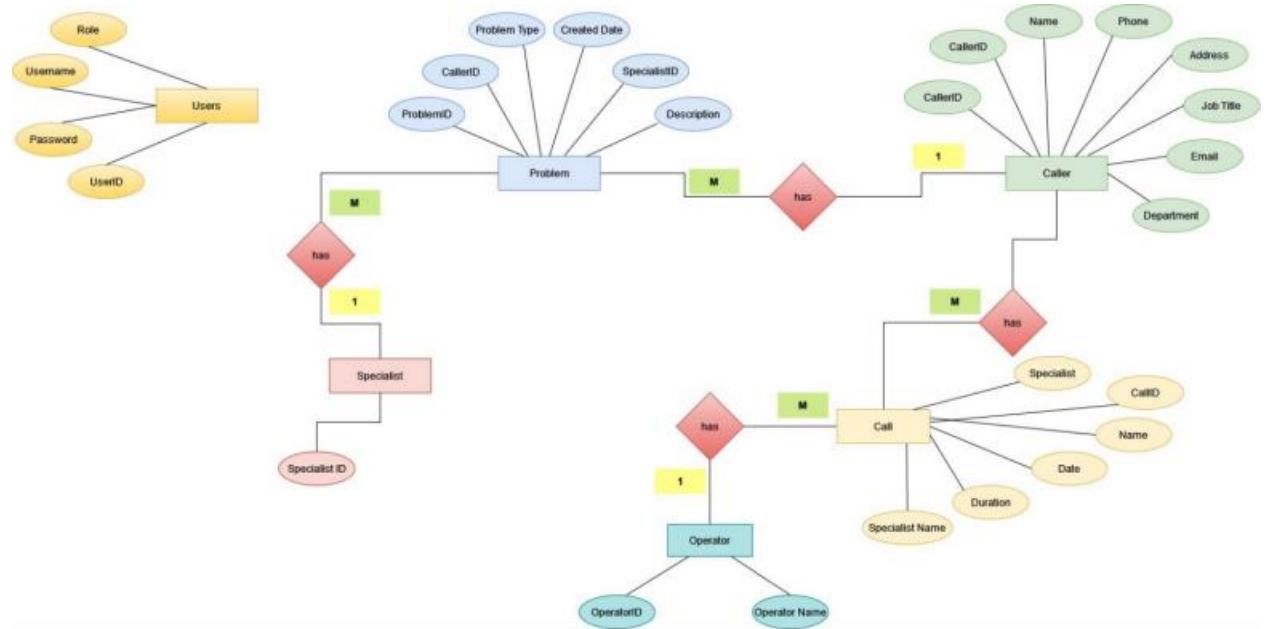


Fig 1: ERD Diagram

Enhanced Entity Relational Diagram

The Entity Relationship Diagram stands as a vital tool for database design, analysis, and troubleshooting. By visually representing the connections between entities, attributes, and relationships, ER Diagrams provide a concise and structured overview of a system's data structure. With their standardized symbolism and linguistic-like structure, ER Diagrams foster effective communication and collaboration among stakeholders. When used in conjunction with complementary diagrammatic techniques, they enable a comprehensive understanding of data flow within complex systems, facilitating informed decision-making and optimized database design.

(Lucidchart, 2023)

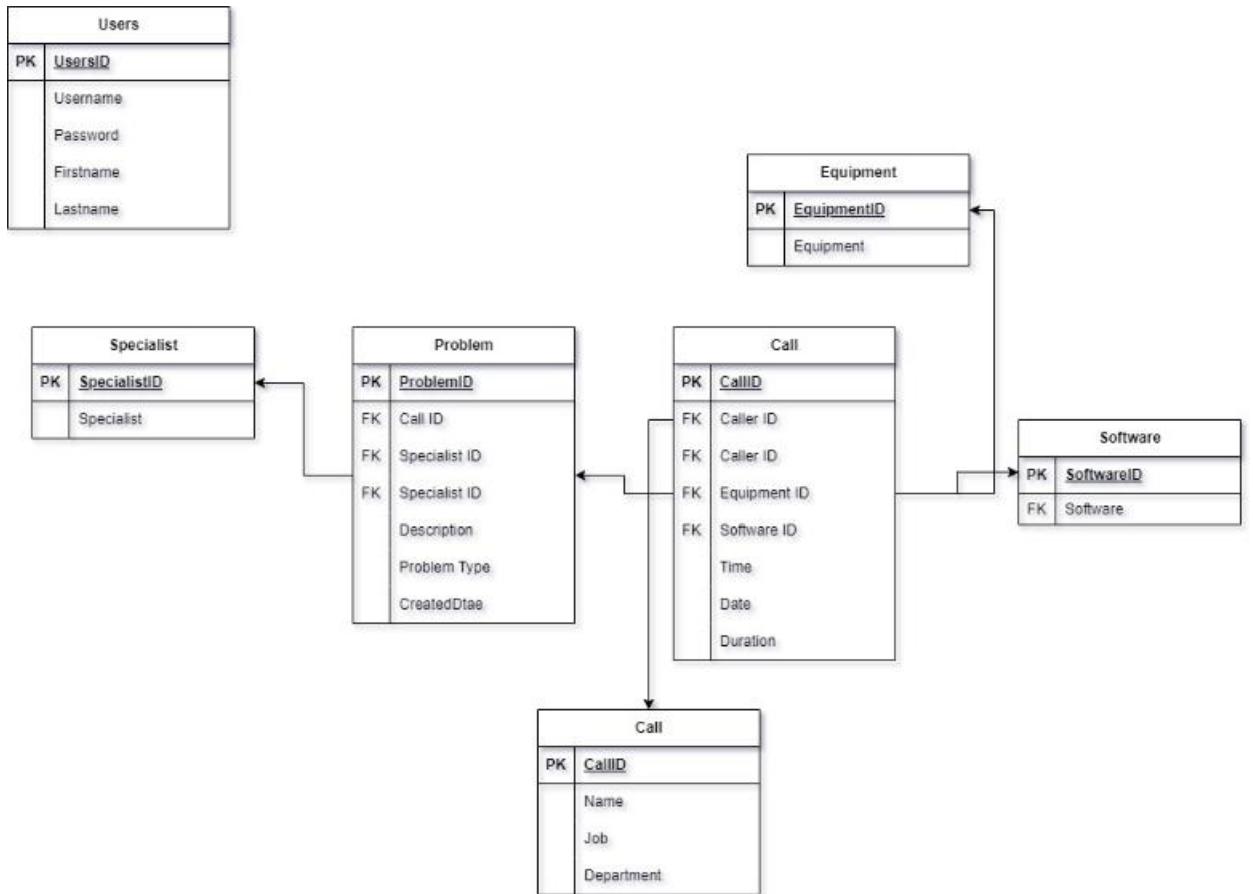


Fig 2: Relation Database Diagram

NORMALIZATION

Normalization is a procedure that involves identifying and removing redundancies and duplication's within tables, specifically in relation to multiple values. The main goals of normalization are to eliminate data anomalies that can lead to inconsistencies and to produce a logical and consistent data storage structure. A few normalization forms that offer guidelines for arranging data in a normalized fashion include the First Normal Form, Second Normal Form, Third Normal Form, and subsequent forms. The process of normalization holds great significance in the realm of database design as it plays a crucial role in enhancing database performance and scalability by promoting efficient data organization. If we want to reduce the redundancy, the main is primary key to foreign key. If we assign a primary key in one table and put that same primary key as foreign key in other table then, that table will derive the table details.

Advantages of Normalization

Normalization reduces data redundancy and improves data integrity, resulting in a more efficient and streamlined database. It simplifies database design, supports flexible queries, facilitates data updates, and ensures database scalability.

Disadvantages of Normalization

Normalization can lead to increased complexity, reduced flexibility, higher storage requirements, performance overhead, loss of data context, potential data update anomalies, and the need for expert knowledge in database design.

(GeeksforGeeks, 2020)

Types of Normalization

In DBMS, There are several types of normalization, each with its specific rules and guidelines for organizing data in a normalized manner. The types of Normalization:

- 1st Normal Form (1NF)
- 2nd Normal Form (2Nf)
- 3rd Normal Form (3NF)

1st Normal Form (1NF)

1st Normal Form (1NF) represents the fundamental level of normalization within a DBMS. Eliminate the repeating groups such as similar records in all tables will be become unique by primary key in each of the tables and the other tables are depend on the primary key.

For example,

Id	Caller Name	Job Title	Department	Problem
1	Krishna Nepal	Accountant	Administration	Software Crash Virus Threat
2	Ujjwal Bajgain	Receptionist	Front Desk	Internet Issue
3	Parbati Thapa	Manager	Administration	Monitor Replace

Solution,

Id	Caller Name	Job Title	Department	Problem
1	Krishna Nepal	Accountant	Administration	Software Crash
1	Krishna Nepal	Accountant	Administration	Virus Threat
2	Ujjwal Bajgain	Receptionist	Front Desk	Internet Issue
3	Parbati Thapa	Manager	Administration	Monitor Replace

The table has been transformed into 1NF, as all the data is now composed of atomic values, with each cell containing a single value, and there are no instances of repeating data groups.

1NF serves as the initial step in the normalization process, and advancing to subsequent normalization forms is essential to maintain data integrity and consistency.

2nd Normal Form (2NF)

The main key's non-key values are all completely functional. No partial dependencies are allowed. A partial dependency exists when a field is fully dependent on a part of a composite primary key.

Composite primary key is a candidate key that is when two primary keys in a table so one is primary key and another one is composite primary key.

For example,

Caller ID	Caller Name	Caller Email	Problem Type	Problem Description
1	Devendra Gurung	Devendra77@gmail.com	Hardware	Computer is not turning on
1	Devendra Gurung	Devendra77@gmail.com	Software	Website crashes
2	Dinesh Baral	Dinesh55@gmail.com	Software	Application won't install
3	Ram Chandra Bajgain	Ramchandra900@gmail.com	Hardware	Keyboard not responding
3	Ram Chandra Bajgain	Ramchandra900@gmail.com	Hardware	Printer is not working

Solution,

Caller ID	Caller Name	Caller Email
1	Devendra Gurung	Devendra77@gmail.com
2	Dinesh Baral	Dinesh55@gmail.com
3	Ram Chandra Bajgain	Ramchandra900@gmail.com

Caller ID	Problem Type	Problem Description
1	Hardware	Computer is not turning on
1	Software	Website crashes
2	Software	Application won't install
3	Hardware	Keyboard not responding
3	Hardware	Printer not working

The current design has achieved 2NF.

However, it is insufficient to guarantee data consistency and prevent data anomalies entirely. Therefore, progressing to subsequent normalization forms becomes necessary.

3rd Normal Form (3NF)

3rd Normal Form (3NF) extends the principles of 2nd Normal Form (2NF) by tackling the problem of transitive dependencies. In 3NF, a table should be free of any transitive dependencies. Transitive dependencies occur when a non-primary key column relies on another non-primary key column, rather than solely on the primary key. To attain 3NF, a table must already comply with the requirements of 2NF, and all non-primary key columns must exhibit direct dependency on the primary key.

Example,

Id	Caller Name	Job Title	Department
1	Raju Poudel	Accountant	Administration
2	Lila Bajgain	Receptionist	Front Desk
3	Devdas Thapa	Manager	Administration

Id	Problem
1	Software Crash
1	Virus Threat
2	Internet Issue
3	Monitor Replace

Solution,

CallerId	Caller Name	Job Title	Department
1	Raju Poudel	Accountant	Administration
2	Lila Bajgain	Receptionist	Front Desk
3	Devdas Thapa	Manager	Administration

ProblemId	Problem
1	Software Crash
2	Virus Threat
3	Internet Issue
4	Monitor Replace

The current design has achieved 3NF.

However, even though 3NF provides significant improvements, it is still insufficient to ensure complete data consistency and prevent all data anomalies.

(DatabaseTown, 2023)

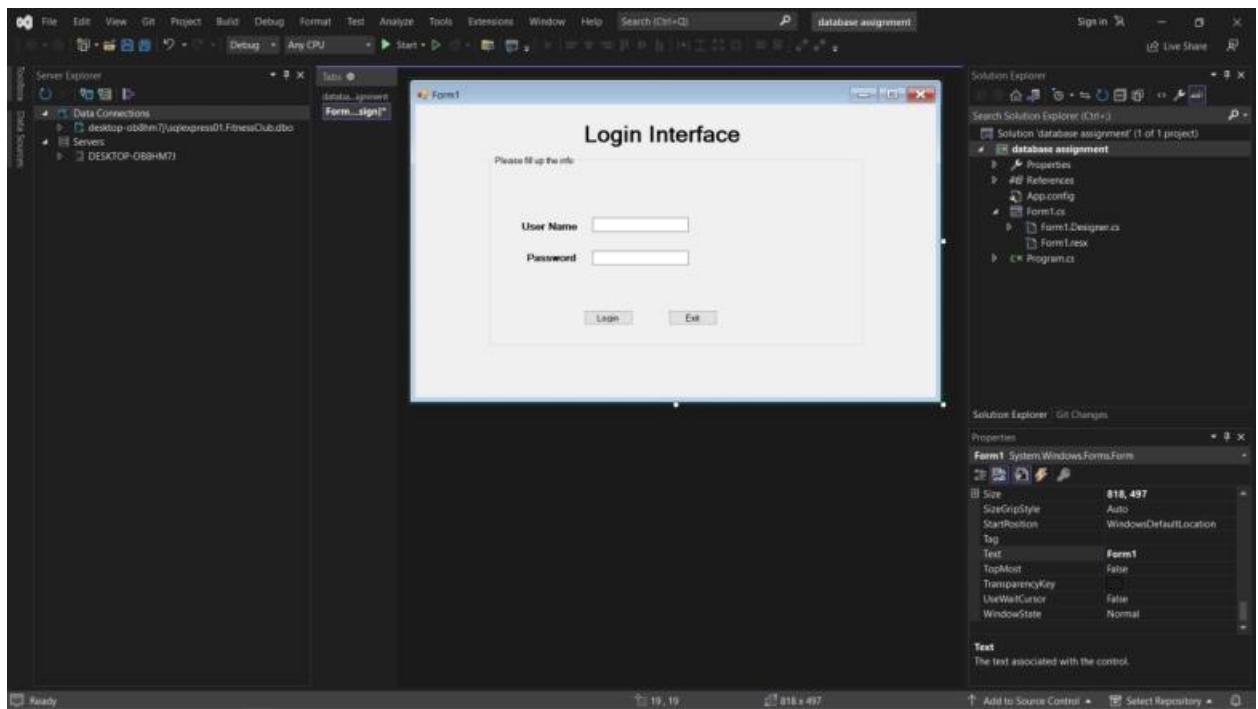
Interface Diagram

An interface diagram serves as a visual representation that depicts the design of an application. Its purpose is to assist United Limited in comprehending the architecture of the database before it is implemented in a practical setting. By evaluating the advantages and disadvantages of the design, an interface diagram offers an opportunity to make necessary adjustments before progressing further with development. Consider it akin to a blueprint for the design, as it showcases the user's perspective and enables developers to assess its usability, ultimately determining the most optimal course of action. This diagram holds significant importance in the design process as it facilitates clear communication among stakeholders and ensures that application development aligns with the user's requirements.

An example of interface diagram of United Limited's helpdesk:

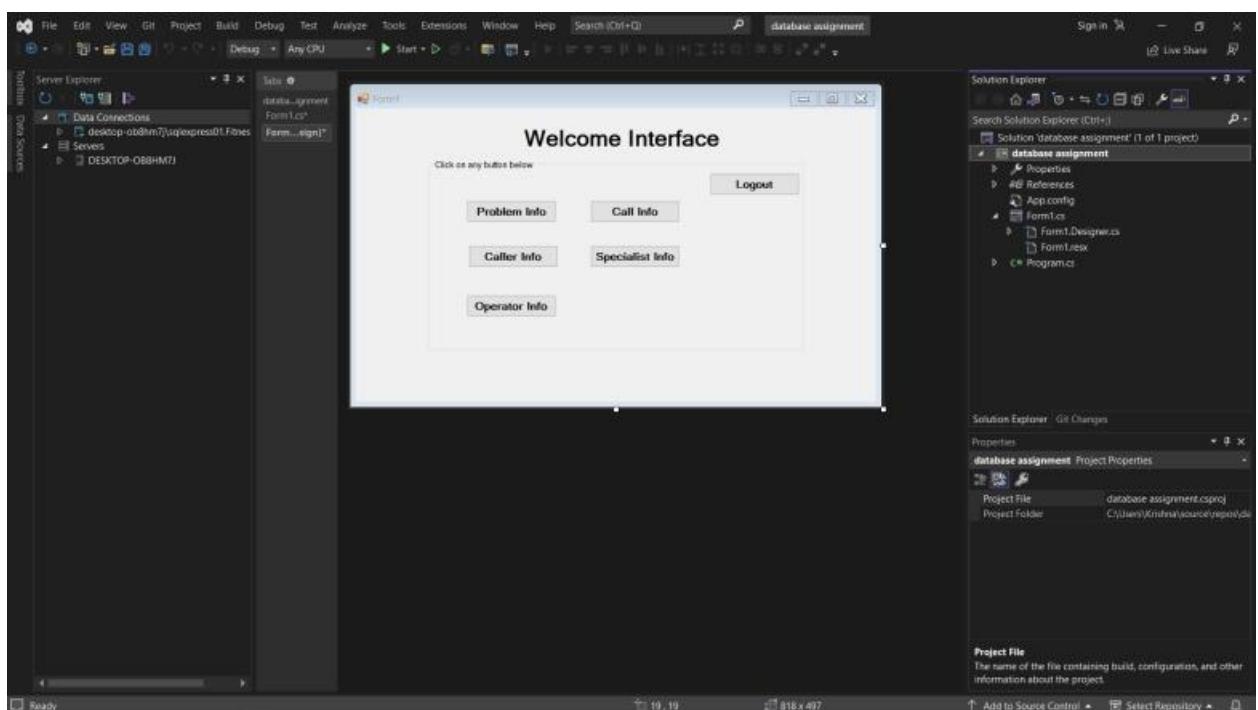
Upon successful authentication, the user is redirected to a distinct interface, accompanied by a new screen or page that enables further engagement with the helpdesk system.

- Login Interface

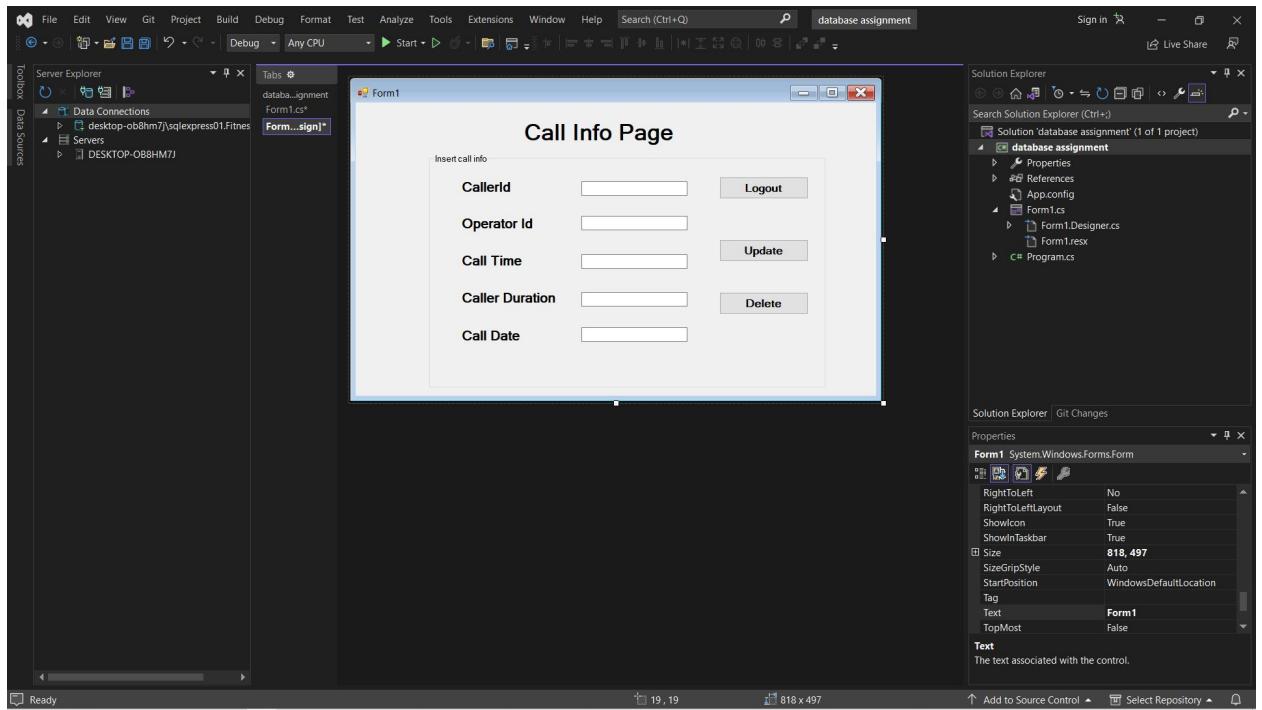


● Welcome Interface

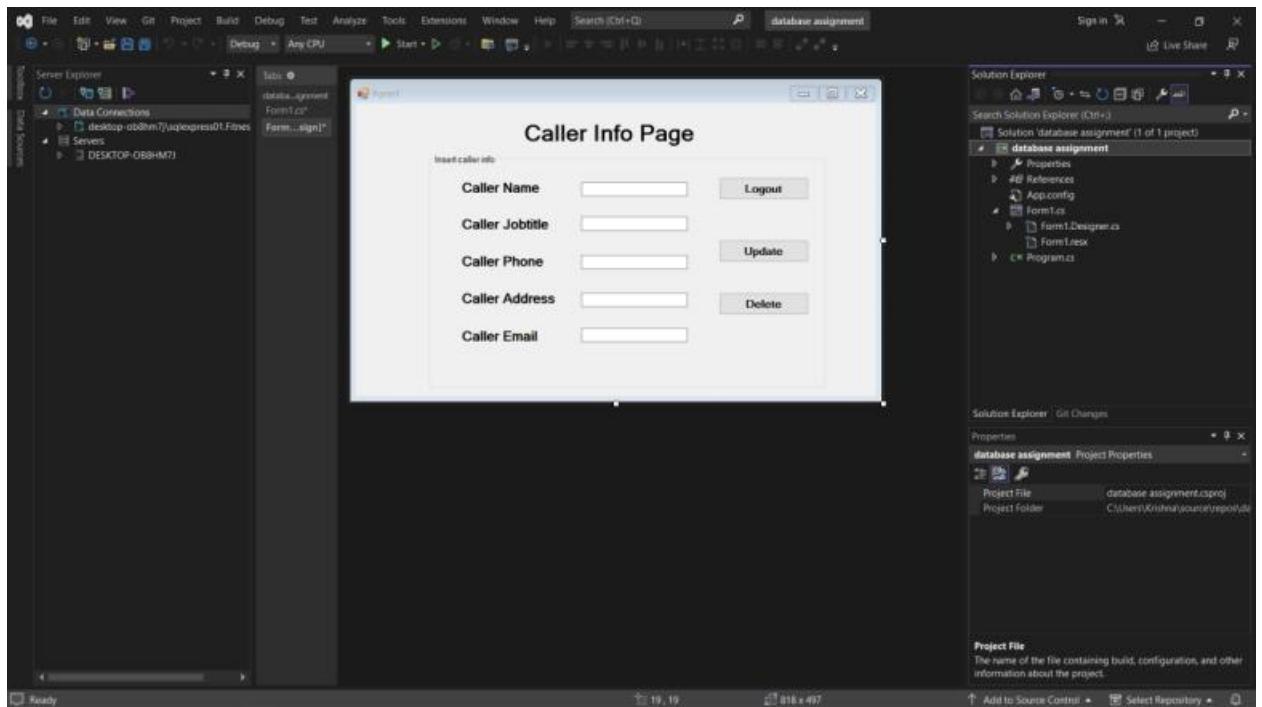
Upon accessing the welcome interface, the user is presented with a range of options regarding the forms they may wish to complete. By clicking on one of the buttons displayed on the subsequent page, the user is seamlessly directed to the corresponding form. This streamlined navigation process ensures that the user is guided towards their desired form of interaction, allowing for a tailored and personalized experience.



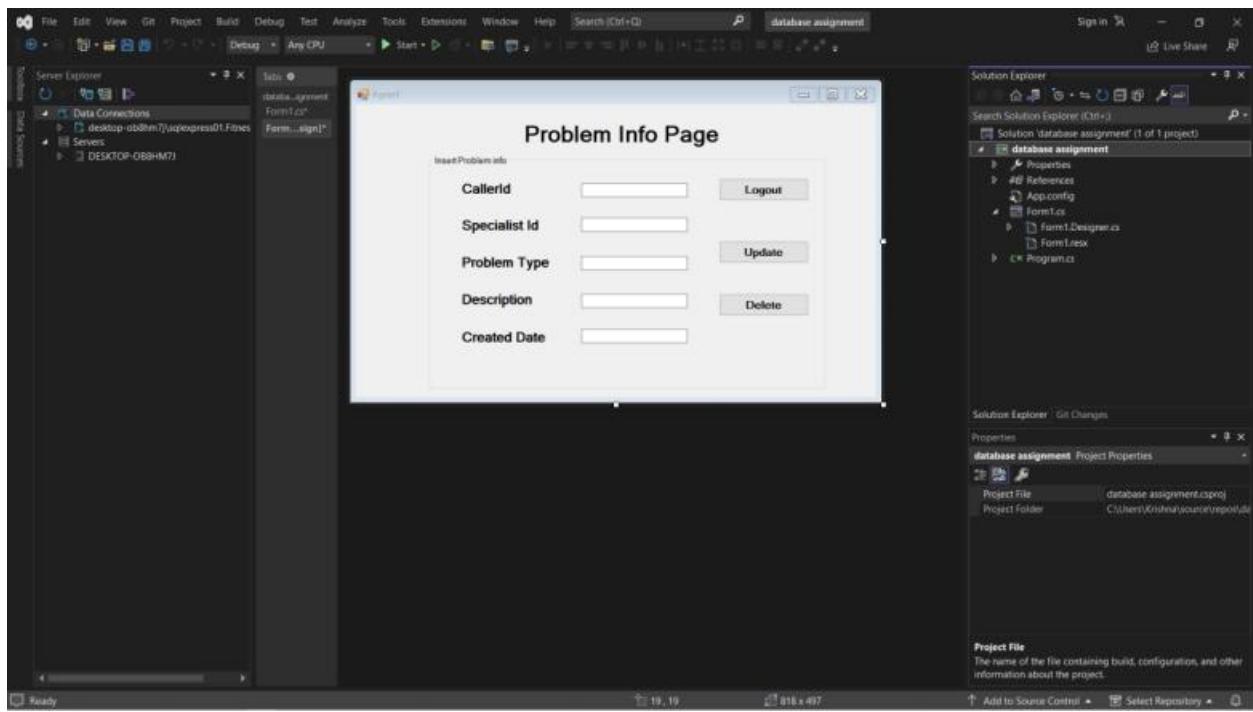
● Call Info Page



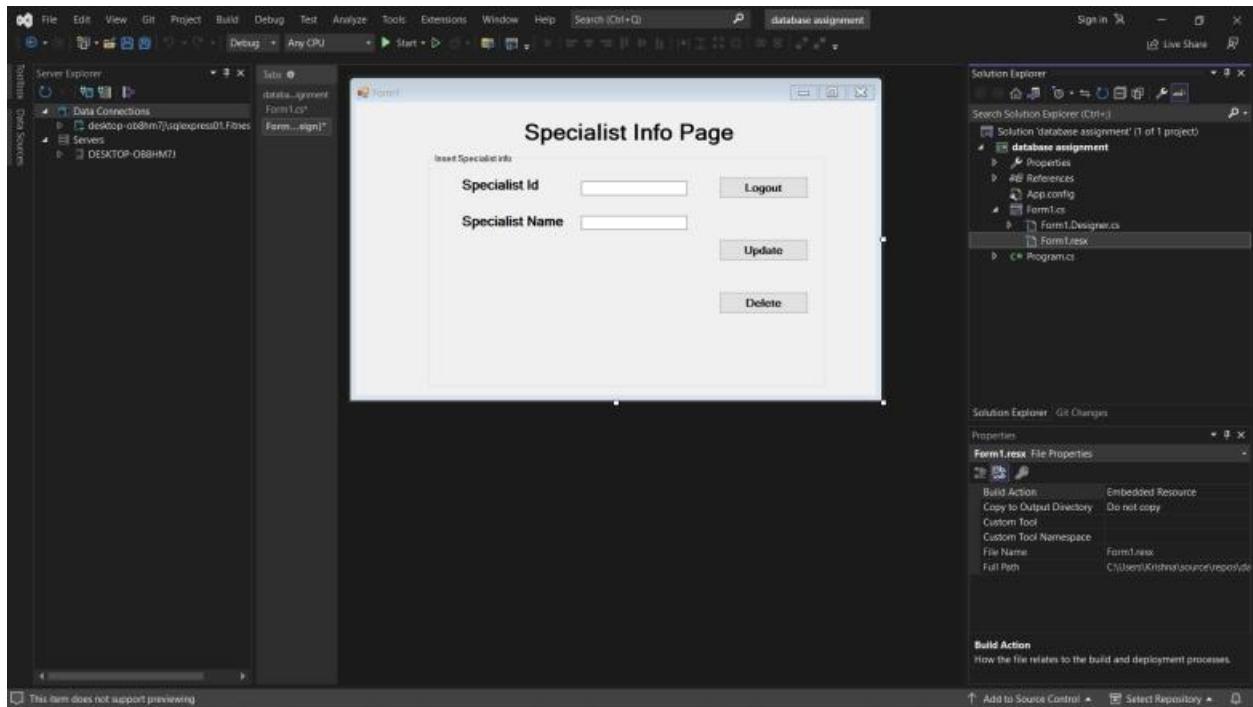
● Caller Info Page



● Problem Info Page



● Specialist Info Page



Data Validation mechanism

Data validation involves the examination and verification of collected data prior to its utilization. Each activity involving data manipulation has to include data validation in order to ensure trustworthy results, whether it includes data collection, processing, or arrangement for presentational purposes. Despite the urge to skip validation owing to time restrictions, the best and most accurate results require it.

To verify logical consistency while inputting and saving data, the system has several checks. The process of data validation has been significantly accelerated by technological improvements. Modern data integration solutions typically integrate and automate data validation as a vital step throughout the whole workflow to reduce the need for a lot of human effort. Recognizing the significance of data quality, data validation is now considered an essential process. Data cleaning will cost more and perhaps cause more issues if it is completed later in the process.

Data validation has become a crucial step inside businesses engaged in data-related activities, such as data gathering, processing, and analysis. Due to the fact that it lays the way for performing meaningful analyses utilizing real datasets, it is widely recognized as the cornerstone of good data management.

By ensuring the accuracy, completeness, and consistency of data, organizations can confidently leverage the information at their disposal, enabling informed decision-making and driving positive outcomes.

United Limited implements a comprehensive data verification procedure to ensure the accuracy and quality of the data. The objective is to maintain logical consistency in the data that is entered and stored within the system. This involves meticulously implementing the data verification process as intended and subsequently reviewing the system report to validate the integrity of the data. By following this procedure, United Limited can have confidence in the correctness and reliability of the data, enabling informed decision-making and efficient operations.

Data Type Check

United Limited has implemented a meticulous data type verification system to ensure precision and compatibility of supplied data. When data is input, the system verifies that it corresponds to the required data types. For example, date and time values for the proper data types and numeric values for the int data type are checked. By enforcing these restrictions, data integrity is upheld, avoiding errors and inconsistencies. This verification process acts as a safeguard against mismatched or incompatible data types that could lead to corruption or misinterpretation. The system maintains a high standard of data quality, consistency, and reliability while preventing inappropriate data from compromising storage integrity. United Limited's data type verification system is crucial for upholding accuracy, compatibility, and effective data analysis and decision-making.

Length Check

United Limited prioritizes data integrity and incorporates a length check as part of their validation measures. By using this procedure, it is made sure that the data is within the desired character range and does not go above the limit set for particular fields. The length check guarantees data consistency, standardization, and adherence to established character limits. It prevents issues arising from data exceeding or falling short of the specified limit, reducing the risk of inaccuracies or data loss. By implementing the length check, United Limited upholds data integrity, ensures accuracy and completeness, and enhances the overall quality and reliability of their stored data.

Format Check

United Limited prioritizes data accuracy and employs a format check as part of its validation process. Making sure that data entering follows a specific structure or pattern is made possible by this safety measure. Using an input mask, the system verifies that the data accurately complies with the required format specifications. The format check acts as a safeguard against data entered in incorrect formats, promoting consistency and reducing errors. It contributes to data integrity and reliability by ensuring that stored data remains accurate and conforms to the specified format requirements.

Conclusion

In conclusion, the design of the database for United Limited's help-desk was carried out meticulously, taking into account both user and system requirements. Utilizing tools such as ER diagrams and interface diagrams, we ensured a comprehensive and efficient database structure. The implementation of normalization techniques guaranteed that the database was properly organized and free from data redundancies. Overall, the database design for United Limited's help-desk successfully met the objectives of addressing requirements and ensuring optimal functionality.

Task 2

DBMS

A complex piece of software called a database management system (DBMS) makes it easy to create, maintain, and modify databases, enabling effective data administration and organization inside a business. DBMS provides various services such as database creation, data entry, and data manipulation activities like modifications, updates, and deletions.

Three widely used DBMS software solutions—Microsoft Access, Oracle, and SQL—each have unique features and capabilities. For United Limited, the SQL software has been carefully chosen and employed to develop and optimize their database infrastructure. SQL's robust capabilities and extensive support for managing and querying relational databases make it an excellent fit for United Limited's specific needs. By leveraging the power of SQL software, I have successfully constructed a

tailored and robust database system for United Limited, guaranteeing efficient data storage, retrieval, and manipulation.

Implementation of query against help-desk system

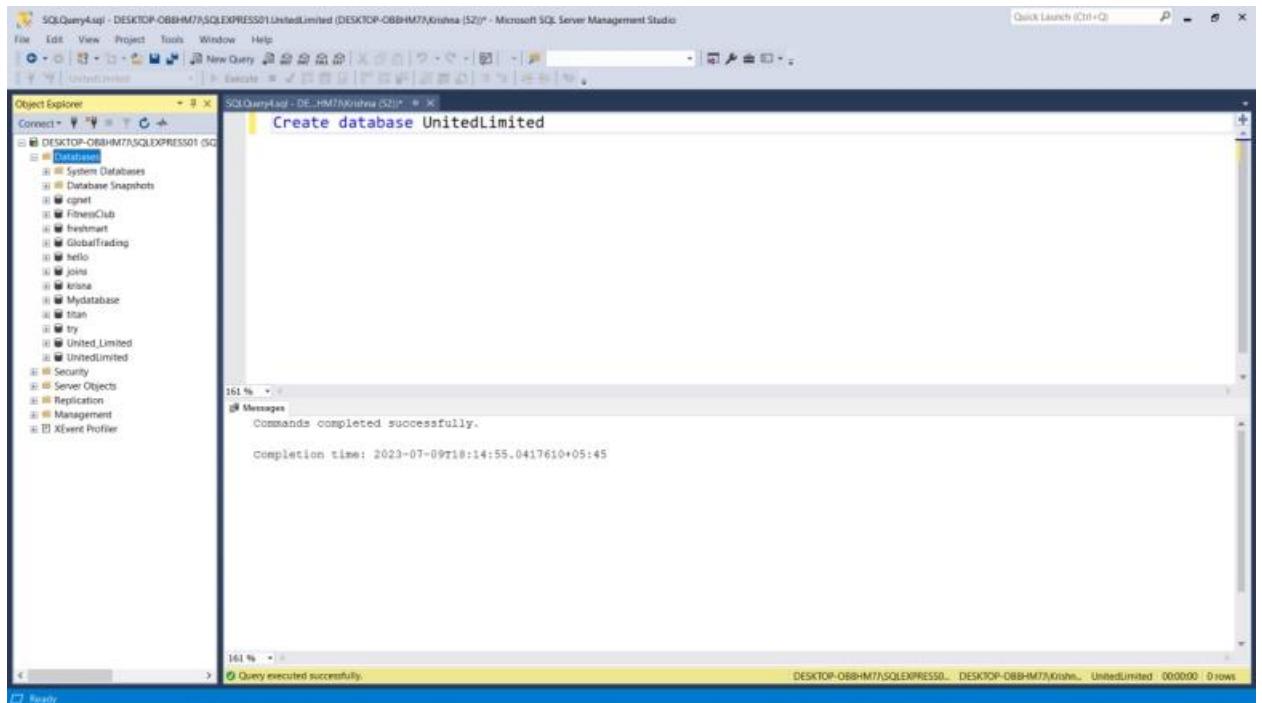
During the implementation process, I have executed queries against the help-desk system to retrieve the desired output in accordance with the previously established design and plan. These queries were designed to extract specific information from the database, perform calculations or aggregations, and generate reports or results that align with the requirements of the help-desk system. By executing these queries, I have been able to validate the functionality and accuracy of the database design, ensuring that it effectively supports the operations of the help-desk system.

Creating Database

Specifically, the "Create table" query is utilized to generate the necessary database structure for United Limited. This query enables the creation of tables within the database, defining their respective fields, data types, and any constraints or relationships required. By executing this query, the database is established with the appropriate structure that facilitates efficient storage and retrieval of data related to the helpdesk system.

Query: Create Database UnitedLimited

Output:

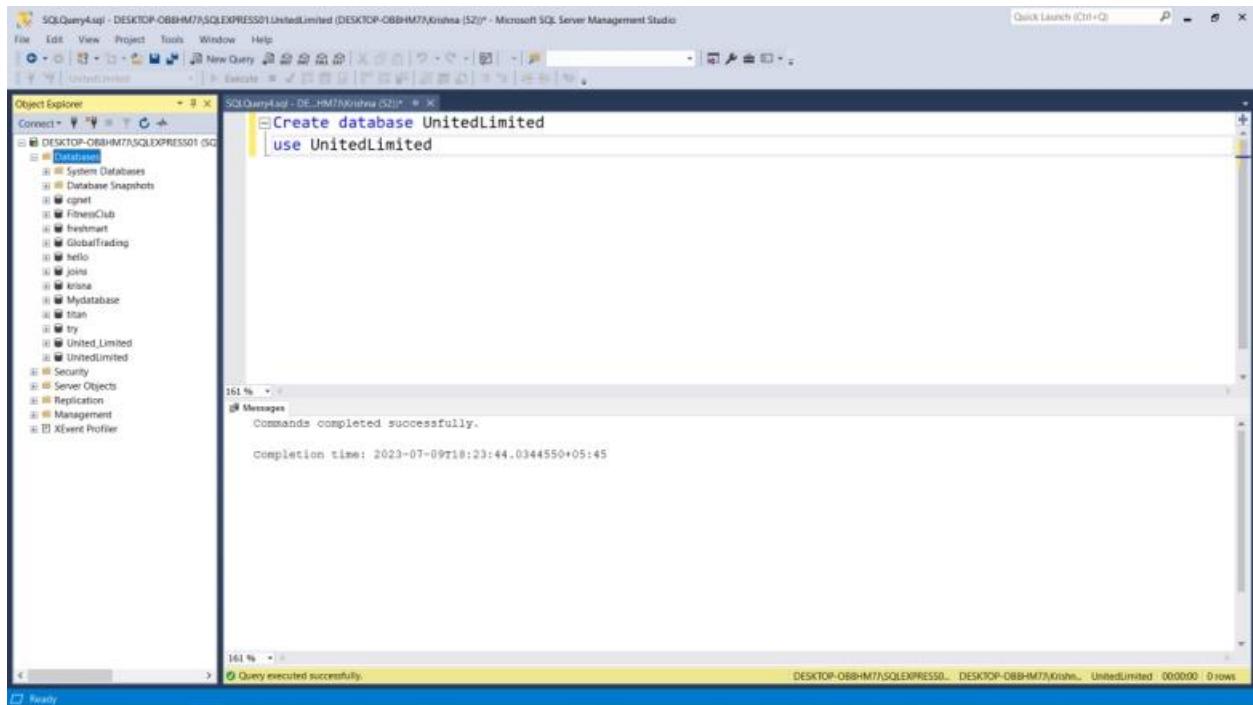


The selected name has been successfully used to create the IT Helpdesk database for United Limited.

Use Database

Query: Use database UnitedLimited

Output:



After creating and using the database, we need to create the tables according to the ER diagram.

Creating Table Users

The user table was created by the following query.

```
Create table Users(  
    UserId int identity(1,1) primary key,  
    UserName Varchar(100) not null,  
    Password Varchar(50) not null,  
);
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'UnitedLimited' is selected. In the main query window, a T-SQL script is being run to create a table:

```
Create table Users(
    UserId int identity(1,1) primary key,
    UserName Varchar(100) not null,
    Password Varchar(50) not null,
);
```

The execution results show the command completed successfully with a completion time of 2023-07-09T10:20:07.0744900+05:45.

Creating Table Callers

The Caller table was created with the following query.

```
Create table Callers(
    CallerId int identity(1,1) primary key,
    CallerName Varchar(50) not null,
    CallerJobTitle Varchar(50),
    CallerPhone bigint not null,
    CallerAddress Varchar(50) not null,
    CallerEmail Varchar(50) not null,
);
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'UnitedLimited' is selected. In the center pane, a query window titled 'SQLQuery4.sql - DESKTOP-OB8HM7\Krishna (S2) - Microsoft SQL Server Management Studio' contains the following T-SQL code:

```
Create table Callers(
    CallerId int identity(1,1) primary key,
    CallerName Varchar(50) not null,
    CallerJobTitle Varchar(50),
    CallerPhone bigint not null,
    CallerAddress Varchar(50) not null,
    CallerEmail Varchar(50) not null,
);
```

Below the code, the 'Messages' pane displays the execution results:

```
Commands completed successfully.  
completion time: 2023-07-09T10:33:14.4050351+05:45
```

The status bar at the bottom indicates 'Ready'.

Creating Table Operators

A table operator was created with the following query.

```
Create Table Operators(  
    OperatorId int identity(1,1) primary key,  
    SpecialistName Varchar(50) not null,  
);
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the connection is to DESKTOP-OB8HM7\SQLEXPRESS01 (S0) under the UnitedLimited database. In the center pane, a query window titled 'SQLQuery4.sql' contains the following SQL code:

```
Create Table Operators(
    OperatorId int identity(1,1) primary key,
    SpecialistName Varchar(50) not null,
);
```

Below the code, the 'Messages' pane displays the output:

```
Commands completed successfully.  
Completion time: 2023-07-09T18:36:32.9373267+05:45
```

The status bar at the bottom right shows the session details: DESKTOP-OB8HM7\SQLEXPRESS01, DESKTOP-OB8HM7\Krishna, UnitedLimited, 00:00:00, 0 rows.

Creating Table specialists

A specialist table was created with the following query.

```
Create Table specialist(
    SpecialistId int identity(1,1) primary key,
    SpecialistName Varchar(50) not null,
    SolutionInfo Varchar(300) not null,
);
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a connection to 'DESKTOP-OBBHM7A\SQLEXPRESS01' is selected. In the center pane, a query window titled 'SQLQuery4 - DESKTOP-OBBHM7A\Krishna (S2)' contains the following T-SQL code:

```
Create Table specialist(
    SpecialistId int identity(1,1) primary key,
    SpecialistName Varchar(50) not null,
    SolutionInfo Varchar(300) not null,
);
```

Below the code, the 'Messages' pane displays the output:

```
Commands completed successfully.  
Completion time: 2023-07-09T10:39:38.7094012+05:45
```

The status bar at the bottom right shows: DESKTOP-OBBHM7A\SQLEXPRESS01 DESKTOP-OBBHM7A\Krishna UnitedLimited 00:00:00 0 rows.

Creating Table Calls

A call table was created with the following query.

```
Create table Calls(
    CallId int identity(1,1) primary key,
    CallerId int foreign key references Callers(CallerId),
    OperatorId int foreign key references Operators(OperatorId),
    CallTime time not null,
    CallDuration time not null,
    CallDate date not null,
);
```

Output :

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'UnitedLimited' is selected. In the center pane, a query window titled 'SQLQuery4.sqi - DESKTOP-OB8HM7\Krishna (S2) - Microsoft SQL Server Management Studio' contains the following T-SQL code:

```
Create table Calls(
    CallId int identity(1,1) primary key,
    CallerId int foreign key references Callers(CallerId),
    OperatorId int foreign key references Operators(OperatorId),
    CallTime time not null,
    CallDuration time not null,
    CallDate date not null,
);
```

Below the code, the 'Messages' pane displays the execution results:

```
Commands completed successfully.  
completion time: 2023-07-09T10:47:04.7146202+05:45
```

The status bar at the bottom indicates 'Query executed successfully.'

Creating Table Problem

A table problem was created with the following query.

```
Create table Problem(
    ProblemId int identity(1,1) primary key,
    CallerId int foreign key references Callers(CallerId),
    SpecialistId int foreign key references Specialist(SpecialistId),
    ProblemType Varchar(50) not null,
    ProblemDescription Varchar(300) not null,
);
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'UnitedLimited' is selected. In the main query window, a 'Create table' script is being run:

```
CREATE table Problem(
    ProblemId int identity(1,1) primary key,
    CallerId int foreign key references Callers(CallerId),
    SpecialistId int foreign key references Specialist(SpecialistId),
    ProblemType Varchar(50) not null,
    ProblemDescription Varchar(300) not null,
);
```

The 'Messages' pane at the bottom right shows the command completed successfully with a completion time of 2023-07-09T10:50:28.8800024+05:45.

United Limited's database helpdesk was successfully created without any issues. From now on, we will verify the effect through implementation.

Insert Data into Users table

Here's the query that inserts data into the user table is:

```
insert into users
(UserName, Password)
Values
('Krishna','Krishna123'),
('Radha','Radha321');
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'UnitedLimited' database is selected. In the main query window, the following T-SQL code is executed:

```
insert into users
  (UserName, Password)
Values
  ('Krishna','Krishna123'),
  ('Radha','Radha321');
```

The output pane shows the results of the execution:

```
(2 rows affected)
```

```
Completion time: 2023-07-09T19:05:39.9399252+05:45
```

At the bottom, a status bar indicates: DESKTOP-OB8HM77\SOLEXPRESSO1 - DESKTOP-OB8HM77\Krishna - UnitedLimited - 000000 - 0 rows.

Insert data into Callers Table

The query that inserts data into calling table:

```
insert into Callers

(CallerName, CallerJobTitle, CallerPhone, CallerAddress, CallerEmail)

Values

('Basanta puri','Doctor','9845534902','Dhamak','Basantadhamak11@gmail.com'),

('Madan Nepal','Engineer','9846534872','Tandi','MadanChitwan22@gmail.com');
```

Output :

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists the database structure, including the 'UnitedLimited' database selected. The central pane displays a query window titled 'SQLQuery4.sql - [DE_HMT\Kishna (S2)]'. The query is:

```
insert into Callers
(CallerName, CallerJobTitle, CallerPhone, CallerAddress, CallerEmail)
Values
('Basant Puri', 'Doctor', '9845534902', 'Dhamak', 'Basantadhamak11@gmail.com'),
('Madan Nepal', 'Engineer', '9846534872', 'Tandi', 'MadanChitwan22@gmail.com');
```

The status bar at the bottom indicates 'Query executed successfully.' and '0 rows affected'.

Inserting Data into Operators Table

Here's the query that inserts data into the operators table:

```
insert into Operators  
  (SpecialistName)  
Values  
  ('Shyam Devkota'),  
  ('Lalumaya Adhikari');
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'UnitedLimited' is selected. A query window titled 'SQLQuery4 - DESKTOP-OB8HM7\Krishna (S2) - Microsoft SQL Server Management Studio' contains the following T-SQL code:

```
insert into Operators
(SpecialistName)
Values
('Shyam Devkota'),
('Lalumaya Adhikari');
```

The execution results pane shows the message: 'Query executed successfully.' and 'Completion time: 2023-07-09T19:20:46.0627456+05:45'. The status bar at the bottom indicates 'Ready'.

Inserting data into Specialist table

Here's the query that inserts data into specialist table:

```
insert into Specialist
(SpecialistName, SolutionInfo)
Values
('Bebika Thapa', 'wwwwww'),
('Baburam Pant', 'wwwwww');
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'UnitedLimited' database is selected. In the center pane, a query window titled 'SQLQuery4sql - DESKTOP-OB8HM7\Krishna (S2) - Microsoft SQL Server Management Studio' contains the following T-SQL code:

```
insert into Specialist
(SpecialistName, SolutionInfo)
Values
('Bebika Thapa', 'wwwwww'),
('Baburam Pant', 'wwwwww');
```

The execution results show:

- 150 %
- 2 rows affected
- Completion time: 2023-07-09T19:23:28.4806102+05:45

At the bottom, a green status bar indicates: Query executed successfully.

Insert Data into Call Table

The query that inserts data into Call table:

```
insert into Calls

(CallId, OperatorId, CallTime, CallDuration, CallDate)

Values

('1','1','12:01','00:03:44','2080-2-3'),

('1','1','11:33','00:01:58','2080-2-3');
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "SQLQuery4 - DESKTOP-OB8HM7\Krishna (S2)\ - Microsoft SQL Server Management Studio". The left pane is the Object Explorer, showing a tree structure of databases, tables, and other objects under "DESKTOP-OB8HM7\Krishna (S2)\UnitedLimited". The right pane contains a query window titled "SQLQuery4 - DESKTOP-OB8HM7\Krishna (S2)\ - Microsoft SQL Server Management Studio". The query is:

```
insert into Calls
(CallerId, OperatorId, CallTime, CallDuration, CallDate)
Values
('1','1','12:01','00:03:44','2080-2-3'),
('1','1','11:33','00:01:58','2080-2-3');
```

The status bar at the bottom indicates "Query executed successfully." and "Completion time: 2023-07-09T19:29:24.9526340+05:45".

Insert Data into Problem Table

The query that inserts data into Problem table:

```
insert into Problem

(CallId, SpecialistId, ProblemType, ProblemDescription)

Values

('1','1','Hardware','fffffff'),
('2','2','Software','fffffff');
```

Output:

SQLQuery4 - DESKTOP-OB8HM7\Krishna (S2)\ - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Untitled\Untitled

Object Explorer

SQLQuery4 - DESKTOP-OB8HM7\Krishna (S2)\ -

Connect ▾ New Query Task List Execute

Object Explorer

DB DESKTOP-OB8HM7\SQLEXPRESS01\UnitedLimited

Database

System Databases

Database Snapshots

cgnft

FitnessClub

freshmat

GlobalTrading

hello

join

Krisha

Mydatabase

titan

try

United_Limited

UnitedLimited

Database Diagrams

Tables

System Tables

FileTables

External Tables

Graph Tables

dbo.Callers

dbo.Calls

dbo.Operations

dbo.Problem

dbo.specialist

dbo.Users

Views

External Resources

Synonyms

Programmability

Query Store

Service Broker

Storage

Security

Messages

insert into Problem
(CallerId, SpecialistId, ProblemType, ProblemDescription)
Values
('1' , '1' , 'Hardware' , 'fffffff'),
('2' , '2' , 'Software' , 'fffffff');

150 %

Completion time: 2023-07-09T19:34:56.9979846+05:45

Query executed successfully.

DESKTOP-OB8HM7\SQLEXPRESS01\UnitedLimited 000000 0 rows

Ready

Retrieving Data

Here is the Query to get the data:

Select * from [Table Name]

Below is an example of retrieving data from each table:

For Problem Table

Syntax of the problem table query:

```
Select * from Problem
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'UnitedLimited'. The central pane contains a query window with the following text:

```
Select * from Problem
```

The results pane below shows the output of the query:

	ProblemId	CalledId	SpecialistId	ProblemType	ProblemDescription
1	1	1	1	Hardware	Computer issue
2	2	2	2	Software	Software bug

A status bar at the bottom indicates: 'Query executed successfully.' and 'DESKTOP-DBBM7\SOLEXPRESSO01 - DESKTOP-DBBM7\Krishna - UnitedLimited : 000000 : 2 rows'.

For Calls table

Query:

```
Select * from Calls
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left is the Object Explorer pane, which lists various databases, tables, and system objects. In the center is the Results pane where the query output is displayed. The query itself is 'Select * from Calls'. The results show two rows of data with columns: CallId, CallEnd, OperatorId, CallTime, CallDuration, and CallDate.

	CallId	CallEnd	OperatorId	CallTime	CallDuration	CallDate
1	1	5	1	12:01:00.0000000	00:03:44.0000000	2080-02-03
2	2	5	1	11:33:00.0000000	00:01:58.0000000	2080-02-03

For Specialist Table

Query:

```
Select * from Specialist
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists several databases, including 'UnitedLimited'. The 'UnitedLimited' database is expanded, showing tables like 'Specialist', 'Employee', 'Customer', etc. A query window titled 'SQLQuery4.sql - DESKTOP-OB8HM7\Krishna (S2) - Microsoft SQL Server Management Studio' contains the SQL command: 'Select * from Specialist'. The results pane shows the following data:

	SpecialistId	SpecialistName	SolutionInfo
1	Bebika Thapa	wwwwww	
2	Rahulraam Pant	wwwwww	

Below the results, a message bar says 'Query executed successfully.' and the status bar indicates 'DESKTOP-OB8HM7\SQLEXPRESS01 DESKTOP-OB8HM7\Krishna UnitedLimited : 00:00:00 , 2 rows'.

For Operator Table

Query:

```
Select * from operators
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists several databases, including 'UnitedLimited'. The 'UnitedLimited' database is expanded, showing tables like 'Specialist', 'Employee', 'Customer', etc. A query window titled 'SQLQuery4.sql - DESKTOP-OB8HM7\Krishna (S2) - Microsoft SQL Server Management Studio' contains the SQL command: 'Select * from operators'. The results pane shows the following data:

	OperatorId	SpecialistName
1	Shyam Devkota	
2	Lakumaya Adhikari	

Below the results, a message bar says 'Query executed successfully.' and the status bar indicates 'DESKTOP-OB8HM7\SQLEXPRESS01 DESKTOP-OB8HM7\Krishna UnitedLimited : 00:00:00 , 2 rows'.

For Callers Table

Query:

```
Select * from Callers
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists databases, tables, and other objects. The central pane displays the results of the query "Select * from Callers". The results show two rows of data:

CallerId	CallerName	CallerJobTitle	CallerPhone	CallerAddress	CallerEmail
1	Basant Nepal	Doctor	9845534502	Dhamak	BasantChamak11@gmail.com
2	Madan Nepal	Engineer	9845534572	Taxi	MadanChheen22@gmail.com

At the bottom, a message says "Query executed successfully." and the status bar indicates "DESKTOP-DBBHMT7\SOLEXPRESSO\Krishna 000000 : 2 rows".

For Users Table

Query:

```
Select * from Users
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists several databases, including 'DESKTOP-OBHHM7\SQLEXPRESS01' and 'UnitedLimited'. The 'UnitedLimited' database is expanded, showing tables like 'Users', 'try', 'United_Limited', and 'UnitedLimited'. The 'Users' table is selected in the center pane, and its schema is displayed. A query window titled 'SQLQuery4sql - DE...HM7\Krishna (S2)*' contains the command 'Select * from Users'. The results pane shows the data from the 'Users' table:

UserId	UserName	Password
1	Krishna	Krishna123
2	Radha	Radha321

The status bar at the bottom indicates 'Query executed successfully.' and provides details about the session: DESKTOP-OBHHM7\SQLEXPRESS01, DESKTOP-OBHHM7\Krishna, UnitedLimited, 000000, 2 rows.

Updating Data

Update command modifies data of one or more records.

Syntax of the UPDATE:

```
UPDATE [table_name] SET [column_name = value] WHERE[condition]
```

```
Update Calls
```

```
set Calltime = '04:44'
```

```
where CallId = 9
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure of 'UnitedLimited'. In the center, the 'SQL Query1' window contains the following T-SQL code:

```
Update Calls
set Calltime = '04:44'
where CallId = 9
```

Below the code, the results pane shows the output:

```
(0 rows affected)
Completion time: 2023-07-09T19:54:58.3757096+05:45
```

At the bottom of the window, a yellow status bar indicates: 'Query executed successfully.'

Delete data

Here's the query to remove the data from the calling table :

```
Delete from Calls
```

```
Where CallId = 7
```

Output :

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays a tree view of databases, including 'DESKTOP-OB8HM77\SQLEXPRESS01' and 'UnitedLimited'. The 'Tables' node under 'UnitedLimited' is expanded, showing various table names like 'Calls', 'dbo.Calls', 'dbo.Calls', 'dbo.Operations', 'dbo.Problems', 'dbo.specialist', and 'dbo.Users'. In the center, the 'SQLQuery4.asp - DE... (S2) []' window contains a T-SQL query:

```
>Delete from Calls  
Where CallId = 7
```

The results pane shows the output of the query:

(0 rows affected)

Completion time: 2023-07-09T20:02:29.1671333+05:45

At the bottom, a status bar indicates: DESKTOP-OB8HM77\SQLEXPRESS01 DESKTOP-OB8HM77\Krishna UnitedLimited 000000 0 rows.

For Security and Maintenance

While creating a database may seem straightforward, ensuring robust security and effective maintenance is of utmost importance. To address these critical aspects, we have implemented a series of enhanced queries specifically designed to optimize database maintenance and bolster security measures. By incorporating these new queries, we strive to guarantee the utmost data protection and seamless management of the database.

Sort the data

Any procedure that includes putting data in a meaningful order to aid interpretation, analysis, or visualization is referred to as data sorting. Sorting is a typical tactic used when dealing with research data to present the data in a way that makes it simpler to understand the story the data is saying. Both raw data (across all entries) and

aggregated data (in a table, graphic, or other aggregated or summed output) can be sorted.

In addition to the more common methods of sorting data by actual numbers, counts, or percentages, data can also be organized according to variable value labels. For each value choice in a categorical investigation, the researcher may be able to preserve information known as value labels from some systems. Most software products also provide sorting by a number of other criteria. A data collection having region and country fields may be sorted with region being the primary sort and country being the secondary sort, according to a changeable priority order. Each region that has been sorted will then receive the county sort.

(Ali, 2018)

Here is the query that sorts the Data:

```
Select * from Calls  
order by Calldate ASC
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer on the left, there are several databases listed under 'DESKTOP-OB8HM77\SQLEXPRESS01'. In the center, a query window titled 'SQLQuery4.asp - DESKTOP-OB8HM77\Krishna (S2) []' contains the following SQL code:

```
Select * from Calls  
order by CallDate ASC
```

The results pane below shows the output of the query:

	CallId	CallerId	OperatorId	CallTime	CallDuration	CallDate
1	1	1	1	12:01:08.0000000	00:01:44.0000000	2000-02-03
2	2	1	1	11:33:06.0000000	00:01:58.0000000	2000-02-03

At the bottom of the results pane, it says 'Query executed successfully.' and shows the status bar with 'DESKTOP-OB8HM77\SQLEXPRESS01 DESKTOP-OB8HM77\Krishna UnitedLimited 0000000 2 rows'.

Joins

Rows from two or more tables can be combined using the SQL statement "JOIN" based on a common column. These commands are primarily utilized when a user intends to extract data from tables that exhibit one-to-many or many-to-many relationships.

By employing JOINS, SQL users can efficiently retrieve data by linking related information from multiple tables. This allows for comprehensive analysis and reporting, leveraging the interconnections between tables to gather meaningful insights and produce accurate results. Whether it involves combining customer information with their respective orders or linking products with their associated categories, JOINS facilitate the consolidation of data from multiple sources, enhancing the data retrieval capabilities of SQL queries.

Types of Joins

Inner Joins

Right Joins

Left Joins

Full Outer Joins

Inner Joins

As the name implies, an INNER join only links two tables together when the values in the two tables match. When performing an INNER join between Table 1 and Table 2, only the tuples or rows that possess matching values in both tables will be included in the output. This type of join effectively filters and combines the data from both tables based on the shared values, ensuring that only the corresponding records are included in the result set. By using INNER join, SQL users can extract data that satisfies the condition of having matching values in both tables, facilitating comprehensive data analysis and reporting.

Query:

```
Select * from Calls  
Inner join problem  
on Calls. CallId = Problem.ProblemId  
Select * from Problem
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'UnitedLimited' database is selected. In the center pane, a query window displays the following T-SQL code:

```

Select * from Calls
Inner join problem
on Calls.CallId = Problem.ProblemId
Select * from Problem

```

The results pane shows two result sets. The first result set, 'Calls', has columns: CallId, CalleId, OperatorId, CallTime, CallDuration, CallDate, ProblemId, CalledId, SpecialistId, ProblemType, and ProblemDescription. It contains two rows:

CallId	CalleId	OperatorId	CallTime	CallDuration	CallDate	ProblemId	CalledId	SpecialistId	ProblemType	ProblemDescription
1	1	1	12:01:08.0000000	00:00:44.0000000	2000-02-03	1	1	1	Hardware	AAA
2	2	1	11:33:08.0000000	00:01:58.0000000	2000-02-03	2	2	2	Software	BBB

The second result set, 'problem', has columns: ProblemId, CalledId, SpecialistId, ProblemType, and ProblemDescription. It also contains two rows:

ProblemId	CalledId	SpecialistId	ProblemType	ProblemDescription
1	1	1	Hardware	AAA
2	2	2	Software	BBB

At the bottom of the results pane, a message says "Query executed successfully."

This is an example of an Inner Joins between the problem table and call table.

Left Joins

Regardless of whether there is a match in the right table, the LEFT JOIN makes sure that all rows from the left table are included in the output. The suitable entries from the right table are connected to the left table if a match is discovered. The appropriate table columns in the result set will, however, have NULL values if no match is discovered.

Query:

```

Select * from Calls
Left join problem
on Calls.CallId = Problem.ProblemId
Select * from Problem

```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'UnitedLimited' is selected. In the center pane, a query window displays the following T-SQL code:

```
Select * from Calls
Left join problem
on Calls.CallId = Problem.ProblemId
Select * from Problem
```

The results pane shows two tables. The first table, 'Calls', has four columns: CallId, CallTime, CallDuration, and CallDate. The second table, 'Problem', has five columns: ProblemId, CallId, SpecialistId, ProblemType, and ProblemDescription. The data is as follows:

CallId	CallTime	CallDuration	CallDate	ProblemId	CallId	SpecialistId	ProblemType	ProblemDescription
1	12:01:00.0000000	00:03:44.0000000	2000-02-03	1	1	1	Hardware	AAA
2	11:33:00.0000000	00:01:58.0000000	2000-02-03	2	2	2	Software	BBB

ProblemId	CallId	SpecialistId	ProblemType	ProblemDescription
1	1	1	Hardware	AAA
2	2	2	Software	BBB

At the bottom of the results pane, a message says "Query executed successfully."

This is an example of a Left Joins between the problem table and call table.

Right Joins

Regardless of whether there is a match in the left table, RIGHT JOIN makes sure that all rows from the right table are included in the output. If a match is discovered, the right table is connected with the relevant records from the left table. The left table columns in the result set, however, will have NULL values if no match is discovered.

Query:

```
Select * from Calls
Right join problem
on Calls.CallId = Problem.ProblemId
Select * from Problem
```

Output:

```

Select * from Calls
Right join problem
on Calls.CallId = Problem.ProblemId
Select * from Problem

```

CallId	Called	OperatorId	CallTime	CallDuration	CallDate	ProblemId	Called	SpecialistId	ProblemType	ProblemDescription
1	1	1	12:01:08.0000000	00:00:44.0000000	2000-02-03	1	1	1	Hardware	AAA
2	2	1	11:33:08.0000000	00:01:58.0000000	2000-02-03	2	2	2	Software	BBB

ProblemId	Called	SpecialistId	ProblemType	ProblemDescription
1	1	1	Hardware	AAA
2	2	2	Software	BBB

Query executed successfully.

This is an example of a Right Joins between the problem table and call table.

Full Outer Joins

Every row from both tables is present in the result of a complete join. The relevant records from both tables are merged if a match is discovered. The result set will still include NULL values for the fields from the database without a matching record if there is no match for a specific record in either the left or right table.

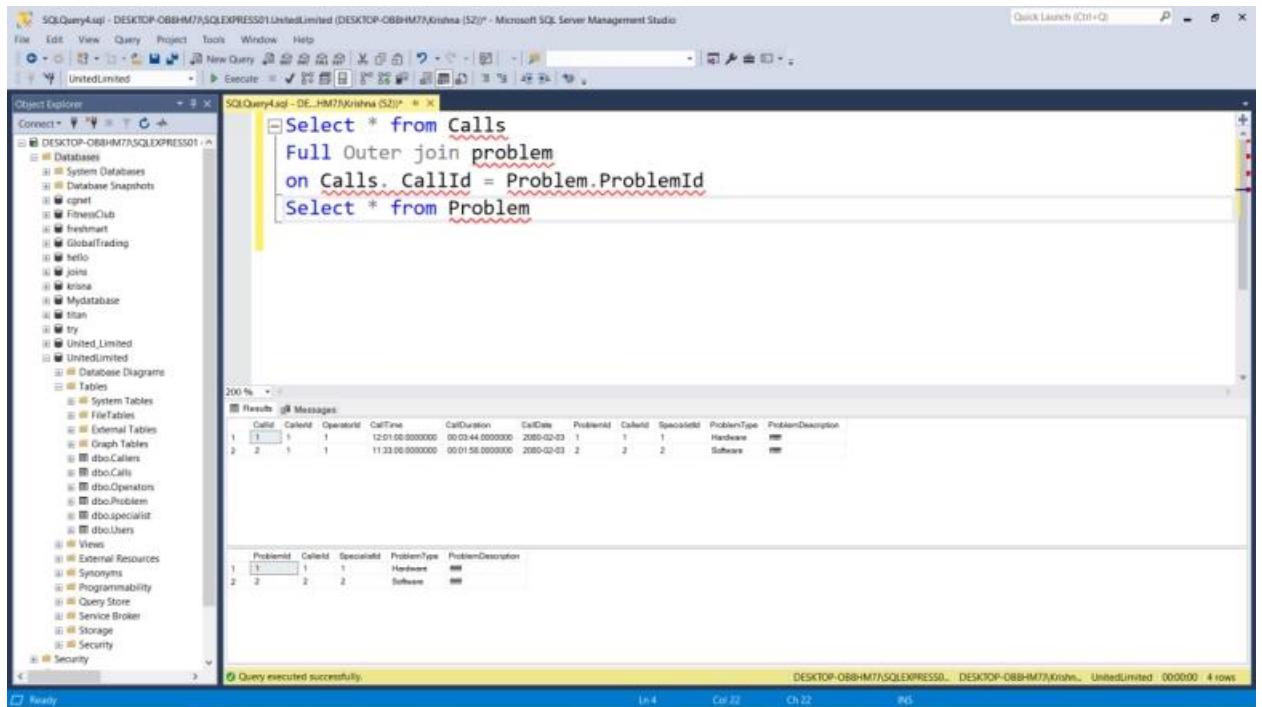
Query:

```

Select * from Calls
Full Outer join problem
on Calls.CallId = Problem.ProblemId
Select * from Problem

```

Output:



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'United_Limited' database is selected. In the center pane, a query window titled 'SQLQueryLast - DE... (S2) [x]' contains the following T-SQL code:

```
Select * from Calls
Full Outer join problem
on Calls.CallId = Problem.ProblemId
Select * from Problem
```

The 'Results' tab displays two tables. The first table, 'Calls', has four rows:

	CallId	Caller	OperatorId	CallTime	CallDuration	CallDate	ProblemId	CalledId	SpecialistId	ProblemType	ProblemDescription
1	1	1	1	12:01:00.0000000	00:03:44.0000000	2000-02-03	1	1	1	Hardware	AAA
2	2	1	1	11:33:00.0000000	00:01:58.0000000	2000-02-03	2	2	2	Software	BBB
3											

The second table, 'problem', has two rows:

	ProblemId	CalledId	SpecialistId	ProblemType	ProblemDescription
1	1	1	1	Hardware	AAA
2	2	2	2	Software	BBB

At the bottom of the results pane, a message says 'Query executed successfully.'

This is an example of a Full Outer Joins between the problem table and call table.

(Sahiti Kappagantula, 2019)

Union

The SQL Union operator is used to integrate the results of two or more tables and erase duplicate data. A join generates new columns, but a union generates new rows based on the combined data. To reduce data duplication, this may be used to consolidate several smaller tables into one large table. To be connected using union, the tables must have the same data type and number of columns. Otherwise, the union operator cannot be used with relational databases.

Query:

```
Select * from specialist  
Union all  
Select * from specialist1
```

Output:

```
Create table specialist1  
([SpecialistId] int identity(1,1) primary key,  
[SpecialistName] Varchar(50) not null,  
[SolutionInfo] Varchar(300))  
Insert into specialist1 ([SpecialistName], SolutionInfo)  
values ('Babita Chatterjee', Null);  
Select * from specialist  
Union all  
Select * from specialist1
```

	SpecialistId	SpecialistName	SolutionInfo
1	1	Bekika Thapa	wwwwww
2	2	Balaram Pant	wwwwww
3	1	Babita Chatterjee	NULL

Query executed successfully.

I altered the users table by deleting the column roles and running the union because it requires the same data types and quantity of columns in both tables.

Trigger

A trigger, as used in the context of databases, is a group of instructions that are "fired" or triggered by a specific event, frequently a command delivered via the database's Data Manipulation Language (DML). The technique we use to add, remove, retrieve from, and edit data within a database is known as DML. The most frequent DML commands are SELECT, INSERT, UPDATE, and DELETE, while different database vendors could add additional extensions to the language. It is possible to programme any of these instructions to activate AFTER, BEFORE, or INSTEAD of a trigger.

Triggers are most frequently used in databases to ensure that data remains constant throughout the whole database, regardless of where it is stored, and to automate recurring, related tasks.

(Database Triggers: Examples & Overview Video, 2023)

Query:

```
Create trigger tr_Calls
on Calls
for insert
as begin
Select * from inserted
end
insert into Calls values('1', '1','11:21', '00:03:26', '2080-1-2');
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The Object Explorer on the left lists several databases: System Databases, Database Snapshots, master, msdb, tempdb, FitnessClub, hsmart, GlobalTrading, help, krishna, Mydatabase, stan, try, United_Limited, and UnitedLimited. The central pane displays a query window with the following T-SQL code:

```
Create trigger tr_Calls
on Calls
for insert
as begin
Select * from inserted
end
insert into Calls values('1', '1','11:21', '00:03:26', '2080-1-2');
```

Below the code, the Messages pane shows the results of the execution:

```
Commands completed successfully.
Completion time: 2023-07-09T22:14:57.1684033+05:40
```

At the bottom of the screen, a status bar indicates "Query executed successfully." and shows the system information: DESKTOP-OB8HM7\SQLEXPRESS0, DESKTOP-OB8HM7\Krishna, UnitedLimited, 00:0000, 0 rows.

Store Procedure

A database will record a series of logical steps and SQL instructions as a stored procedure. It enables the integration and coordinated execution of business and application logic within the database. Less information would need to be sent across great distances between the database and the application, allowing for more efficient handling of complex procedures and computations.

Using SQL or other database technologies, stored procedures may be created and managed. They support result sets, input parameters, and output parameters. Stored procedures can speed things up by minimizing the number of times the database and the application interact. They also assist in maintaining data integrity standards and implementing security policies inside the database environment.

They are useful for software development and maintenance since they can be independently tested and versioned. Compared to including queries in a graphical user interface (GUI), they offer advantages. Due to their modular design, stored procedures make application troubleshooting and problem solving easier. Efficiency gains can be made utilizing customized stored procedures without altering the GUI's source code. There are many advantages to utilizing stored procedures in database systems since scripting them is typically simpler than making queries using the GUI.

Query:

```
Create procedure spGetCalls
as
begin
Select CallTime, CallDuration, CallDate from Calls
end
exec spGetCalls
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'UnitedLimited' is selected. In the center pane, a query window displays the creation of a stored procedure:

```
=Create procedure spGetCalls  
as  
begin  
Select CallTime, CallDuration, CallDate from Calls  
end  
exec spGetCalls
```

Below the code, the 'Messages' tab shows the results of the execution:

```
Commands completed successfully.  
Completion time: 2020-07-09T02:44:50.8437787+05:45
```

At the bottom of the window, a status bar indicates: DESKTOP-OB8HM7\Krishna (54) - DESKTOP-OB8HM7\Krishna - UnitedLimited - 000000 - 0 rows.

Store Procedure on save

A type of storage activity that may occur automatically whenever new data is added, updated, or discarded. The same as Trigger. If you declare this operation before adding data to the database, the database system will do it automatically. It is mostly used for data protection and upkeep.

Query:

```

Create procedure sp_save
(@specialistName varchar(50),
@SolutionInfo Varchar(300)
)
as
begin
insert into Specialist values (@SpecialistName, @SolutionInfo)
end
exec sp_save'Krishna','null'

```

Select * from Specialist

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure, including databases like 'United', 'United.Limited', and 'United_Limited'. The 'United_Limited' database is selected. In the center, the 'SQLQuery6' query editor contains the T-SQL code for creating the 'sp_save' stored procedure. The code defines parameters for specialist name and solution info, includes an 'as' clause, a 'begin' block with an 'insert' statement into the 'Specialist' table, an 'end' block, and an 'exec' statement calling the procedure with 'Krishna' and 'null'. Below the code, the 'Messages' pane shows the output: 'Commands completed successfully.' and 'Completion time: 2013-07-09T22:31:09.4254477+05:00'. At the bottom, the status bar indicates '16 19 Col 25 Ch 25 145 0 Rows'.

```

CREATE PROCEDURE sp_save
    (@specialistName varchar(50),
     @SolutionInfo Varchar(300))
AS
BEGIN
    INSERT INTO Specialist VALUES (@SpecialistName, @SolutionInfo)
END
EXEC sp_save 'Krishna', 'null'

```

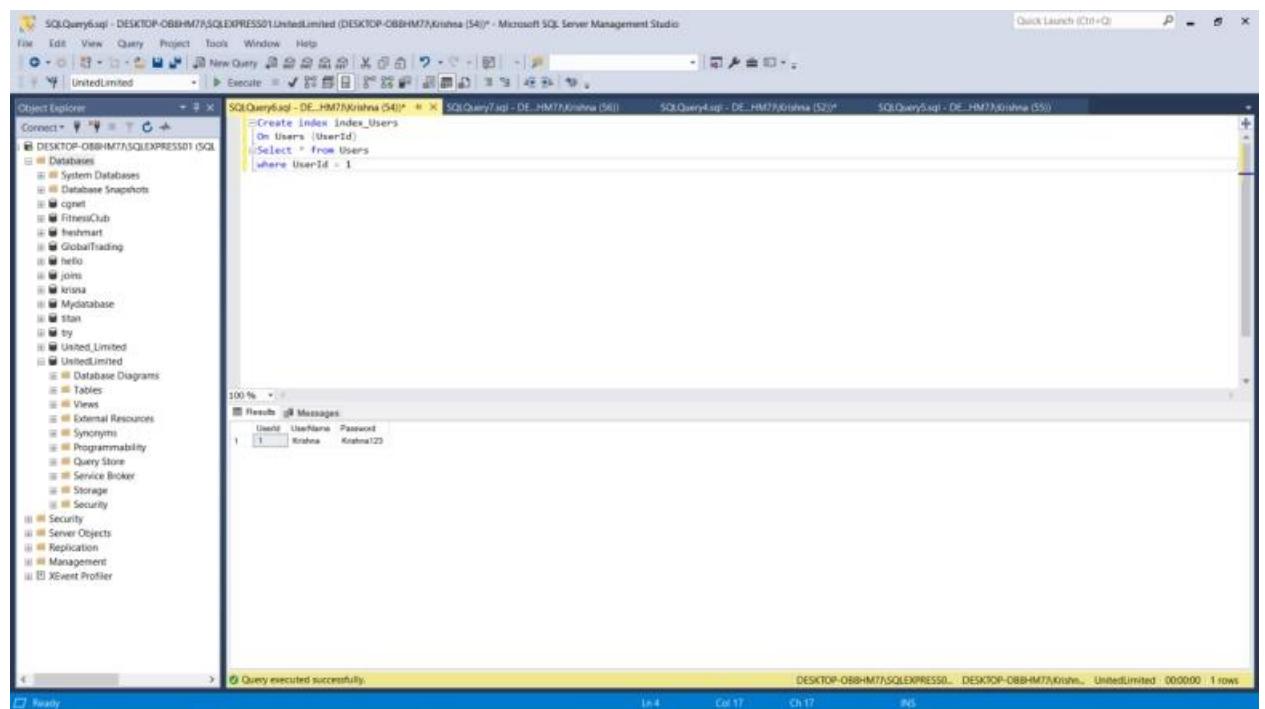
Index

Indexes help databases retrieve data more quickly. This is crucial during times of organizational stress. It is straightforward to get data from each column by using an index. It also has an adverse effect on insert, update, and delete operations since they need to be updated each time the table is changed.

Query:

```
Create index index_Users  
On Users (UserId)  
  
Select * from Users  
  
where UserId = 1
```

Output:



The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The Object Explorer on the left lists several databases, including 'United.Limited'. The central pane contains four tabs: 'SQLQuery6.sql - DE_HM7\Krishna (54)', 'SQLQuery7.sql - DE_HM7\Krishna (56)', 'SQLQuery8.sql - DE_HM7\Krishna (52)', and 'SQLQuery9.sql - DE_HM7\Krishna (55)'. The first tab is active and displays the following T-SQL code:

```
Create index index_Users  
On Users (UserId)  
  
Select * from Users  
  
where UserId = 1
```

The 'Results' tab below shows the output of the 'Select' query:

Userid	UserName	Password
1	Krishna	Krishna123

A status bar at the bottom indicates 'Query executed successfully.' and '1 row(s) affected'.

Group By

To group rows in a certain column that have the same value, use the group by function. After that, each group is given summary statistics using the aggregate function.

Query:

```
Select UserName  
from Users  
group by UserName  
having UserName = 'Krishna'
```

Output :

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists databases like DESKTOP-OB8HM7\Krishna, FitnessClub, freshmart, GlobalTrading, hello, joins, krishna, Mydatabase, stan, try, and United_Limited. The central pane displays a query window with the following T-SQL code:

```
Select UserName  
from Users  
group by UserName  
having UserName = 'Krishna'
```

The results pane below shows a single row of data:

UserName
Krishna

A status bar at the bottom indicates "Query executed successfully." and "Ready".

BackUp

Backup

It provide the extra

```
backup database UnitedLimited  
to disk ='C:\united limited\backup.bak'
```

```
drop database UnitedLimited
```

```
restore database UnitedLimited  
from disk ='C:\united limited\backup.bak'
```

Output:

The screenshot shows a SQL query window with the following content:

```
backup database UnitedLimited  
to disk ='C:\united limited\backup.bak'  
  
drop database UnitedLimited  
  
restore database UnitedLimited  
from disk ='C:\united limited\backup.bak'
```

Below the query window, the "Messages" pane displays the following output:

100 % ↻
(Messages)
Processed 640 pages for database 'UnitedLimited', file 'UnitedLimited' on file 1.
Processed 2 pages for database 'UnitedLimited', file 'UnitedLimited_log' on file 1.
RESTORE DATABASE successfully processed 642 pages in 0.022 seconds (227.805 Mb/sec).
Completion time: 2023-05-11T10:10:18.0000874+05:45

At the bottom of the screen, there is a status bar with the text "Query executed successfully."

Create and Implement a test plan with effective tests.

- a) A realistic assessment of database health should include extreme and inaccurate data in addition to true data.
- b) Testing should assess the breadth, usability, and conformance to the informational needs of the programme.
- b) Tests must offer the necessary information and components for a successful deployment.

As part of this project, you will have the opportunity to conduct a variety of tests to thoroughly examine your database, ensure that it is user-friendly, and that it meets your objectives.

Test plan

A test plan is a document designed to detail the goals, strategies, and resources needed to complete a project. Wikipedia (2023).

United Limited's database testing programme assist in ensuring that your database satisfies user and system requirements while running efficiently.

Why test plan is necessary for United Limited ?

The United Limited database test plan is critical for determining the efficacy of the system.

To confirm that no fields are missing.

To analyse the efficacy of the database.

It makes it easy for others.

- To comprehend the database's peculiarities.
- Be a resource for those who work for the organization.
- To assure the database's usability.
- To authenticate the data's legitimacy.

Testing

The database is the heart of every software programme, hence optimal database performance is essential. Database testing ensures that all of the database's components are functional. This is accomplished in a controlled testing environment. The method is frequently referred to as back-end testing or data testing. When assessing a database, the triggers, tables, and structure must all be looked at. At several stages, the accuracy and integrity of the data are also verified. Complex queries are used to load or stress test the database in order to do this. By doing this, it is feasible to assess the system's responsiveness and adaptability to different situations. It is a multi-tiered method since database testing typically deals with a wide variety of challenges. The database itself, the business layer, the data access layer, and the user interface layer should all undergo thorough testing. Together, these layers provide an application that is extensive and adaptable. When creating software, the functioning of every component must be ensured.

(Reqtest, 2020)

Unit Testing

A particular kind of software test that concentrates on a software product's constituent parts is known as a unit test. The goal is to confirm that each and every line of software code works as planned. A unit in the source code of an application may be a function, method, module, object, or another sort of entity. An object in the code is evaluated as part of a unit test to make sure it was written correctly, without mistakes, and that it produces the desired results when given the correct inputs. Since they are written as code that remains in the code-base alongside the application code they are testing, unit tests are commonly generated by developers throughout the development phase of a project. Unit tests may be organized and executed using a number of frameworks, according to developers.

(Dizdar, 2022)

Integration test

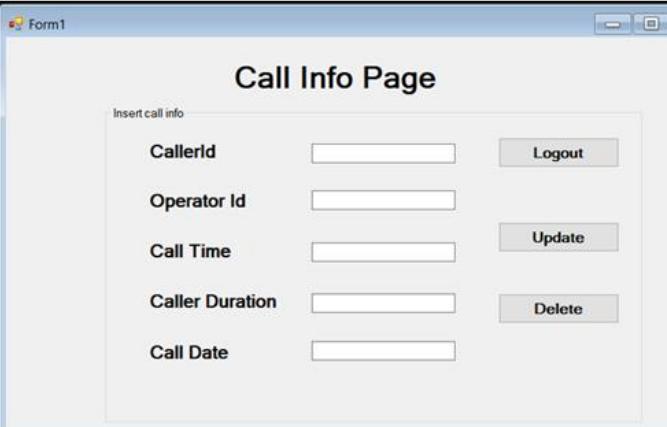
Every piece of software is composed of several modules, each of which has a certain function. Testing professionals do integration testing even when a software module runs successfully on its own to ensure that the programme will continue to run flawlessly when all modules have been combined. Integrity testing evaluates how effectively various parts come together to form a single entity. This testing shows how easily modifications to the software's numerous interrelated components work. Integrity testing is used to find bugs and issues in the different user interfaces of the programme.

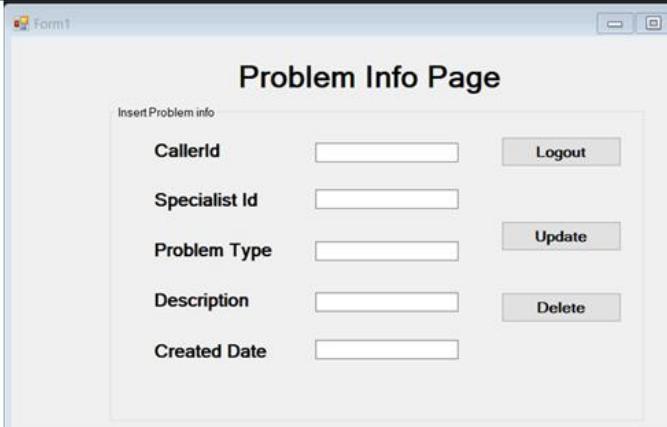
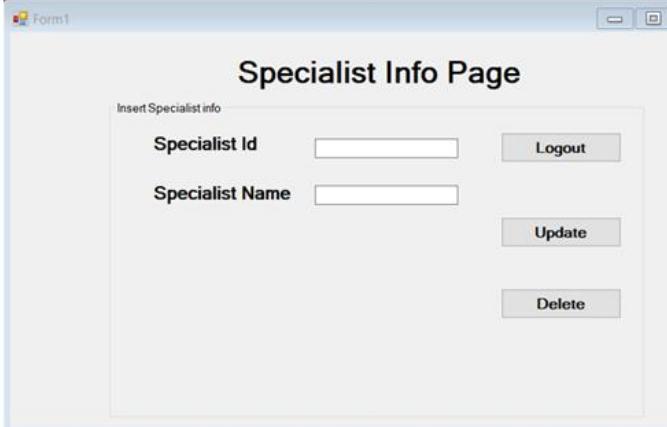
Stress test

Stress testing, a sort of non-functional testing, determines if a system can withstand excessive network traffic, under-clocking, which reduces a computer's power consumption, over-clocking, which boosts a computer's performance, and resource utilization demands at their maximum levels. Stress testing is often known as "endurance testing." In order to assess the reliability and stability of the AUT (Application Under Test), this testing strategy places a significant burden on its resources. It allows testers to ascertain the system's endurance capacity or point of no return before failure. In order to find any technological errors or faults that might cause SLA deviations, the goal of this testing is to ascertain whether the Service Level Agreement (SLA) is being adhered to. This testing frequently helps testers find certain critical errors that, in some cases, can be hard to find using conventional testing approaches. It provides a way to precisely gauge the system's performance. Additionally, it effectively evaluates how the system responds and manages faults under demanding and demanding conditions.

(Reqtest, 2019)

What was tested?	Expected/Actual Output	Remarks
Login Interface		The Login Interface will allow the user to login to the programmed and open the welcome form.
Welcome Interface		The application's Welcome Interface can be used to create a new user.

Caller Info page		<p>The Caller Info Page, which is incorporated into the main form, is used to enter caller information on the application.</p>
Call Info Page		<p>The Call Info Page, which is incorporated into the main form, is used to enter the Call specifics and information.</p>

Problem Info Page		The Problem Info Page is used to enter all relevant information regarding the situation.
Specialist Info Page		The Specialist Info Page is used to enter information on specialists.

Data Type Check

Insert Check

Project Name	United Limited
Test case ID	1
Test Engineer	Krishna Prasad Baigai
Test Scenario	Insert data into database with correct datatype
Test Step	Insert the data into the table with the correct SQL syntax
Test Data	Users (Krishna, Krishna123),(Radha, Radha321)
Expected Result	System should show success message on output
Action Result	Success message shown
Status	Pass

Query:

```
insert into users
(UserName, Password)
Values
('Krishna','Krishna123'),
('Radha','Radha321');
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists databases, including 'United.Limited'. The central pane displays a query window with the following SQL code:

```
insert into users
(UserName, Password)
Values
('Krishna','Krishna123'),
('Radha','Radha321');
```

Below the query window, the 'Messages' pane shows the results of the execution:

- (2 rows affected)
- Completion time: 2023-07-09T19:05:39.9389252+05:45

At the bottom of the screen, a status bar indicates 'Ready'.

Project Name	United Limited
Test case ID	2
Test Engineer	Krishna Prasad Bajgai
Test Scenario	Insert data into database with correct datatype
Test Step	Insert the data into the table with the correct SQL syntax
Test Data	Callers('Basantapuri','Doctor','9845534902','Dhamak','Basantadhamak11@gmail.com'), ('Madan Nepal','Engineer','9846534872','Tandi','MadanChitwan22@gmail.com')
Expected Result	System should show success message on output
Action Result	Error message was shown
Status	Pass

Query:

```
insert into Callers
(CallName, CallerJobTitle, CallerPhone, CallerAddress, CallerEmail)
Callers
('Basantapuri','Doctor','9845534902','Dhamak','Basantadhamak11@gmail.com'),
('Madan Nepal','Engineer','9846534872','Tandi','MadanChitwan22@gmail.com');
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The title bar reads "SQLQuery6sql - DESKTOP-OB8HM7\Krushna (61)\ - Microsoft SQL Server Management Studio". The main window has two panes: "Object Explorer" on the left and a "Query" window on the right. The "Object Explorer" pane shows a connection to "DESKTOP-OB8HM7\SQLEXPRESS01 (SG)". The "Query" window contains the following SQL code:

```
insert into Callers
(CallerName, CallerJobTitle, CallerPhone, CallerAddress, CallerEmail)
values
('Basanta puri', 'Doctor is from a kantipur hospital who lives in pokhara district
', '9845534902', 'Dhamak', 'Basantadhamak11@gmail.com')
```

The status bar at the bottom indicates "Query completed with errors." and "Compilation time: 2022-07-12T00:26:51.3899700+05:00".

Data Length Testing

Query:

```
insert into Callers
(CallerName, CallerJobTitle, CallerPhone, CallerAddress, CallerEmail)
values
('Basanta puri', 'Doctor is from a kantipur hospital who lives in pokhara district
', '9845534902', 'Dhamak', 'Basantadhamak11@gmail.com')
```

Output:

```

File Edit View Query Project Tools Window Help
Quick Launch (Ctrl+Q) P X
Object Explorer + x SQLQuery6sql - DESKTOP-OB8HM7\Krishna (70)\ - Microsoft SQL Server Management Studio
Connect + 49 + x DESKTOP-OB8HM7\Krishna (70)\ - UnitedLimited (DESKTOP-OB8HM7\Krishna (70)\ - Microsoft SQL Server Management Studio)
Database Security Server Objects Replication Management XEvent Profiler
+ DESKTOP-OB8HM7\SQLEXPRESS01 (SG)
+ Database
+ Security
+ Server Objects
+ Replication
+ Management
+ XEvent Profiler
+ Execute ✓ 100% 100% 100%
+ Messages
Msg 2013, Level 16, State 1
String or binary data would be truncated in table 'UnitedLimited.dbo.Callers', column 'CallerName'. Truncated value: 'Doctor is from a kantipur hospital who lives in pokhara district '.
The statement has been terminated.
Compilation time: 2022-07-16T12:45:26.8871994+05:45
+ Query completed with errors.
DESKTOP-OB8HM7\SQLEXPRESS01 DESKTOP-OB8HM7\Krishna UnitedLimited 000000 0 rows
Ready 165 Col 1 Ch 1 INS

```

```

insert into Callers
(CallName, CallerJobTitle, CallerPhone, CallerAddress, CallerEmail)
values
('Basanta puri', 'Doctor', '9845534902', 'Dhamak', 'Basantadhamak11@gmail.com')

```

The screenshot shows the Microsoft SQL Server Management Studio interface. A query is being run against the 'UnitedLimited' database. The query inserts data into the 'Callers' table. The output window displays a success message: 'Msg 2013, Level 16, State 1 String or binary data would be truncated in table 'UnitedLimited.dbo.Callers', column 'CallerName'. Truncated value: 'Doctor is from a kantipur hospital who lives in pokhara district '. The statement has been terminated.' followed by the compilation time information.

Project Name	United Limited
Test case ID	4
Test Engineer	Krishna Prasad Bajgai
Test Scenario	Insert data into database with correct data length
Test Step	Insert the data into the Callers table with character less than 50
Test Data	('Basanta puri', 'Doctor', '9845534902', 'Dhamak', 'Basantadhamak11@gmail.com'),
Expected Result	Success message should be shown as output.
Action Result	Success message was shown
Status	Pass

Query:

```

insert into Callers
(CallName, CallerJobTitle, CallerPhone, CallerAddress, CallerEmail)
values
('Basanta puri', 'Doctor', '9845534902', 'Dhamak', 'Basantadhamak11@gmail.com')

```

Output:

```

File Edit View Query Project Tools Window Help
New Query Execute
Object Explorer
Connect > DESKTOP-OB8HM7\SQLEXPRESS01 UnitedLimited (DESKTOP-OB8HM7\Krishna (70)) - Microsoft SQL Server Management Studio
Object Explorer
SClQuery6sql - DESKTOP-OB8HM7\Krishna (70) * < X
Insert into Callers
(
    CallerName, CallerJobTitle, CallerPhone, CallerAddress, CallerEmail
)
values
(
    'Basanta puri', 'Doctor', '9845534902', 'Basantadhamak11@gmail.com'
)

100 % < > Messages
1 row affected
Compilation time: 2022-07-12T13:00:48.2443198+05:00
100 % < > Query executed successfully.
DESKTOP-OB8HM7\SQLEXPRESS01 DESKTOP-OB8HM7\Krishna UnitedLimited 000000 0 rows
Ready Col 1 Ch 1 INS

```

Not Null Testing

Project Name	United Limited
Test case ID	5
Test Engineer	Krishna Prasad Bajgai
Test Scenario	Insert data into database with correct data type and SQL syntax
Test Step	Insert the data with null value in not null domain name with the correct SQL syntax in the table
Test Data	(‘Basanta puri’, ‘Doctor’ , ‘9845534902’ , ‘Basantadhamak11@gmail.com’),
Expected Result	Error message should be shown as output.
Action Result	Error message was shown
Status	Pass

Query:

```

insert into Callers
values
('Basanta puri', 'Doctor', '9845534902', 'Basantadhamak11@gmail.com')

```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'SQLQuery6.sql - DESKTOP-OB8HM7\Krishna (70)\UnitedLimited - Microsoft SQL Server Management Studio' is open. The query is:

```
insert into Callers
values
('Ravanta puri', 'Doctor', '9849534982', 'Ravantadhamok11@gmail.com')
```

The status bar at the bottom indicates 'Query completed with errors'. The error message in the messages pane is:

```
Msg 213, Level 16, State 1, Line 1
Column name or number of supplied values does not match table definition.
```

Project Name	United Limited
Test case ID	6
Test Engineer	Krishna Prasad Bajgai
Test Scenario	Insert data into database with correct data type and SQL syntax
Test Step	Insert the data into problem table which have foreign key integration from Caller table with 100 value in CallerId
Test Data	('100', '2', 'Hardware', 'App is not installing')
Expected Result	Error message should be shown as output.
Action Result	Error message was shown
Status	Pass

Query:

```
insert into Problem

(CallerdId, SpecialistId, ProblemtType, Problemdescription)

values

('100', '2', 'Hardware', 'App is not installing')
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. In the Object Explorer, the connection is to DESKTOP-DBBHM7\SQLEXPRESS01\UnitedLimited. The current database is UnitedLimited. In the center pane, a query window titled 'SQLQuery6.sql - DESKTOP-DBBHM7\SQLEXPRESS01\UnitedLimited (DESKTOP-DBBHM7\Krishna (70)) - Microsoft SQL Server Management Studio' contains the following SQL code:

```
insert into Problem
(CallerId, SpecialistId, ProblemType, ProblemDescription)
values
('100', '2', 'Hardware', 'App is not installing')
```

In the 'Messages' pane below, an error message is displayed:

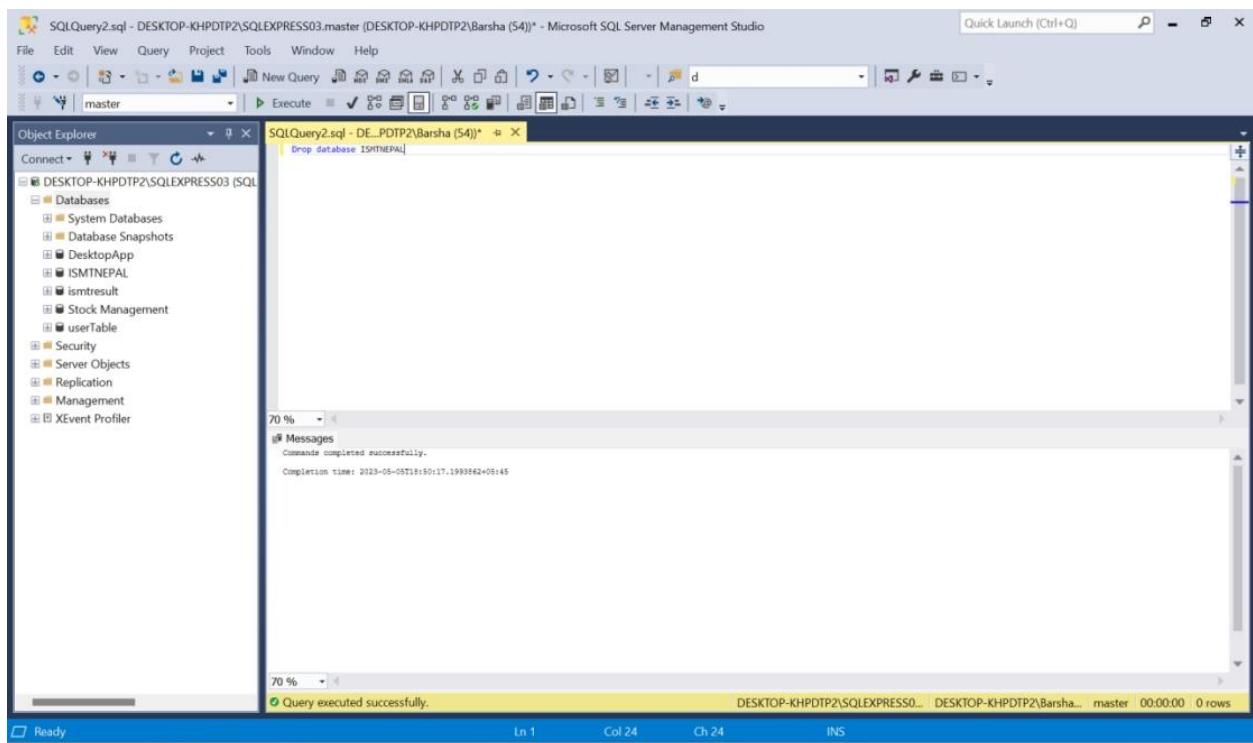
```
Msg 647, Level 16, State 1, Line 1
The INSERT statement conflicted with the CHECK constraint "CN_Problem_CallerId_MustBeValid". The conflict occurred in database "UnitedLimited", table "dbo.Problem", column 'CallerId'.
The statement has been terminated.
```

At the bottom of the screen, a status bar shows 'Ready' and other system information.

Recovery Testing

Project Name	United Limited
Test case ID	7
Test Engineer	Krishna Prasad Bajgai
Test Scenario	Recover the deleted database
Test Step	Delete the database <u>UnitedLimited</u> then try to recover it from the backup drive
Test Data	Restore database <u>UnitedLimited</u> From disk = 'd:/backup/backup.bak' (‘1’,’1’,’11:33’,’00:04:31’,’2080-1-3’)
Expected Result	Database should be recover successfully
Action Result	Success message should be shown as an output
Status	Pass

Output:



Primary Key testing

Project Name	United Limited
Test case ID	8
Test Engineer	Krishna Prasad Baigai
Test Scenario	Insert data into the database with correct data type an SQL syntax but insert same data multiple time
Test Step	Insert the same data multiple time into Calls table
Test Data	('1','1','11:33','00:04:31','2080-1-3') ('1','1','11:33','00:04:31','2080-1-3') ('2','2','12:46','00:02:51','2080-1-3')
Expected Result	<u>CallId</u> should be different for every values and <u>success</u> A message should be shown
Action Result	Success message with different <u>CallId</u> was shown
Status	Pass

Query:

```
insert into Calls  
  
(CallerId, OperatorId, CallTime, CallDuration, CallDate)  
  
Values  
  
('1','1','11:33','00:04:31','2080-1-3'),  
('1','1','11:33','00:04:31','2080-1-3'),  
('2','2','12:46','00:02:51','2080-1-3');
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the connection is to DESKTOP-OB8HM7\SQLEXPRESS01\UnitedLimited (DESKTOP-OB8HM7\Krishna (61)). The main window has two panes: Object Explorer on the left and a Results pane at the bottom. In the Object Explorer, under the 'Tables' node, there is a 'Calls' table. The Results pane displays the output of an INSERT query into the 'Calls' table. The query is:

```
insert into Calls  
(CallerId, OperatorId, CallTime, CallDuration, CallDate)  
Values  
(1, '1', '11:33', '00:04:33', '2080-1-3'),  
(1, '1', '11:33', '00:04:33', '2080-1-3'),  
(2, '2', '12:46', '00:02:53', '2080-1-3')
```

The results show three rows inserted into the 'Calls' table:

CallId	CallerId	OperatorId	CallTime	CallDuration	CallDate
1	1	2	12:48:00	00:02:51	2080-01-03
2	4	1	11:33:00	00:04:31	2080-01-03
3	3	1	11:32:00	00:04:31	2080-01-03

At the bottom of the screen, a status bar shows 'Query executed successfully.' and the command history: DESKTOP-OB8HM7\SQLEXPRESS01\UnitedLimited - 000000 : 3 rows.

Performance Testing

Query:

Query:

```
insert into Specialist  
(SpecialistName, SolutionInfo)  
values  
('Krishna Bajgai', 'fff'),  
('Radha Bajgai','fff'),  
('Krishna Bajgai', 'fff'),  
('Radha Bajgai','fff')
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the connection is to DESKTOP-OB8HM7\SQLEXPRESS01\UnitedLimited (DESKTOP-OB8HM7\Krishna (61)). The Object Explorer on the left shows the database structure, including the United_Limited database and its tables like dbo.Specialist. The main results pane displays the execution of an INSERT query into the Specialist table, inserting 100 rows. The status bar at the bottom shows "Query executed successfully".

```
Object Explorer
File Edit View Query Project Tools Window Help
New Query New Item Task List Execute
Object Explorer
Connect
DESKTOP-OB8HM7\SQLEXPRESS01\UnitedLimited
Databases
System Databases
Database Snapshots
cnpt
FitnessClub
freshmart
GlobalTrading
hello
john
irina
Mydatabase
itan
try
United_Limited
UnitedUnlimited
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Calls
dbo.Calls
dbo.Operators
dbo.Problem
dbo.specialist
dbo.specialist1
dbo.Users
Views
External Resources
Synonyms
Programmability
Query Store
Service Broker
Storage
Security
100 %
Messages
114 rows affected
Compilation time: 2019-07-03T14:01:44.2983180+05:00
Query executed successfully.
```

Evaluate Database effectiveness according to user needs and make recommendations for improvement

A database developer at United Limited, I am committed to ensuring the efficient functioning of the database and meeting the needs of our users. We prioritize data security, performance, and user-friendly interfaces while continuously striving for improvements to enhance usability and accommodate future growth. The database's design incorporates various techniques, including ER diagrams and DFDs, and employs different types of queries to manipulate data effectively. By adhering to best practices and continuously evaluating and enhancing the database, we ensure that United Limited has a reliable and scalable database solution to support its operations.

Querying process of Selecting

Query:

```
SQLQuery4.sql - DE...HM7\Krishna (52)* * X
Select * from Problem
```

Output:

Results Messages

	ProblemId	CallerId	SpecialistId	ProblemType	ProblemDescription
1	1	1	1	Hardware	fffffff
2	2	2	2	Software	fffffff

Query:

SQLQuery4.sql - DE...HM7\Krishna (52)* Select * from Users

Output:

Results Messages

	UserId	UserName	Password
1	1	Krishna	Krishna123
2	2	Radha	Radha321

Query:

SQLQuery4.sql - DE...HM7\Krishna (52)* Select * from Calls

Output:

Results Messages

	CallId	CallerId	OperatorId	CallTime	CallDuration	CallDate
1	1	1	1	12:01:00.0000000	00:03:44.0000000	2080-02-03
2	2	1	1	11:33:00.0000000	00:01:58.0000000	2080-02-03

Query:

SQLQuery4.sql - DE...HM7\Krishna (52)* Select * from Callers

Output:

Results Messages

	CallerId	CallerName	CallerJobTitle	CallerPhone	CallerAddress	CallerEmail
1	1	Basanta puri	Doctor	9845534902	Dhamak	Basantadhamak11@gmail.com
2	2	Madan Nepal	Engineer	9846534872	Tandi	MadanChitwan22@gmail.com

Query:

```
SQLQuery4.sql - DE_HM7\Krishna (52)*  * X
Select * from operators
```

Output:

Results		Messages
	OperatorId	SpecialistName
1	1	Shyam Devkota
2	2	Lalumaya Adhikari

Query:

```
SQLQuery4.sql - DE_HM7\Krishna (52)*  * X
Select * from Specialist
```

Output:

Results		Messages	
	SpecialistId	SpecialistName	SolutionInfo
1	1	Bebika Thapa	wwwwww
2	2	Baburam Pant	wwwwww

Merits:

Constrains:

Update callers Set CallerName

Query:

```
Update Callers set CallerName = 'Binod Chaulagai'
Select * from Callers
```

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the connection is to 'DESKTOP-OB8HM7\SQLEXPRESS01.UnitedLimited'. The 'UnitedLimited' database is selected. In the center pane, a query window titled 'SQLQuery6.sql - DE...HM7\Krishna (61)*' contains the following SQL code:

```
Update Callers set CallerName = 'Binod Chaulagai'  
Select * from Callers
```

Below the code, the results pane shows a table with two rows of data:

CallerId	CallerName	CallerJobTitle	CallerPhone	CallerAddress	CallerEmail
1	Binod Chaulagai	Doctor	9846534902	Dhamak	Basantidhamak11@gmail.com
2	Binod Chaulagai	Engineer	9846534872	Tandi	MedanChitwan22@gmail.com

At the bottom of the results pane, a message bar indicates: 'Query executed successfully.'

Update Specialist set SpecialistName

Query:

```
Update Specialist set SpecialistName = 'Binod Chaulagai'  
where SpecialistId = 2  
Select * from Specialist
```

Output:

SQLQuery6.sql - DESKTOP-O88HM7\SQLEXPRESS01.UnitedLimited (DESKTOP-O88HM7\Krishna (61)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

UnitedLimited Execute

Object Explorer

Connect ▾

DESKTOP-O88HM7\SQLEXPRESS01 (SQ)

- Databases
- Security
- Server Objects
- Replication
- Management
- XEvent Profiler

SQLQuery6.sql - DE...HMT7\Krishna (61)) # ×

```
Update Specialist set SpecialistName = 'Binod Chaulagai'  
where SpecialistId = 2  
Select * from Specialist
```

100 %

Results Messages

	SpecialistId	SpecialistName	SolutionInfo
1	1	Bebika Thapa	wwwwww
2	2	Binod Chaulagai	wwwwww

Query executed successfully.

Ready

Effectiveness of testing with explanation of choice of the test data used

I well appreciate the necessity for comprehensive testing to certify the system's right operation and meet end-user needs as a dedicated database developer for United Limited's IT Help Desk. Testing includes a wide range of functional and non-functional tests to ensure that the system operates as anticipated and generates the desired outputs.

In order to simulate real-world events and take into account a variety of inputs and outputs, it is essential to develop tests that incorporate a variety of data types, data lengths, and other components. This eliminates any potential null checks and ensures that the system will only accept authentic and accurate input. All tests are conducted in environments that closely mimic the setup used in production in order to ensure consistent behaviour across different configurations.

Furthermore, the database is populated with substantial amounts of data to gauge the system's performance under realistic conditions. Performance tests are conducted to validate the system's ability to handle the anticipated workload efficiently. This ensures that the IT help-desk system can effectively manage the expected volume of user interactions without compromising its performance.

Testing is essential for finding and fixing problems, enhancing the overall quality of the system, reducing risks, guaranteeing compliance with industry standards, and enhancing system performance. The capacity to guarantee that the IT help-desk system performs as intended and fulfills the expectations of its users is one of the crucial variables that substantially influences the performance of United Limited's database. This objective must be accomplished using efficient testing techniques.

Any problems may be found, fixed, and possibilities for system improvement can be taken advantage of by carefully choosing the right test data and properly reviewing the test outcomes. In the end, this results in the creation of an IT help-desk system that successfully, consistently, and successfully responds to user requests.

Expert testing is a crucial phase in the development of a reliable and effective IT help-desk solution by United Limited. It's essential to guarantee system dependability, satisfy customer requests, and provide an enjoyable user experience.

Effectiveness of unit testing:

By increasing the stability and dependability of the programme, unit testing is essential to its development. Early testing of tiny pieces of the code allows for prompt identification of any faults, making their resolution simpler and more affordable. This repetitive testing procedure increases the software's overall quality by verifying that each component works as intended and lowering the likelihood of mistakes. Unit testing aids in separating the precise parts of the code that require attention when issues do arise, making the debugging process easier. Running tests after making modifications also ensures that the current functionality is preserved and prevents the emergence of new problems. Unit testing is a helpful activity that enhances the stability and integrity of software development. Unit tests are thought to improve the quality of the code. By testing individual functions or small blocks of code, developers may quickly identify any possible effective barriers and minor flaws inside a unit before commencing integration testing.

Conclusion:

After lengthy testing, it was discovered that the Help Desk system is perfectly running in line with the defined parameters provided in the business proposal. During the testing phase, a few minor issues like incorrect input validation and usability issues with the user interface were found. However, these issues are easily resolved and don't have a significant impact on the system's overall functionality.

Overall, it has been shown that the system is trustworthy, trustworthy, and appropriate for usage in a genuine situation. The Help Desk system has successfully helped clients with their technology issues and concerns, according to testing. The operations and

software have shown to be dependable even under high user loads, producing an excellent customer experience with prompt and accurate service.

Despite the discovery of a few minor interface and data entry vulnerabilities, the right actions have already been taken to promptly resolve these issues. The Help Desk system will offer customers and support agents a logical and user-friendly experience after these problems have been corrected.

The testing findings demonstrate the Help Desk system's high level of dependability and compliance with the standards necessary for an essential customer support tool. Once the issues found have been resolved, the Help Desk system will expedite routine technical assistance for customers and provide valuable direction to your committed support staff.

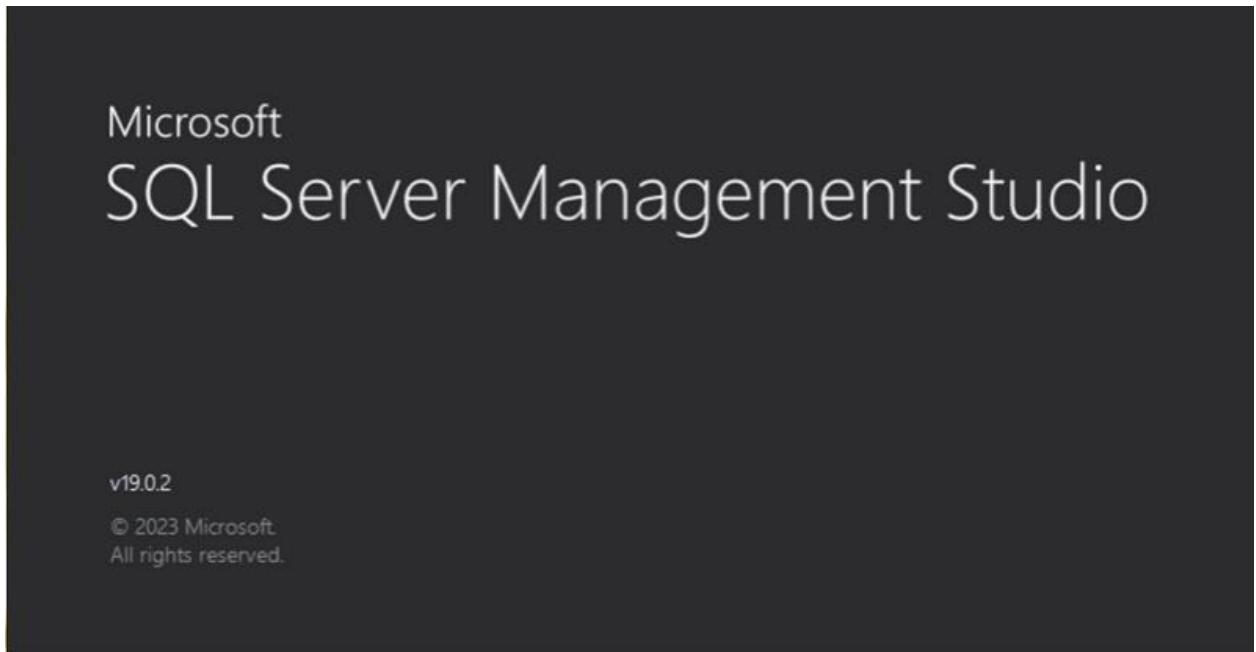
Task-5

Introduction

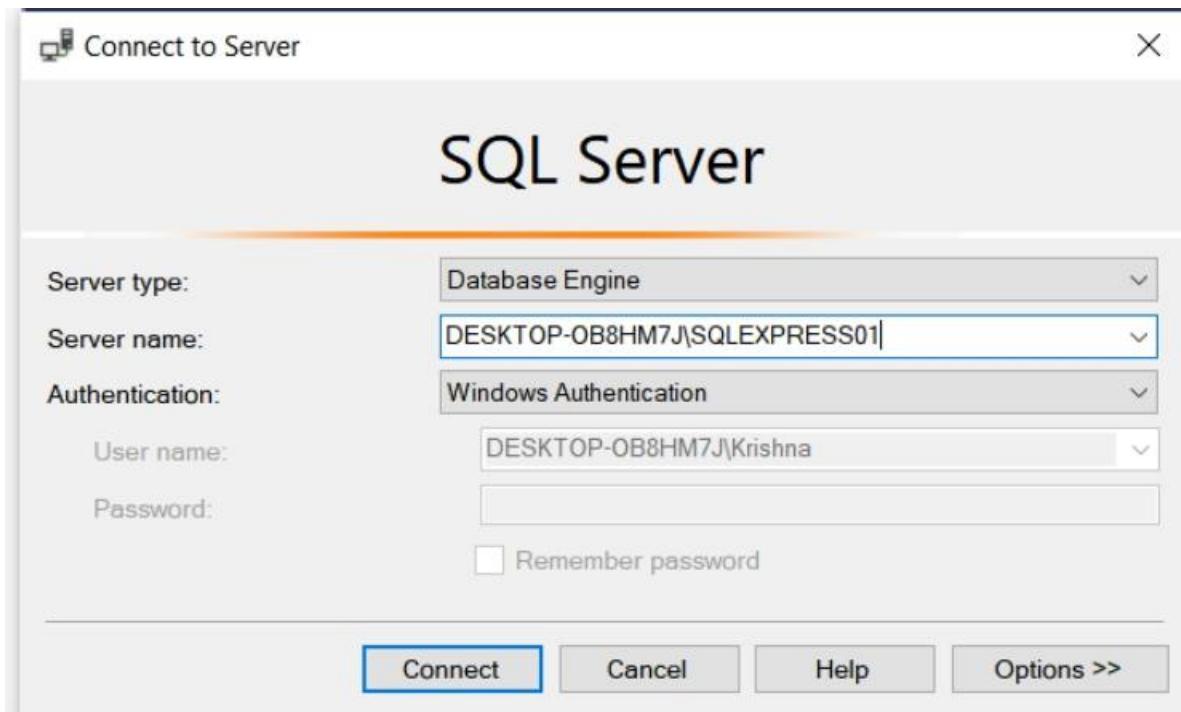
This report provides a detailed step-by-step procedure for effectively utilizing a database management system using Microsoft SQL Server Management Studio. The outlined steps will guide you through the process of utilizing the software to its fullest potential.

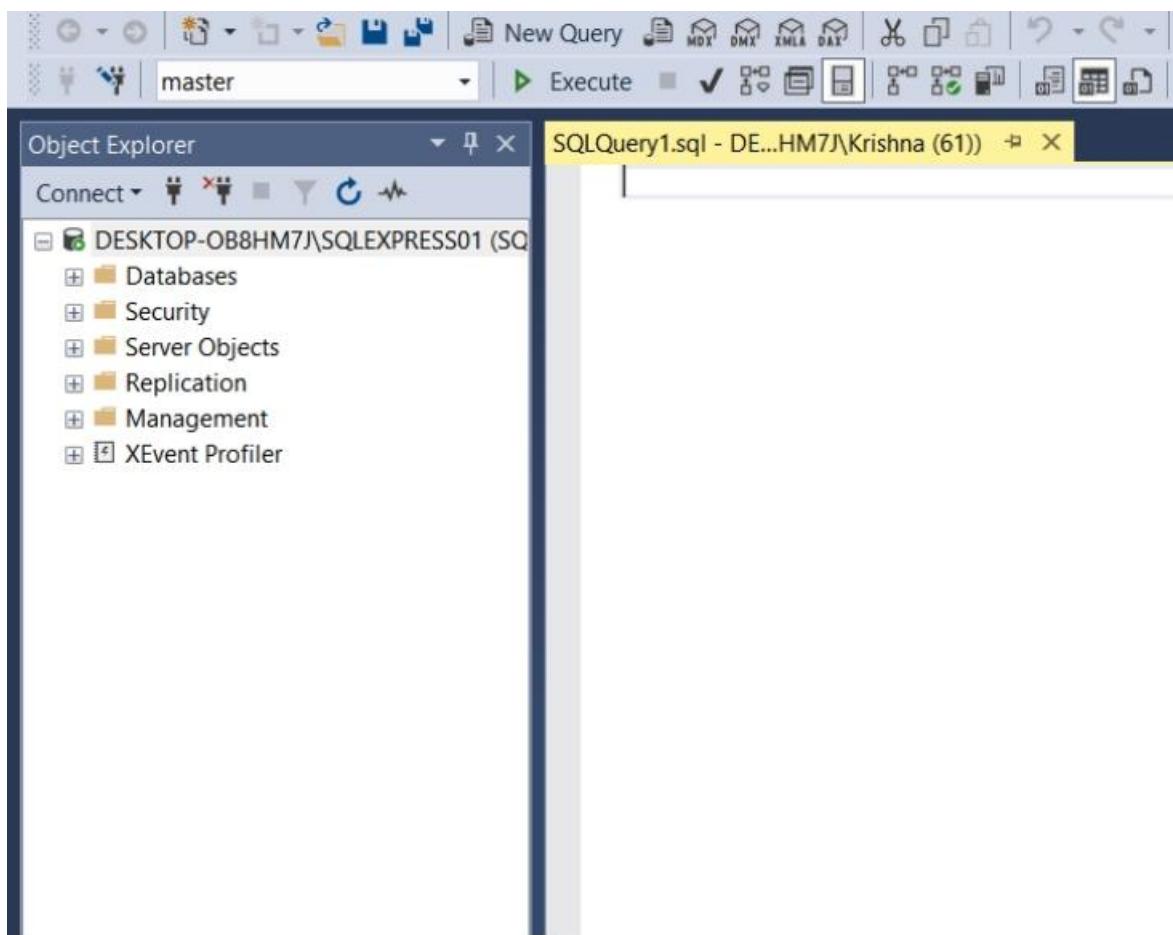
User Documentation

Start SQL server management studio

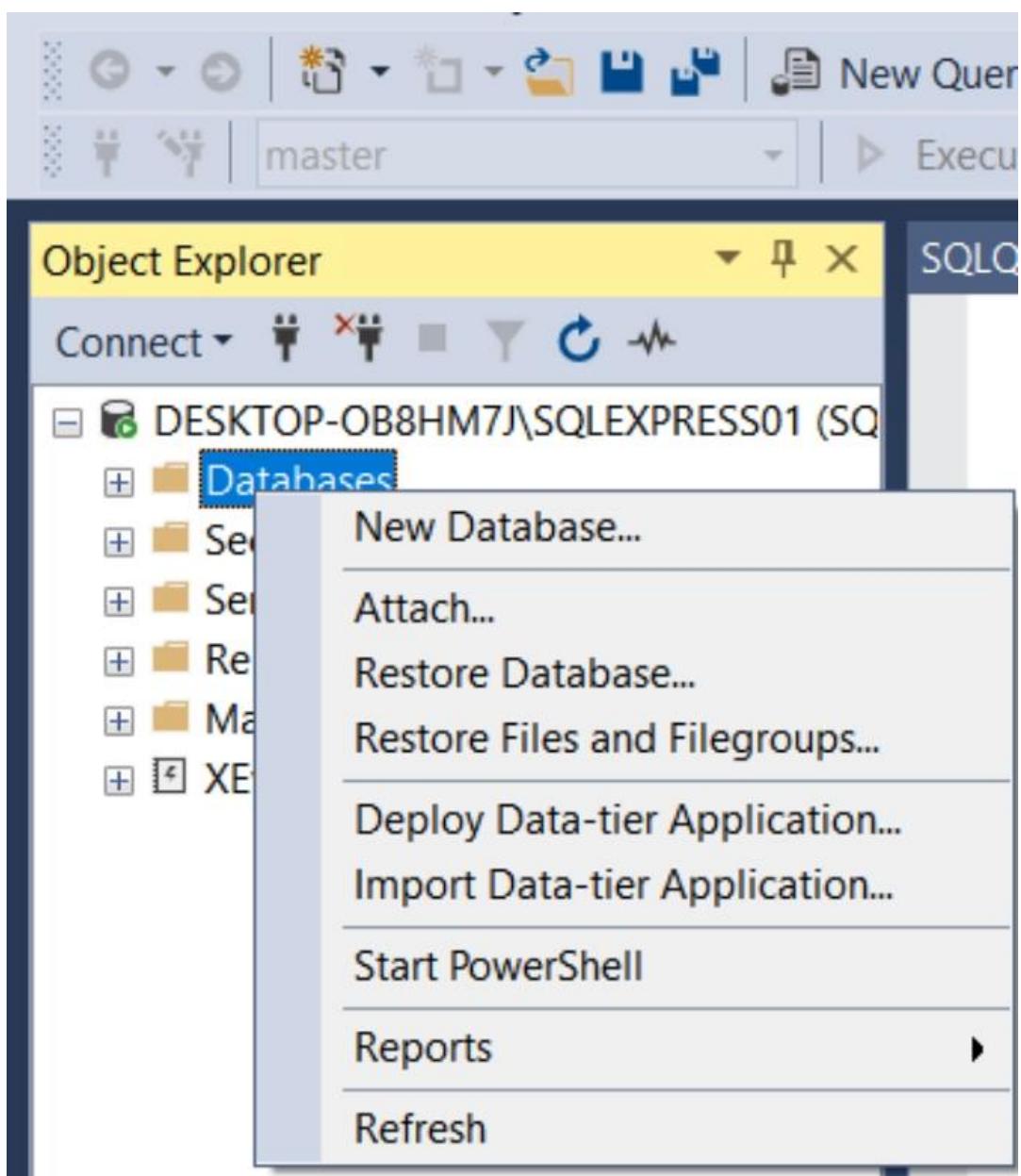


Connecting SQL

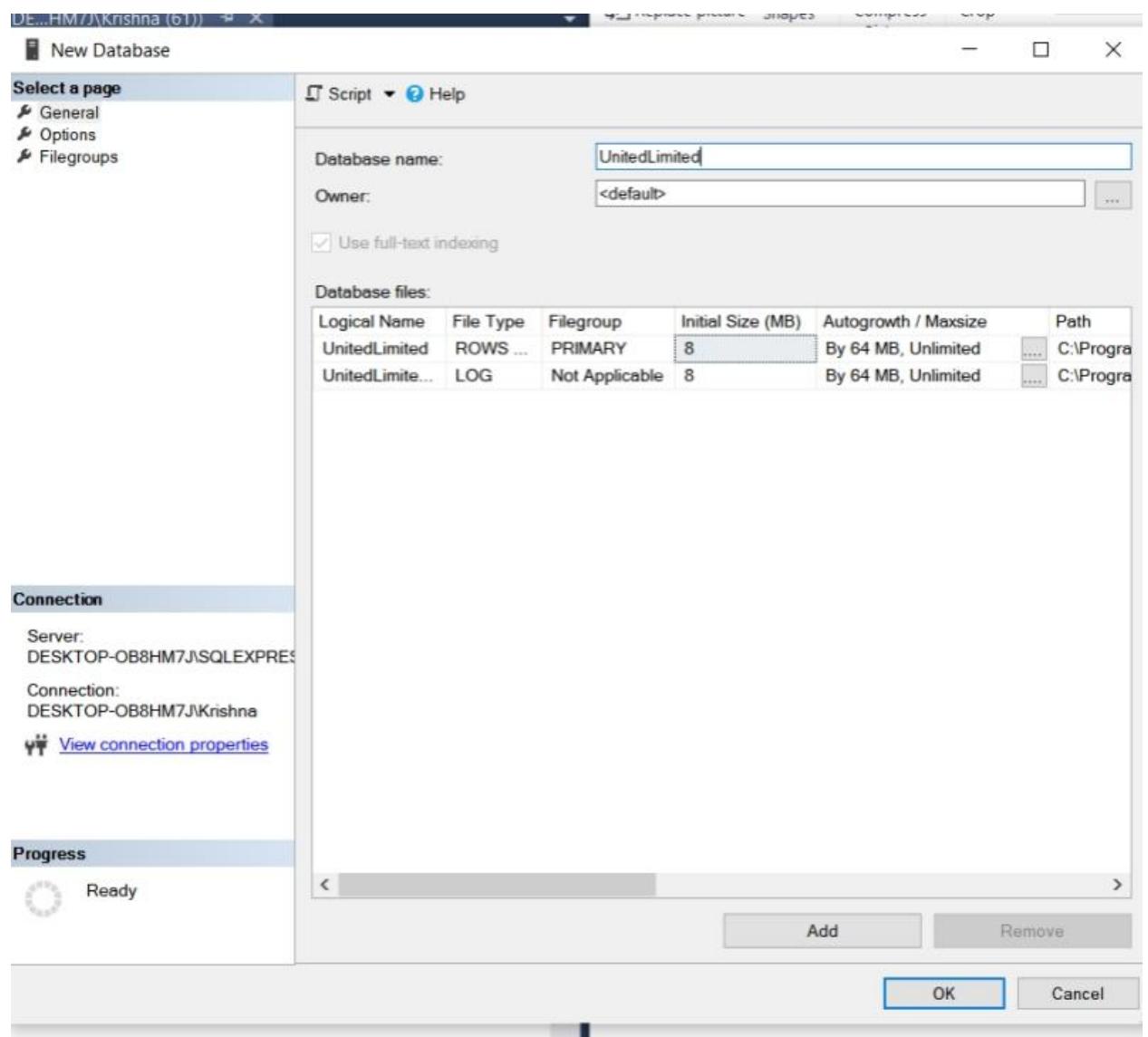


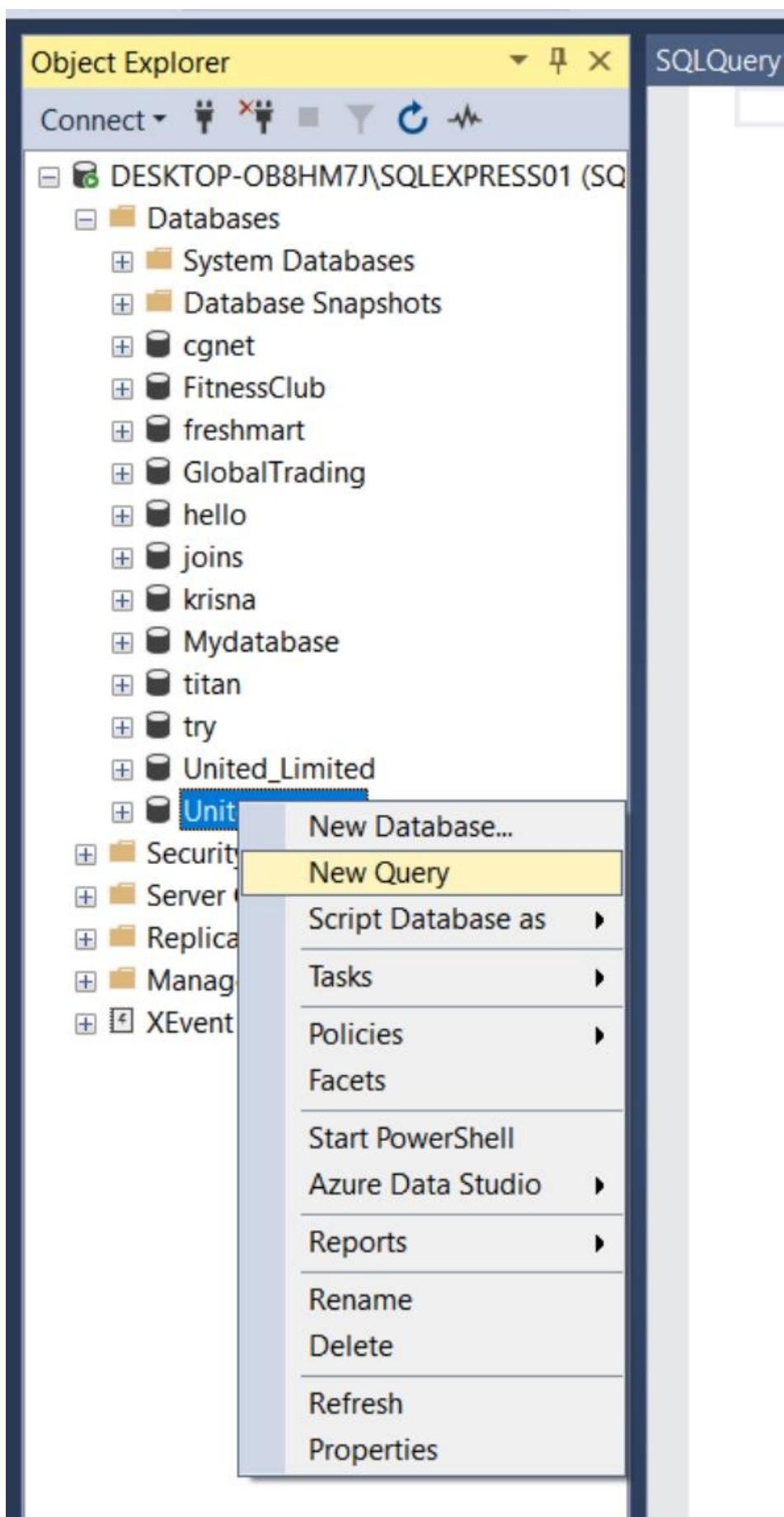


Right Click on Database



Enter the Database name and Click Add.





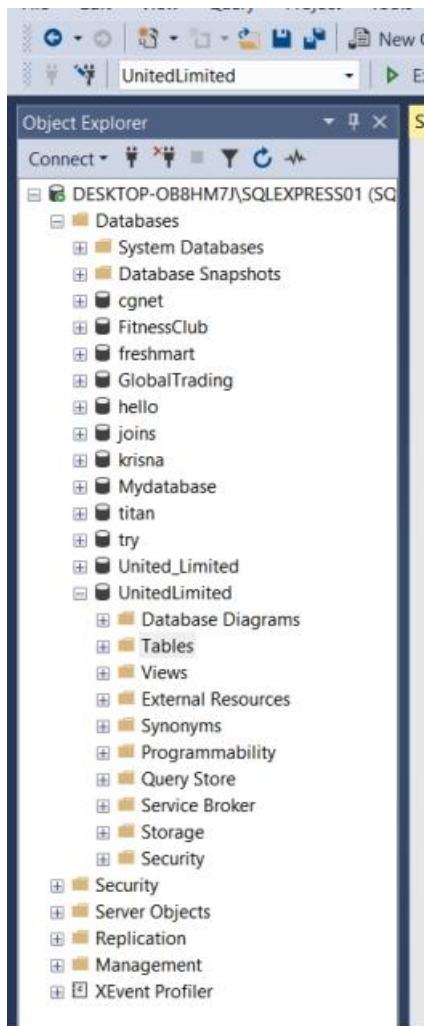
Create Query

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The title bar reads "SQLQuery1.sql - DESKTOP-HZFEHSL\SQLEXPRESS.UnitedLimited (DESKTOP-HZFEHSL\user [54]) - Microsoft SQL Server Management Studio". The Object Explorer sidebar shows a connection to "DESKTOP-HZFEHSL\SQLEXPRESS (SQL)". The main query window contains the following SQL code:

```
create database UnitedLimited
use UnitedLimited

create table Users(
    UserId int identity(1,1) primary key,
    UserName varchar(100) not null,
    password varchar(50) not null,
    role varchar(50) not null)
```

The status bar at the bottom right indicates "Activate Windows". Below the status bar, the taskbar shows the weather (24°C, Partly sunny), a search bar, and several pinned application icons.



Right click the table then check Select top 1000 rows again

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The Object Explorer on the left lists several databases, including DESKTOP-OB8HM7J\SQLEXPRESS01 and UnitedLimited. The UnitedLimited database is expanded, showing tables such as dbo.Callers, dbo.Calls, dbo.Operators, dbo.Problem, dbo.specialist, dbo.specialist1, and dbo.Users. The central pane contains a query window with the following T-SQL script:

```
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [CallerId]
    ,[CallerName]
    ,[CallerJobTitle]
    ,[CallerPhone]
    ,[CallerAddress]
    ,[CallerEmail]
FROM [UnitedLimited].[dbo].[Callers]
```

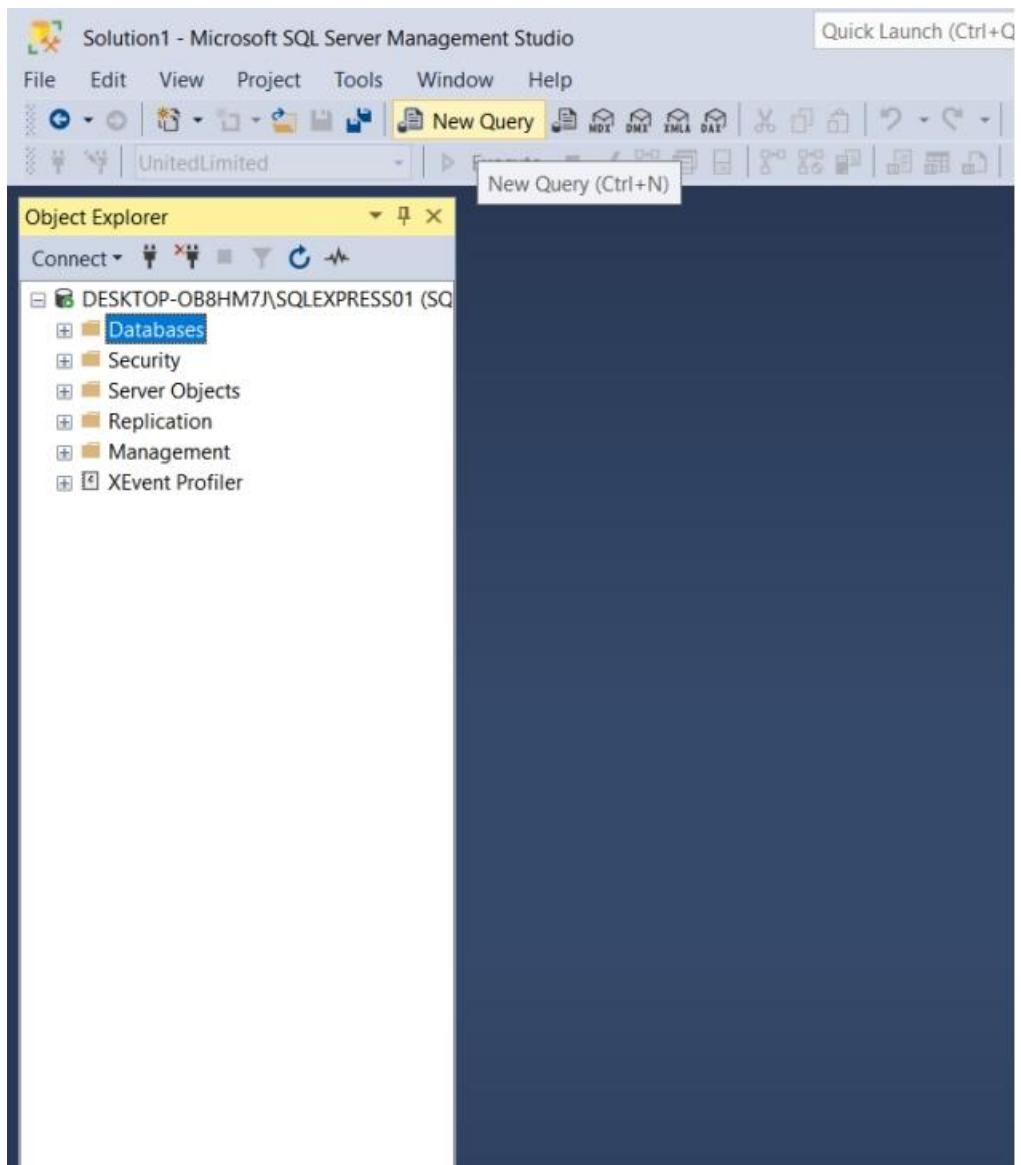
The results pane shows two rows of data from the Callers table:

	CallerId	CallerName	CallerJobTitle	CallerPhone	CallerAddress	CallerEmail
1	1	Basanta puri	Doctor	9845534902	Dhamak	Basantdhamak11@gmail.com
2	2	Madan Nepal	Engineer	9846534872	Tandi	MadanChitan22@gmail.com

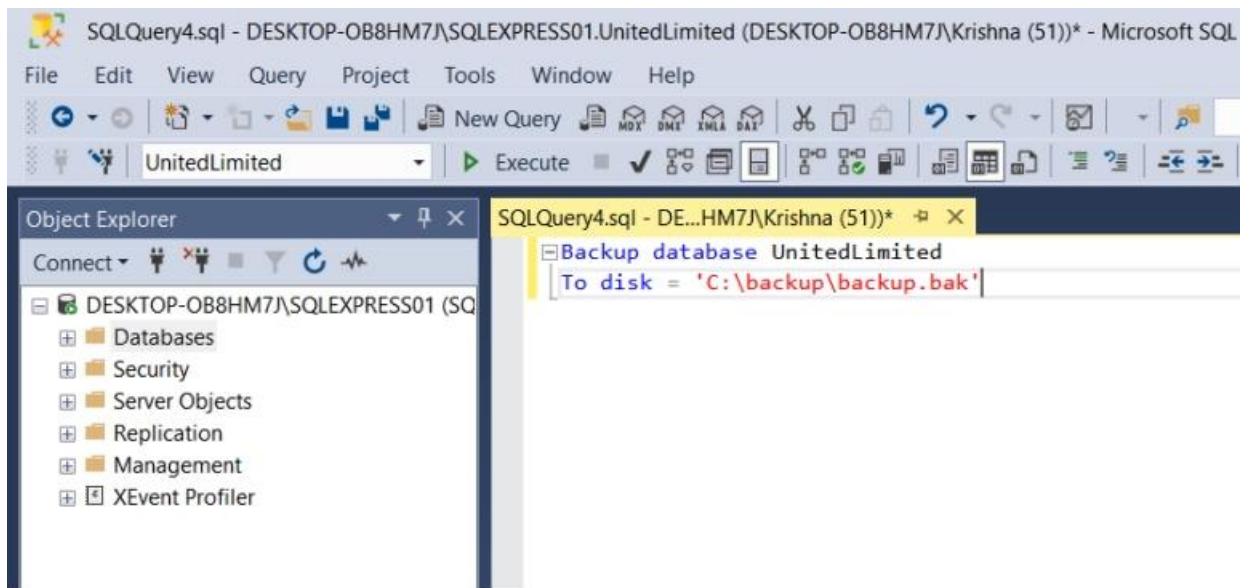
A status bar at the bottom indicates "Query executed successfully." and "Ln 1".

Backup Database

Click on new Query

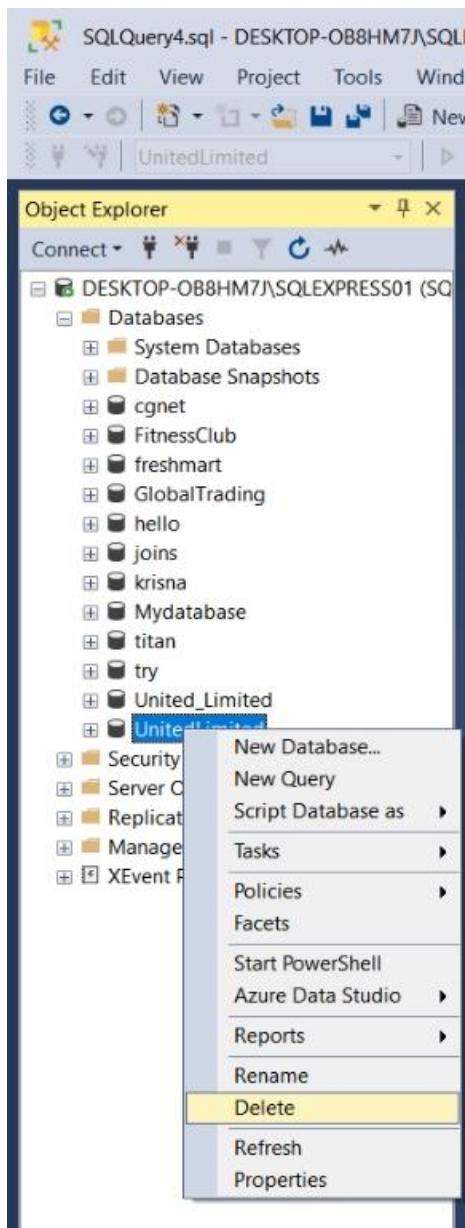


Sql Database for Back Up

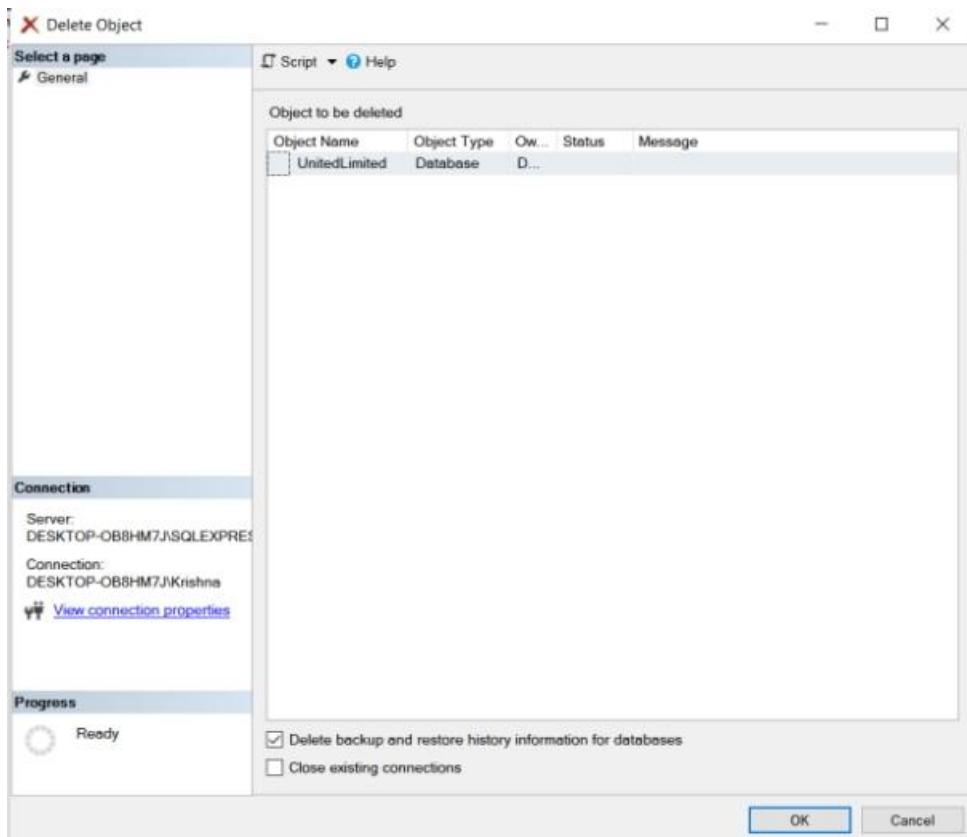


Delete Database

Click on Delete

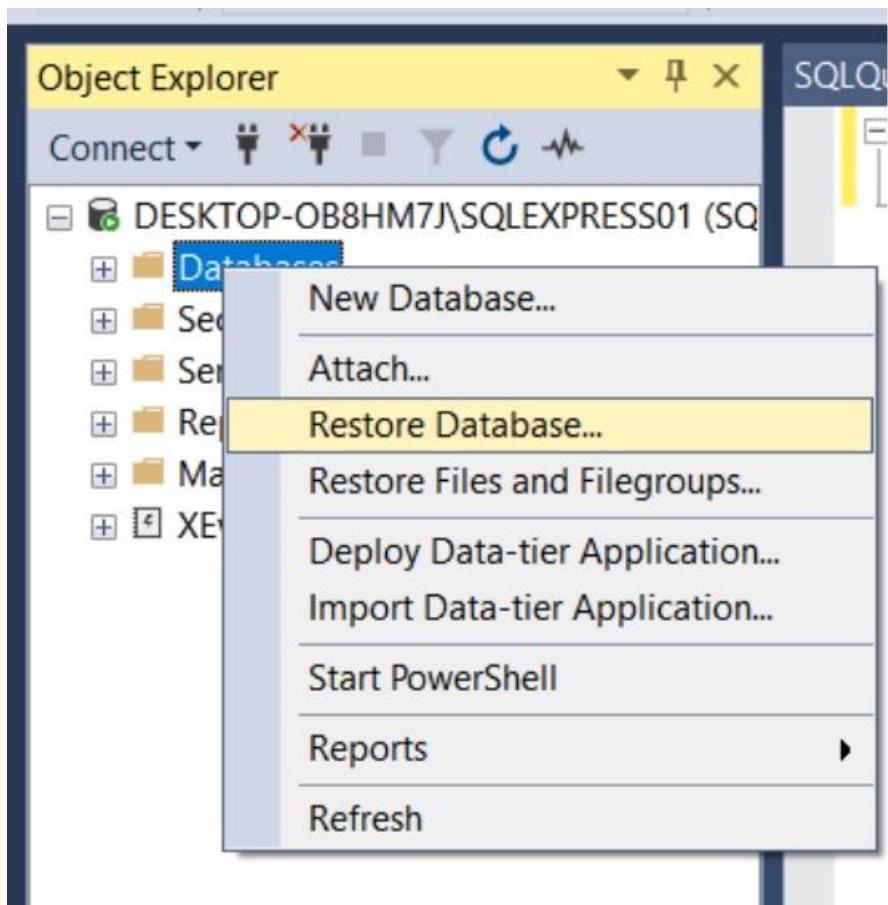


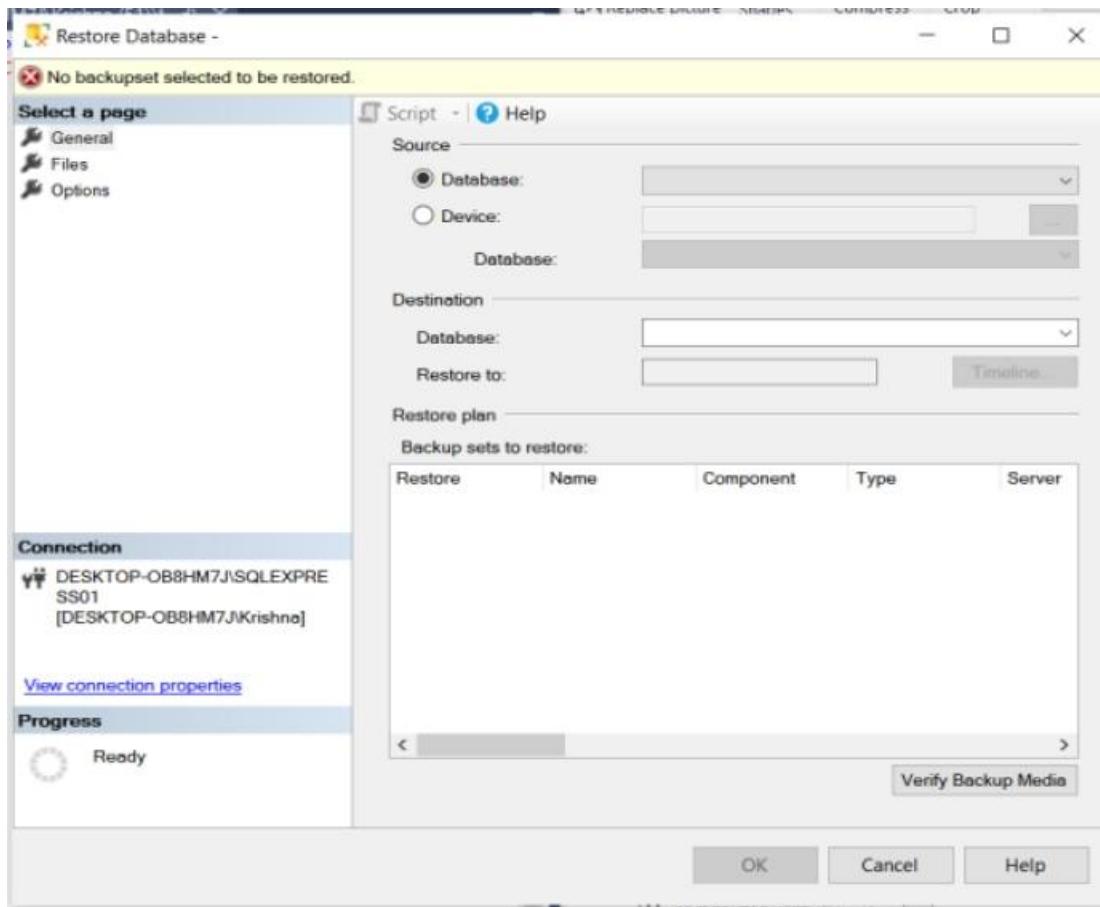
Click On OK:



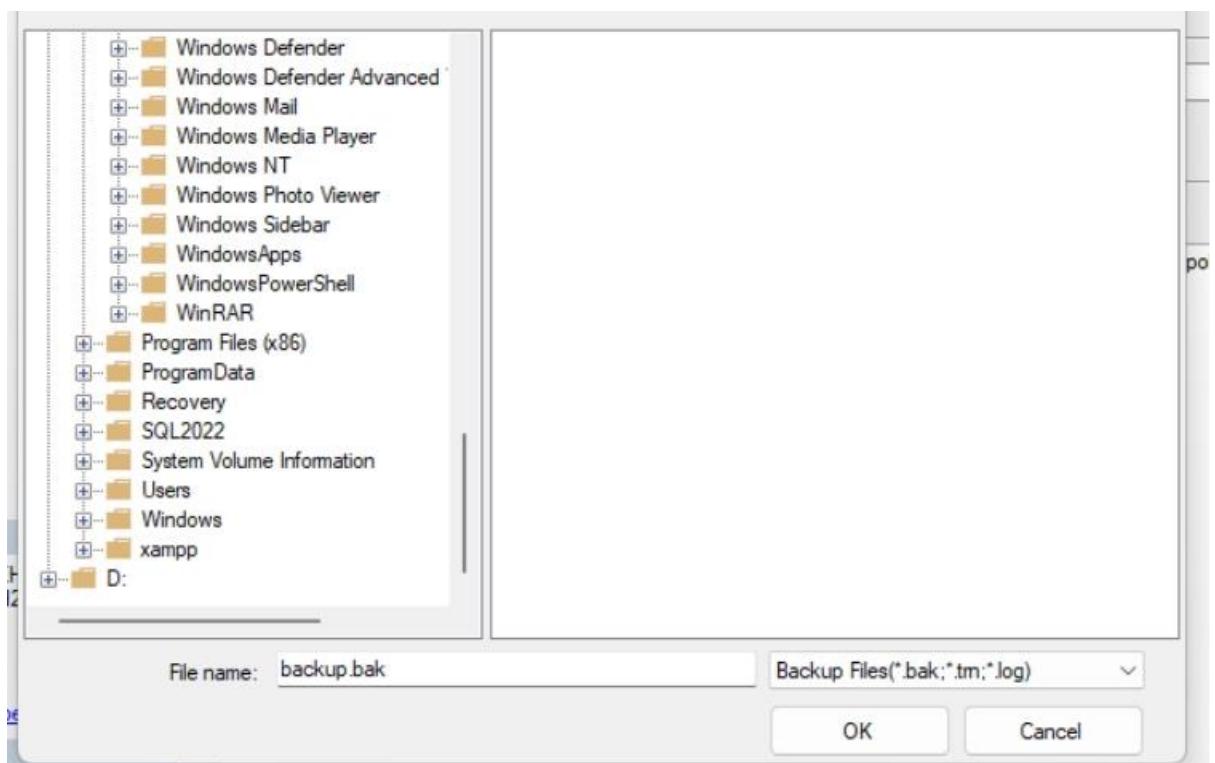
Restore Database

Click on Restore

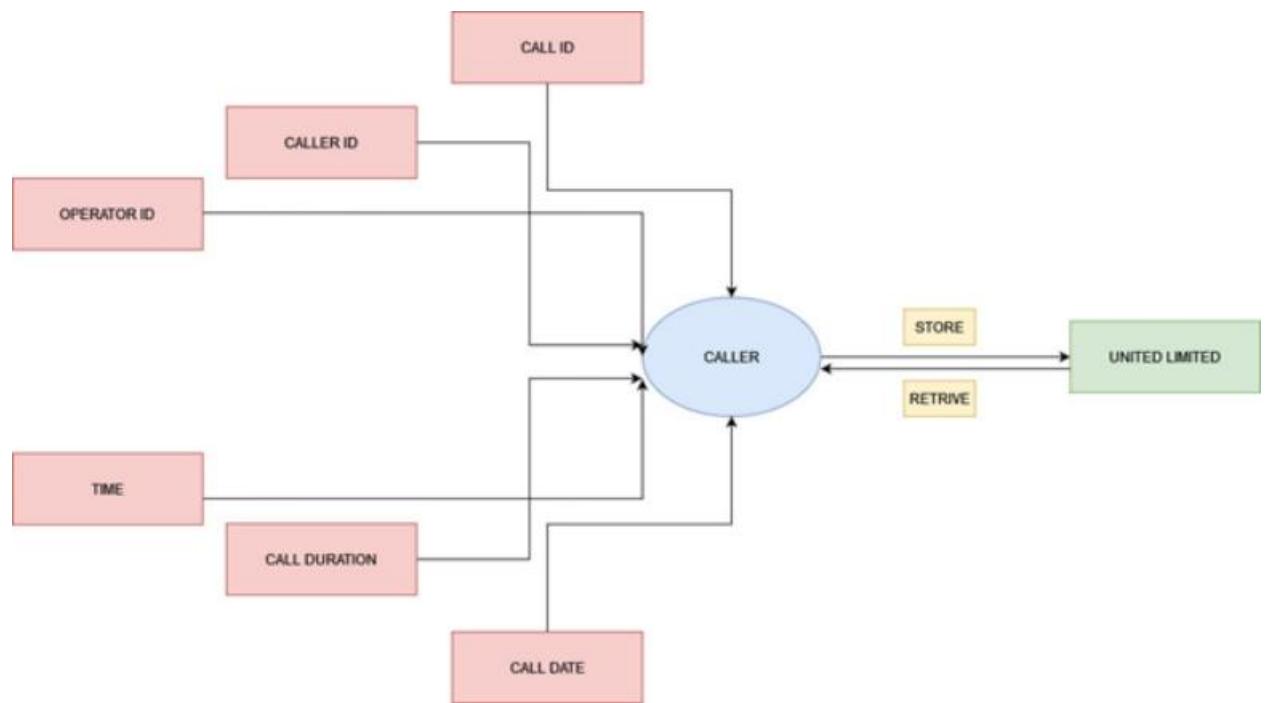


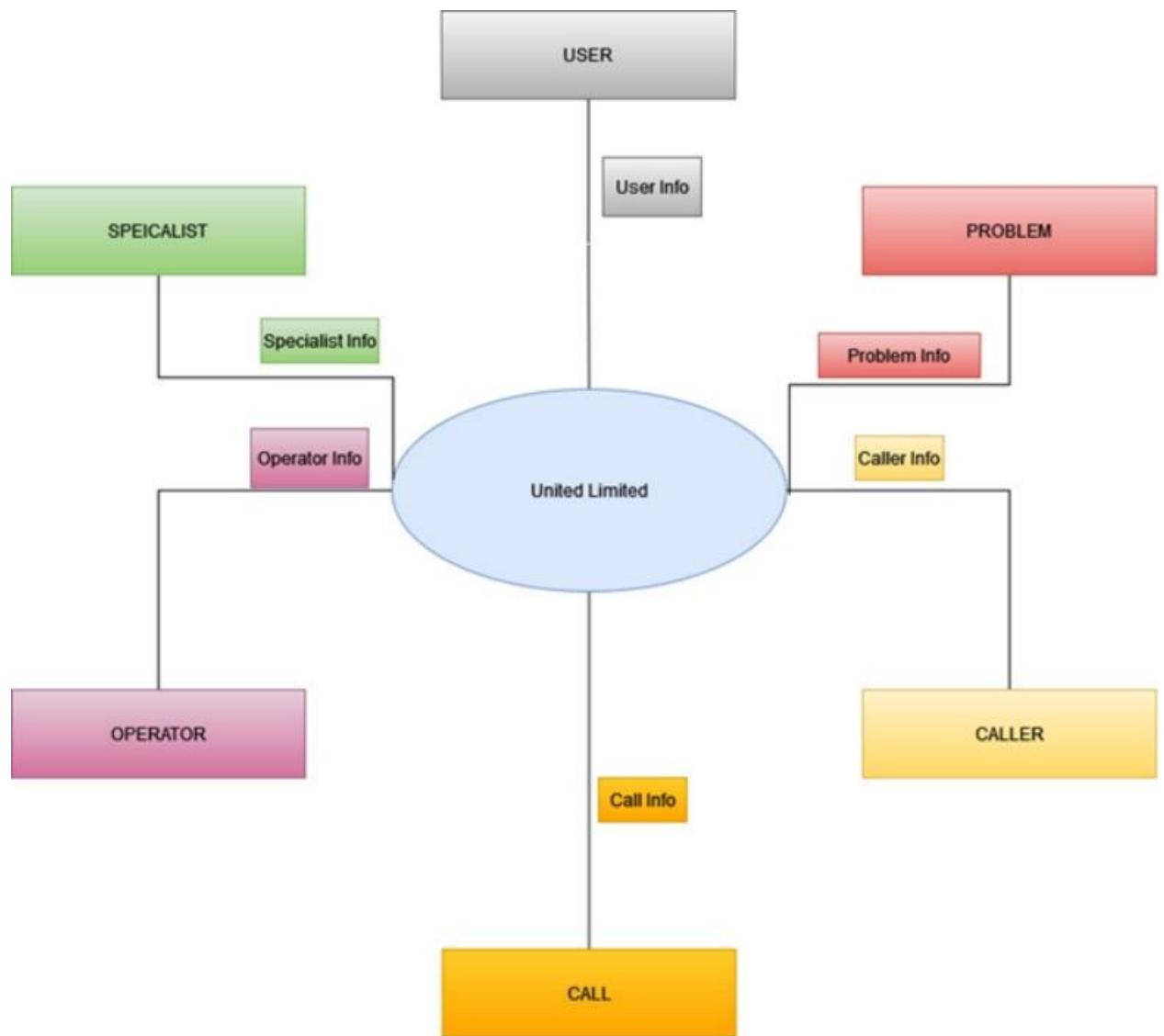


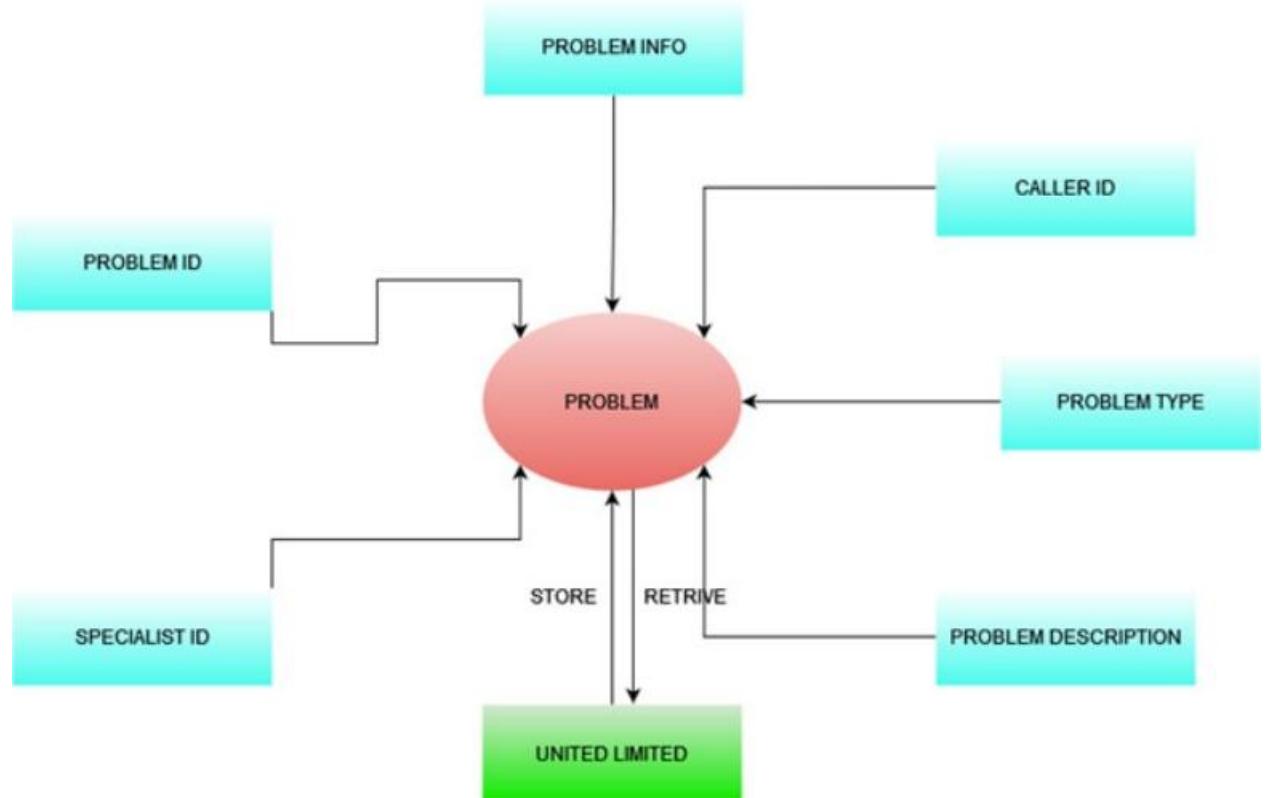
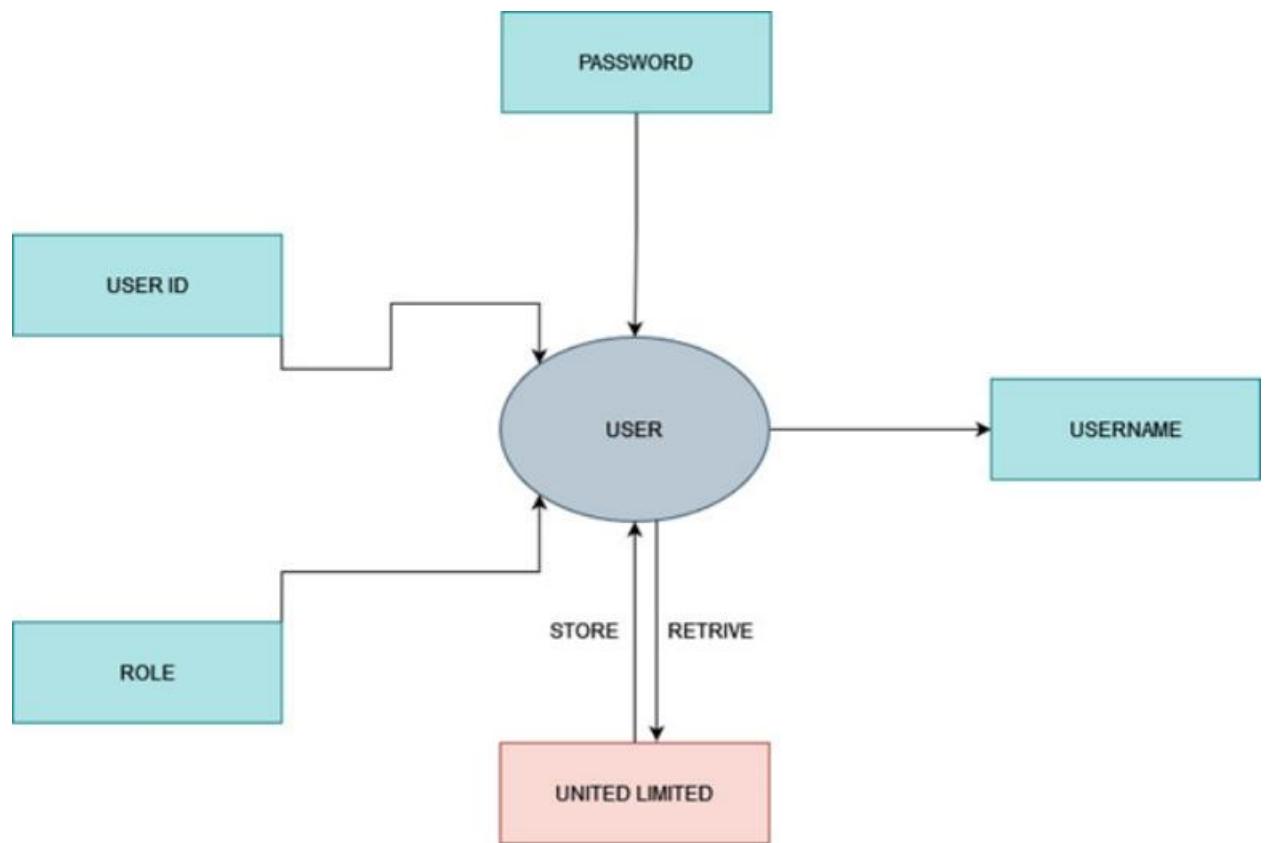
Click ok to restore the Database

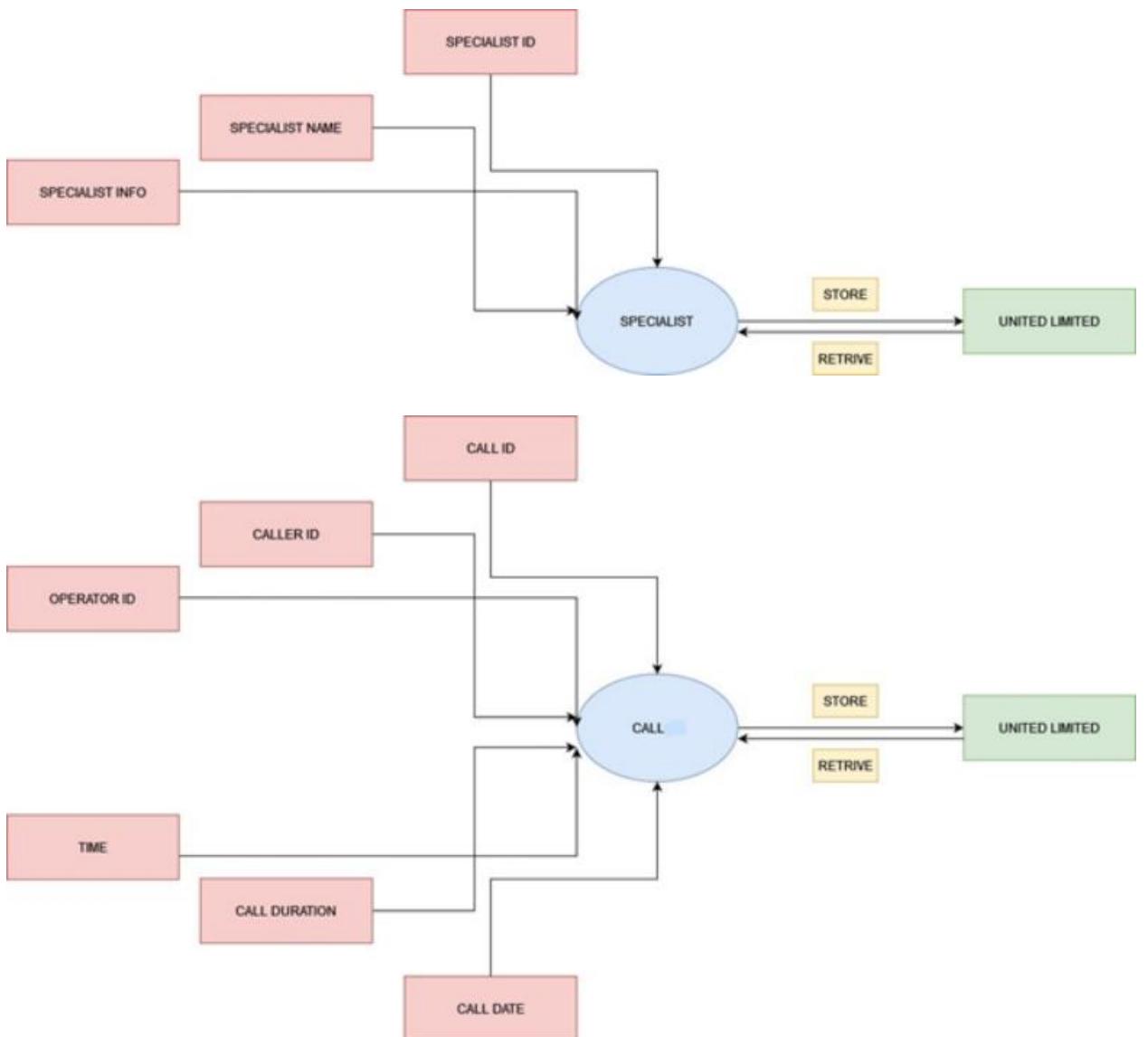


Flowcharts Diagram









References:

- Secoda.co. (2023). *A relational database is a type of database that stores and provides access to data points that are related to one another. Learn more here.* [online] Available at: <https://www.secoda.co/glossary/relational-database> [Accessed 14 Jul. 2023].
- Secoda.co. (2023). *A data dictionary is a central source of information about the data in your organization, business or enterprise. Learn more here.* [online] Available at: <https://www.secoda.co/glossary/data-dictionary> [Accessed 14 Jul. 2023].
- Guru99. (2023). *Entity Relationship (ER) Diagram Model with DBMS Example.* [online] Available at: <https://www.guru99.com/er-diagram-tutorial-dbms.html#4> [Accessed 14 Jul. 2023].
- Lucidchart. (2023). *What is an Entity Relationship Diagram (ERD)?* [online] Available at: <https://www.lucidchart.com/pages/er-diagrams> [Accessed 14 Jul. 2023].
- GeeksforGeeks. (2020). *Advantages and Disadvantages of Normalization.* [online] Available at: <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-normalization/> [Accessed 14 Jul. 2023].
- DatabaseTown. (2023). *Normalization in DBMS | Types of Normalization with Examples - DatabaseTown.* [online] Available at: <https://databasetown.com/types-of-normalization-in-DBMS-with-examples/> [Accessed 14 Jul. 2023].
- Sahiti Kappagantula (2019). *Microsoft SQL Server Certification Course.* [online] Edureka. Available at: <https://www.edureka.co/blog/sql-joins-types#FULL%20JOIN> [Accessed 15 Jul. 2023].
- TIBCO Software. (2022). *What is Data Validation?* [online] Available at: <https://www.tibco.com/reference-center/what-is-data-validation> [Accessed 15 Jul. 2023].
- Ali, T. (2018). *What is Data Sorting?* [online] Displayr. Available at: <https://www.displayr.com/what-is-data-sorting/> [Accessed 17 Jul. 2023].

Database Triggers: Examples & Overview Video (2023). *Database Triggers: Examples & Overview - Video & Lesson Transcript* | Study.com. [online] study.com. Available at: <https://study.com/academy/lesson/database-triggers-examples-overview.html> [Accessed 17 Jul. 2023].

Reqtest. (2020). *A Complete Guide To Database Testing With Examples*. [online] Available at: <https://reqtest.com/en/knowledgebase/database-testing/> [Accessed 18 Jul. 2023].

Dizdar, A. (2022). *Unit Testing: Definition, Examples, and Critical Best Practices*. [online] Bright Security. Available at: <https://brightsec.com/blog/unit-testing/> [Accessed 18 Jul. 2023].

Reqtest. (2019). Stress Testing Tutorial: Understanding the Basics. [online] Available at: <https://reqtest.com/en/knowledgebase/stress-testing-tutorial/> [Accessed 18 Jul. 2023].