

|  |  |
| --- | --- |
| Reg Number | R231567R |
| Full Name | **CHAWATAMA TADIWANASHE T** |
| Program | **HCS (COMPUTER SCIENCE)** |
| SDO | **MS MLANGA** |

**Main Steps in Database Design**

1. **Determine the purpose of your database:** The first step is to clearly define the purpose and goals of the database, for example when designing a database for a library the purpose would be to keep track of books, their authors and people who have used them. So, this step will involve understanding and documenting, the requirements of the database, what kind of data will be stored, users who will access, and what functionality the database should provide. This step helps to establish the overall direction and scope of the database design and to capture all the necessary data elements and their relationships.
2. **Analyse the data:** This involves analysing the data that is looking at the data in detail, identifying errors of duplication or inconsistency and determining the entities (e.g., customers, books, orders) and their attributes (e.g., name, address, author). Entities like "Book," "Customer," and "Order" are identified, along with their attributes like book title, author, customer name, etc. Relationships between entities, such as "a customer can place multiple orders," are also defined. Once data is analysed it is then used to create a data model.
3. **Data Modelling:** a data model is a visual representation of the structure of a database. Data modelling also involves create Entity Relationship diagrams, which shows relationships between different entities. Entities, attributes and relationships in the data are identified, and then mapped to tables and relationships. For example, in our library database, we would create a model for each entity that is books, authors and borrowers. For each entity we will the define attributes that is for the book we will set bookID, name, author and so on This process is very important as it makes it easier to understand and use data.
4. **Design database Schema:** schemas are set of rules that define the structure of data in databases, they are defined in data definition language. Designing schema includes defining unique keys for tables and relationships between the tables. In our library example the schema would include books table, authors table, borrowers table and so on. The relationships between the entities will be defined such as how books are related to authors or borrowers. They are very important as they promote consistency and accuracy to the data, and makes it easier to query the data, for example we could set constraints that will restrict entry of books into the database if the value for the author is null.
5. **Create the database:** firstly, a database management system is chosen, such as MySQL, mongo and so on. The we create the database using those query languages, and also create tables inside our database for example, we will first create a library database then create books table, authors table and borrower table, inside the library database and apply normalisation. Once the database is created it is the populated with data.
6. **Test and validate the database:** The goal of this step is to ensure the correctness and efficiency of the database. Testing involves verifying that the database functions correctly, checking to see if It meets the requirements specified in the earlier steps and Optimization which involves analysing the performance of the database and making any necessary adjustments to improve its efficiency. For example, we would create test data and insert it our tables, then run queries against the database to make sure that data is being retrieved correctly. The goal is to ensure that the database performs well under different scenarios and can handle the expected workload.
7. **Maintenance and evolution:** The final goal is to ensure the ongoing maintenance and evolution of the database. This involves monitoring the database's performance, making necessary modifications based on changing requirements, and ensuring data integrity and security over time.

**QUESTION (B)**

b. What is UML? How does database design fit into the overall design of data intensive software?

System. [10]

UML (Unified Modelling Language) is a standardized language used in software engineering to design, visualize, and document software systems. UML provides a set of graphical notations and diagrams that show the structure, behaviour, and interactions of a software system.

Database design plays a crucial role in data-intensive software systems. These are systems that often involve storing, managing and processing large amounts of data in order to provide their services, and the database design directly impacts these systems because the way data is stored and organised can have a big impact on the overall system’s performance. Here's how database design fits into the overall design of such systems:

* **Data modelling:** IT is the first step in database design, during which the needs and guidelines for the software system are converted into a conceptual data model. The limitations and relationships between the different data elements are represented by this model. It serves as the basis for the database schema and aids in comprehending the data requirements.
* **Schema Design:** The database schema needs to be designed after the data model has been established. The database's structure, including its tables, columns, relationships, keys, and constraints, is specified by the schema. It establishes the structure and storage of the data in the database. A well-designed schema supports the necessary functions of the software system and guarantees data integrity and effective retrieval.
* **Performance Optimization:** Performance optimization is taken into account when designing databases. It entails creating indexes, refining queries, and taking denormalization and data normalization strategies into account. Large amounts of data can be handled by data-intensive software with faster response times by optimizing query performance and creating an efficient database schema. Indexing the data in a database can greatly improve the performance of the system because indexes make if faster to retrieve data from the database, Normalising the data can improve the system performance by reducing the amount of duplicate data and making it easier to update and delete data. Partitioning the data into smaller chunks can improve the systems performance by allowing the database to access only the necessary data.
* **Data Integrity and Consistency:** The software system's database design guarantees the integrity and consistency of the data. It guarantees that the data kept in the database stays correct and dependable by enforcing constraints including primary key constraints, unique constraints, and referential integrity constraints. This is essential for preserving the integrity of the data and avoiding corruption or discrepancies.
* **Scalability and extensibility:** These two factors are taken into account when designing databases. The software system can manage growing data quantities and user loads without compromising speed thanks to a well-designed database schema. It offers the adaptability to take into account upcoming adjustments and additions to the data requirements of the system.
* **Integration with Application Logic:** The software system's application logic and database design are tightly intertwined. It outlines the queries, transactions, and data manipulation procedures that the software does to access and modify data. In addition to guaranteeing effective communication between the application code and the database, the design should be in line with the functional needs of the software.