1. What is difference between Truncate and Delete?

Ans: Difference between Truncate and Delete-

Truncate	Delete
It is a DDL command.	It is DML command.
It is used to delete all the record from table.	It is used to delete all the record or specific record from the table.
It will delete all the record at a time.	It will delete all the record page by page.
It will work fast compare to delete.	It will work slow compare with truncate.

2. What is Primary and Foreign key?

Ans:

Primary Key-

- Primary Key constraint will not allow duplicate values.
- Primary Key constraint will not allow null values.
- We can't apply more than one primary key constraint on a single table.
- Primary Key constraint will have constraint name.

Foreign Key-

A foreign key constraint is used to establish the relation between two or more tables.

Note:

- One table should contain primary key and other table should contain foreign key.
- A common column should be in both tables.
- and common column datatype of both tables should be same.

3. What are Indexes? Define its types.

Ans:

Indexes-

- An Index in SQL is similar like an index in a book.
- Indexes are used for- FAST ACCESSING OF DATA.

Indexes are two types-

- ✓ **Clustered Index**
- ✓ **Non-Clustered Index**

A. Clustered Index-

- Whenever we apply primary key constraint on a column in a table then automatically clustered index will be applied on primary key column.
- And that clustered index will arrange the data in Ascending Order.
- We can't apply more than one clustered index on a single table.

Syntax-

Create clustered index index_name on tablename(columnname)

Example-

Create clustered index index1 on employee (eno)

B. Non-Clustered Index-

- Whenever we apply unique key constraint on a column in a table then automatically non-clustered index will be applied.
- And that non- clustered index will not arrange the data in Ascending Order.
- We can apply more than one non- clustered index on a single table.

Syntax-

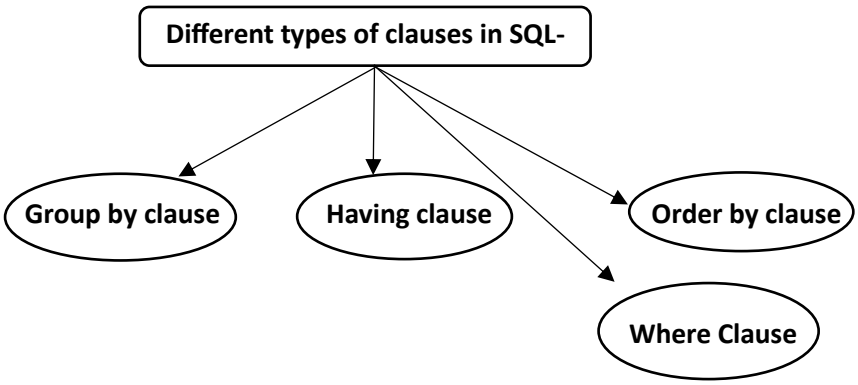
Create non_clustered index index_name on tablename(columnname)

Example-

Create non_clustered index index1 on employee (eno)

4. Explain the different types of clauses in SQL.

Ans:



A. Group by clause-

- It will group the common set of values as single group in a single column.
- It will always work with aggregation functions.

Example-

Select Ename, min(salary) as 'min sal' , max(salary) as 'max sal' from employee group by Ename

Note: with group by clause we can't apply where condition

B. Having clause-

It is used to search conditions for a group or an aggregate. It is generally used in Group by clause. If we are not using group by clause then we can use having clause just like where clause.

Example

Given below is an example for the use of Having clause −

Create table product (pid number (10), company varchar2(20), qty number (10), rate number (10), cost number (10));
insert into product values (1,'LANCO', 20,450,4000);
insert into product values(2,'METRO',30,400,3000);
insert into product values(3,'Tata',40,350,4500);
insert into product values(4,'Tata',30,250,5000);
insert into product values(5,'METRO',22,400,4500);

select * from product;

SELECT company, COUNT (*) FROM product group by company
having count (*)>1;

C. Order by clause-

- It is used to display the data either in Ascending or descending order.
- By default, order by clause arrange the data in ascending order.

Syntax-

Select column1, column2

From table_name where condition

Order by column1, column2…..ASC|DESC

Example 1-

Select * from product

Order by company

Note: By default, it is showing in Ascending order.

Example 2-

Select * from product

Order by company DESC

D. Order by clause-

- The SQL WHERE clause is used to specify a condition while fetching the data from a single table or by joining with multiple tables
- If the given condition is satisfied, then only it returns a specific value from the table.
- You should use the WHERE clause to filter the records and fetching only the necessary records.
- The WHERE clause is not only used in the SELECT statement, but it is also used in the UPDATE, DELETE statement, etc.

Syntax- select comlum1 , column2, column3

From table name where [condition]

Example- Consider the CUSTOMERS table having the following records –

```
+----+----------+-----+-----------+----------+
| ID | NAME     | AGE | ADDRESS   | SALARY   |
+----+----------+-----+-----------+----------+
|  1 | Ramesh   |  32 | Ahmedabad |  2000.00 |
|  2 | Khilan   |  25 | Delhi     |  1500.00 |
|  3 | kaushik  |  23 | Kota      |  2000.00 |
|  4 | Chaitali |  25 | Mumbai    |  6500.00 |
|  5 | Hardik   |  27 | Bhopal    |  8500.00 |
|  6 | Komal    |  22 | MP        |  4500.00 |
|  7 | Muffy    |  24 | Indore    | 10000.00 |
+----+----------+-----+-----------+----------+
```

The following code is an example which would fetch the ID, Name and Salary fields from the CUSTOMERS table, where the salary is greater than 2000 –

Query-

```
SELECT ID, NAME, SALARY

FROM CUSTOMERS

WHERE SALARY > 2000;
```

Output-

```
+----+----------+----------+
| ID | NAME     | SALARY   |
+----+----------+----------+
|  4 | Chaitali |  6500.00 |
|  5 | Hardik   |  8500.00 |
|  6 | Komal    |  4500.00 |
|  7 | Muffy    | 10000.00 |
+----+----------+----------+
```

5. What is Stored Procedure? Define its types.

Ans:

Stored Procedure:

- Stored Procedure is set of pre-compiled SQL statement which will gets executed when we call it.
- Whenever we pass any SQL Query, 3 types of operations will be done-

    - ✓ Syntax checking
    - ✓ Proper plan is selected like any indexes applied or not.
    - ✓ Query execution will be done.

Types of Stored Procedure:

- **Pre-defined Stored Procedure**
- **User-defined Stored Procedure**

➢ **Pre-defined Stored Procedure-**

It is created by Microsoft.
All the system defined SPs are available under- Master Database of SQL with programmability option with Stored Procedure.

➢ **User-defined Stored Procedure-**

SP that was created by developer on the user requirement.

6. Explain Exception Handling.

Ans:

Exception Handling-

- Whenever we writes any SP , the procedure will be compiled at the time of compilation of the procedure.
- Compile will check for syntax errors and display to the programmers.
- Programmers will read the error msg and rectify it.

Note: Exception is nothing but runtime error occur at the time of executing the program.

For handling this type of error, we use exception handling.

We can handle run-time error in 2 ways-

a. **Logical Implementation**
b. **Try-Catch Implementation**

Logical Implementation-

- Programmer must analyze that if any runtime error, they have to identify and handle the runtime error by writing some logic.0000000000032.000
- It is difficult for the programmer to write the logic each and every time so reduce the burden on the programmer.
- That's why Microsoft has given the Try-Catch implementation.

Try- Catch Implementation-

- **Try-** Declare and set the variable and print the variable.
- **Catch-** Catch the occurs if it is in program otherwise it will be executed.

7. Define Trigger and Explain types of Triggers.

Ans:

Trigger:

It is a special type of stored procedure which will gets invoked immediately after performing DML operations.

Types-

- **DDL Trigger**
- **DML Trigger**
- **Instead of Trigger.**

DDL Trigger- It is used to create, modify, drop the trigger.

DML Trigger-

It will be invoked immediately after performing DML Operations

like – insert, update and delete.

Syntax-

To create a trigger-

```
Create trigger trigger name on tablename after insert/ delete/update
As
Begin
End
```

Example-

create trigger to display record in inserted successfully.

```
Create trigger t1 on employee
after insert
As
Begin
Print ('Record is inserted successfully');
end
```

8. What is view in SQL?

Ans:

View-

- View is an imaginary table or virtual table.
- View is used to hide the confidential data from the table.
- View is stored select query.

Type-

- **Simple View**
- **Complex View**

Simple View- Simple view that was created on single table is called as Simple view.

Complex View- Complex view that was created on multiple tables is called as complex view.

Syntax-

To create a view-

```
Create view view_name
As
Selected query
```

Example-

```
Create view ITdept
As
Select *from employee where dname='IT'
```

9. What are functions? Define its types.

Ans:

Function:

It is a sub program which is used to perform some operation and return some value.

Types of function-

- **System-defined functions-**
  System functions are built-in database functions.

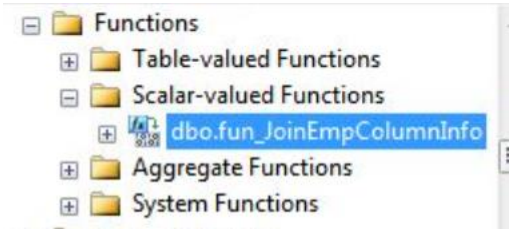
- **User-defines functions-** User-defined functions are create by a user.

  a) **Scaler valued function-**
  Now we are getting an employee table with two different data joined and displayed in a single column data row. Here create a two-column join function as in the following:

```
create function fun_JoinEmpColumnInfo
    (
        @EmpContact nchar (15),
        @EmpEmail nvarchar (50),
        @EmpCity varchar (30)
     )
        returns nvarchar (100)
          as
        begin
return (select @EmpContact+ ' ' +@EmpEmail + ' ' +      @EmpCity)
         end
```

After running this query- we will see a created scalar function in the database.

Like that-



  b) **Table values function-** In this type of function, we select a table data using a user-created function. A function is created using the Create function SQL command. The following query creates a new user-defined function.

```
Create function Fun_EmployeeInformation ()
    returns table
     as
 return (select * from Employee)
```

After running this query, we will see- a newly created function in the database.

10. What is Aggregate Function?

Ans:

Aggregate Function- A function which will process on multiple rows at a time and return only one value is called as multi row function or aggregate function.

Types of Aggregate function-

I. **SUM ()-**
Sum function is used to calculate the sum of all selected columns. It works on numeric fields only.

Example- we have below table name is "PRODUCT_MAST "like-

PRODUCT_MAST

PRODUCT	COMPANY	QTY	RATE	COST
Item1	Com1	2	10	20
Item2	Com2	3	25	75
Item3	Com1	2	30	60
Item4	Com3	5	10	50
Item5	Com2	2	20	40
Item6	Cpm1	3	25	75
Item7	Com1	5	30	150
Item8	Com1	3	10	30
Item9	Com2	2	25	50
Item10	Com3	4	30	120

I want to sum of cost columns from above table so for that we need to write query like that-

Query-

```
SELECT SUM(COST)
FROM PRODUCT_MAST;
```

Output- 670

II. **MAX ()-**
MAX function is used to find the maximum value of a certain column. This function determines the largest value of all selected values of a column.

Example-

Query-

```
SELECT MAX(RATE)
FROM PRODUCT_MAST;
```

Output- 30

III. AVG ()-

The AVG function is used to calculate the average value of the numeric type. AVG function returns the average of all non-Null values.

Example-

Query-

```
SELECT AVG(COST)
 FROM PRODUCT_MAST;
```

Output- 67.00

IV. MIN ()-

MIN function is used to find the minimum value of a certain column. This function determines the smallest value of all selected values of a column.

Example-

Query-

```
SELECT MIN(RATE)
 FROM PRODUCT_MAST;
```

Output- 10

COUNT ()-

- COUNT function is used to Count the number of rows in a database table. It can work on both numeric and non-numeric data types.
- COUNT function uses the COUNT (*) that returns the count of all the rows in a specified table. COUNT (*) considers duplicate and Null.

Example-

Query-

```
SELECT COUNT (*)
 FROM PRODUCT_MAST;
```

Output- 10

11. Explain Mathematical functions in SQL?

Ans: Mathematical functions in SQL-

Mathematical functions are very important in SQL to implement different mathematical concepts in queries.

Some of the major mathematical functions in SQL are as follows –

1. **ABS(X)**

This function returns the absolute value of X. For example −

```
Select abs (-6);
```

This returns 6.

2. **MOD (X, Y)**

The variable X is divided by Y and their remainder is returned. For example −

```
Select mod (9,5);
```

This returns 4.

3. **SIGN(X)**

This method returns 1 if X is positive, -1 if it is negative and 0 if the value of X is 0. For example −

```
Select sign (10);
```

This returns 1.

4. **FLOOR(X)**

This returns the largest integer value that is either less than X or equal to it. For example −

```
Select floor (5.7);
```

This returns 5.

5. **CEIL(X)**

This returns the smallest integer value that is either more than X or equal to it. For example −

```
Select ceil (5.7);
```

This returns 6.

6. **POWER (X, Y)**

This function returns the value of x raised to the power of Y For example −

```
Select power (2,5);
```

This returns 32.

7. **ROUND(X)**

This function returns the value of X rounded off to the whole integer that is nearest to it. For example –

```
Select round (5.7);
```

This returns 6.

8. **SQRT(X)**

This function returns the square root of X. For example –

```
Select sqrt (9);
```

This returns 3.

9. **ASIN(X)**

This function accepts a Sin value as the input and returns the angle in radians. For example –

```
Select Asin (0);
```

This returns 0.

10. **ACOS(X)**

This function accepts a Cos value as the input and returns the angle in radians. For example –

```
Select acos (1);
```

This returns 0.

11. **ATAN(X)**

This function accepts a Tan value as the input and returns the angle in radians. For example –

```
Select atan (0);
```

This returns 0.

12. **SIN(X)**

This function accepts an angle in radians as its parameter and returns its Sine value. For example –

```
Select sin (0);
```

This returns 0.

13. **COS(X)**

This function accepts an angle in radians as its parameter and returns its Cosine value. For example –

```
Select cos (0);
```

This returns 1.

14. **TAN(X)-**

This function accepts an angle in radians as its parameter and returns its Tan value. For example –

```
Select tan (0);
```

12.Define Transaction Control Language in SQL?

Ans: Transaction Control Language (TCL):

- TCL stands for Transaction Control Language.
- This command is used to manage the changes made by DML statements.
- TCL allows the statements to be grouped together into logical transactions.

TCL commands are as follows:
1. COMMIT
2. SAVEPOINT
3. ROLLBACK
4. SET TRANSACTION

1. COMMIT COMMAND

- COMMIT command saves all the work done.
- It ends the current transaction and makes permanent changes during the transaction.

Syntax:
commit;

2. SAVEPOINT COMMAND

- SAVEPOINT command is used for saving all the current point in the processing of a transaction.
- It marks and saves the current point in the processing of a transaction.

  Syntax:
  SAVEPOINT <savepoint_name>

  Example:
  SAVEPOINT no_update;

- It is used to temporarily save a transaction, so that you can rollback to that point whenever necessary.

3. ROLLBACK COMMAND

- ROLLBACK command restores database to original since the last COMMIT.
- It is used to restores the database to last committed state.

Syntax:
ROLLBACK TO SAVEPOINT <savepoint_name>;

Example:
ROLLBACK TO SAVEPOINT no_update;

4. SET TRANSACTION

- SET TRANSACTION is used for placing a name on a transaction.

  Syntax:
  SET TRANSACTION [Read Write | Read Only];

- You can specify a transaction to be read only or read write.
- This command is used to initiate a database transaction.
-

13. What are Differences between ROLLBACK and COMMIT commands.

ROLLBACK	COMMIT
ROLLBACK command is used to undo the changes made by the DML commands.	The COMMIT command is used to save the modifications done to the database values by the DML commands.
It rollbacks all the changes of the current transaction.	It will make all the changes permanent that cannot be rolled back.
Syntax: DELETE FROM table_name ROLLBACK	Syntax: COMMIT;

14. What are Differences between Functions and Stored Procedures.

Answer:

Function	Stored Procedure
Always returns a single value; either scalar or a table.	Can return zero, single or multiple values.
Functions are compiled and executed at run time.	Stored procedures are stored in parsed and compiled state in the database.
Only Select statements. DML statements like update & insert are not allowed.	Can perform any operation on database objects including select and DML statements.
Allows only input parameters. Does not allow output parameters.	Allows both input and output parameters
Does not allow the use of Try...Catch blocks for exception handling.	Allows use of Try...Catch blocks for exception handling.
Cannot have transactions within a function.	Can have transactions within a stored procedure.
Cannot call a stored procedure from a function.	Can call a function from a stored procedure.
Temporary tables cannot be used within a function. Only table variables can be used.	Both table variables and temporary tables can be used.
Functions can be called from a Select statement.	Stored procedures cannot be called from a Select/Where or Having statements. Execute statement has to be used to execute a stored procedure.
Functions can be used in JOIN clauses.	Stored procedures cannot be used in JOIN clauses

15. Define Normalization and its types.

Ans:

Normalization:

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies.
- Normalization divides the larger table into smaller and links them using relationships.
- The normal form is used to reduce redundancy from the database table.

Types of Normal Forms:

Normalization works through a series of stages called Normal forms. The normal forms apply to individual relations. The relation is said to be in particular normal form if it satisfies constraints.

1) **First Normal Form(1NF)**

- A relation will be 1NF if it contains an atomic value.
- It states that an attribute of a table cannot hold multiple values. It must hold only single-valued attribute.
- First normal form disallows the multi-valued attribute, composite attribute, and their combinations.

2) **Second Normal Form (2NF**

- In the 2NF, relational must be in 1NF.
- In the second normal form, all non-key attributes are fully functional dependent on the primary key

3) **Third Normal Form(3NF)**

- A relation will be in 3NF if it is in 2NF and not contain any transitive partial dependency.
- 3NF is used to reduce the data duplication. It is also used to achieve the data integrity.
- If there is no transitive dependency for non-prime attributes, then the relation must be in third normal form.

4) Boyce Codd normal form (BCNF)

- A relation will be in 3NF if it is in 2NF and not contain any transitive partial dependency.
- 3NF is used to reduce the data duplication. It is also used to achieve the data integrity.
- If there is no transitive dependency for non-prime attributes, then the relation must be in third normal form.

5) Fourth Normal Form(4NF)

- BCNF is the advance version of 3NF. It is stricter than 3NF.
- A table is in BCNF if every functional dependency X → Y, X is the super key of the table.
- For BCNF, the table should be in 3NF, and for every FD, LHS is super key.

6) Fifth Normal Form(5NF)

- A relation is in 5NF if it is in 4NF and not contains any join dependency and joining should be lossless.
- 5NF is satisfied when all the tables are broken into as many tables as possible in order to avoid redundancy.
- 5NF is also known as Project-join normal form (PJ/NF).

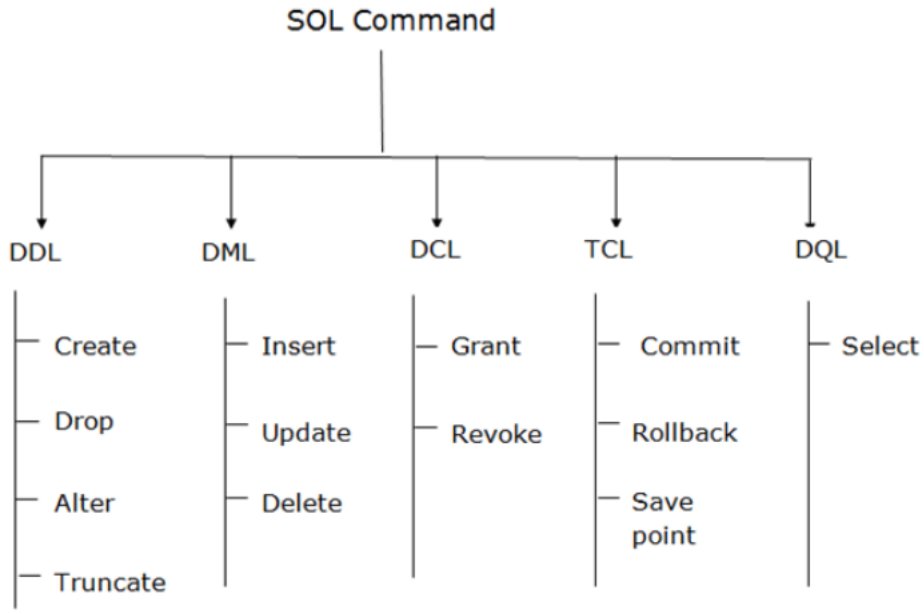
16. Explain SQL Commands.

Ans: SQL Commands-

- SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data.
- SQL can perform various tasks like create a table, add data to tables, drop the table, modify the table, set permission for users.

Types of SQL Commands

There are **five** types of SQL commands: **DDL, DML, DCL, TCL, and DQL.**



1. Data Definition Language (DDL)

- DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.
- All the command of DDL is auto-committed that means it permanently save all the changes in the database.

2. Data Manipulation Language (DML)

- DML commands are used to modify the database. It is responsible for all form of changes in the database.
- The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.

3. Data Control Language (DCL)

- DCL commands are used to grant and take back authority from any database user.

4. Transaction Control Language (TCL)

- TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only.
- These operations are automatically committed in the database that's why they cannot be used while creating tables or dropping them.

5. Data Query Language

- DQL is used to fetch the data from the database.

It uses only one command:

- **SELECT**