Assignment 1

1. What is a relational database management system (RDBMS)? What are the advantages of a database management system over a file system?

Ans. A relational database is a type of database that stores and provides access to data points that are related to one another. Relational [databases](https://www.oracle.com/in/database/what-is-database/) are based on the relational model, an intuitive, straightforward way of representing data in tables. In a relational database, each row in the table is a record with a unique ID called the key. The columns of the table hold attributes of the data, and each record usually has a value for each attribute, making it easy to establish the relationships among data points.

Advantages of DBMS over File system:

Data redundancy and inconsistency –

Redundancy is the concept of repetition of data i.e. each data may have more than a single copy. The file system cannot control the redundancy of data as each user defines and maintains the needed files for a specific application to run. There may be a possibility that two users are maintaining the same files data for different applications. Hence changes made by one user do not reflect in files used by second users, which leads to inconsistency of data. Whereas DBMS controls redundancy by maintaining a single repository of data that is defined once and is accessed by many users. As there is no or less redundancy, data remains consistent.

Data sharing –

The file system does not allow sharing of data or sharing is too complex. Whereas in DBMS, data can be shared easily due to a centralized system.

Data concurrency –

Concurrent access to data means more than one user is accessing the same data at the same time. Anomalies occur when changes made by one user get lost because of changes made by another user. The file system does not provide any procedure to stop anomalies. Whereas DBMS provides a locking system to stop anomalies to occur.

Data searching –

For every search operation performed on the file system, a different application program has to be written. While DBMS provides inbuilt searching operations. The user only has to write a small query to retrieve data from the database.

Data integrity –

There may be cases when some constraints need to be applied to the data before inserting it in the database. The file system does not provide any procedure to check these constraints automatically. Whereas DBMS maintains data integrity by enforcing user-defined constraints on data by itself.

System crashing –

In some cases, systems might have crashed due to various reasons. It is a bane in the case of file systems because once the system crashes, there will be no recovery of the data that’s been lost. A DBMS will have the recovery manager which retrieves the data making it another advantage over file systems.

Data security –

A file system provides a password mechanism to protect the database but how long can the password be protected? No one can guarantee that. This doesn’t happen in the case of DBMS. DBMS has specialized features that help provide shielding to its data.

1. In a database management system, explain the ACID properties.

Ans. A transaction is a single logical unit of work that accesses and possibly modifies the contents of a database. Transactions access data using read and write operations. In order to maintain consistency in a database, before and after the transaction, certain properties are followed. These are called ACID properties.

**Atomicity**

By this, we mean that either the entire transaction takes place at once or doesn’t happen at all. There is no midway i.e. transactions do not occur partially. Each transaction is considered as one unit and either runs to completion or is not executed at all. It involves the following two operations. Atomicity is also known as the ‘All or nothing rule’.

—Abort: If a transaction aborts, changes made to the database are not visible.

—Commit: If a transaction commits, changes made are visible.

**Consistency**

This means that integrity constraints must be maintained so that the database is consistent before and after the transaction. It refers to the correctness of a database.

**Isolation**

This property ensures that multiple transactions can occur concurrently without leading to the inconsistency of the database state. Transactions occur independently without interference. Changes occurring in a particular transaction will not be visible to any other transaction until that particular change in that transaction is written to memory or has been committed. This property ensures that the execution of transactions concurrently will result in a state that is equivalent to a state achieved these were executed serially in some order.

**Durability**

This property ensures that once the transaction has completed execution, the updates and modifications to the database are stored in and written to disk and they persist even if a system failure occurs. These updates now become permanent and are stored in non-volatile memory. The effects of the transaction, thus, are never lost.

1. Explain the concept of normalization.

Ans. It is the process of organizing the data in a database. It helps in removing the duplicate values in the database. Normalization divides the large table into smaller tables and links them using relationships.

The normal form is used to reduce redundancy from the database table. Normalization is the name given to the process of simplifying the relationship among data elements in a record.

In simple words we can say,

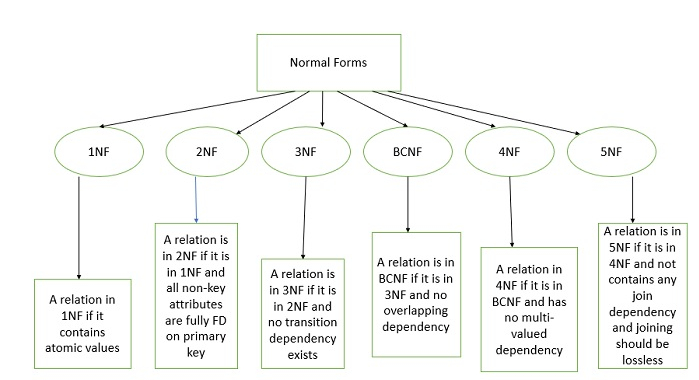
Normalization is the process of organizing data to minimize.

* Redundancy/duplication/repetition.
* Insertion, deletion, updating anomalies.

**Normal forms**

There are six Normal forms which are as follows −

* First Normal Form (1NF)
* Second Normal Form (2NF)
* Third Normal Form (3NF)
* Boyce-Codd Normal Form (BCNF)
* Fourth Normal Form (4NF)
* Fifth Normal Form (5NF)



## Objective of Normalization

The main objectives of using the normalization technique are as follows −

* It provides a formal framework for analyzing the relations based on the key attributes and their functional dependencies.
* Freeing the relations from insertion, update and delete anomalies.
* Reducing the need of restructuring the tables.

While decomposing, the normalization process should ensure the following two properties are satisfied −

* **Lossless join or nonadditive property** − It guarantees that the spurious tuples are not generated with respect to the relation schemas created after decomposition.
* **Dependency preservation property** − It ensures that every functional dependency is represented in some of the individual relations resulting after decomposition.
* **Denormalization** − It is the process of storing the join of higher normal form relations as a base relation- which is in a lower normal form.

1. Explain the many types of query languages used in relational databases. DQL, DML, DCL, and DDL are some examples.

Ans. Data Query Language (DQL) is used to fetch the data from the database. It uses only one command:

### SELECT:

This command helps you to select the attribute based on the condition described by the WHERE clause.

Data Definition Language helps you to define the database structure or schema. Let’s learn about DDL commands with syntax.

### CREATE

CREATE statements is used to define the database structure schema.

### DROP

Drops commands remove tables and databases from RDBMS.

### ALTER

Alters command allows you to alter the structure of the database.

### TRUNCATE:

This command is used to delete all the rows from the table and free the space containing the table.

Data Manipulation Language (DML) allows you to modify the database instance by inserting, modifying, and deleting its data. It is responsible for performing all types of data modification in a database.

There are three basic constructs which allow database program and user to enter data and information are:

Here are some important DML commands in SQL:

* INSERT
* UPDATE
* DELETE

DCL (Data Control Language) includes commands like GRANT and REVOKE, which are useful to give “rights & permissions.” Other permission controls parameters of the database system.

### Examples of DCL commands:

Commands that come under DCL:

* Grant
* Revoke

1. What is the difference between the main key and a composite key? Give instances of how primary key and composite are used.

## Ans. Primary Key

First, a primary key uniquely identifies each record in a database table. Any individual key that does this can be called a candidate key, but only one can be chosen by database engineers as a primary key.

Composite Key

Next, there's the composite key, which is composed of two or more attributes that collectively uniquely identify each record.

An example would be a list of homes on a real estate market. In a well-ordered database, there should be a primary key that uniquely identifies each record.

How this works may have to do with the sophistication of the database.

In some cases, the homes may only be uniquely identified by a mortgage number — all other data (towns, streets, house numbers) is not unique to each record. The mortgage number would be the primary key. Suppose, however, that an MLS realtor’s listing technology assigns its own unique numbers to the records in the table.

Then, there will be two keys that developers might identify as “candidate keys”:

* The mortgage numbers.
* The MLS numbers.

One of them will qualify as the “primary key” in what some would consider an arbitrary way.

A composite key, then, would be the combination of two keys.

For example: the combination of house number and street might qualify as a composite key, given that the market listings are local. If so, then when someone searches using both the house number and the street, they should only get one single record returned.

Foreign Key

Meanwhile, if there is a key in a linked table, such as a buyer’s table that references the primary key, that will be a foreign key.

While a primary key and a composite key might do the same things, the primary key will consist of one column, where the composite key will consist of two or more columns.

The relationship between a primary key and a foreign key is quite different. The key thing to understand here is that the primary key in one database table becomes a foreign key in another database table, and vice versa.

A foreign key in a database table is taken from some other table and applied in order to link database records back to that foreign table.

The foreign key in the database table where it resides is actually the primary key of the other table.

Here's an example to make this clearer: If we have a database table labeled, say, "orders," and we have a foreign key that's labeled "customers," we could perhaps link that foreign key to a primary key, identifying each customer in a separate table.

In sum, all of this interconnection of key categories shows the detailed nature of creating sophisticated patterns and relationships in relational database systems.

The database engineer knows these categories and attributes intimately, and applies them in order to help ensure that the SQL queries and other data retrieval methods triggered by end users are effective.

Along with database normalization and promoting structured data, these key relationships are core parts of database system integrity and data storage design best practices.

1. Create a table with a primary key, a column default value, and a column unique constraint in SQL.

Ans. create table ORDERS

(

orderid int not null,

customerID int,

employeeid int,

orderdate datetime,

shipperid int,

PRIMARY KEY(orderid)

)

INSERT into ORDERS values(901,1,1,sysdate(),92801);

INSERT into ORDERS values(902,2,2,sysdate(),92802);

INSERT into ORDERS values(903,3,3,sysdate(),92803);

INSERT into ORDERS values(904,4,4,sysdate(),92804);

INSERT into ORDERS values(905,5,5,sysdate(),92805);

INSERT into ORDERS values(906,6,6,sysdate(),92806);