

WORKSHEET 6 SQL

ASSIGNMENT-6

1. Which of the following are TCL commands?

Ans:

A,C,D.

2. Which of the following are DDL commands?

Ans:

A,C,D

3. Which of the following is a legal expression in SQL?

Ans:

B

4. DCL provides commands to perform actions like-?

Ans:

C

5. Which of the following should be enclosed in double quotes?

Ans:

B

6. Which of the following command makes the updates performed by the transaction permanent in the database?

Ans:

B

7. A subquery in an SQL Select statement is enclosed in:

Ans:

A

8. The result of a SQL SELECT statement is a :-

Ans:

C

9. Which of the following do you need to consider when you make a table in a SQL?

Ans:

D

10. If you don't specify ASC and DESC after a SQL ORDER BY clause, the following is used by___?

Ans:

A

11. What is denormalization?

Ans:

Denormalization is a database optimization technique in which we add redundant data to one or more tables. This can help us avoid costly joins in a relational database. Note that denormalization does not mean 'reversing normalization' or 'not to normalize'. It is an optimization technique that is applied after normalization.

Basically, The process of taking a normalized schema and making it non-normalized is called denormalization, and designers use it to tune the performance of systems to support time-critical operations.

In a traditional normalized database, we store data in separate logical tables and attempt to minimize redundant data. We may strive to have only one copy of each piece of data in a database.

Example: Suppose after normalization we have two tables first, Student table and second, Branch table. The student has the attributes as Roll_no , Student-name , Age , and Branch_id .

Student table

| Roll_no | Student_name | Age | Branch_id |
|---------|--------------|-----|-----------|
| 1 | Andrew | 18 | 10 |
| 2 | Angel | 19 | 10 |
| 3 | Priya | 20 | 10 |
| 4 | Analisa | 21 | 11 |
| 5 | Anna | 21 | 12 |

The branch table is related to the Student table with Branch_id as the foreign key in the Student table.

Branch table

| Branch_id | Branch_name | HOD |
|-----------|-------------|--------|
| 10 | CSE | Mr.abc |
| 11 | EC | Dr.xyz |
| 12 | EX | Dr.pqr |

If we want the name of students along with the name of the branch name then we need to perform a join operation. The problem here is that if the table is large we need a lot of time to perform the join operations. So, we can add the data of Branch_name from Branch table to the Student table and this will help in reducing the time that would have been used in join operation and thus optimize the database

1. Advantages of Denormalization:

Query execution is fast since we have to join fewer tables.

2. Disadvantages of Denormalization

As data redundancy is there, update and insert operations are more expensive and take more time. Since we are not performing normalization, so this will result in redundant data.

Data Integrity is not maintained in denormalization. As there is redundancy so data can be inconsistent.

12. What is a database cursor?

Ans:

A database cursor can be thought of as a pointer to a specific row within a query result. The pointer can be moved from one row to the next. Depending on the type of cursor, you may be even able to move it to the previous row. SQL result is like a bag, you get to hold a whole bunch of rows at once, but not any of them individually; whereas, a cursor is like a pair of tweezers. With it, you can reach into the bag and grab a row, and then move onto the next.

Types of Cursors:

The type of cursors you can define are broken in two main categories: scrolling capabilities and ability to detect changes made to the database.

Let's first talk about scrolling capabilities. Cursors can be defined with two main scrolling capabilities, FORWARD_ONLY or SCROLL.

FORWARD_ONLY – The cursor starts on the first row and end on the last. The cursor can only move to the next row in the result.

SCROLL – the cursor can use operations, such as FIRST, LAST, PRIOR, NEXT, RELATIVE, ABSOLUTE to navigate the results.

13. What are the different types of the queries?

Ans:

Navigational search queries:

A navigational query is a search query entered with the intent of finding a particular website or webpage. For example, a user might enter "youtube" into Google's search bar to find the YouTube site rather than entering the URL into a browser's navigation bar or using a bookmark. In fact, "facebook" and "youtube" are the top two searches on Google, and these are both navigational queries.

Informational search query: When someone enters an informational search query into Google or another search engine, they're looking for information – hence the name.

They are probably not looking for a specific site, as in a navigational query, and they are not looking to make a commercial transaction. They just want to answer a question or learn how to do something.

Transactional search queries: A transactional search query is a query that indicates an intent to complete a transaction, such as making a purchase. Transactional search queries may include exact brand and product names (like “samsung galaxy s3”) or be generic (like “iced coffee maker”) or actually include terms like “buy,” “purchase,” or “order.” In all of these examples, you can infer that the searcher is considering making a purchase in the near future, if they’re not already pulling out their credit card. In other words, they’re at the business end of the conversion funnel. Many local searches (such as “Denver wine shop”) are transactional as well.

Vertical searches are a subset of transactional search queries, and they represent people looking to make a transaction in a specific industry. These include local searches, restaurant searches, hotel searches, flight searches, etc. Google’s moves in recent years to directly target vertical searches have led to accusations of antitrust violations.

14. Define constraint?

Ans:

Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.

constraints are a set of rules implemented on tables in relational databases to dictate what data can be inserted, updated or deleted in its tables. This is done to ensure the accuracy and the reliability of information stored in the table. Constraints enforce limits to the data or type of data that can be inserted/updated/deleted from a table. The purpose of constraints is to maintain the data integrity during an update/delete/insert into a table. Once the constraint is placed, if any operation in the database does not follow the rules specified by the constraint, the particular operation is aborted. In this article, we will go through what SQL constraints are, what are the different kinds of SQL constraints are commonly used and how to implement and get rid of them. First, however, we will take a brief look into why they are needed.

import constraint.

define a variable as our problem.

add variables and their respective intervals to our problem.

add built-in/custom constraints to our problem.

fetch the solutions.

go through the solutions to find the ones we need.

15. What is auto increment?

Ans:

Auto-increment allows a unique number to be generated automatically when a new record is inserted into a table.

Often this is the primary key field that we would like to be created automatically every time a new record is inserted. The auto increment in SQL is a feature that is applied to a field so that it can automatically generate and provide a unique value to every record that you

enter into an SQL table. This field is often used as the PRIMARY KEY column, where you need to provide a unique value for every record you add. However, it can also be used for the UNIQUE constraint columns.