**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?

Answers:

**RDBMS (relational database management system):**

A relational [database](https://searchdatamanagement.techtarget.com/definition/database) management system (RDBMS) is a collection of programs and capabilities that enable IT teams and others to create, update, administer and otherwise interact with a [relational database](https://searchdatamanagement.techtarget.com/definition/relational-database). RDBMSes store data in the form of tables, with most commercial relational database management systems using [Structured Query Language](https://searchsqlserver.techtarget.com/definition/SQL) (SQL) to access the database. However, since SQL was invented after the initial development of the relational model, it is not necessary for RDBMS use.

The RDBMS is the most popular database system among organizations across the world. It provides a dependable method of storing and retrieving large amounts of data while offering a combination of system performance and ease of implementation.

# Advantage of DBMS over file system:

There are several advantages of Database management system over file system. Few of them are as follows:

* **No redundant data**: Redundancy removed by data [**normalization**](https://beginnersbook.com/2015/05/normalization-in-dbms/). No data duplication saves storage and improves access time.
* **Data Consistency and Integrity**: As we discussed earlier the root cause of data inconsistency is data redundancy, since data normalization takes care of the data redundancy, data inconsistency also been taken care of as part of it
* **Data Security**: It is easier to apply access constraints in database systems so that only authorized user is able to access the data. Each user has a different set of access thus data is secured from the issues such as identity theft, data leaks and misuse of data.
* **Privacy**: Limited access means privacy of data.
* **Easy access to data** – Database systems manages data in such a way so that the data is easily accessible with fast response times.
* **Easy recovery**: Since database systems keeps the backup of data, it is easier to do a full recovery of data in case of a failure.
* **Flexible**: Database systems are more flexible than file processing systems.

# 2.In a database management system, explain the ACID properties.

# Ans:

ACID Properties in DBMS

**ACID (Atomicity, Consistency, Isolation, Durability)** is a set of properties of database transactions intended to guarantee validity even in the event of errors, power failures, etc.

In the context of databases, a sequence of database operations that satisfies the ACID properties, and thus can be perceived as a single logical operation on the data, is called a transaction.

For example, a transfer of funds from one bank account to another involves debiting from one account and crediting to another, and this whole process is a single transaction.

* **Atomicity**  
  All statements of a transaction must succeed completely, or fail completely in each and every situation, including power failures, errors and crashes. Example - Debiting and crediting in a money transfer transaction, both must happen either together or not at all.
* **Consistency**  
  The database must remain in a consistent state after any transaction. Data in the database should not have any changes other than intended after the transaction completion.
* **Isolation**  
  Isolation ensures that concurrent execution of transactions leaves the database in the same state that would have been obtained if the transactions were executed sequentially.
* **Durability**  
  Durability guarantees that once a transaction has been committed, it will remain committed even in the case of a system failure which actually means recording the completed transactions (or their effects) in non-volatile memory.

Mark as Completed

3.Explain the concept of normalization.

Ans:

### Introduction

This concept document provides a brief discussion of the topic of data normalization as it applies to the development of the [Artifact: Data Model](http://home.iscte-iul.pt/~hro/RUPSmallProjects/core.base_rup/workproducts/rup_data_model_65B46980.html). It does not provide a full treatment of normalization, because the subject is quite broad and has been documented in many texts on database design. In [[NBG01](http://home.iscte-iul.pt/~hro/RUPSmallProjects/core.base_rup/customcategories/references_56F06DFD.html#NBG01)], normalization is defined as "an analytic technique used to produce a correct relational database design." In practice, normalization is a procedure for eliminating redundancy in the Data Model by means of applying restrictive rules. Elimination of data redundancy in the tables of the Data Model helps enforce referential integrity of the data in the database.

Normalization is usually performed on the Data Model after an initial version of the tables and their relationships has been developed in the model. The exact timing of when to apply normalization depends on the specific project situation and is up to the [Database Designer](http://home.iscte-iul.pt/~hro/RUPSmallProjects/core.base_rup/roles/rup_database_designer_524DC34F.html). The normalization process is applied to the tables in the Data Model in series of steps in which each step applies rules that are stricter than the last.

### Levels of Normalization

Normalization is hierarchically classified into numeric forms, with the most common being first, second, and third normal form. Each level of normalization is more restrictive than the previous. The first three hierarchical levels of normalization are:

* **First Normal Form**-Repeating groups of data columns in tables have been eliminated such that data is organized into atomic units.
* **Second Normal Form**-Data is in first normal form, and redundancy on primary key fields has been eliminated such that column values are wholly dependent on the primary key field.
* **Third Normal Form**-Data is in second normal form, and each column is not dependent on any other non-key column.

Other levels of normalization are possible but are not covered in this discussion.  Information on additional levels of normalization can be found in [[DAT99](http://home.iscte-iul.pt/~hro/RUPSmallProjects/core.base_rup/customcategories/references_56F06DFD.html#DAT99)]. The exact level of normalization to apply to the Data Model is a decision that the database designer must make based on the specifics of the project situation.

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

Ans:

SQL commands are divided into four subgroups, DDL, DML, DCL, and TCL.

DDL:

DDL is short name of **Data Definition Language,** which deals with database schemas and descriptions, of how the data should reside in the databas

* [CREATE](https://www.w3schools.in/mysql/php-mysql-create/) - to create a database and its objects like (table, index, views, store procedure, function, and triggers)
* ALTER - alters the structure of the existing database
* DROP - delete objects from the database
* TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed
* COMMENT - add comments to the data dictionary
* RENAME - rename an object

DML:

DML is short name of **Data Manipulation Language** which deals with data manipulation and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE, etc., and it is used to store, modify, retrieve, delete and update data in a database.

* [SELECT](https://www.w3schools.in/mysql/php-mysql-select/) - retrieve data from a database
* [INSERT](https://www.w3schools.in/mysql/php-mysql-insert/) - insert data into a table
* [UPDATE](https://www.w3schools.in/mysql/php-mysql-update/) - updates existing data within a table
* [DELETE](https://www.w3schools.in/mysql/php-mysql-delete/) - Delete all records from a database table
* MERGE - UPSERT operation (insert or update)
* CALL - call a PL/SQL or Java subprogram
* EXPLAIN PLAN - interpretation of the data access path
* LOCK TABLE - concurrency Control

## DCL:

DCL is short name of **Data Control Language** which includes commands such as GRANT and mostly concerned with rights, permissions and other controls of the database system.

* GRANT - allow users access privileges to the database
* REVOKE - withdraw users access privileges given by using the GRANT command

## TCL:

TCL is short name of Transaction Control Language which deals with a transaction within a database.

* COMMIT - commits a Transaction
* ROLLBACK - rollback a transaction in case of any error occurs
* SAVEPOINT - to rollback the transaction making points within groups
* SET TRANSACTION - specify characteristics of 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:

**Primary key** is that column of the table whose every row data is uniquely identified. Every row in the table must have a primary key and no two rows can have the same primary key. Primary key value can never be null nor can be modified or updated.

**Composite Key** is a form of the candidate key where a set of columns will uniquely identify every row in the table.

A composite key in SQL can be defined as a combination of multiple columns, and these columns are used to identify all the rows that are involved uniquely. Even though a single column can’t identify any row uniquely, a combination of over one column can uniquely identify any record. In other words, the combination key can also be described as a primary key that is being created by using multiple columns. However, the data types of different columns could differ from each other. You can also combine all the foreign keys to create a composite key in[SQL](https://www.simplilearn.com/want-to-launch-career-as-sql-specialist-article).

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

Ans:

# SQL Server CREATE TABLE Statement

This [SQL tutorial](http://ramkedem.com/en/sql-tutorial/) explains how to use the **CREATE TABLE statement** in SQL Server. This tutorial is the first part of two posts describing DDL (Data Definition Language) statements in SQL Server.

The DDL statements are a subset of SQL statements used to create, modify, or remove database structures. In this post you will learn how to create and delete tables.

This tutorial allows you to become familiar with the following topics:

* [Creating new tables](https://ramkedem.com/en/sql-server-create-table/#CreatingNewTables)
  + [Defining Data Types](https://ramkedem.com/en/sql-server-create-table/#DataTypes)
  + [Defining Default Value](https://ramkedem.com/en/sql-server-create-table/#DefaultValue)
  + [Defining Constraints](https://ramkedem.com/en/sql-server-create-table/#Constraints)
  + [Column Level Constraints](https://ramkedem.com/en/sql-server-create-table/#ColumnLevelConstraints)
    - [Primary Key](https://ramkedem.com/en/sql-server-create-table/#PrimaryKey)
    - [Unique](https://ramkedem.com/en/sql-server-create-table/#Unique)
    - [Not Null](https://ramkedem.com/en/sql-server-create-table/#NotNull)
    - [Check](https://ramkedem.com/en/sql-server-create-table/#Check)
    - [Foreign Key](https://ramkedem.com/en/sql-server-create-table/#ForeignKey)
  + [Table Level Constraints](https://ramkedem.com/en/sql-server-create-table/#TableLevelConstraints)
* [Drop Existing Table](https://ramkedem.com/en/sql-server-create-table/#DropExistingTables)

The [next post](http://ramkedem.com/en/sql-server-alter-table/) will describe how to use the SQL Server ALTER TABLE statement.

## **SQL Server CREATE TABLE Statement**

SQL Server CREATE TABLE statement is used to create new tables in the database.

## **Data Types**

|  |  |  |
| --- | --- | --- |
| **Column Type** | **Description** | **Example** |
| **VARCHAR (**size**)** | String column. The value within the brackets indicates the maximum size of each field in the column (in characters) | VARCHAR(3) → ‘ABC’  VARCHAR(3) → ‘AB’ |
| **Decimal (p,s)** | Numeric column. **P**recision – number of digits, **S**cale – how many of the digits are located after the decimal point | DECIMAL(5,2) → 476.29  DECIMAL(5,2) → 6.29 |
| **DATE** | Date format column | ‘YYYY-MM-DD’ |