# Top Accenture SQL Interview Questions and Answers

## What is a UNIQUE constraint?

- **Definition**: A constraint that ensures all values in a column or a set of columns are unique.
- **Purpose**: Prevents duplicate entries in the specified column(s).
- **Example**: ALTER TABLE table_name ADD CONSTRAINT unique_constraint UNIQUE (column_name);

## What is a Query?

- **Definition**: A request for data or information from a database.
- **Types**:
    - **SELECT**: Retrieves data from a database.
    - **INSERT**: Adds new data.
    - **UPDATE**: Modifies existing data.
    - **DELETE**: Removes data.

## What is Data Integrity?

- **Definition**: The accuracy and consistency of data within a database.
- **Types**:
    - **Entity Integrity**: Ensures each row is uniquely identifiable.
    - **Referential Integrity**: Ensures relationships between tables remain consistent.
    - **Domain Integrity**: Ensures data values are within a specific range or format.

## What is the difference between Clustered and Non-clustered index?

- **Clustered Index**:
    - **Definition**: Sorts and stores data rows in the table based on the index key.
    - **Characteristics**: Only one clustered index per table; affects physical order of rows.
- **Non-clustered Index**:
    - **Definition**: Creates a separate structure from the data rows to store index keys and pointers.
    - **Characteristics**: Multiple non-clustered indexes can exist on a table; does not affect physical order of rows.

## What is an Index? Explain its different types.

- **Definition**: A database object that improves query performance by allowing faster data retrieval.

- **Types**:
    - **Clustered Index**: Defines the physical order of data in a table.
    - **Non-clustered Index**: Creates a logical order of data without changing the physical order.
    - **Unique Index**: Ensures all values in the index are unique.
    - **Composite Index**: Indexes multiple columns to improve performance on queries involving those columns.

## What is a Cross-Join?

- **Definition**: A type of join that returns the Cartesian product of two tables.
- **Characteristics**: Combines every row from the first table with every row from the second table.
- **Example**: SELECT * FROM table1 CROSS JOIN table2;

## What is a Self-Join?

- **Definition**: A join where a table is joined with itself.
- **Purpose**: Useful for comparing rows within the same table.
- **Example**: SELECT A.*, B.* FROM employees A JOIN employees B ON A.manager_id = B.employee_id;

## What is a Join? List its different types.

- **Definition**: A SQL operation that combines rows from two or more tables based on a related column.
- **Types**:
    - **INNER JOIN**: Returns rows with matching values in both tables.
    - **LEFT JOIN (or LEFT OUTER JOIN)**: Returns all rows from the left table and matched rows from the right table.
    - **RIGHT JOIN (or RIGHT OUTER JOIN)**: Returns all rows from the right table and matched rows from the left table.
    - **FULL JOIN (or FULL OUTER JOIN)**: Returns rows with a match in one of the tables.

## What is Pattern Matching in SQL?

- **Definition**: Pattern matching is used to search for specific patterns within column data.
- **Key Operators**:
    - LIKE: Searches for a specified pattern in a column.
    - %:  Represents zero or more characters in a pattern.
    - _: Represents a single character in a pattern.

## How to create empty tables with the same structure as another table?

- **Using CREATE TABLE AS SELECT**

CREATE TABLE new_table AS SELECT * FROM existing_table WHERE 1=0;

- **Using CREATE TABLE with LIKE**:

CREATE TABLE new_table LIKE existing_table;

## What is a Recursive Stored Procedure?

- **Definition**: A stored procedure that calls itself to perform repetitive tasks.
- **Usage**: Useful for tasks like traversing hierarchical data (e.g., organizational charts).

**What is a Stored Procedure?**

- **Definition**: A precompiled SQL statement or set of statements stored in the database.
- **Benefits**: Can be executed repeatedly, improves performance, and provides modular code.

## What is Collation? What are the different types of Collation Sensitivity?

- **Definition**: Collation determines how data is sorted and compared in SQL.
- **Types of Sensitivity**:
    - **Case Sensitivity**: Differentiates between uppercase and lowercase letters (e.g.,COLLATE SQL_Latin1_General_CP1_CI_AS).
    - **Accent Sensitivity**: Differentiates between accented and unaccented characters (e.g., COLLATE SQL_Latin1_General_CP1_CS_AS).
    - **Kana Sensitivity**: Differentiates between Hiragana and Katakana characters.
    - **Width Sensitivity**: Differentiates between single-byte and double-byte characters.

## What are the differences between OLTP and OLAP?

- **OLTP (Online Transaction Processing)**:
    - **Purpose**: Handles day-to-day operations and transactions.
    - **Characteristics**: Fast query processing, high transaction volume, normalized data.
- **OLAP (Online Analytical Processing)**:
    - **Purpose**: Used for complex queries and data analysis.
    - **Characteristics**: Optimized for read-heavy operations, supports multidimensional analysis, denormalized data.

## What is OLTP?

- **Definition**: OLTP stands for Online Transaction Processing.
- **Purpose**: Manages and processes transaction-oriented applications.

- **Characteristics**: High performance, quick query processing, and supports large numbers of simultaneous transactions.

## What is a User-defined Function? What are its various types?

- **Definition**: A function created by users to perform operations and return values.
- **Types**:
    - **Scalar Functions**: Return a single value (e.g., RETURN).
    - **Table-Valued Functions**: Return a table (e.g., RETURN TABLE).
    - **Inline Table-Valued Functions**: Return a table from a single SELECT statement.

## What is a Foreign Key?

- **Definition**: A column or a set of columns in one table that uniquely identifies a row in another table.
- **Purpose**: Ensures referential integrity by linking related data across tables.
- **Example**: FOREIGN KEY (column_name) REFERENCES other_table(column_name);

## What is a Subquery? What are its types?

- **Definition**: A query nested inside another query used to perform operations that require data from multiple queries.
- **Types**:
    - **Scalar Subquery**: Returns a single value (e.g., used in SELECT, WHERE clauses).
    - **Column Subquery**: Returns a single column of values (e.g., used in IN clause).
    - **Row Subquery**: Returns a single row of values (e.g., used in comparison with a row).
    - **Table Subquery**: Returns a set of rows and columns (e.g., used in the FROM clause).

## What is a Primary Key?

- **Definition**: A unique identifier for each record in a table.
- **Characteristics**: Must be unique and not null; ensures each row can be uniquely identified.
- **Example**: PRIMARY KEY (column_name);

## What are Constraints in SQL?

- **Definition**: Rules applied to columns in a table to ensure data accuracy and integrity.
- **Types**:
    - **NOT NULL**: Ensures that a column cannot have NULL values.
    - **UNIQUE**: Ensures all values in a column are unique.
    - **PRIMARY KEY**: Uniquely identifies each record in a table.

- **FOREIGN KEY**: Ensures referential integrity between tables.
- **CHECK**: Ensures that all values in a column satisfy a specific condition.
- **DEFAULT**: Provides a default value for a column when none is specified.

## What are Tables and Fields?

- **Tables**: Structures in a database that store data in rows and columns.
- **Fields**: Columns within a table that store specific pieces of data.

## What is the difference between SQL and MySQL?

- **SQL**: Structured Query Language used to manage and manipulate relational databases.
- **MySQL**: An open-source relational database management system that uses SQL to interact with the database.

## What is SQL?

- **Definition**: Structured Query Language, used for querying and managing relational databases.
- **Purpose**: Allows users to create, read, update, and delete data within a database.

## What is RDBMS? How is it different from DBMS?

- **RDBMS (Relational Database Management System)**:
  - **Definition**: A type of DBMS that stores data in structured tables with relationships between them.
  - **Examples**: MySQL, PostgreSQL, Oracle.
- **DBMS (Database Management System)**:
  - **Definition**: A software system for managing databases, which can be relational or non-relational.
  - **Examples**: MongoDB (NoSQL), SQLite (lightweight SQL).

## What is DBMS?

- **Definition**: Database Management System, software used to manage and interact with databases.
- **Purpose**: Provides tools for storing, retrieving, and manipulating data.

## What is a Database?

- **Definition**: A structured collection of data stored and managed electronically.
- **Purpose**: Organizes and allows efficient access, management, and manipulation of data.

## What is the SELECT statement?

- **Definition**: A SQL command used to query and retrieve data from a database.
- **Example**

```
SELECT column1, column2
FROM table_name
WHERE condition;
```