**SQL 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) Relational Database Management system is a software or program that is used to maintain a relational Database. It is used to store, manage, query and retrieve data stored in relational database.

In File system allows access to single files or tables at a time. In File System data is directly stored in a set of files. It contains flat files that have no relation to other files.

Advantages of RDBMS 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 data of the same file 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 into 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.
* **Backup:**It creates a backup subsystem to restore the data if required.
* **Interfaces**: It provides different multiple user interfaces like graphical user interface and application program interface.
* **Easy Maintenance**: It is easily maintainable due to its centralized nature.

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

Ans) A database management system must ensure ACID properties of transactions to maintain data. The ACID properties elaborate as:

1. **Atomicity (A)** – Each transaction in database must be atomic i.e. either all actions are carried out or none are. In case of system crash, there should be no effect of incomplete transactions.
2. **Consistency (C)**- Each transaction, run by itself with no concurrent execution of other transactions, must preserve the consistency of the database.
3. **Isolation (I)** – Transactions must be isolated or protected from the effects of concurrently scheduling other transactions.
4. **Durability (D)** – A transaction’s effect should persist even if the system crashes before all changes are reflected on disk. This property is known as durability.
5. Explain the concept of normalization.

Ans) Normalization is the process of organizing the data in the database.

* Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies.
* Normalization divides the larger table into smaller and links them using relationships.
* The normal form is used to reduce redundancy from the database table.

**Following are the various types of Normal forms:**

|  |  |
| --- | --- |
| **Normal Form** | **Description** |
| [1NF](https://www.javatpoint.com/dbms-first-normal-form) | A relation is in 1NF if it contains an atomic value. |
| [2NF](https://www.javatpoint.com/dbms-second-normal-form) | A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key. |
| [3NF](https://www.javatpoint.com/dbms-third-normal-form) | A relation will be in 3NF if it is in 2NF and no transition dependency exists. |
| BCNF | A stronger definition of 3NF is known as Boyce Codd's normal form. |
| [4NF](https://www.javatpoint.com/dbms-forth-normal-form) | A relation will be in 4NF if it is in Boyce Codd's normal form and has no multi-valued dependency. |
| [5NF](https://www.javatpoint.com/dbms-fifth-normal-form) | A relation is in 5NF. If it is in 4NF and does not contain any join dependency, joining should be lossless. |

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

Ans:) These SQL commands are mainly categorized into four categories as:

1. DDL – Data Definition Language
2. DQl – Data Query Language
3. DML – Data Manipulation Language
4. DCL – Data Control Language

**DDL (Data Definition Language):**

Data Definition Language actually consists of the SQL commands that can be used to define the database schema.

* [CREATE](https://www.geeksforgeeks.org/sql-create/): This command is used to create the database or its objects (like table, index, function, views, store procedure, and triggers).
* [DROP](https://www.geeksforgeeks.org/sql-drop-truncate/): This command is used to delete objects from the database.
* [ALTER](https://www.geeksforgeeks.org/sql-alter-add-drop-modify/): This is used to alter the structure of the database.
* [TRUNCATE](https://www.geeksforgeeks.org/sql-drop-truncate/): This is used to remove all records from a table, including all spaces allocated for the records are removed.
* [COMMENT](https://www.geeksforgeeks.org/sql-comments/): This is used to add comments to the data dictionary.
* [RENAME](https://www.geeksforgeeks.org/sql-alter-rename/): This is used to rename an object existing in the database.

### ****DQL (Data Query Language):****

* [SELECT](https://www.geeksforgeeks.org/sql-select-clause/): It is used to retrieve data from the database.

### ****DML(Data Manipulation Language):****

List of DML commands:

* [INSERT](https://www.geeksforgeeks.org/sql-insert-statement/) : It is used to insert data into a table.
* [UPDATE](https://www.geeksforgeeks.org/sql-update-statement/): It is used to update existing data within a table.
* [DELETE](https://www.geeksforgeeks.org/sql-delete-statement/) : It is used to delete records from a database table.

### ****DCL (Data Control Language):****

List of  DCL commands:

* [**GRANT:**](https://www.geeksforgeeks.org/mysql-grant-revoke-privileges/)This commandgives users access privileges to the database.
* [**REVOKE:**](https://www.geeksforgeeks.org/difference-between-grant-and-revoke/)This command withdraws the user’s access privileges given by using the GRANT command.

### ****TCL (Transaction Control Language):****

List of TCL commands:

* [**COMMIT**](https://www.geeksforgeeks.org/sql-transactions/)**:**Commits a Transaction.
* [**ROLLBACK**](https://www.geeksforgeeks.org/sql-transactions/)**:**Rollbacks a transaction in case of any error occurs.
* [**SAVEPOINT**](https://www.geeksforgeeks.org/sql-transactions/)**:**Sets a savepoint within a transaction.
* [**SET TRANSACTION:**](https://www.geeksforgeeks.org/sql-transactions/)Specify characteristics for the transaction.

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

Ans:) A **PRIMARY KEY** constraint uniquely identifies each record in a table. A Primary keys column must contain unique values and cannot have null values.A table can have only one primary key, which may consist of single or multiple columns.

A **COMPOSITE KEY** is a combination of two or more columns in a table that can be used to uniquely identify each row in the table when the columns are combined uniqueness is guaranteed, but when it taken individually it does not guarantee uniqueness.Note that the columns that make up a composite key can be of different data types.So basically, primary key becomes the composite key when more than column is used to uniquely identify each row in the table.

Suppose you have a sample data of customers which contains column as **CustName** , **Cust\_booldGroup** , **CustMailId** and you are asked to create a table with primary key column in order to identify each records in table uniquely.

As per given sample data, you simply cannot create a primary key either on **custname**, or **cust\_bloodgroup** as there may be chances that two customers might have same name or blood group.

Now have another option to create a primary key including both **custname** and **cust\_bloodgroup** columns but it can be a risky that that two customers can have same name and blood group combinations.

So, it will be a better way to create a primary key including all three columns to make a strong primary key that uniquely identify each row in a table.

So, in order to create a primary key on table column when you include more than one column this is called as a **composite key**. So basically, it is a approach choosing a primary key columns as a combitions of multiple columns when one column is not sufficient to identify each rows in a table uniquely.

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

Ans)

create table employee(

emp\_id int not null,

emp\_name varchar2(255),

location varchar2(255) default 'Bangalore',

dept\_id int unique,

salary int,

primary key(emp\_id)

);