**1.What is Database?**

Ans: An organized collection of data. A method to manipulate and access the data.

**2.What is DBMS?**

Ans: DBMS is called Database Management system (Oracle, MySQL, PostgreSQL etc.).

**3.What is RDBMS?**

Ans: A type of database systems that stores data in structured tables (using row and columns) and uses SQL for managing and querying data. In short RDBMS - relational database management system.

**4.What is SQL?**

Ans: Structured Query Language Which is used to talk to our database. Ex: SELECT \* FROM hr;

**5.What is Command line interface for PostgreSQL?**

Ans: SQL shell (psql) in windows (‘\! cls’ to clear the cli,’\list’ to see all the list).

**6.How can we create database in PostgreSQL in CLI?**

Ans: CREATE DATABASE TEST;

**7.How to login via CLI?**

Ans: (psql -U postgres) U stand for username postgres/username then password. Must add the path location to the environment variable to access psql in windows CLI

**8.Quey for database list in PostgreSQL?**

Ans: SELECT datname FROM pg\_database;

**9.How to change database In PostgreSQL?**

Ans: \ c database name;

**10.What is table?**

Ans: Table is a collection of related data held in table format within database.

**11. Table View Query?**

Ans: select \* from person;

**12.Table insert Query?**

Ans: insert into person (id,name,city)

Values (101,’Raju’, ’Dhaka’);

**13.What is Table update Query?**

Ans: update person set name=’Raju’ where id=111;

**14. What is table delete Query?**

Ans: delete from person where name=’Raju’;

**15.Data types?**

Ans: DECIMAL(5,2) Example: 155.38☑, 119.12☑,28.15 ☑, 1150.15%⮾.

**16.What is Constraint?**

Ans: A constraint in PostgreSQL is a rule applied to a column. Ex: primary-key, not null, unique values etc.

**17. Set default value?**

Ans: create table customers

(acc\_no int primary key,

name varchar(100) not null,

acc\_type varchar(50) not null default ‘Savings’);

**18. How can we use Auto increment in PostgreSQL?**

Ans: Create table employees (

id serial/bigserial primary key,

firstname varchar(50),

lastname varchar(50),

);

**19.Table create SQL?**

Ans: create table employees (

emp\_id serial primary key,

fname varchar(50)not null,

lname varchar(50)not null,

email varchar(100) not null unique, dept varchar(50),

salary decimal(10,2) default 30000.00,

hire\_date date not null default current\_date);

**20.What is clauses?**

Ans: SQL Quary with condition called clauses. Ex: (Where, Distinct, Order By, Limit, Like).

**21.Find out only HR and IT dept details from employees table.**

**Ans:** select \* from employees where dept=’HR’ or dept=’IT’;

**22.Find employees whose salary is more than 50000.**

**Ans:** SELECT \* FROM employees where salary > 50000.00;

**23. What is relational operator in PostgreSQL?**

Ans: In PostgreSQL, relational operators (e.g., =, <>, >, <, >=, <=) are used to compare values in SQL queries, primarily in the WHERE clause, such as WHERE age > 30 AND salary >= 60000.

**24. What is Logical operator in PostgreSQL?**

Ans: In PostgreSQL, logical operators like **AND**, **OR**, and **NOT** are used to combine conditions in SQL queries, e.g., WHERE age > 30 AND salary > 50000.

**25.What will be the way to find out 3 department via SQL?**

**Ans:** select \* from employees where dept=’IT’ or dept=’HR’ or dept=Finance ;

/select \* from employees where in (‘IT’, ’HR’, ’Finance’);

**26.Find employees whose salary is more than 40000 and less than 61000.**

**Ans:**  SELECT \* FROM employees where salary between 40000.00 and 61000.00;

**27. How to find Unique values from column?**

**Ans:** select distinct dept from employees;

**28. How to find anyone by his names character?**

**Ans:** select \* from employees where fname like '%N%';(

* Starts with ‘A’: LIKE ‘A%’
* Ends with ‘A’: LIKE ‘%A’
* Contains(anywhere) with ‘a’: Like ‘%a%’
* Second character is ‘a’: LIKE ‘\_a%’
* Case-insensitive contains ‘john’: LIKE ‘%jhon%’

**29. How to find 2charecter departments name in employees table?**

**Ans:** select\* from employees where dept like ‘\_\_**’;**

**30.What is the use of Aggregate function?**

**Ans:** Aggregate function is used to calculate data (COUNT, SUM, AVG, MIN, MAX)

* Select count(emp\_id) from employees;
* Select sum(salary) from employees;
* Select avg(salary) from employees;
* Select min(salary) from employees;
* Select max(salary) from employees;

**31. How does work group by in SQL?**

Ans: In SQL, the GROUP BY clause groups rows with the same values in specified columns and allows aggregate functions (e.g., SUM, COUNT) to summarize data within each group. For example: SELECT dept, SUM (dept) AS department

FROM employees GROUP BY dept;

**32. How Can we find how many employees in each department in employees table?**

Ans: select dept, count(emp\_id) from employees’ group by dept;

**33.How can we find total salary by department in employees table?**

Ans: select dept, sum(salary) from employees group by dept;

**34.Use Of CONCAT in SQL?**

Ans: CONCAT is used to add to string column EX:

* select concat ('hello',' world');
* select fname||lname from employees;
* select fname||' '||lname from employees;

**35. What is the way to concat to string and add a separation in between in PostgreSQL?**

Ans:

* select Concat\_ws('\_ ',fname,lname) from employees;
* select Concat\_ws(':','one','two','three') from employees;

**36.Uses of substring?**

**Ans:**

* select substring('Hello buddy!',1,5);
* select substr('Hello buddy!',7,12)**;**

**37.Uses of Replace in PostgreSQL.**

**Ans:** REPLACE (str, from\_str, to\_str);

REPLACE (‘Hey Buddy’, ‘Hey’, ‘Hello’);

select dept, replace (dept, 'IT', 'TECH') from employees;

dept | replace

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

* IT | TECH
* HR | HR
* IT | TECH
* Finance | Finance
* HR | HR
* Marketing | Marketing
* IT | TECH
* IT | TECH
* Finance | Finance
* Marketing | Marketing

**38.Reverce function uses**

**Ans:** select reverse(‘hello’); result: olleH.

**39.Uses of length.**

**Ans:**  select fname, length(fname) from employees;

fname | length

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

Raj | 3

Priya | 5

Arjun | 5

Suman | 5

**40.Find out those employees whose first name length getter than 5 character?**

**Ans:** SELECT \*

FROM employees

WHERE LENGTH (first\_name) > 5;

**41.UPPER and LOWER uses.**

**Ans**: select upper/lower(fname) from employees;

**42.Uses of LEFT, RIGHT, TRIM, POSITION**

**Ans:**

* select left(fname,2) from employees;
* select right(fname,2) from employees;
* select trim (' Alright! '); ---------- Alright!
* Select position (‘Al’ in ‘Alright!’);

**43.Find Different type of department in database?**

**Ans:** select distinct (dept) from employees;

**44. Display records with high-low salary.**

**Ans;** select \* from employees order by salary desc;

**45. How to see only top 3 records from a table?**

**Ans:** select \* from employees limit 3;

**46. Show records where first name starts with letter ‘A’**

**Ans:** select \* from employees where fname like 'A%';

**47.Show records where length of the lname is 4 character.**

**Ans:**

* select \* from employees where length(lname)=4;
* select \* from employees where lname like ‘\_\_\_\_’;

**48. Find Total no. of employees in database?**

**Ans:** select count(emp\_id) from employees;

**49. Find no. of employees in each department.**

**Ans:** select dept, count(dept) from employees group by dept;

**50. Find lowest salary paying.**

**Ans:**

* select min(salary) from employees;
* select \* from employees order by salary limit 1;
* select \* from employees where salary=(select min(salary) from employees);

**51. Find highest salary paying.**

**Ans:**

* select max(salary) from employees;
* select \* from employees order by salary desc limit 1;
* **select \* from employees where salary= (select max(salary) from employees);**

**52. Find total salary paying in IT department?**

**Ans:**

* select dept, sum(salary) from employees group by dept having dept='IT';
* select sum(salary) from employees where dept='HR';

**53. Average salary paying in each department**

**Ans:** select dept, avg(salary) from employees group by dept;

**54. How to add column in database?**

**Ans:** ALTER TABLE person ADD COLUMN age INT**;**

**55. How to drop column in PostgreSQL database?**

**Ans:**  ALTER TABLE person DROP COLUMN age;

**56. How to add column in table?**

Ans: Alter table personal

add column gender varchar (50);

**57.How to drop column in table?**

Ans: Alter table personal

drop column gender;

**58.How to Rename Column name in table?**

Ans: Alter table personal

rename column name to fname;

**59.How to Rename a table name?**

**Ans:**

* ALTER TABLE personal RENAME TO person;
* RENAME TABLE personal TO person;

**60. How to change data type limit in column?**

**Ans:** Alter table personal\_info

alter column fname

set data type varchar (200);

**61. Use Case in PostgreSQL.**

**Ans:** select fname, salary,

Case

When salary >= 50000 then ‘high’

Else ‘low’

End as sal\_cat

From employees;

**62. Make 10% bonus for every employees.**

**Ans:** select fname, salary,

case when salary>0

then round(salary\*.10)

end as bonus

from employees;

**63. Group by and count by sal\_cat.**

**ANS:** select

case

when salary>50000 then 'High'

when salary between 48000 and 50000 then 'Mid'

else 'Low‘

end as sal\_cat, count(emp\_id)

from employees

group by sal\_cat;

sal\_cat | count

High | 5

Mid | 3

Low | 2

**64.Cross join.**

**Ans: select \* from customers cross join orders;**

**cust\_id | cust\_name | ord\_id | ord\_date | price | cust\_id**

**---------+-----------+--------+------------+--------+---------**

**1 | Raju | 1 | 2024-01-01 | 250.00 | 1**

**2 | Sham | 1 | 2024-01-01 | 250.00 | 1**

**3 | Paul | 1 | 2024-01-01 | 250.00 | 1**

**4 | Alex | 1 | 2024-01-01 | 250.00 | 1**

**1 | Raju | 2 | 2024-01-15 | 300.00 | 1**

**2 | Sham | 2 | 2024-01-15 | 300.00 | 1**

**3 | Paul | 2 | 2024-01-15 | 300.00 | 1**

**4 | Alex | 2 | 2024-01-15 | 300.00 | 1**

**1 | Raju | 3 | 2024-02-01 | 150.00 | 2**

**2 | Sham | 3 | 2024-02-01 | 150.00 | 2**

**3 | Paul | 3 | 2024-02-01 | 150.00 | 2**

**4 | Alex | 3 | 2024-02-01 | 150.00 | 2**

**1 | Raju | 4 | 2024-03-01 | 450.00 | 3**

**2 | Sham | 4 | 2024-03-01 | 450.00 | 3**

**3 | Paul | 4 | 2024-03-01 | 450.00 | 3**

**4 | Alex | 4 | 2024-03-01 | 450.00 | 3**

**1 | Raju | 5 | 2024-04-04 | 550.00 | 2**

**2 | Sham | 5 | 2024-04-04 | 550.00 | 2**

**3 | Paul | 5 | 2024-04-04 | 550.00 | 2**

**4 | Alex | 5 | 2024-04-04 | 550.00 | 2**

**(20 rows)**

**65. Inner join.**

**Ans: select \* from customers c**

**inner join**

**orders o**

**on c.cust\_id = o.cust\_id;**

**cust\_id | cust\_name | ord\_id | ord\_date | price | cust\_id**

**---------+-----------+--------+------------+--------+---------**

**1 | Raju | 1 | 2024-01-01 | 250.00 | 1**

**1 | Raju | 2 | 2024-01-15 | 300.00 | 1**

**2 | Sham | 3 | 2024-02-01 | 150.00 | 2**

**3 | Paul | 4 | 2024-03-01 | 450.00 | 3**

**2 | Sham | 5 | 2024-04-04 | 550.00 | 2**

**(5 rows)**

**66. Use of Foreign key.**

**Ans:** CREATE TABLE orders (

ord\_id SERIAL PRIMARY KEY,

ord\_date DATE NOT NULL,

price NUMERIC NOT NULL,

cust\_id INTEGER NOT NULL,

FOREIGN KEY (cust\_id) REFERENCES

customers (cust\_id)

);

**67.**