SQL Assignment 6

11. What is denormalization?

Ans. In SQL, denormalization is the process of intentionally adding redundant data to a database table to improve performance or simplify querying. This technique is used to avoid the performance issues that can occur when querying large and complex normalized database tables.

Normalization is the process of organizing data in a database by dividing it into two or more tables and establishing relationships between them. Normalization helps to eliminate redundancy and reduce data inconsistency, but it can also make querying the data more complex and time-consuming.

Denormalization involves adding redundant data to a database table to improve query performance by reducing the need for complex joins or subqueries. This can result in faster query performance, especially when working with large datasets or complex queries.

There are several ways to denormalize a database table, including adding redundant columns, creating summary tables, or duplicating data in multiple tables. However, denormalization can also increase data redundancy and the risk of data inconsistency, so it should be used carefully and only in cases where it provides significant performance benefits.

Overall, denormalization is a trade-off between query performance and data consistency, and it should be used judiciously in order to optimize query performance without compromising the integrity of the data.

12. What is a database cursor?

Ans. A database cursor is a programming mechanism used to traverse records in a database. It is a database object that is used to process or manipulate individual rows returned by a database query.

When a query is executed against a database, the database server returns a result set, which is a collection of records that meet the criteria of the query. A cursor allows the programmer to navigate through this result set and access each record one at a time.

A cursor is typically used in database applications to loop through the records in a result set, perform some operation on each record, and then move on to the next record. This can be useful for tasks like generating reports, processing data, or updating records.

In general, cursors provide a powerful and flexible way to work with data in a database, but they can also be resource-intensive and slow down performance if not used correctly. As a result, many modern databases provide alternative mechanisms, such as set-based operations, to achieve the same result without using a cursor.

13. What are the different types of queries?

In SQL, there are several types of queries that can be used to manipulate and retrieve data from a database. The most common types of queries are:

1. SELECT: The SELECT statement is used to retrieve data from one or more tables in a database. It allows you to specify which columns you want to retrieve, and also allows you to apply filters and sorting to the data.
2. INSERT: The INSERT statement is used to add new rows of data to a table in a database. You specify the table name and the values you want to insert into the table.
3. UPDATE: The UPDATE statement is used to modify existing rows of data in a table in a database. You specify the table name, the column to be updated, and the new value.
4. DELETE: The DELETE statement is used to remove rows of data from a table in a database. You specify the table name and the condition that determines which rows to delete.
5. JOIN: The JOIN statement is used to combine data from two or more tables in a database. You specify the tables to join and the join condition, which is used to match rows from one table with rows from another table.
6. UNION: The UNION statement is used to combine the results of two or more SELECT statements into a single result set. The SELECT statements must have the same number of columns and compatible data types.
7. GROUP BY: The GROUP BY statement is used to group rows of data based on one or more columns. It allows you to perform aggregate functions, such as SUM, COUNT, and AVG, on the grouped data.
8. HAVING: The HAVING statement is used in conjunction with the GROUP BY statement to filter the grouped data based on a condition.

These are some of the most commonly used types of queries in SQL, and understanding how to use them is essential for working with a database.

14. Define constraint?

Ans. In SQL, a constraint is a rule or restriction that is applied to a column or a set of columns in a table. Constraints are used to enforce data integrity and to ensure that the data stored in a database is accurate and consistent.

There are several types of constraints that can be used in SQL, including:

9. NOT NULL: This constraint ensures that a column cannot contain NULL values.

10. UNIQUE: This constraint ensures that each value in a column is unique and cannot be repeated.
11. PRIMARY KEY: This constraint is used to define a column or set of columns that uniquely identify each row in a table.
12. FOREIGN KEY: This constraint is used to establish a relationship between two tables. It ensures that the values in a column in one table match the values in a column in another table.
13. CHECK: This constraint is used to specify a condition that must be met for the data in a column to be valid.
14. DEFAULT: This constraint is used to specify a default value for a column when a new row is inserted into a table.

Constraints can be applied when a table is created, or they can be added later using an ALTER TABLE statement. Constraints can help to ensure data integrity and prevent data inconsistencies or errors, which is essential for the accuracy and reliability of a database.

15. What is auto increment?

Ans. In SQL, auto-increment is a feature that allows a database system to automatically generate a unique, sequential number for a column in a table. This is commonly used to create a primary key for a table that is guaranteed to be unique for each row in the table.

Auto-increment is typically implemented using a special data type, such as INTEGER, that is defined as an identity column or a sequence. When a new row is added to the table, the database system automatically generates a new, unique value for the column with the auto-increment feature.

For example, if a table has a column named "ID" with auto-increment enabled, the first row inserted into the table will be assigned the value 1 for the ID column. The second row inserted will be assigned the value 2, and so on.

Auto-increment is a convenient way to ensure that each row in a table has a unique identifier. It can simplify the process of adding new data to a database, and it can also improve the performance of certain operations, such as searching for specific rows in a large table.