## NAME: Dipesh Ramesh Limaje

**INTERNSHIP BATCH : 33**

## TOPIC: SQL

**SME : Mr. Shwetank Mishra**

**WORKSHEET NO : 6**

Q1 Ans A, C, D Commit, Rollback, and Savepoint.

Q2 Ans A,C, D Alter Drop Create

Q3 Ans B SELECT NAME FROM SALES;

Q4 Ans C Authorizing Access and other control over Database

Q5 Ans B Column Alias

Q6 Ans B COMMIT

Q7 Ans A Parenthesis - (...).

Q8 Ans C TABLE

Q9 Ans D All of the mentioned

Q10 Ans A ASC

**Q11] 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. It is an optimization technique that is applied after normalization.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.

**Pros of Denormalization:**

1] Retrieving data is faster since we do fewer joins

2] Queries to retrieve can be simpler(and therefore less likely to have bugs)

**Cons of Denormalization:**

1] Updates and inserts are more expensive.

2] Denormalization can make update and insert code harder to write.

3] Data may be inconsistent.

4] Data redundancy necessitates more storage.

**Q12] What is a database cursor?**

ANS Cursor is a Temporary Memory or Temporary Work Station. It is Allocated by Database Server at the Time of Performing DML(Data Manipulation Language) operations on Table by User. Cursors are used to store Database Tables. There are 2 types of Cursors: Implicit Cursors, and Explicit Cursors. These are explained as following below.

**1] Implicit Cursors:**

Implicit Cursors are also known as Default Cursors of SQL SERVER. These Cursors are allocated by SQL SERVER when the user performs DML operations.

**2] Explicit Cursors :**

Explicit Cursors are Created by Users whenever the user requires them. Explicit Cursors are used for Fetching data from Table in Row-By-Row Manner.

**Q13] What are the different types of the queries?**

**ANS**

**1. Creating a table (CREATE TABLE)**

To create a brand new, empty table, you will use the SQL keyword "CREATE TABLE."

**2. Inserting records in a table (INSERT INTO)**

You can easily insert rows of data, or records, into a new or existing table. To do this, you will use the INSERT INTO command followed by the name of the table you are working with.

**3. Viewing all records from a table (SELECT)**

The SELECT statement represents one of the simplest and most frequently used in SQL. It allows you to view all of the records from a table that you specify.

**4. Arranging the records in a table (ORDER BY)**

You use the ORDER BY clause with your SELECT query to arrange the displayed results in a particular order.

**5. Viewing only selected records from a table (SELECT COUNT)**

If a huge number of rows exist in a table, and you do not want all the records to fill your display screen, then SQL gives an option to view only selected rows. You can use the SELECT COUNT clause to do this.

**6. Deleting records from a table (DELETE)**

To delete selected rows from a table

**7. Changing data in existing records in a table (UPDATE)**

The UPDATE clause allows you to change information for records that already exist in your table.

**8. Adding or deleting columns in a table (ALTER TABLE)**

If you would like to create new columns in an existing table in your database, you can use the ALTER TABLE command.

**Q14] Define constraint?**

**ANS** SQL 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.

**Types of SQL Constraints**

SQL constraints can be at a column or a table level. Column level constraints apply to specific columns in a table and do not specify a column name except the check constraints.
Following is a list of the most commonly used column and table level SQL constraints:
1] NOT NULL Constraint
2] UNIQUE Constraint
3] DEFAULT Constraint
4] CHECK Constraint
5] PRIMARY KEY Constraint
6] FOREIGN KEY Constraint

**Q15] What is auto increment?**

**ANS**

Auto Increment is a field used to generate a unique number for every new record added into a table. This is generally used for the primary key column as it becomes easy for the developers to automatically generate a unique number for every new record.
Auto increment columns are typically used as the primary key of a table. Primary keys serve as a unique identifier for each row in the table and ensure that each row can be easily retrieved and updated. When a new row is inserted into a table with an auto increment column, the database automatically generates a new unique number for the column and assigns it to the new row.

For example, if a table has an auto increment column named "ID", the first row inserted into the table might have an ID value of 1, the second row inserted might have an ID value of 2, and so on. The database automatically keeps track of the next available value for the auto increment column and assigns it to new rows as they are inserted.
In some databases, you can specify the starting value and the increment step for the auto increment column. For example, you could specify that the auto increment column should start at 1000 and increase by 10 each time a new row is inserted.

Using auto increment columns can greatly simplify database development and administration. For example, you don't need to worry about manually assigning unique values to each row, and you don't need to worry about the possibility of duplicate values being inserted into the table.