BEGINNER LEVEL

1.**What is postgresql?**

- *This is the most successful open source database in the world. This is also used to create advanced applications.*

2.**What is the Maximum size of the table in postgresql?**

- *The Mazimum size of the table in postgresql is 32TB*

3. **How to start the Database server*?***

- /usr/local/etc/rc.d/010.pgsql.sh start
- /usr/local/etc/rc.d/postgresql start

4. How to stop the Database server?

- /usr/local/etc/rc.d/010.pgsql.sh stop
- /usr/local/etc/rc.d/postgresql stop

5. How to check Whether Postgresql Server is Up And Running?

- /usr/local/etc/rc.d/010.pgsql.sh status
- /usr/local/etc/rc.d/postgresql status

6. What are the languages postgresql supports?

- *It supports a language of its own known as PL/pgSQL and it supports internal procedural languages.*

7. How to start the PostgreSQL database?

 # service postgresql start

Starting PostgreSQL: ok

8. How to stop the PostgreSQL database?

 # service postgresql stop

Stopping PostgreSQL: server stopped ok

9. How to restart the PostgreSQL database?

 # service postgresql restart

Restarting PostgreSQL: server stopped ok

*10.*How to create the PostgreSQL Database?

- *Creating the database in the PSQL prompt, with createuser command.*

 # CREATE DATABASE mydb WITH OWNER vishal;
CREATE DATABASE

11.How to I get the list of databases in PostgreSQL database?

```
# \l
List of databases
Name | Owner | Encoding
————————-+————————-+————————-
backup | postgres | UTF8
mydb | vishal | UTF8
postgres | postgres | UTF8
template0 | postgres | UTF8
template1 | postgres | UTF8
```

12.How to Delete/Drop an exixtingPostgreSQL database?

 # \l
List of databases
Name | Owner | Encoding
———-+———-+———-
backup | postgres | UTF8
mydb | ramesh | UTF8
postgres | postgres | UTF8
template0 | postgres | UTF8
template1 | postgres | UTF8

- *# DROP DATABASE mydb;*
  DROP DATABASE

13.What is Table?

- *A Table refers to a collection of data in an organised manner in forms of rows and column*
  Eg:Table: StudentInformation

14.What is Field?

- *A fields refers to the number of columns in the table.*

Field: Stu Id, Stu Name, Stu Marks

15. What are joins in SQL?

A JOIN clause is used to combine rows from two or more tables, based on a related column between them. It is used to merge two tables or retrieve data from there. There are 4 joins in SQL namely:

- *Inner Join*
- *Right Join*
- *Left Join*
- *Full Join*

16.What is a Primary Key?

- A Primary key is a column (or collection of columns) or a set of columns that uniquely identifies each row in the table.
- Uniquely identifies a single row in the table
- Null values not allowed

Student Table	
Stu_ID	Stu_Name
1	John
2	Jack
3	Tyler
4	Sofia

Example: In the Student table, Stu_ID is the primary key.

17.What are Constraints?

- *Constraints are used to specify the limit on the data type of the*

table. It can be specified while creating or altering the table statement. The sample of constraints are:

- *NOT NULL*
- *CHECK*
- *DEFAULT*
- *UNIQUE*
- *PRIMARY KEY*
- *FOREIGN KEY*

*18.***What is meant by DELETE Query?**

- *PostgreSQL* ***DELETE*** *Query is used to delete the existing records from a table. You can use WHERE clause with DELETE query to delete selected rows. Otherwise, all the records would be deleted.*

19.What is meant by Truncate?

- *It removes all rows from a set of tables. It has the same effect as an unqualified DELETE on each table, but since it does not actually scan the tables it is faster. Furthermore,*
- *it reclaims disk space immediately, rather than requiring a subsequent VACUUM operation. This is most useful on large tables.*

20.What is the difference between DELETE and TRUNCATE?

DELETE	TRUNCATE
Delete command is used to delete a row in a table.	*Truncate is used to delete all the rows from a*

	table.
You can rollback data after using delete statement.	*You cannot roll back data*
It is a DML command	*It is a DDL command.*
It is slower than truncate statement.	*It is faster.*

21.What is a Unique key?

- *Uniquely identifies a single row in the table.*
- *Multiple values allowed per table.*
- *Null values allowed.*

22.What is a Foreign key?

- *It maintains referential integrity by enforcing a link between the data in two tables.*
- *The foreign key in the child table references the primary key in the parent table.*
- *It constraint prevents actions that would destroy links between the child and parent tables.*

23.What do you mean by data integrity?

- *Data Integrity defines the accuracy as well as the consistency of the data stored in a database.*

24.. Write a SQL query to display the current date?

- *In SQL, there is a built-in function called GetDate() which helps to return the current timestamp/date.*

25. What is an Index?

- *An index refers to a performance tuning method of allowing faster retrieval of records from the table.*
- *An index creates an entry for each value and hence it will be faster to retrieve data.*

26. What do you mean by Denormalization?

- *Denormalization refers to a technique which is used to access data from higher to lower forms of a database.*
- *It helps the database managers to increase the performance of the entire infrastructure as it introduces redundancy into a table.*

27.What are Entities?

- *A person, place, or thing in the real world about which data can be stored in a database. Tables store data that represents one type of entity.*

- *For example – A bank database has a customer table to store customer information.*
- *Customer table stores this information as a set of attributes (columns within the table) for each customer.*

28.What is Relationship?

- *Relation or links between entities that have something to do with each other*
- *For example – The customer name is related to the customer account number and contact information, which might be in the same table.*
- *There can also be relationships between separate tables (for*

29.What is Normalization?

- *Normalization is the process of organizing data to avoid duplication and redundancy.*
- *Undesirable characteristics like Insertion, Update and Deletion Anomalies.*
- *It is a multi-step process that puts data into tabular form, removing duplicated data from the relation tables.*

30.What is the difference between DROP &TRUNCATE command?

- *DROP command removes a table and it cannot be rolled back from the database.*
- *TRUNCATE command removes all the rows from the table.*

31. What is ACID property in a database?

- *Atomicity, Consistency, Isolation, Durability.*
- *It is used to ensure that the data transactions are processed reliably in a database system.*

32. What is Atomicity?

- *Atomicity refers to the transactions that are completely done or failed where transaction refers to a single logical operation of a data.*
- *It means if one part of any transaction fails*
- *The entire transaction fails and the database state is left unchanged.*

33. What is consistency?

- *Consistency ensures that the data must meet all the validation rules. In simple words*
- *you can say that your transaction never leaves the database without completing its state.*

34. What is isolation?

- *This property ensures that multiple transactions can occur concurrently without leading to the inconsistency of database state.*

35. What is Durability?

- *Durability means that if a transaction has been committed,*

- *it will occur whatever may come in between such as power loss, crash or any sort of error.*

36. What do you mean by "Trigger" in SQL?

- *Trigger in SQL is are a special type of stored procedures that are defined to execute automatically in place or after data modifications.*
- *It allows you to execute a batch of code when an insert, update or any other query is executed against a specific table.*

37. What are the different operators available in SQL?

- *There are 3 operators available in SQL, namely:*
- *Arithmetic Operators*

- *Logical Operators*
- *Comparison Operators*

38. What are the most important commands in postgresql?

- *SELECT*
- *UPDATE*
- *DELETE*
- *INSERT INTO*
- *CREATE DATABASE*
- *ALTER DATABASE*
- *CREATE TABLE*
- *ALTER TABLE*
- *DROP TABLE*
- *CREATE INDEX*
- *DROP INDEX*

39. What the SELECT Statement can do?

- ***SELECT*** *is the most common statement used, amd it allows us to retrieve information from table.*

40. What is the syntax for SELECT Statement?(For ex we can take table t1 here)

- SELECT * FROM t1

41. Why we use INSERT INTO Statement?

- *The INSERT INTO statement is used to insert new records in a table.*

42. What is the syntax for INSERT INTO Statement?

- *INSERT INTO table name VALUES(value1,value2,value3,...);*

43. What is meant by SELECT DISTINCT Statement?

- *The SELECT DISTINCT statement is used to return only distinct (different) values.*
- *Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.*

44. What is the Syntax for SELECT DISTINCT Statement?

- *SELECT DISTINCT column1,column2,...*

  FROM table name;

45. What is COUNT()?

- *The COUNT() function returns the number of rows that matches a specified criteria.*

46. What is AVG()?

- *The AVG() function returns the average value of a numeric column.*

47. What is SUM()?

- *The SUM() function returns the total sum of a numeric column.*

48. What is meant by LIKE Operator?

- *The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.*
- *There are two wildcards often used in conjunction with the LIKE operator:*

- *% - The percent sign represents zero, one, or multiple characters*
- *_ - The underscore represents a single character*

49. What is meant by IN operator?

- *The IN operator allows you to specify multiple values in a WHERE clause.*
- *The IN operator is a shorthand for multiple OR conditions.*

50. What is a NULL Value?

- *A field with a NULL value is a field with no value.*
- *If a field in a table is optional, it is possible to insert a new record or update a record without adding a value to this field.*

- *Then, the field will be saved with a NULL value.*

INTERMEDIATE LEVEL

1. **What is inner Join?**

- *Inner Join is the most common type of join.*
- *It is used to return all the rows from multiple tables where the join condition is satisfied.*

2. **What is Right Join?**

- *Right Join is used to return all the rows from the right table*
- *but only the matching rows from the left table where the join condition is fulfilled.*

3. **What is Left Join?**

- *Left Join is used to return all the rows from the left table*
- *but only the matching rows from the right table where the join condition is fulfilled.*

4. **What is Full Join?**

- *Full join returns all the records when there is a match in any of the tables.*
- *Therefore, it returns all the rows from the left-hand side table and all the rows from the right-hand side table.*

5. What is meant by NOT NULL Constraint?

- *The NOT NULL constraint enforces a column to NOT accept NULL values.*

- *The NOT NULL constraint enforces a field to always contain a value.*
- *This means that you cannot insert a new record, or update a record without adding a value to this field.*

6. What is meant by CHECK Constraints?

- *The CHECK constraint is used to limit the value range that can be placed in a column.*
- *If you define a CHECK constraint on a single column it allows only certain values for this column.*
- *If you define a CHECK constraint on a table it can limit the values in certain columns*

based on values in other columns in the row.

7. What is DEFAULT Constraints?

- *The DEFAULT constraint is used to insert a default value into a column.*
- *The default value will be added to all new records, if no other value is specified.*

8. What is meant by UNIQUE Constraints?

- *The UNIQUE constraint uniquely identifies each record in a database table.*
- *The UNIQUE and PRIMARY KEY constraints both provide a*

guarantee for uniqueness for a column or set of columns.

- *A PRIMARY KEY constraint automatically has a UNIQUE constraint defined on it.*

9. What is meant by PRIMARY KEY Constraints?

- *The PRIMARY KEY constraint uniquely identifies each record in a database table.*
- *Primary keys must contain UNIQUE values.*
- *A primary key column cannot contain NULL values.*

10. **What is meant by FOREIGN KEY Constraints?**

- *A FOREIGN KEY in one table points to a PRIMARY KEY in another table.*
- *The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables.*

11. What is Unique Index?

- *This index does not allow the field to have duplicate values if the column is unique indexed.*
- *If a primary key is defined, a unique index can be applied automatically.*

12. What is meant by clusterd index?

- *This index reorders the physical order of the table and searches based on the basis of key values.*
- *Each table can only have one clustered index.*

13. What is meant by Non-Clusterd index?

- *Non-Clustered Index does not alter the physical order of the table and maintains a logical order of the data.*
- *Each table can have many nonclustered indexes.*

14. **What is meant by Modification State indexes?**

- *A modification state index is a system-generated index*

- *that is used in the implementation of currently committed semantics for index scans on column-organized tables.*

15. What is meant by Bidirectional Indexes?

- *Bidirectional indexes allow scans in both the forward and reverse directions.*

16. What is meant by Expression Based indexes?

- *With expression-based indexes, you can create an index that includes expressions.*
- *The performance of queries that involves expressions is improved if the database manager chooses an*

index that is created on the same expressions.

17. What is the advantages of Normalization?

- ***Better Datab****ase organization*
- *More Tables with smaller rows*
- *Efficient data access*
- *Greater Flexibility for Queries*
- *Quickly find the information*

18. What is the types of Arithmetic Operators in Postgresql?

- *Addition(+)*
- *Subtraction(-)*
- *Multiplication(*)*
- *Division(/)*
- *Modulo(%)*

19. **What is VARBINARY(size)?**

- *Equal to VARCHAR(), but stores binary byte strings.*
- *The size parameter specifies the maximum column length in bytes.*

20. What is menat by VHARCHAR(size)?

- *A VARIABLE length string (can contain letters, numbers, and special characters).*
- *The size parameter specifies the maximum column length in characters - can be from 0 to 65535.*

21. What is CHAR(size)?

- *FIXED length string(can contain letters,numbers,and special characters).*

- *The Size parameters specifies the column length in charcters-can be from 0 to 255.*
- *The Default is 1.*

22. What is SET(val1,val2,val3,…)?

- *A string object that can have 0 or more values, chosen from a list of possible values.*
- *You can list up to 64 values in a SET list.*

23. **What is meant by BIT(size)?**

- *A bit-value type. The number of bits per value is specified in size.*
- *The size parameter can hold a value from 1 to 64. The default value for size is 1.*

24. What is meant by DATE data type?

- *A date.Format:YYYY-MM-DD*
- *The Supported range is from '1000-01-01' to '9999-12-31.*

25. What is meant by TIME(fsp)?

- *A time. Format: hh:mm:ss. The supported range is from '-838:59:59' to '838:59:59'*

26. What is meant by sql_variant?

- *Stores upto 8000 bytes of data of variuos data types,except text,ntext,timestamp*

27. What is uniqueidentifier?

- *It Stores a globally unique identifier (GUID)*

28. What is meant by cursor?

- *Cursor Stores a reference to a cursor used for database operations.*

29.What is subquery in SQL?

- *A subquery is a query inside another query.*
- *where a query is defined to retrieve data or information back from the database.*
- *In a subquery, the outer query is called as the main query whereas the inner query is called subquery.*

30. What are the different types of a subquery?

- *There are two types of subquery*
- ***Correlated subquery**: These are queries which select the data from a table referenced in the outer query*
- *It is not considered as an independent query as it refers to another table and refers the column in a table.*
- ***Non-Correlatedsubquery**: This query is an independent query*
- *where the output of subquery is substituted in the main query.*

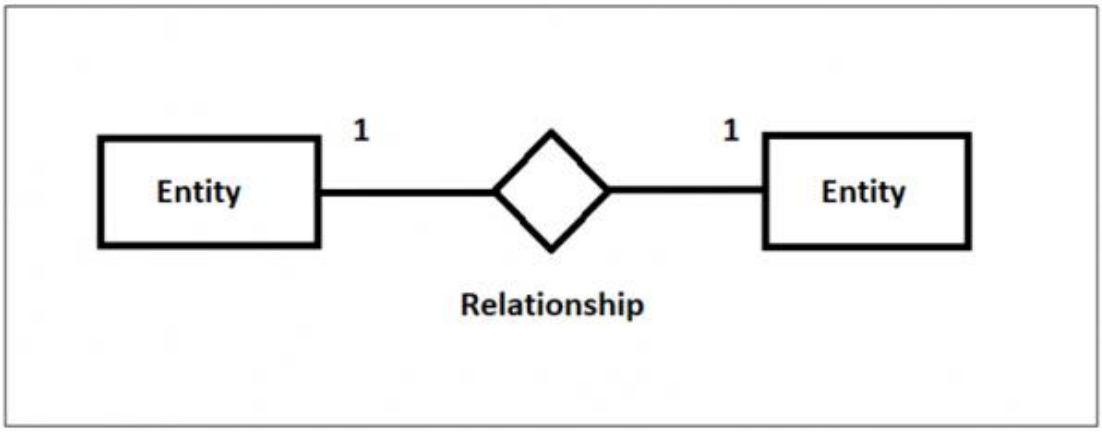
31.Definition - What does *One-to-Many Relationship* mean?

- *In relational databases, a one-to-many relationship occurs when a parent record in one table can potentially reference several child records in another table.*
- *In a one-to-many relationship, the parent is not required to have child records*
- *therefore, the one-to-many relationship allows zero child records, a single child record or multiple child records.*
- *The important thing is that the child cannot have more than one parent record.*
- *The opposite of a one-to-many relationship is a many-to-many*

relationship, in which a child record can link back to several parent records.

32.What is meant by One-to-one relationship?

- ***One-to-One relationship*** *in DBMS is a relationship between an instance of an entity with another*



33. What is meant by Many to Many realtionship?

- *Multiple records in one table are related to multiple records in another table.*

34. What is the main difference between 'BETWEEN' and 'IN' condition operators?

- *BETWEEN operator is used to display rows based on a range of values in a row whereas the IN condition operator is used to check for values contained in a specific set of values.*
- ***Example of BETWEEN:***
- *SELECT * FROM Students where ROLL_NO BETWEEN 10 AND 50;*
- ***Example of IN:***
- *SELECT * FROM students where ROLL_NO IN (8,15,25);*

35.What do you mean by recursive stored procedure?

- *Recursive stored procedure refers to a stored procedure which calls by itself until it reaches some boundary condition.*
- *This recursive function or procedure helps the programmers to use the same set of code n number of times.*

36.What are the various levels of constraints?

- *They are tht representation of a column to enforce data entity and consistency.*
- *There are two levels of a constraint, namely:*
- *column level constraint*
- *table level constraint*

37.How can you fetch alternate records from a table?

- *You can fetch alternate records i.e both odd and even row numbers.*
- *For example- To display even numbers, use the following command:*
- *Select studentId from (Select rowno, studentId from student) where mod(rowno,2)=0*
- *Now, to display odd numbers:*
- *Select studentId from (Select rowno, studentId from student) where mod(rowno,2)=1*

38.How can you select unique records from a table?

- *You can select unique records from a table by using the DISTINCT keyword.*

- *Select DISTINCT studentID from Student*
- *Using this command, it will print unique student id from the table Student.*

39.How can you fetch first 5 characters of the string?

- *There are a lot of ways to fetch characters from a string. For example:*
- *Select SUBSTRING(StudentName,1,5) as studentname from student*

40.What is a View?

- *A view is a virtual table which consists of a subset of data contained in a table. Since views are not present, it takes less space to store.*

- *View can have data of one or more tables combined and it depends on the relationship.*

41.What are Views used for?

- *A view refers to a logical snapshot based on a table or another view. It is used for the following reasons:*
- *Restricting access to data.*
- *Making complex queries simple.*
- *Ensuring data independence.*
- *Providing different views of same data.*

42. What do you mean by Collation?

- *Collation is defined as a set of rules that determine how data can be sorted as well as compared. Character data is sorted using the rules that define the correct*

character sequence along with options specifying case-sensitivity, character width etc.

43. What is a Datawarehouse?

- *Datawarehouse refers to a central repository of data where the data is assembled from multiple sources of information.*
- *Those data are consolidated, transformed and made available for the mining as well as online processing.*
- *Warehouse data also have a subset of data called Data Marts*

44. What is GROUP BY Statement?

- *The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".*

- *The GROUP BY statement is often used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to group the result-set by one or more columns.*

45. What is the syntax for GROUP BY Statement?

```
SELECT column1, column2
FROM table_name
WHERE [ conditions ]
GROUP BY column1, column2
ORDER BY column1, column2
```

46. What is meant by ORDER BY Statement?

- ***ORDER BY*** *clause is used to sort the data in ascending or descending order, based on one or more columns.*

- *Some databases sort the query results in an ascending order by default.*

47.What is the syntax for ORDER BY Statement?

```
SELECT column-list
FROM table_name
[WHERE condition]
[ORDER BY column1, column2, ..
columnN] [ASC | DESC];
```

48.What is Temporary Tables?

- *There are RDBMS, which support temporary tables. Temporary Tables are a great feature that lets you **store and process intermediate results** by*

using the same selection, update, and join capabilities that you can use with typical SQL Server tables.

- *The temporary tables could be very useful in some cases to keep temporary data. The most important thing that should be known for temporary tables is that they will be deleted when the current client session terminates.*

49. What is LIKE clause?

- **LIKE** clause is used to compare a value to similar values using wildcard operators. There are two wildcards used in conjunction with the LIKE operator.
- The percent sign (%)
- The underscore (_)

- The percent sign represents zero, one or multiple characters. The underscore represents a single number or character. These symbols can be used in combinations.

50.What is the Syntax for LIKE clause?

```
SELECT FROM table_name
WHERE column LIKE 'XXXX%'

or

SELECT FROM table  name
WHERE column LIKE '%XXXX%'

or

SELECT FROM table  name
WHERE column LIKE 'XXXX  '

or

SELECT FROM table  name
WHERE column LIKE '  XXXX'

or

SELECT FROM table  name
WHERE column LIKE '_XXXX_'
```

51.What is known as UPDATE Query?

- **UPDATE** Query is used to modify the existing records in a table.
- You can use the WHERE clause with the UPDATE query to update the selected rows, otherwise all the rows would be affected.

EXPERT LEVEL QUESTIONS

1. What is meant by NOT NULL Constraint?

- **NOT NULL** constraint restricts a column from having a NULL value.
- Once **NOT NULL** constraint is applied to a column, you cannot pass a null value to that column.
- It enforces a column to contain a proper value.
- One important point to note about this constraint is that it cannot be defined at table level.

Example using `NOT NULL` constraint

```
CREATE TABLE Student(s_id int NOT NULL, Name varchar(60), Age int);
```

- The above query will declare that the **s_id** field of **Student** table will not take NULL value.

2. What is meant by UNIQUE Constraint?

- **UNIQUE** constraint ensures that a field or column will only have unique values.
- A **UNIQUE** constraint field will not have duplicate data.
- This constraint can be applied at column level or table level.

Using `UNIQUE` constraint when creating a Table (Table Level)

- Here we have a simple CREATE query to create a table, which will have a column **s_id** with unique values.

```
CREATE TABLE Student(s_id int NOT NULL UNIQUE, Name varchar(60), Age int);
```

- The above query will declare that the **s_id** field of **Student** table will only have unique values and wont take NULL value.

Using `UNIQUE` constraint after Table is created (Column Level)

```
ALTER TABLE Student ADD UNIQUE(s_id);
```

- The above query specifies that **s_id** field of **Student** table will only have unique value.

3. What is meant by Primary Key Constraint?

- Primary key constraint uniquely identifies each record in a database.
- A Primary Key must contain unique value and it must not contain null value.
- Usually Primary Key is used to index the data inside the table.

Using PRIMARY KEY constraint at Table Level

```
CREATE table Student (s_id int PRIMARY KEY, Name varchar(60) NOT NULL, Age int);
```

- The above command will creates a PRIMARY KEY on the s_id.

Using PRIMARY KEY constraint at Column Level

```
ALTER table Student ADD PRIMARY KEY (s_id);
```

- The above command will creates a PRIMARY KEY on the s_id.

4. What is meant by FOREIGN KEY Constraint?

- FOREIGN KEY is used to relate two tables.
- FOREIGN KEY constraint is also used to restrict actions that would destroy links between tables.

Customer_Detail Table

c_id	Customer_Name	address
101	Adam	Noida
102	Alex	Delhi
103	Stuart	Rohtak

Order_Detail Table

Order_id	Order_Name	c_id
10	Order1	101
11	Order2	103

12	Order3	102

- In **Customer_Detail** table, **c_id** is the primary key which is set as foreign key in **Order_Detail** table.
- The value that is entered in **c_id** which is set as foreign key in **Order_Detail** table must be present in **Customer_Detail** table where it is set as primary key.
- This prevents invalid data to be inserted into **c_id** column of **Order_Detail** table.
- If you try to insert any incorrect data, DBMS will return error and will not allow you to insert the data.

Using FOREIGN KEY constraint at Table Level

```
CREATE table Order_Detail(
   order_id int PRIMARY KEY,
   order_name varchar(60) NOT NULL,
   c_id int FOREIGN KEY REFERENCES Customer_Detail(c_id));
```

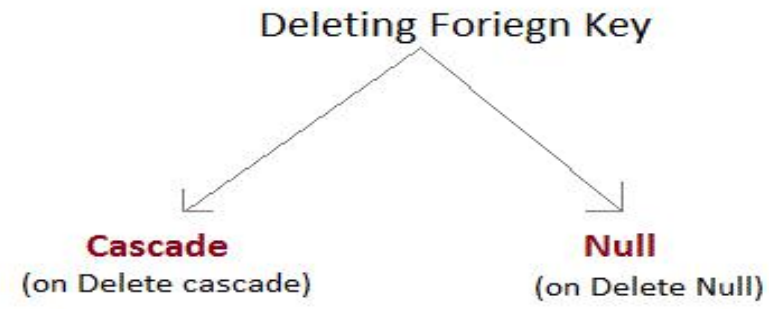
- In this query, **c_id** in table Order_Detail is made as foriegn key, which is a reference of **c_id** column in Customer_Detail table.

Using FOREIGN KEY constraint at Column Level

```
ALTER table Order_Detail ADD FOREIGN KEY (c_id) REFERENCES Customer_Detail(c_id);
```

- **5.What is meant by Behaviour of Foriegn Key Column on Delete**
- There are two ways to maintin the integrity of data in Child table, when a particular record is deleted in the main table.
- When two tables are connected with Foriegn key, and certain data in the main table is deleted, for which a record exits in the child table, then we must have some

mechanism to save the integrity of data in the child table.



- **On Delete Cascade :** This will remove the record from child table, if that value of foriegn key is deleted from the main table.
- **On Delete Null :** This will set all the values in that record of child table as NULL, for which the value of foriegn key is deleted from the main table.
- If we don't use any of the above, then we cannot delete data from the main table for which data in

child table exists. We will get an error if we try to do so.

```
ERROR : Record in child table exist
```

6. What is meant by CHECK Constraint?

- **CHECK** constraint is used to restrict the value of a column between a range.
- It performs check on the values, before storing them into the database.
- Its like condition checking before saving data into a column.

Using `CHECK` constraint at Table Level

```
CREATE table Student(
    s_id int NOT NULL CHECK(s_id > 0),
    Name varchar(60) NOT NULL,
    Age int);
```

- The above query will restrict the **s_id** value to be greater than zero.

Using `CHECK` constraint at Column Level

```
ALTER table Student ADD CHECK(s_id > 0);
```

7. What is meant by Functions in Postgresql?

- It provides many built-in functions to perform operations on data.
- These functions are useful while performing mathematical calculations, string concatenations, sub-strings etc.
- Functions are divided into two categories,
- Aggregate Functions
- Scalar Functions

8.What is meant by Aggregate Functions?

- These functions **return a single value** after performing calculations on a group of values.

9.What is meant byAVG()function?

- Average returns average value after calculating it from values in a numeric column.

Its general **syntax** is

```
SELECT AVG(column_name) FROM table_name
```

Using AVG() function

Consider the following **Emp** *table*

eid	name	age	salary
401	Anu	22	9000
402	Shane	29	8000
403	Rohan	34	6000
404	Scott	44	10000
405	Tiger	35	8000

Query to find average salary will be

```
SELECT avg(salary) from Emp;
```

Result of the above query will be

avg(salary)
8200

10. What is meant by COUNT() Function?

Count returns the number of rows present in the table either based on some condition or without condition.

Its general **syntax** is

```
SELECT COUNT(column_name) FROM table-name
```

Using COUNT() function

Consider the following **Emp** table

eid	name	age	salary
401	Anu	22	9000
402	Shane	29	8000
403	Rohan	34	6000
404	Scott	44	10000
405	Tiger	35	8000

- Query to count employees, satisfying specified condition is,

```
SELECT COUNT(name) FROM Emp WHERE salary = 8000;
```

- Result of the above query will be,

count(name)
2

Example of COUNT(distinct)

Consider the following **Emp** table

eid	name	age	salary
401	Anu	22	9000
402	Shane	29	8000
403	Rohan	34	6000
404	Scott	44	10000
405	Tiger	35	8000

Query is

```
SELECT COUNT(DISTINCT salary) FROM emp;
```

- Result of the above query will be

count(distinct salary)
4

11. What is meant by FIRST() ?

- First function returns first value of a selected column
- **Syntax** for FIRST function is,

```
SELECT FIRST(column_name) FROM table-name;
```

Using FIRST() function

- Consider the following **Emp** table

eid	name	age	salary
401	Anu	22	9000
402	Shane	29	8000
403	Rohan	34	6000
404	Scott	44	10000
405	Tiger	35	8000

- Query will be

```
SELECT FIRST(salary) FROM Emp;
```

- And the result will be

first(salary)
9000

12.What is meant LAST()?

- LAST function returns the return last value of the selected column.

Syntax of LAST function is,

```
SELECT LAST(column_name) FROM table-name;
```

Using LAST() function

Consider the following **Emp** table

eid	name	age	salary
401	Anu	22	9000
402	Shane	29	8000
403	Rohan	34	6000
404	Scott	44	10000
405	Tiger	35	8000

Query will be

```
SELECT LAST(salary) FROM emp;
```

Result of the above query will be

last(salary)
8000

12. What is meant by MAX()?

- MAX function returns maximum value from selected column of the table.

Syntax of MAX function is,

```
SELECT MAX(column_name) from table-name;
```

Using MAX() function

Consider the following **Emp** table

eid	name	age	salary
401	Anu	22	9000
402	Shane	29	8000
403	Rohan	34	6000
404	Scott	44	10000
405	Tiger	35	8000

Query to find the Maximum salary will be,

```
SELECT MAX(salary) FROM emp;
```

Result of the above query will be

MAX(salary)
10000

13. What is meant by MIN()?

- MIN function returns minimum value from a selected column of the table.

Syntax for MIN function is

```
SELECT MIN(column_name) from table-name;
```

Using MIN() function

Consider the following **Emp** table

eid	name	age	salary
401	Anu	22	9000
402	Shane	29	8000
403	Rohan	34	6000
404	Scott	44	10000
405	Tiger	35	8000

Query to find minimum salary is,

```
SELECT MIN(salary) FROM emp;
```

Result will be

MIN(salary)
6000

14. What is meant by SUM()?

- SUM function returns total sum of a selected columns numeric values.

Syntax for SUM is

```
SELECT SUM(column_name) from table-name;
```

Using SUM() function

Consider the following **Emp** table

eid	name	age	salary
401	Anu	22	9000
402	Shane	29	8000
403	Rohan	34	6000
404	Scott	44	10000
405	Tiger	35	8000

Query to find sum of salaries will be

```
SELECT SUM(salary) FROM emp;
```

Result of above query is

SUM(salary)
41000

15. What is meant by UCASE()?

- UCASE function is used to convert value of string column to Uppercase characters.

Syntax of UCASE

```
SELECT UCASE(column_name) from table-name;
```

Using UCASE() function

Consider the following **Emp** table

eid	name	age	salary
401	anu	22	9000
402	shane	29	8000
403	rohan	34	6000
404	scott	44	10000
405	Tiger	35	8000

Query for using UCASE is

```
SELECT UCASE(name) FROM emp;
```

Result is

UCASE(name)
ANU
SHANE
ROHAN
SCOTT
TIGER

16. What is meant by LCASE()?

- LCASE function is used to convert value of string columns to Lowecase characters.

Syntax for LCASE is

```
SELECT LCASE(column_name) FROM table-name;
```

Using LCASE() function

Consider the following **Emp** table

eid	name	age	salary
401	Anu	22	9000
402	Shane	29	8000
403	Rohan	34	6000
404	SCOTT	44	10000
405	Tiger	35	8000

- Query for converting string value to Lower case is

```
SELECT LCASE(name) FROM emp;
```

Result will be

LCASE(name)
anu
shane
rohan
scott

tiger

17. What is meant by MID()?

- MID function is used to extract substrings from column values of string type in a table.

Syntax for MID function is,

```
SELECT MID(column_name, start, length) from table-name;
```

Using MID() function

Consider the following **Emp** table

eid	name	age	salary
401	anu	22	9000
402	shane	29	8000
403	rohan	34	6000
404	scott	44	10000
405	Tiger	35	8000

Query will be

```
SELECT MID(name,2,2) FROM emp;
```

Result will come out to be

MID(name,2,2)
nu
ha

oh
co
ig

18. What is meant by ROUND()?

- ROUND function is used to round a numeric field to number of nearest integer.
- It is used on Decimal point values.

Syntax of Round function is

```
SELECT ROUND(column_name, decimals) from table-name;
```

Using ROUND() function

Consider the following **Emp** table

eid	name	age	salary
401	anu	22	9000.67
402	shane	29	8000.98
403	rohan	34	6000.45
404	scott	44	10000
405	Tiger	35	8000.01

Query is

```
SELECT ROUND(salary) from emp;
```

Result will be

ROUND(salary)
9001
8001
6000
10000
8000

19. What is meant by Cross JOIN?

- This type of JOIN returns the cartesian product of rows from the tables in Join.
- It will return a table which consists of records which combines each row from the first table with each row of the second table.

Cross JOIN Syntax is

```
SELECT column-name-listFROM table-name1 CROSS JOIN table-name2;
```

Example of Cross JOIN

Following is the **class** table

ID	NAME
1	abhi
2	adam
4	alex

and the **class_info** table

ID	Address
1	DELHI
2	MUMBAI
3	CHENNAI

Cross JOIN query will be

```
SELECT * FROM

class CROSS JOIN class_info;
```

The resultset table will look like

ID	NAME	ID	Address
1	abhi	1	DELHI
2	adam	1	DELHI
4	alex	1	DELHI
1	abhi	2	MUMBAI
2	adam	2	MUMBAI

4	alex	2	MUMBAI
1	abhi	3	CHENNAI
2	adam	3	CHENNAI
4	alex	3	CHENNAI

- As you can see, this join returns the cross product of all the records present in both the tables.

20. What is meant by INNER Join?

- This is a simple JOIN in which the result is based on matched data as per the equality condition specified in the query.

Inner Join Syntax is

```
SELECT column-name-list FROM table-name1 INNER JOIN table-name2 WHERE table-name1.column-name =
table-name2.column-name;
```

Example of INNER JOIN

Consider a **class** table

ID	NAME
1	abhi
2	adam

3	alex
4	anu

and the **class_info** table

ID	Address
1	DELHI
2	MUMBAI
3	CHENNAI

Inner JOIN query will be

```
SELECT * from class INNER JOIN class_info where class.id = class_info.id;
```

The resultset table will look like

ID	NAME	ID	Address
1	abhi	1	DELHI
2	adam	2	MUMBAI
3	alex	3	CHENNAI

21. What is meant by Natural Join?

- Natural Join is a type of Inner join which is based on column having same name and same

datatype present in both the tables to be joined.

The syntax for Natural Join is

```
SELECT * FROM table-name1 NATURAL JOIN table-name2;
```

Example of Natural JOIN

Here is the **class** table

ID	NAME
1	abhi
2	adam
3	alex
4	anu

and the **class_info** table

ID	Address
1	DELHI
2	MUMBAI
3	CHENNAI

Natural join query will be

```
SELECT * from class NATURAL JOIN class_info;
```

The resultset table will look like

ID	NAME	Address
1	abhi	DELHI
2	adam	MUMBAI
3	alex	CHENNAI

- In the above example, both the tables being joined have **ID** column(same name and same data type)
- hence the records for which value of **ID** matches in both the tables will be the result of Natural Join of these two tables.

22.OUTER JOIN

- Outer Join is based on both matched and unmatched data.
- Outer Joins subdivide further into

- Left Outer Join
- Right Outer Join
- Full Outer Join

22. What is meant by Left Outer Join?

- The left outer join returns a resultset table with the **matched data** from the two tables and then
- The remaining rows of the **left** table and null from the **right** table's columns.

Syntax for Left Outer Join is

```
SELECT column-name-list FROM table-name1 LEFT OUTER JOIN table-name2ON table-name1.column-name =
table-name2.column-name;
```

- To specify a condition, we use the ON keyword with Outer Join.

Left outer Join Syntax

```
SELECT column-name-list FROM table-name1, table-name2 on table-name1.column-name =
table-name2.column-name(+);
```

Example of Left Outer Join

Here is the **class** table

ID	NAME
1	abhi
2	adam
3	alex
4	anu
5	ashish

and the **class_info** table

ID	Address
1	DELHI
2	MUMBAI
3	CHENNAI
7	NOIDA
8	PANIPAT

Left Outer Join query will be

```
SELECT * FROM class LEFT OUTER JOIN class_info ON (class.id = class_info.id);
```

The resultset table will look like

ID	NAME	ID	Address
1	abhi	1	DELHI
2	adam	2	MUMBAI
3	alex	3	CHENNAI
4	anu	null	null
5	ashish	null	null

23. What is meant by Right Outer Join?

- The right outer join returns a resultset table with the **matched data** from the two tables being joined
- Then the remaining rows of the **right** table and null for the remaining **left** table's columns.

Syntax for Right Outer Join is

```
SELECT column-name-list FROM table-name1 RIGHT OUTER JOIN table-name2 ON table-name1.column-name =
table-name2.column-name;
```

Right outer Join Syntax

```
SELECT column-name-list FROM table-name1, table-name2 ON table-name1.column-name(+) =
table-name2.column-name;
```

Example of Right Outer Join

Once again the **class** table

ID	NAME
1	abhi
2	adam
3	alex
4	anu
5	ashish

and the **class_info** table

ID	Address
1	DELHI
2	MUMBAI
3	CHENNAI
7	NOIDA
8	PANIPAT

Right Outer Join query will be

```
SELECT * FROM class RIGHT OUTER JOIN class_info ON (class.id = class_info.id);
```

The resultant table will look like

ID	NAME	ID	Address
1	abhi	1	DELHI
2	adam	2	MUMBAI
3	alex	3	CHENNAI
null	null	7	NOIDA
null	null	8	PANIPAT

24.Full Outer Join

- The full outer join returns a resultset table with the **matched data** of two table then remaining rows of both **left** table and then the **right** table.

Syntax of Full Outer Join is

```
SELECT column-name-list FROM table-name1 FULL OUTER JOIN table-name2ON table-name1.column-name =
table-name2.column-name;
```

Example of Full outer join is

The **class** table

ID	NAME
1	abhi
2	adam
3	alex
4	anu
5	ashish

and the **class_info** table,

ID	Address
1	DELHI
2	MUMBAI
3	CHENNAI
7	NOIDA
8	PANIPAT

Full Outer Join query will be like

```
SELECT * FROM class FULL OUTER JOIN class_info ON (class.id = class_info.id);
```

The resultset table will look like

ID	NAME	ID	Address
1	abhi	1	DELHI
2	adam	2	MUMBAI
3	alex	3	CHENNAI
4	anu	null	null
5	ashish	null	null
null	null	7	NOIDA
null	null	8	PANIPAT

25.Alias - AS Keyword

- **Alias** is used to give an alias name to a table or a column, which can be a resultset table too.
- This is quite useful in case of large or complex queries.
- Alias is mainly used for giving a short alias name for a column or a table with complex names.

Syntax of Alias for table names,

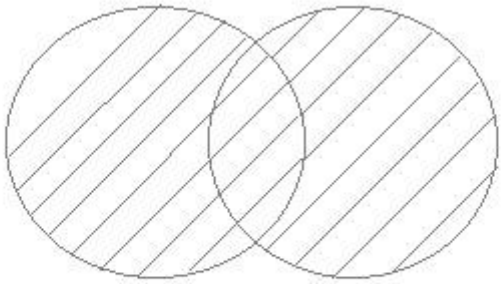
```
SELECT column-name FROM table-name AS alias-name
```

26.What are SET Operations?

- supports few Set operations which can be performed on the table data.
- These are used to get meaningful results from data stored in the table, under different special conditions.
- UNION
- UNION ALL
- INTERSECT
- MINUS

27. What is known as UNION Operation?

- Which combine the results of two more SELECT statements.
- However it will eliminate duplicate rows from its resultset.
- In case of union, number of columns and datatype must be same in both the tables
- on which UNION operation is being applied.



Example of UNION

The **First** table

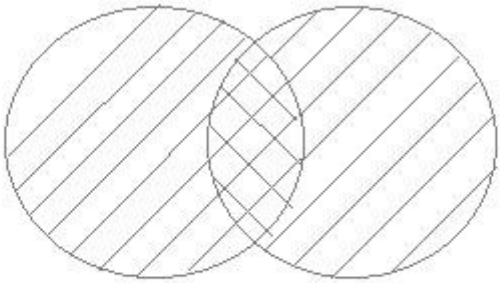
ID	Name
1	abhi
2	adam

The **Second** table

ID	Name
2	adam
3	Chester

28.What is meant by UNION ALL?

- This operation is similar to Union. But it also shows the duplicate rows.



Example of Union All

The **First** table

ID	NAME
1	abhi
2	adam

The **Second** table

ID	NAME
2	adam
3	Chester

Union All query will be like

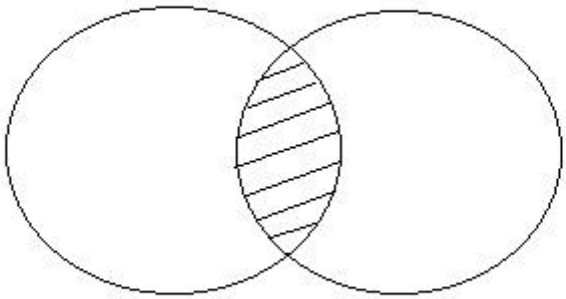
```
SELECT * FROM First UNION ALLSELECT * FROM Second;
```

The resultset table will look like

ID	NAME
1	abhi
2	adam
2	adam
3	Chester

29.What is known as INTERSECT?

- Intersect operation is used to combine two SELECT statements
- But it only retuns the records which are common from
- both SELECT statements. In case of **Intersect** the number of columns and datatype must be same.



Example of Intersect

The **First** table

ID	NAME
1	abhi
2	adam

The **Second** table

ID	NAME
2	adam
3	Chester

Intersect query will be

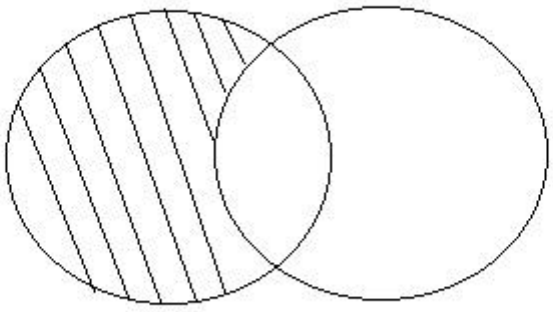
```
SELECT * FROM First INTERSECTSELECT * FROM Second;
```

The resultset table will look like

ID	NAME
2	adam

30.What is know as MINUS?

- The Minus operation combines results of two SELECT statements and return only those in the final result
- which belongs to the first set of the result.



Example of Minus

The **First** table

ID	NAME
1	abhi
2	adam

The **Second** table

ID	NAME
2	adam
3	Chester

Minus query will be

```
SELECT * FROM First

MINUSSELECT * FROM Second;
```

The resultset table will look like

ID	NAME
1	abhi

31.What is Sequence?

- **Sequence** is a feature supported by some database systems to produce unique values on demand. Some DBMS supports AUTO_INCREMENT in place of Sequence.
- AUTO_INCREMENT is applied on columns
- it automatically increments the column value by 1 each time a new record is inserted into the table
- Sequence is also similar to AUTO_INCREMENT but it has some additional features too.

32.How to create a Sequence?

Syntax to create a sequence is,

```
CREATE SEQUENCE sequence-name
    START WITH initial-value
    INCREMENT BY increment-value
    MAXVALUE maximum-value
    CYCLE | NOCYCLE;
```

- The **initial-value** specifies the starting value for the Sequence.
- The **increment-value** is the value by which sequence will be incremented.
- The **maximum-value** specifies the upper limit or the maximum value upto which sequence will increment itself.
- The keyword CYCLE specifies that if the maximum value exceeds the set limit, sequence will restart its cycle from the begining.

- And, NO CYCLE specifies that if sequence exceeds MAXVALUE value, an error will be thrown.

33.How to use the Sequence?

- Let's start by creating a sequence, which will start from 1, increment by 1 with a maximum value of 999.

```
CREATE SEQUENCE seq_1START WITH 1
INCREMENT BY 1
MAXVALUE 999
CYCLE;
```

- Now let's use the sequence that we just created above.

Below we have a **class** table,

ID	NAME
1	abhi

2	adam
4	alex

34.What is VIEW?

- A VIEW in is a logical subset of data from one or more tables.
- View is used to restrict data access.

Syntax for creating a View

```
CREATE or REPLACE VIEW view_name
    AS
    SELECT column_name(s)
    FROM table_name
    WHERE condition
```

- As you may have understood by seeing the above SQL query
- view is created using data fetched from some other table(s).

- It's more like a temporary table created with data.

34. How to create a VIEW?

- Consider following Sale table

oid	order_name	previous_balance	customer
11	ord1	2000	Alex
12	ord2	1000	Adam
13	ord3	2000	Abhi
14	ord4	1000	Adam
15	ord5	2000	Alex

- Query to Create a View from the above table will be

```
CREATE or REPLACE VIEW sale_view AS SELECT * FROM
Sale WHERE customer = 'Alex';
```

The data fetched from

- SELECT statement will be stored in another object called **sale_view**.We can use CREATE and REPLACE

- seperately too, but using both together works better
- as if any view with the specified name exists, this query will replace it with fresh data.

35.How to Display the Displaying a VIEW?

- The syntax for displaying the data in a view is similar to fetching data from a table using
- SELECT statement.

```
SELECT * FROM sale_view;
```

36.What is Force VIEW Creation

- FORCE keyword is used while creating a view, forcefully. This keyword is used to create a View even if the table does not exist.

- After creating a force View if we create the base table and enter values in it, the view will be automatically updated.

Syntax for forced View is

```
CREATE or REPLACE FORCE VIEW view_name AS
    SELECT column_name(s)
    FROM table_name
    WHERE condition;
```

36. How to update a VIEW?

- **UPDATE command for view is same as for tables.**

Syntax to Update a View is,

```
UPDATE view-name SET VALUEWHERE
condition;
```

NOTE:

If we update a view it also updates base table data automatically.

37.What is Read-Only VIEW?

- We can create a view with read-only option to restrict access to the view.
- Syntax to create a view with Read-Only Access

```
CREATE or REPLACE FORCE VIEW view_name AS
    SELECT column_name(s)
    FROM table_name
    WHERE condition WITH read-only;
```

- The above syntax will create view for **read-only** purpose
- we cannot Update or Insert data into read-only view.
- It will throw an **error**.

38.What are the Types of View?

- There are two types of view
- Simple View
- Complex View

Simple View	Complex View
Created from one table	Created from one or more table
Does not contain functions	Contain functions
Does not contain groups of data	Contains groups of data

39. What is meant by common table expression?

- such as WITH expression_name AS (...) SELECT ... and Subqueries such as SELECT ... FROM (SELECT ...) AS subquery_name are tools for

breaking up complex SQL queries, and sometimes the only way to achieve a goal.

- While CTEs are arguably easier to read than subqueries, in Postgres they are an “optimization fence”
- preventing the query optimizer from rewriting queries by moving constraints into or out of the CTE.

40.How to find the largest table in the postgreSQL database?

```
$ /usr/local/pgsql/bin/psql test

Welcome to psql 8.3.7, the PostgreSQL interactive terminal.



Type:  \copyright for distribution terms

       \h for help with SQL commands

       \? for help with psql commands

       \g or terminate with semicolon to execute query

       \q to quit



test=# SELECT relname, relpages FROM pg_class ORDER BY relpages DESC;              relname               |
relpages
```

```
------------$ /usr/local/pgsql/bin/psql test

Welcome to psql 8.3.7, the PostgreSQL interactive terminal.



Type:  \copyright for distribution terms

       \h for help with SQL commands

       \? for help with psql commands

       \g or terminate with semicolon to execute query

       \q to quit



test=# SELECT relname, relpages FROM pg_class ORDER BY relpages DESC;              relname              |
relpages

 pg_proc                          |       50

 pg_proc_proname_args_nsp_index   |       40

 pg_depend                        |       37

----------------------+----------

 pg_proc                          |       50

 pg_proc_proname_args_nsp_index   |       40

 pg_depend                        |       37

 pg_attribute                     |       30
```

- If you want only the first biggest table in the postgres database then append the above query with limit as:

```
# SELECT relname, relpages FROM pg_class ORDER BY relpages DESC limit 1; relname | relpages

---------+----------

 pg_proc |       50
```

```
(1 row)
```

- **relname** – name of the relation/table.
- **relpages** – relation pages (number of pages, by default a page is 8kb)
- **pg_class** – system table, which maintains the details of relations
- **limit 1** – limits the output to display only one row.

41.How to view the indexes of an existing postgreSQL table ?

```
Syntax: # \d table_name
```

- As shown in the example below, at the end of the output you will have a section titled as indexes, if you have index in that table.

- In the example below, table pg_attribute has two btree indexes. By default postgres uses btree index as it good for most common situations.

```
test=# \d pg_attribute   Table "pg_catalog.pg_attribute"

   Column     |   Type   | Modifiers

---------------+----------+-----------

 attrelid      | oid      | not null

 attname       | name     | not null

 atttypid      | oid      | not null

 attstattarget | integer  | not null

 attlen        | smallint | not null

 attnum        | smallint | not null

 attndims      | integer  | not null

 attcacheoff   | integer  | not null

 atttypmod     | integer  | not null

 attbyval      | boolean  | not null

 attstorage    | "char"   | not null

 attalign      | "char"   | not null

 attnotnull    | boolean  | not null

 atthasdef     | boolean  | not null

 attisdropped  | boolean  | not null

 attislocal    | boolean  | not null

 attinhcount   | integer  | not nullIndexes:
```

```
    "pg_attribute_relid_attnam_index" UNIQUE, btree (attrelid, attname)

    "pg_attribute_relid_attnum_index" UNIQUE, btree (attrelid, attnum)
```

42. How to display the plan by executing the query on the server side ?

- This executes the query in the server side, thus does not shows the output to the user.
- But shows the plan in which it got executed.

```
# EXPLAIN ANALYZE query;
```

43. What is meant by PostgreSQL FETCH Command to Limit Query Results?

- **In a very large database, there may be millions of records.**
- **Suppose we only want to look at a small sample of the intended**

query to check that the parameters are accurate.

- PostgreSQL provides the FETCH command for this purpose. The following query returns the first row of the table tbl_scores:

```
SELECT student_id, score

FROM tbl_scores

ORDER BY student_id

FETCH FIRST ROW ONLY;
```

44. What is Advanced Where Clause in Full Outer Join Query?

- The Where clause enables you to set conditions on the data to be returned in a query.
- To fetch a list of department names which have no students

listed, we can use the WHERE clause in this query:

```
SELECT student_name, department_name

FROM tbl_students e

FULL OUTER JOIN tbl_departments d ON
d.department_id = e.department_id

WHERE

 student_name IS NULL;
```

45.How to use the Advanced Subquery in PostgreSQL?

- **Rather than calculating an intermediate result, we can use nested queries in PostgreSQL. These are usually called subqueries.**
- In the previous example, we calculated the average test score in

our sample db. In the following example

- we will return all students whose scores are above average, by putting the average calculation in a subquery:

```
SELECT student_id, score

FROM tbl_scores

WHERE score &amp;amp;gt; (

SELECT AVG (score) FROM tbl_scores;

);
```

46. What is Querying Stats on the Postgre DB?

- **Important to the overall querying capability, PostgreSQL supports a set of calls to physical db properties.**

- **For example, suppose we want to know the largest table in our db. We can run this query:**

```
1SELECT relname, relpages FROM pg_class ORDER BY relpages
 DESC limit 1;
```

- **To understand the system level keywords in this query, have a look at this list:**

  - **relname – table name**
  - **relpages – number of pages**
  - **pg_class – system table names**
  - **limit – limits output to the first result**

47.What is injection?

- Injection is a code injection technique.
- It is the placement of malicious code in strings.

- Injection is one of the most common web hacking techniques.

48.What is having clause?

- HAVING filters records that work on summarized GROUP BY results.
- HAVING applies to summarized group records, whereas WHERE applies to individual records.
- Only the groups that meet the HAVING criteria will be returned.
- HAVING requires that a GROUP BY clause is present.
- WHERE and HAVING can be in the same query.

49. What is WHERE ANY ,ALL clause?

- ANY and ALL keywords are used with a WHERE or HAVING clause.
- ANY and ALL operate on subqueries that return multiple values.
- ANY returns true if any of the subquery values meet the condition.
- ALL returns true if all of the subquery values meet the condition.

50. What is WHERE EXISTS Statement?

- WHERE EXISTS tests for the existence of any records in a subquery.
- EXISTS returns true if the subquery returns one or more records.
- EXISTS is commonly used with correlated subqueries.