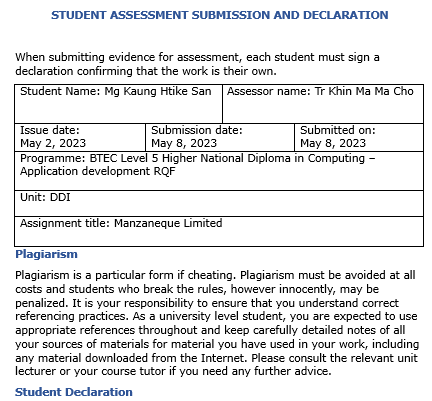
Logo

Description automatically generated with medium confidence



|  |
| --- |
| **Student declaration**  I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.  Student signature:  May 8, 2023 |

**Contents**

[LO1 Use an appropriate design tool to design a relational database system for a substantial problem 2](#_Toc134512889)

[LO2 Develop a fully-functional relational database system, based on an existing system design 10](#_Toc134512890)

[LO3 Test the system against user and system requirements 31](#_Toc134512891)

[LO4 Produce technical and user documentation. 37](#_Toc134512892)

# LO1 Use an appropriate design tool to design a relational database system for a substantial problem

**Design a relational database system with at least four related tables and clear statements of user and system needs using the right tools and techniques.**

**User requirements**

The user requirements specify what the user should perform. When a project's use cases are being discussed, user requirements are typically written. The customer or product managers who are familiar with how the user will utilize the embedded system help define the requirements. Many user requirements concern how a user will communicate with a system and what that user anticipates. User requirements are used to define, create, and verify a manufacturing system (i.e., the vendor design solution that is examined during the design review/qualification process). Depending to scenario, when someone has a trouble at work, they can call the helpdesk. One of the people who work at the help desk will try to answer the question, but if they can't do so right away, the problem is sent to one of several other specialists. Tracking helpdesk enquiries requires an information system. Analysts can then assess equipment performance, helpdesk resource allocation, and employee training needs.

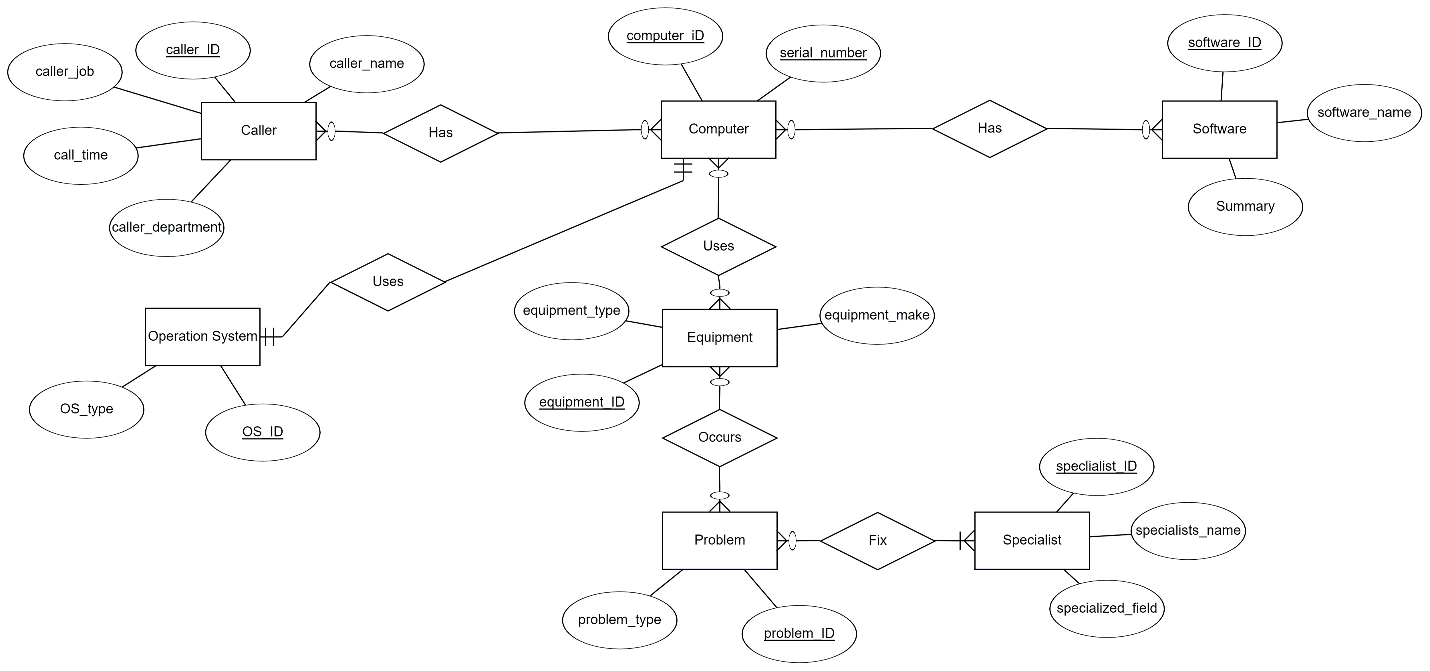
**System requirements**

System requirements define how the user will meet user needs while engaging with the system, as well as nonfunctional requirements such as "the system should handle 100,000 concurrent users." System requirements are statements that identify the capability that a system need in order to meet the needs of the client. Performance, modifiability, availability, scalability, reliability, and other quality standards are critical in system design. These 'ilities' are what we need to assess for a system and decide whether or not our system is appropriately constructed.

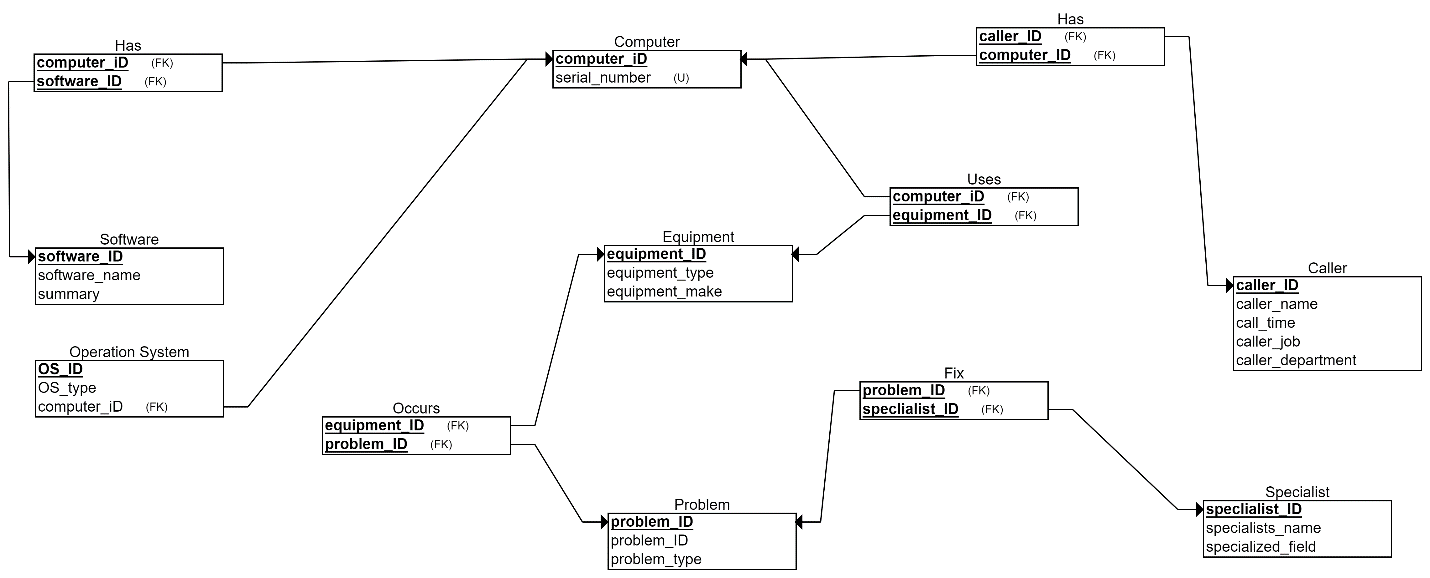
The following are some system requirements.

* When a new contact arrives at the helpdesk, the caller and helpdesk operator's names, the time of the call, the computer's serial number, and, if applicable, the operating system and software being used are recorded.
* The caller's identity will be compared to a personnel database to retrieve the caller's identification number, job title, and department.
* Each specialist will be an authority on one or more varieties of problem.
* The system will also list how many problems each specialist is presently working on so that, if there are multiple specialists for a given problem type, the least busy specialist can be assigned.

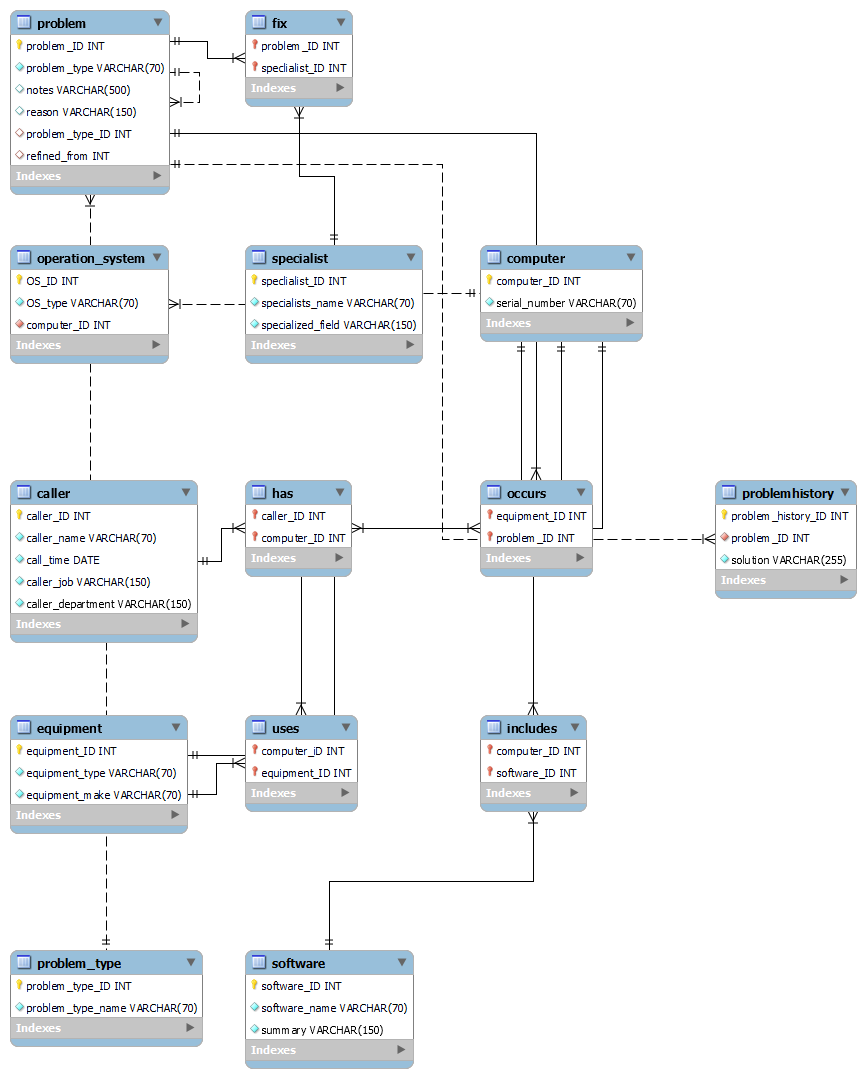
**Conceptual design**



**Logical design**



**Physical design**

****

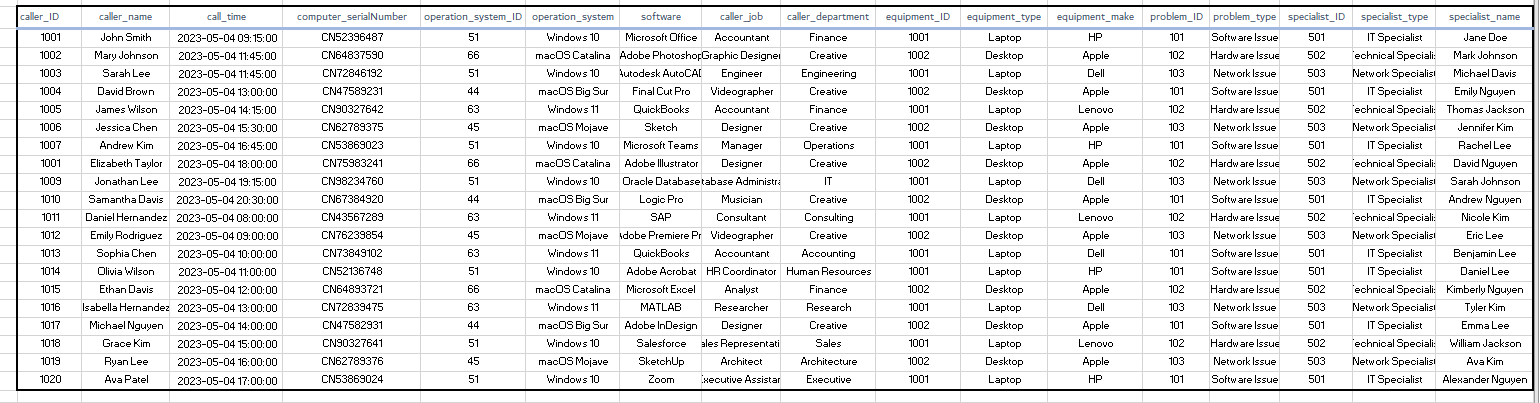
**Interred tables**

|  |  |
| --- | --- |
| **Table Name** | **Information** |
| caller | caller\_ID(PK)  caller\_name  caller\_time  caller\_job  caller\_department |
| specialist | specialist\_ID(PK)  specialists\_name  specialized\_field |
| computer | computer\_ID(PK)  serial\_number |
| equipment | equipment\_ID(PK)  equipment\_type  equipment\_make |
| operation\_system | OS\_ID(PK)  OS\_type  computer\_ID(FK) |
| problem | problem\_ID  problem\_type |
| software | software\_ID(PK)  software\_name  summary |

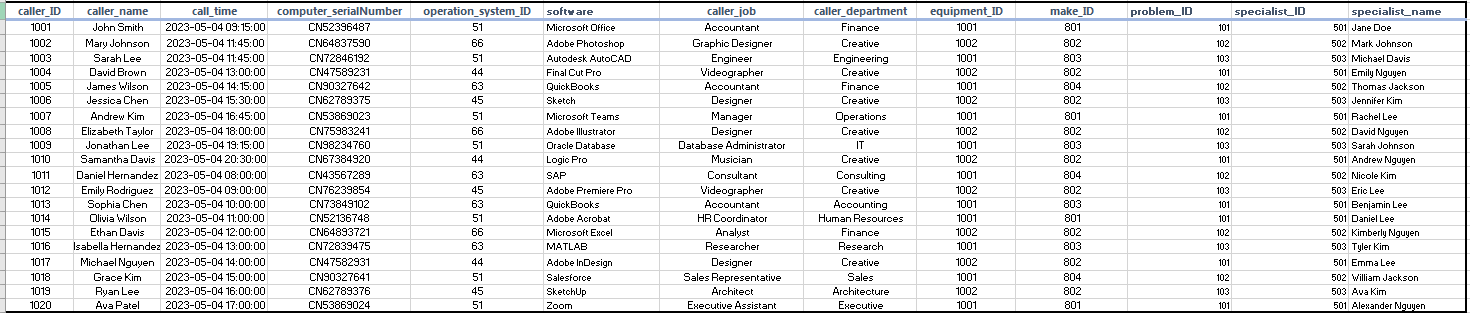
**Normalization**

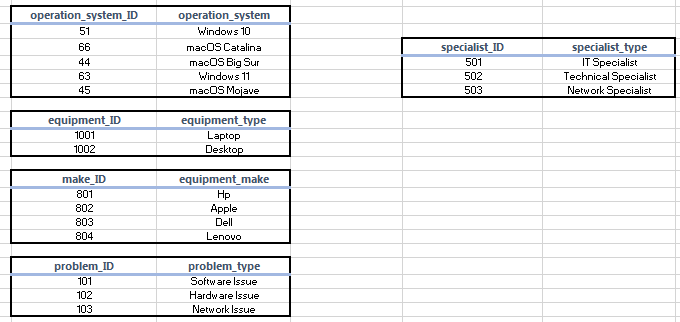
Data normalization is the process of putting data in a standard style that makes it easier to compare and analyze. It means putting the data in a uniform order, which makes it easier to understand and use in many different ways. Normalization can look different based on what data is being looked at and what the goals of the analysis are. In general, it means rescaling the values of the data so that they fall within a certain range or distribution. This can help get rid of outliers and lessen the effect of variables with bigger scales or magnitudes. Normalization is especially helpful in data mining, machine learning, and other data-driven applications where it's important to compare different sets of data in a fair and accurate way. It can also help make statistical studies more accurate and reduce mistakes in how data is interpreted.

**Unnormalized form**

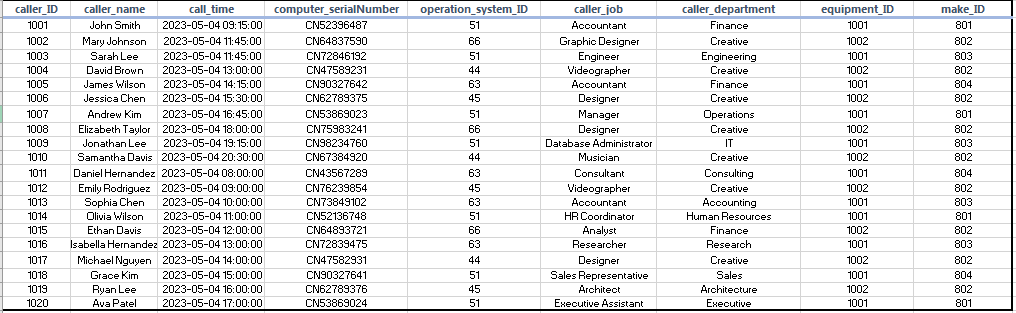


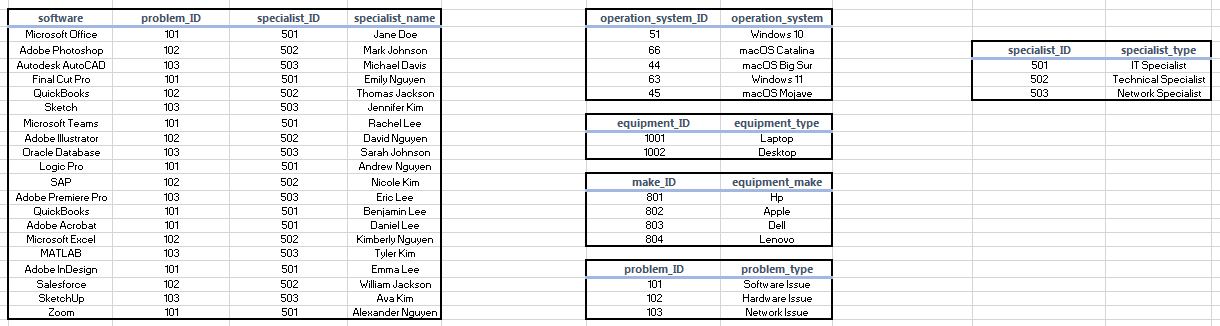
**1 NF**



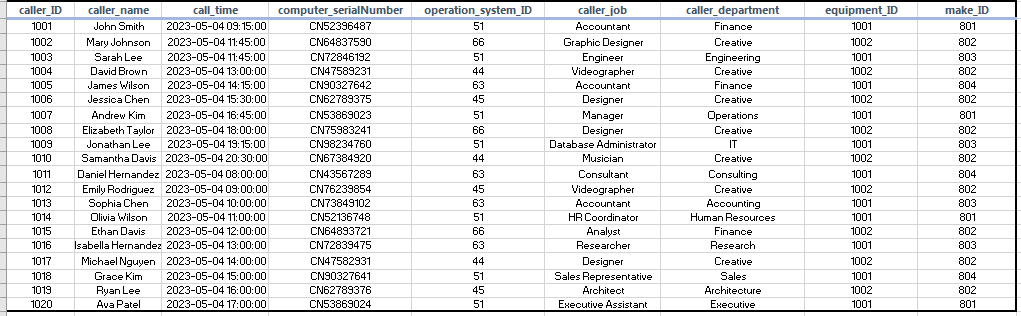


**2 NF**





**3 NF**



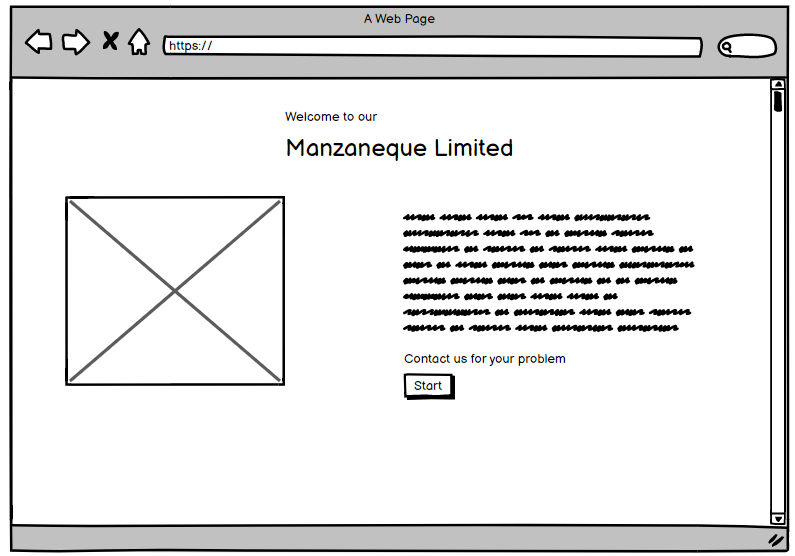


# LO2 Develop a fully-functional relational database system, based on an existing system design

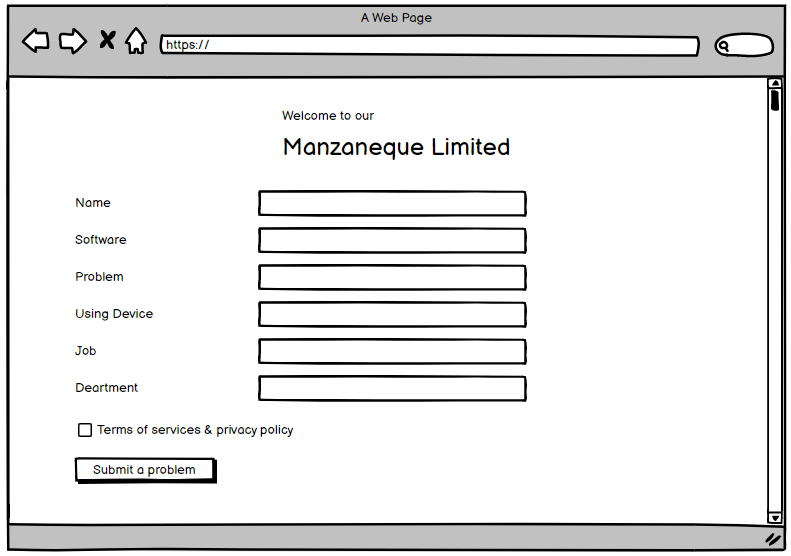
**Develop the database system with evidence of user interface, output and data validations, and querying across multiple tables.**

**User Interface design**

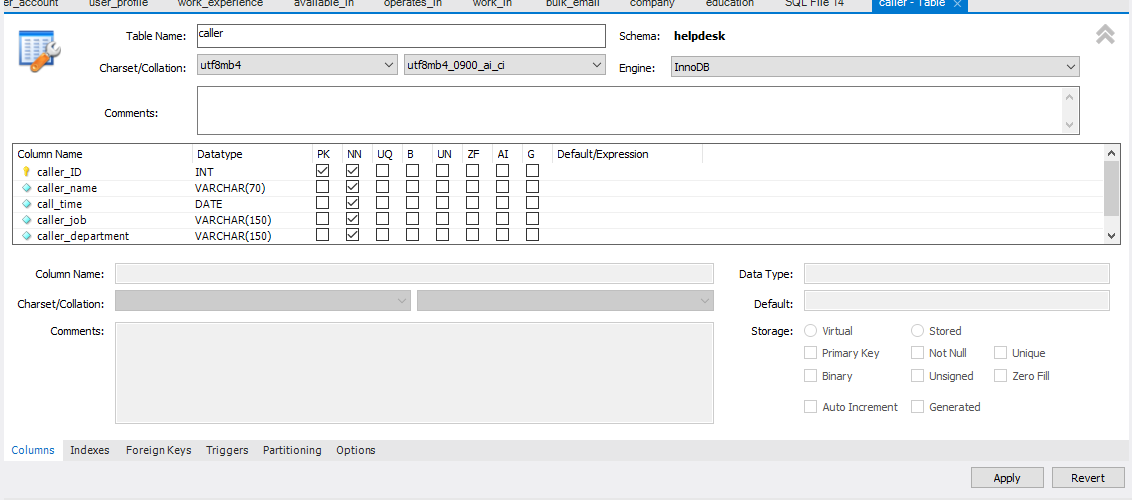
**1st wireframe**



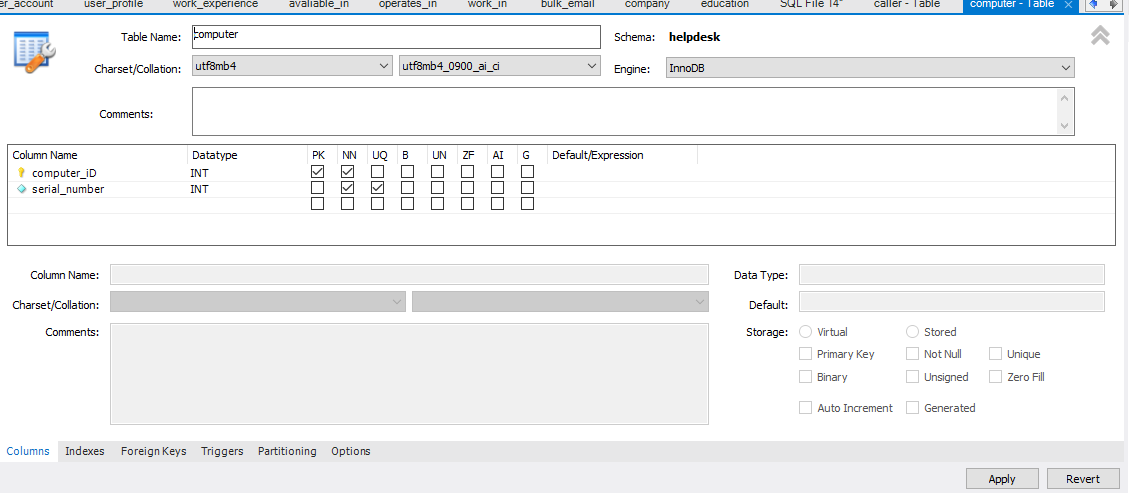
**2nd wireframe**



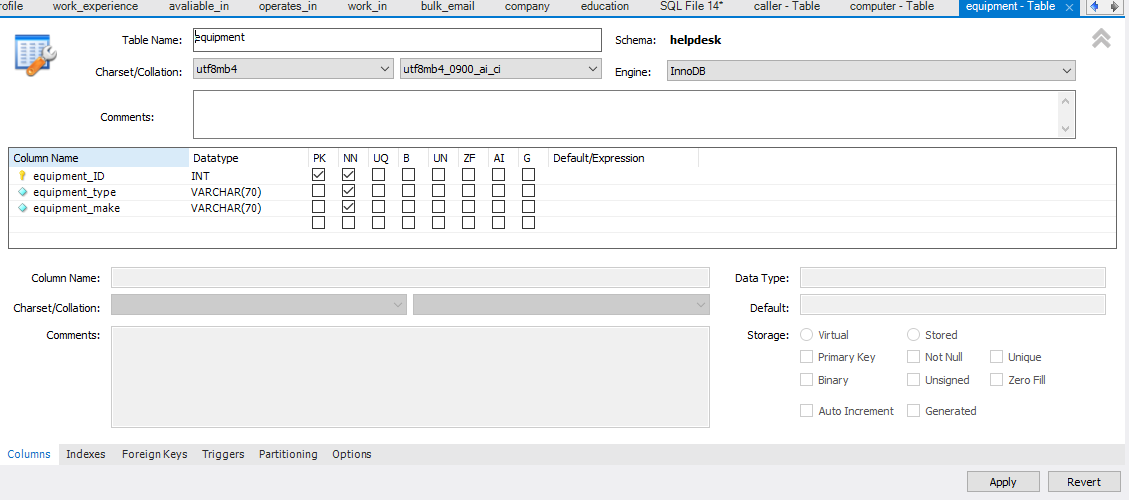
**caller**



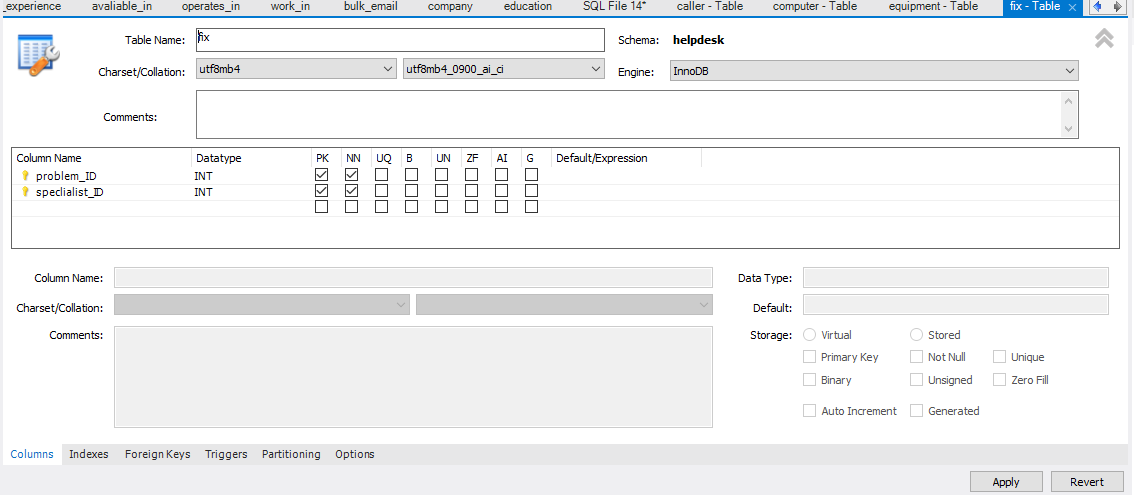
**computer**



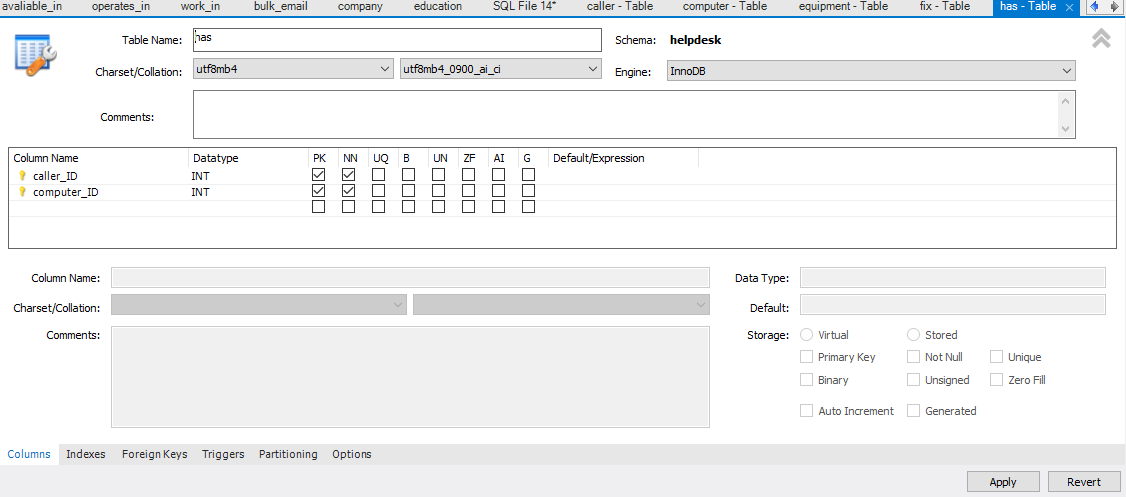
**equipment**



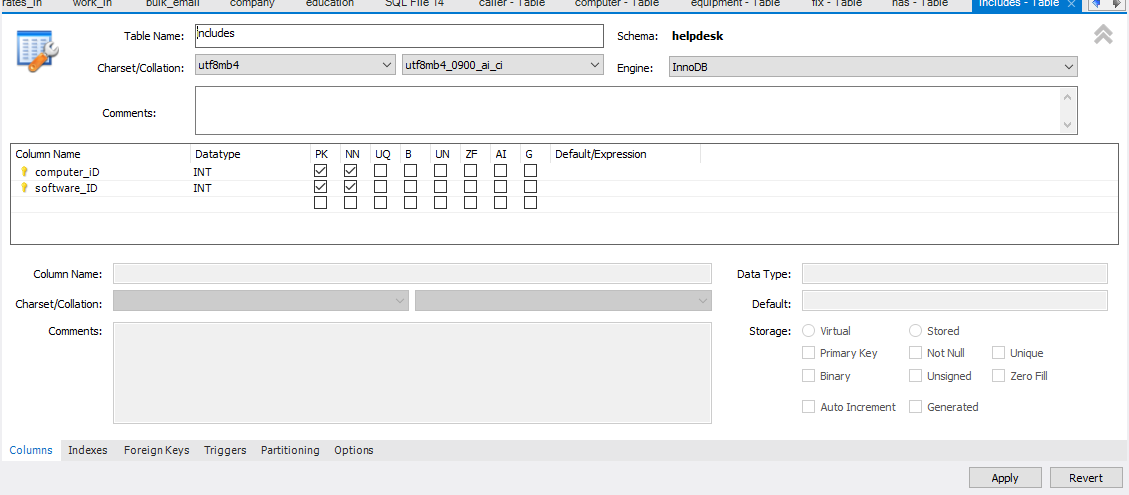
**fix**



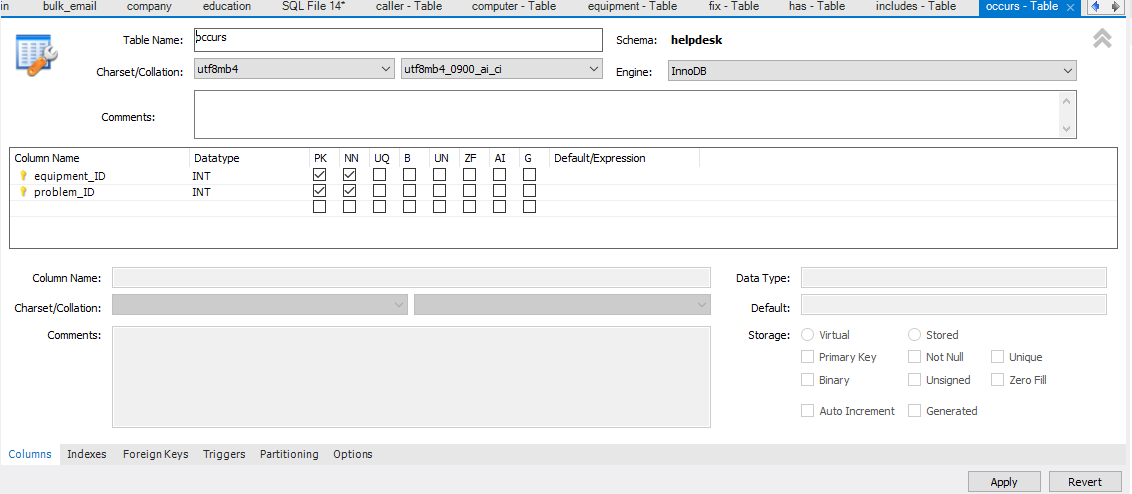
**has**



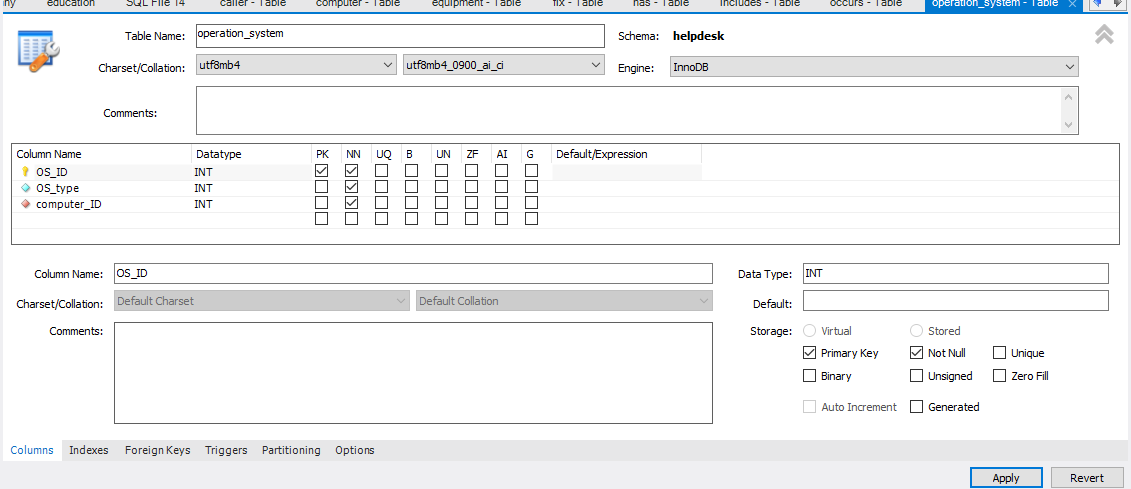
**includes**



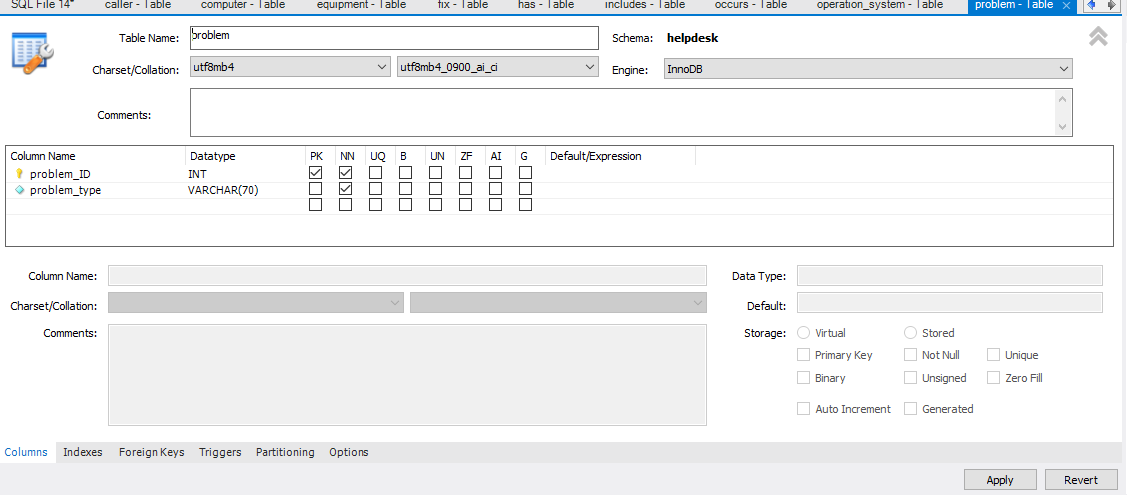
**occurs**



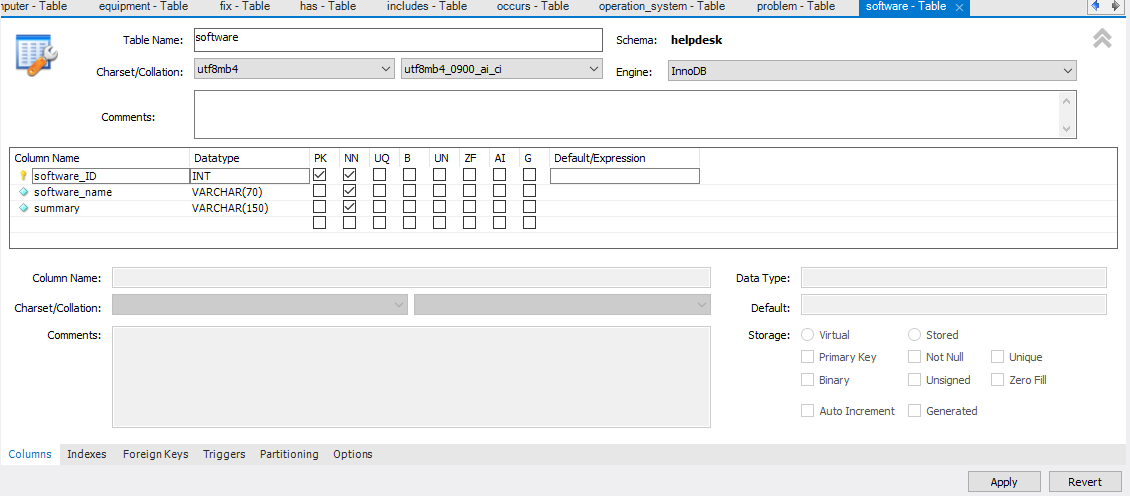
**operation\_system**



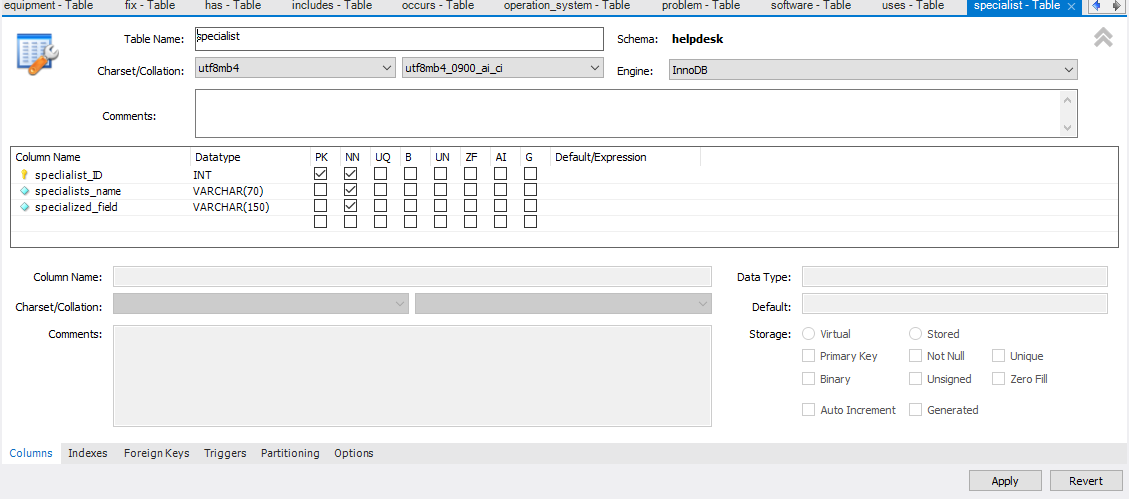
**problem**



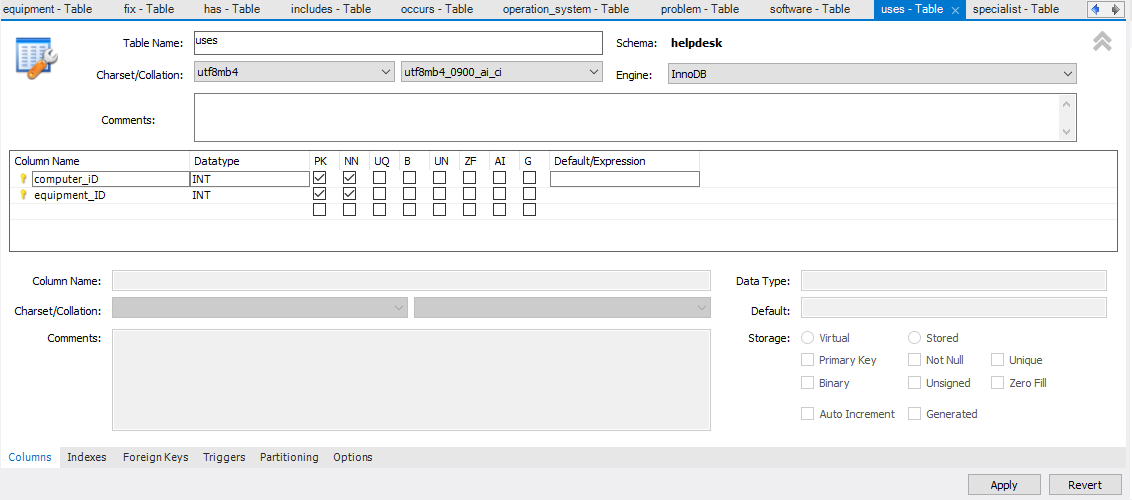
**software**



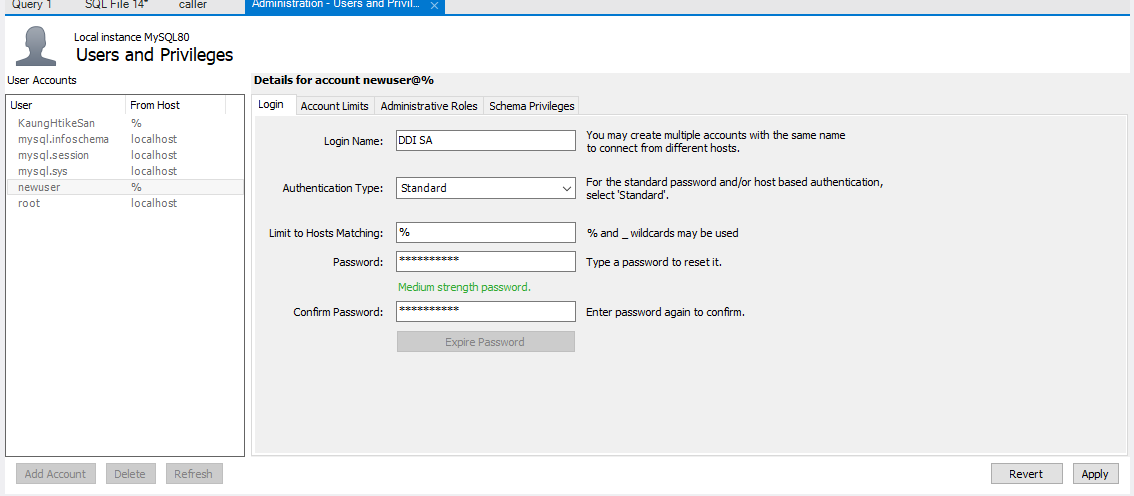
**specialists**

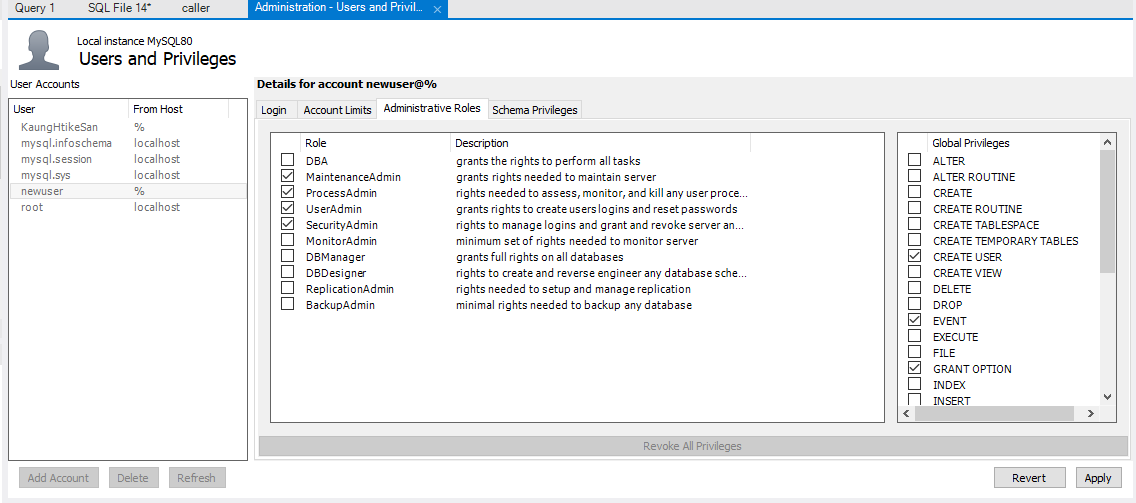


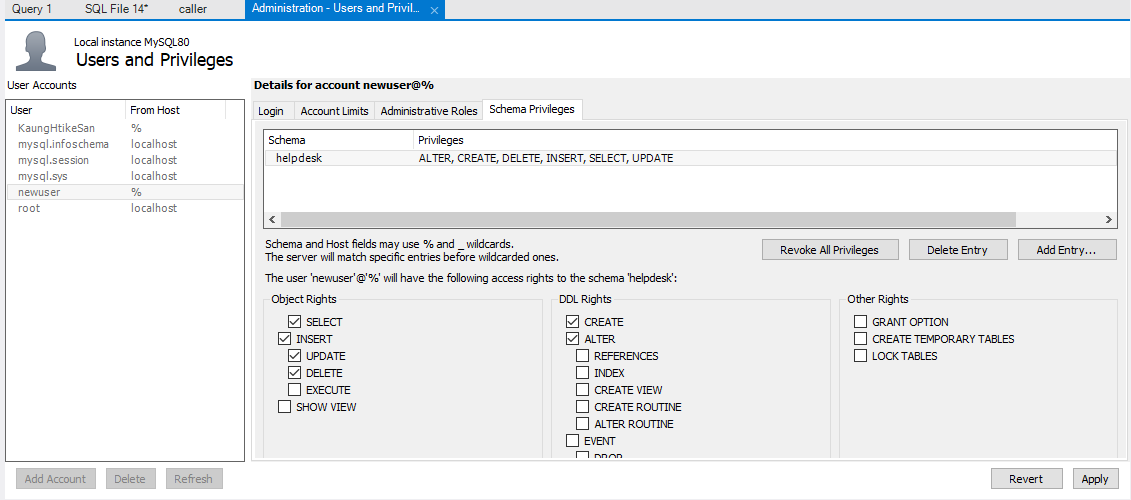
**uses**



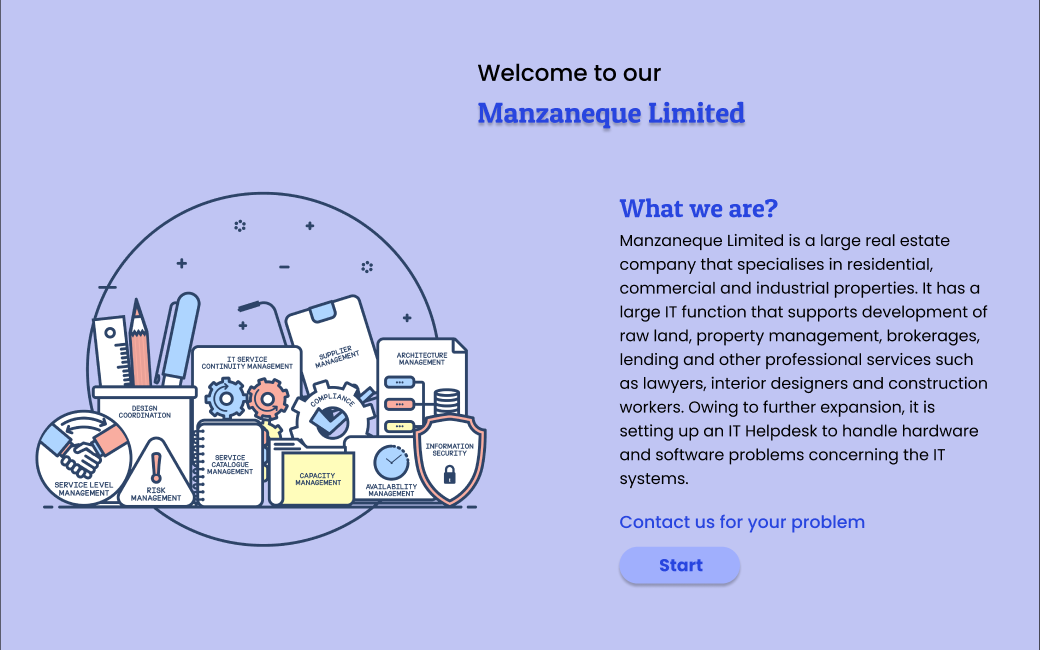
**Implement a fully-functional database system, which includes system security and database maintenance.**

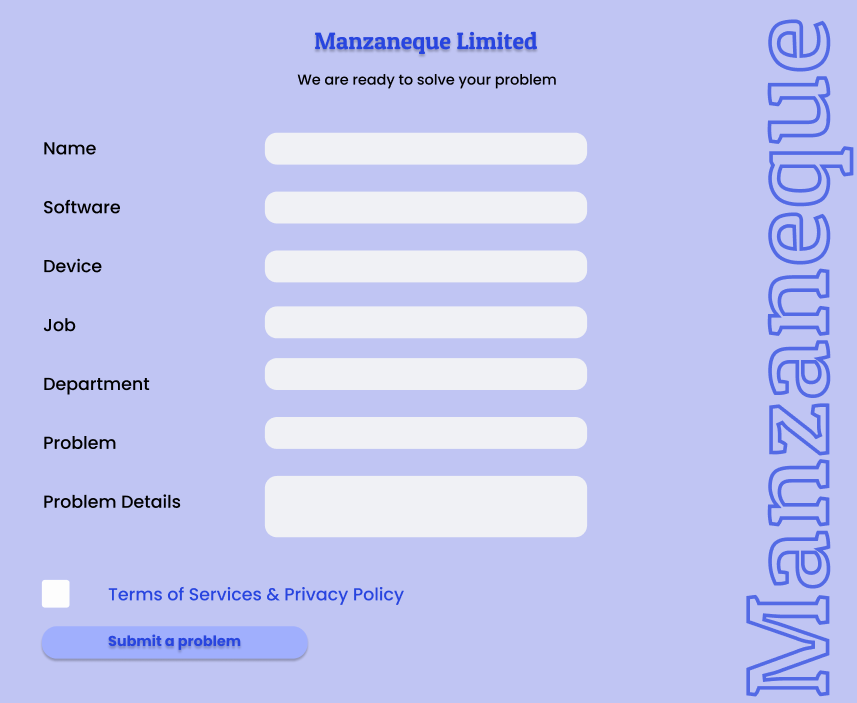






**Interface design**





**Data validation**

Data validation is the process of verifying the completeness, veracity, and structure of data prior to its use in a business process. The outcomes of a data validation operation can provide data for data analytics, business intelligence, and machine learning model training. Validating data at the point of entry aids in minimizing errors, decreasing the need for data purification, and preventing the entry of invalid data, thereby ensuring the database's reliability and consistency. In addition to preventing security breaches, data validation ensures that data is inputted correctly and the database is not susceptible to SQL injection attacks. In conclusion, data validation is necessary to assure the quality and dependability of database data, which is essential for making informed decisions.

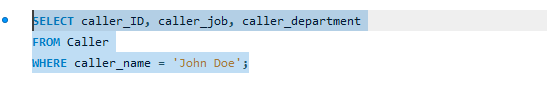
**Implement a query language into the relational database system.**

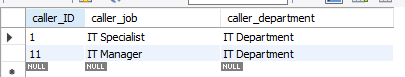
Implementing a query language for a relational database system necessitates the development of a software application that facilitates user interaction with the database through the submission of queries or requests for data retrieval, update, or deletion. SQL is the most common relational database query language. SQL lets users access, write, update, and remove relational database data. An intuitive interface for entering SQL queries, efficient processing of those commands, and clear presentation of the results are all necessary steps in implementing a query language. It is the responsibility of the software to alert the user to any issues that may arise, such as an invalid command or difficulties connecting to a database. Developing a query language is the process of making a software tool that lets users talk to a relational database system using an established syntax like SQL.

**According to the scenario, I did some like the following.**

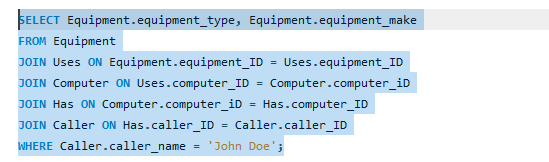
1. When a new contact is received by the helpdesk, the caller and helpdesk operator's names, the time of the call, the serial number of the computer, and, if applicable, the operating system and software being used are recorded.



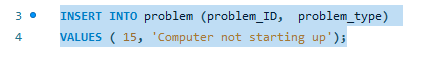
1. The caller's identity will be compared to a personnel database to retrieve the caller's identification number, job title, and department.



1. Their equipment will also be checked against a register of equipment to find the equipment type and make. Their software will be checked to see if it is under a valid license

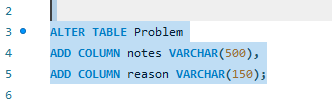


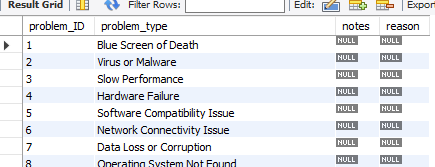


1. Each call is logged and each issue is assigned a problem number, which is given to the caller so that it can be referenced on subsequent conversations regarding the same issue.

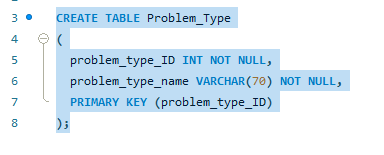


1. The person at the help desk will also make notes and write about the problem. Every call is always logged with a reason, even if it's just a note about how desperate the caller is getting (for example, if it's a follow-up call).

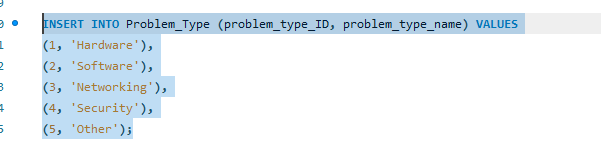


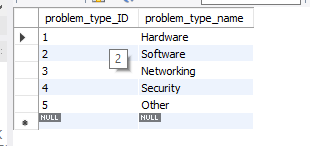


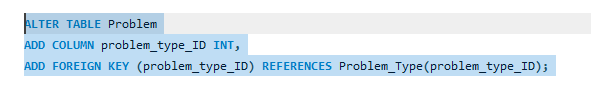
6. The helpdesk operator assigns an issue type from a list when a problem is reported. Operators know what problem type is most relevant and how particular it is.

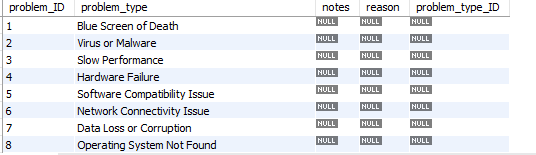




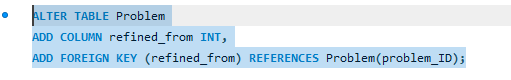


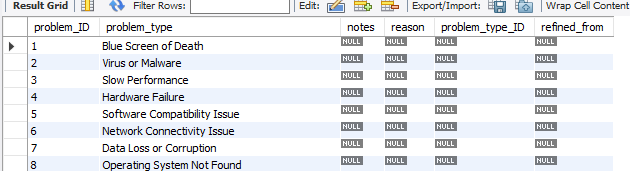




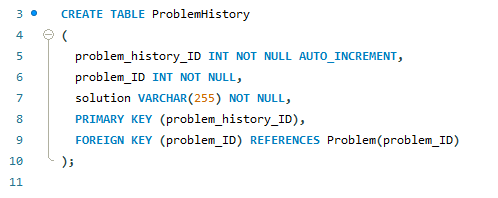


1. Some problem types are refinements of more generic problem types, therefore extra information may change the problem type allocation.





1. Once the problem area has been identified, the helpdesk operator can look up previous problems of the same type to determine if the issue has occurred previously and, if so, how it was resolved.





The above are some examples of inserting query according to scenario.

**Data extraction**

Data extraction is the process of getting different kinds of data from different sources, some of which may not be well organized or arranged at all. There are numerous data extraction strategies available to choose from. In the majority of instances, businesses will extract data in order to further process it or to transfer it to another repository, such as a data warehouse or a data lake, for the purposes of storage. The following are some of the procedures that can be done to guarantee that the data that is extracted is useful and pertinent to the requirements of the organization:

* Define clear objectives
* Identify the right data sources
* Select appropriate query tools
* Develop effective queries
* Validate and refine the results
* Present the data effectively

**Define clear objectives**

Setting clear goals means coming up with specific, measurable goals that a person or group wants to reach. It means figuring out what needs to be done, who should do it, when, and with what tools. Clear goals give people a feeling of direction, focus, and purpose and help make sure that everyone is working toward the same goal. According to scenario, **call\_log**, **computer**, **equipment, operation\_system**, **problem**, **software**, **specialist** are main objectives that I defined.

**Identifying the right data sources**

To find the right data sources, we have to find and choose the ones that are important to the problem we're trying to solve. This means figuring out what information is needed, where it can be found, and how to get to it. It's important to think about the data's quality, clarity, and completeness to make sure it can give useful insights and help make decisions.

**Select appropriate query tools**

When choosing the right query tools, you need to think about the type of data you need to examine, the complexity of your data, the amount of data you have, and your level of experience working with data. Data type & complexity, data size, my skill level, cost are the main methods how I select appropriate query tools.

**Develop effective queries**

Developing efficient queries is important in database management. Queries are used to get specific information from a database, and a badly designed query can lead to wrong or incomplete results, wasted time and effort, and even system errors**.** Creating successful queries means making clear search requests to get the information you need from a database or other source of data I added queries to each of the newly generated tables that I had built.

**Validate and refine the results**

Validating and refining the results in a database means making sure that the query results are correct and make sense, and improving them so that they meet certain standards. We may guarantee that the data you obtain from a database query meets all of our requirements for precision, applicability, and utility by validating and refining the results of the query. We will be able to make judgments that are more informed as a result of this, as well as improve the efficiency of our database system as a whole.

**Present data effectively**

Presenting information in a way that is clear, concise, and easy for the audience to understand is what it means to present data successfully. It means showing information in a way that is easy to understand with the help of charts, graphs, and tables. How well data is presented rests on things like the audience, the goal of the presentation, and the type of data being presented.

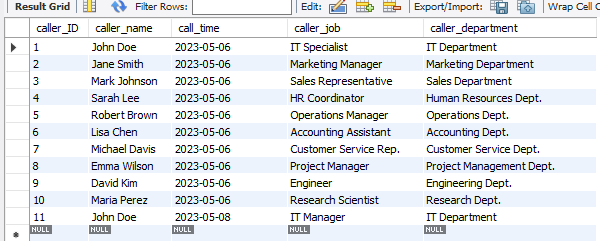
These above are extracted through the use of query tools to produce appropriate management information.

# **LO3 Test the system against user and system requirements**

**Test the system against user and system requirements.**

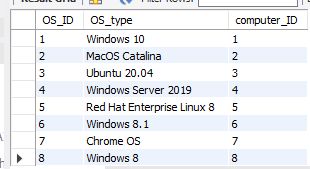
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Test type** | **Test case** | **Test description** | **Expected result** | **Actual result** | **Pass/fail** | **Figure** |
| **1** | Unit Testing | Inserting a new user | To make sure that a new person or employee can be added to the system | It is necessary to add a new user or employee to the table. | A subscriber or employee who has been added to the table. | Pass | Fig 1 |
| **2** | Unit testing | Updating the data in the table | To check if new information can be added to the table | The table should be updated with new data. | The table is updated with new information. | Pass | Fig 2 |
| **3** | Authentication testing | Login to the database with incorrect password | Try to sign in with the wrong password | The database should deny the user access. | The database prevents the user from logging in. | Pass | Fig 3 |
| **4** | Unit testing | Deleting the data | To make sure that the table's data can be erased | The records must be removed from the table. | The information was removed from the table. | Pass | Fig 4 |
| **5** | Structural testing | Data validation | Try to put in wrong information. | The database should reject legitimate data. | The database does not allow invalid data. | Pass | Fig 5 |
| **6** | Structural testing | Data range | Try to add more info than it can hold. | The database should not allow data that exceeds its range. | Database did not recognize data that exceeded its range. | Pass | Fig 6 |

**Fig 1**

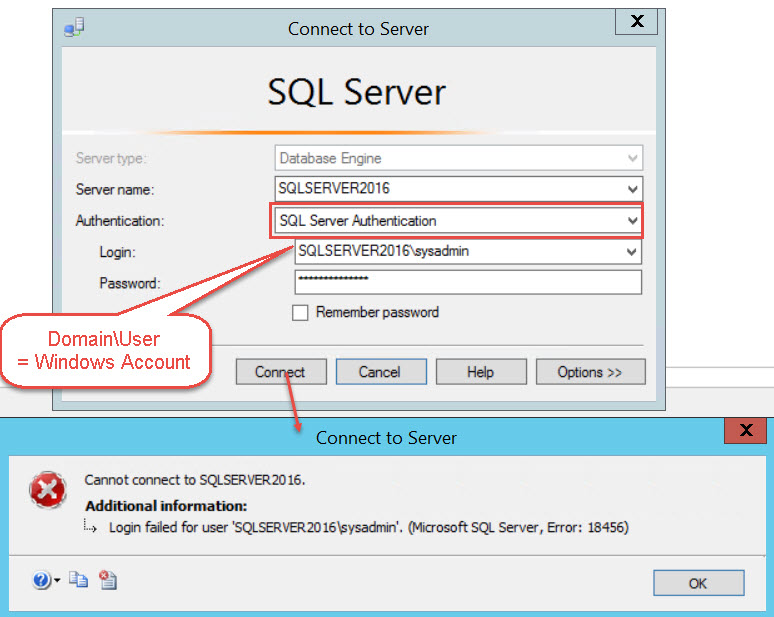


**Fig 2**



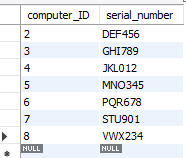


**Fig 3**

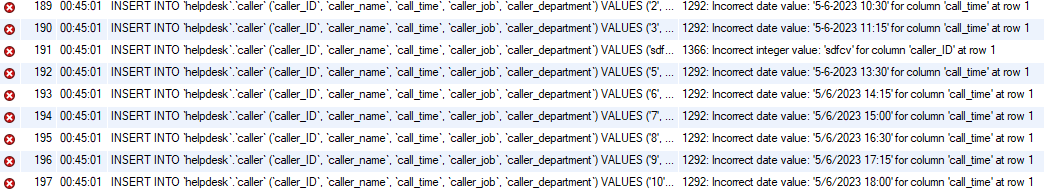


**Fig 4**

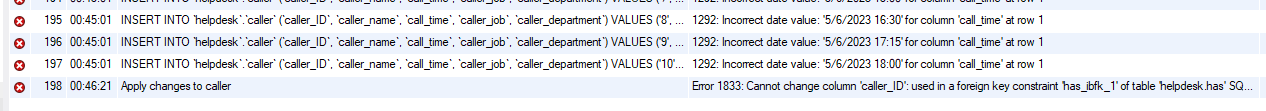




**Fig 5**



**Fig 6**



**Unit testing**

Unit testing is important in database development because it enables developers to test individual units of code and identify any errors or defects before proceeding to larger, more complex portions of the database. In addition, it ensures that each unit of code functions as intended and satisfies the design-phase requirements. By evaluating individual units of code, developers can identify and resolve problems early on, saving time and resources in the long run. It also prevents bugs from becoming more difficult to resolve later in the development process, when multiple code modules are combined. Unit testing ensures that database changes do not degrade extant functionality.

**Structural testing**

In database construction, structural testing is essential because it ensures the correctness and dependability of the database design and implementation. It involves testing the database's internal structures, such as tables, indexes, relationships, and constraints, to ensure that they are correctly implemented and perform as anticipated. By performing structural testing, developers can detect errors and defects early in the development process and validate the integrity of the database schema and logic. This can aid in preventing data corruption, data loss, performance issues, and security vulnerabilities. In addition, structural testing ensures that the database is scalable and capable of handling the expected volume of data and queries.

**Authentication testing**

In database creation, it is essential to conduct authentication testing to ensure that the security measures implemented to protect the database are operating correctly. This type of testing verifies that the authentication mechanisms, such as username and password combinations, access control lists, and encryption protocols, function as intended and provide the necessary security. The database could be susceptible to unauthorized access, data theft, and other security lapses in the absence of authentication testing. By performing authentication testing, any vulnerabilities or deficiencies in the authentication mechanisms can be identified and addressed prior to database deployment, thereby aiding in the protection of the data's confidentiality, integrity, and availability.

# LO4 Produce technical and user documentation.

**Producing technical and user documentation.**

**Technical documentation**

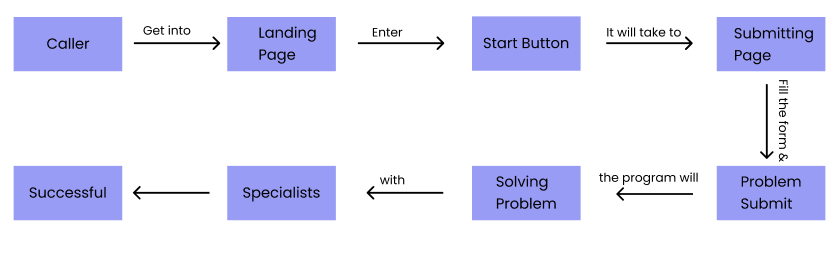
|  |  |
| --- | --- |
| **Product** | **Process** |
| SQL Server | SQL Server operates by default as a Windows service named SQL Server (MSSQLSERVER). Additional SQL Server services, including SQL Server Agent, SQL Server Browser, and SQL Server Integration Services, can be installed and executed as Windows services. |
| Processor | The operating system manages the processor-related services, which typically do not require manual intervention. However, there are performance-related services that can be modified to optimize the system's processor utilization. |
| RAM | I have 8 GB of RAM available to operate a MySQL database efficiently. |
| CPU | A multi-core CPU with a minimum clock speed of 2GHz should be adequate for the majority of minor to medium-sized databases. To ensure optimal performance, a more potent CPU may be necessary for larger databases and environments with significant traffic. In addition, it is essential to ensure that the CPU is compatible with the version of MySQL being used and that the server has sufficient ventilation to prevent overheating. |

**User documentation**

User documentation in a database is the documentation that is produced for the database's end users. It typically contains information on how to use the database, the database's purpose and scope, data entry guidelines, data retrieval procedures, and troubleshooting advice.

For an end-user, if I go to the website, I see landing page. In landing page, some description about company and services are described. There is a button below it. By clicking that button, it will take the user to the problem submitting page. In that page, the user needs to fill his name, problemed software, his current using device, his job, department, problem and problem details. The user also needs to agree the terms of services and privacy policy. After all, by clicking submit a problem, the program will be solved the problem. The user data is also privately stored and secured.

**Data flow diagram**



**Flowchart**

