



cognizant

MOST REPEATED QUES
INTERVIEW QUES

SQL

Structure Query Language (SQL) is a programming language for storing , manipulating and retrieving data in database. **Database** is a system that allow users to store and organise data.

There are mainly 4 types of SQL commands:

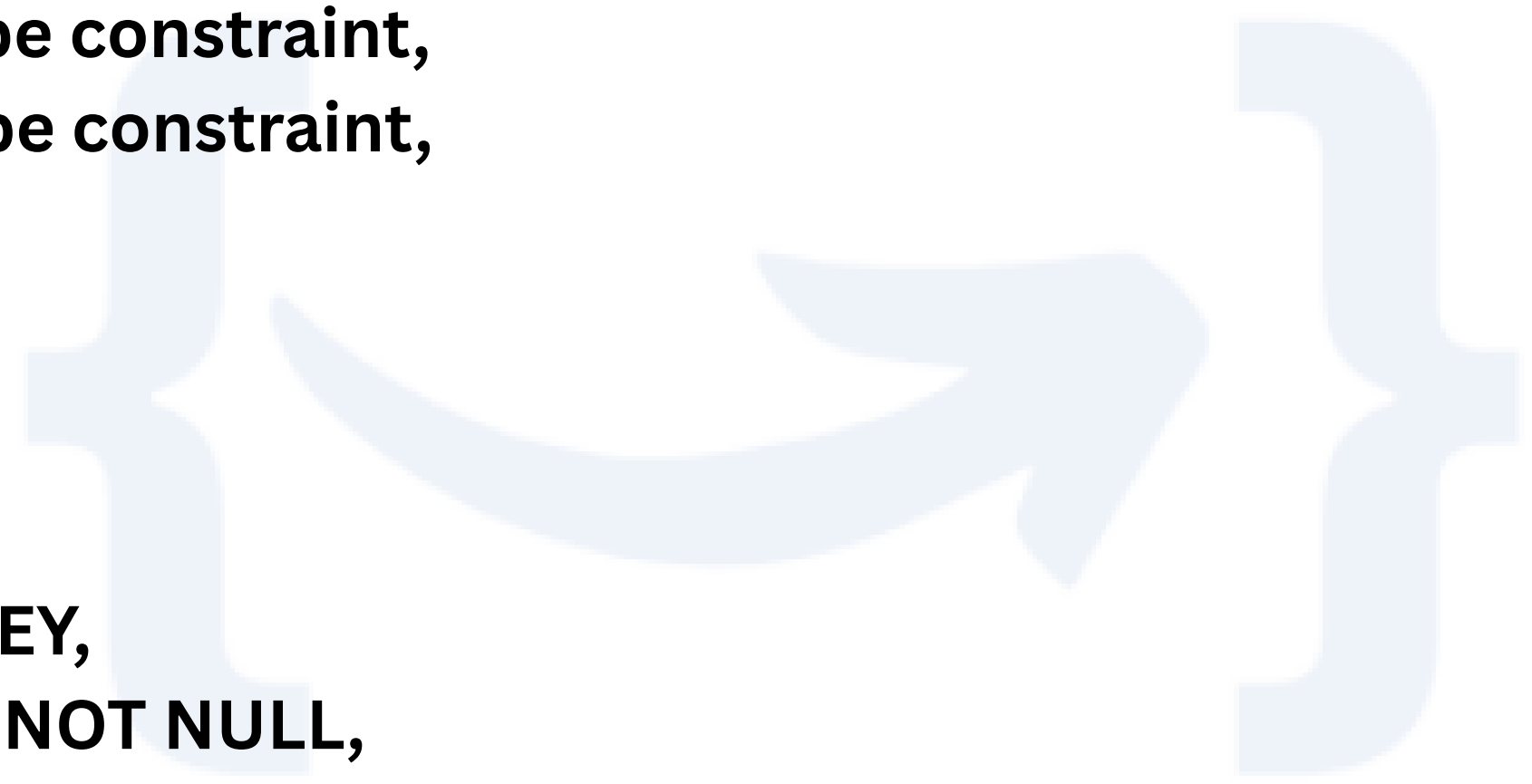
- **DDL (Data Definition Language):** Deals with database schemas and description.
Commands: create, alter,, drop,truncate, rename
- **DML (Data Manipulation Language):** Deals with data manipulaion. Commands: select, insert, update, and delete
- **DCL (Data Control Language):** Deals with permission and other control of database system. Commands: grant and revoke
- **TCL (Transaction Control Language):** Deals with a transaction within a database.
Commands: commit, rollback,savepoint

Create Table

The CREATE TABLE statement is used to create a new table in a database.

```
CREATE TABLE table_name  
(  
    column_name1 datatype constraint,  
    column_name2 datatype constraint,  
    column_name3 datatype constraint,  
);
```

```
CREATE TABLE customer  
(  
    CustID int8 PRIMARY KEY,  
    CustName varchar(50) NOT NULL,  
    Age int NOT NULL,  
    City char(50),  
    Salary numeric  
);
```



MASSIVE SUCCESS RATE



"Transform Your Interview Opportunity into an Offer Letter and Make Your Parents Proud!"

- In-depth Technical Mock
 - Crack coding challenges with real experts.
- HR & Managerial Prep
 - Master behavioral questions and impress TCS Interviewer.
- Full Interview Simulation
 - Ace both technical and HR in one session.
- Resume Review
 - Identify and fix weaknesses for a standout CV.
- Personalized Feedback & Expert Guidance
 - Tailored improvement tips to boost success.

www.primecoding.in

Insert Values In Table

```
INSERT INTO TABLE_NAME  
(column1, column2, column3, ..., columnN)  
VALUES  
(value1, value2, value3, ..., valueN);
```

```
INSERT INTO customer  
(CustID, CustName, Age, City, Salary)  
VALUES  
(1, 'Sham', 26, 'Delhi', 98000),  
(2, 'Laxman', 19, 'Bangalore', 116000),  
(3, 'Puja', 31, 'Mumbai', 60000),  
(4, 'Kitti', 42, 'Pune', 100000);
```

Update Values In Table

```
UPDATE TABLE_NAME SET  
"Column_name1" = 'value1',  
"Column_name2" = 'value2'  
WHERE "ID" = 'value
```

```
UPDATE customer SET  
CustName = 'Xam',  
Age= 32 WHERE  
CustID = 4;
```

ALTER Table

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

- **ALTER TABLE - ADD Column**

```
ALTER TABLE customer  
ADD COLUMN Gender varchar(10)
```

- **ALTER TABLE - MODIFY COLUMN**

```
ALTER TABLE customer  
ALTER COLUMN Gender char(10);
```

- **ALTER TABLE - DROP COLUMN**

```
ALTER TABLE customer  
DROP COLUMN Gender;
```

Delete Values In Table

The DELETE statement is used to delete existing records in a table

```
DELETE FROM table_name WHERE condition;
```

```
DELETE FROM customer WHERE CustID = 3
```

Drop & Truncate Table

The DROP TABLE command deletes a table in the database

```
DROP TABLE table_name;
```

The TRUNCATE TABLE command deletes the data inside a table, but not the table itself

```
TRUNCATE TABLE table_name
```

SELECT Statement

The SELECT statement is used to select data from a database.

```
SELECT column_name FROM table_name;
```

To select all the fields available in the table

```
SELECT * FROM table_name;
```

To select distinct/unique fields available in the table

```
SELECT DISTINCT Column_name FROM table_name;
```

WHERE Clause

The WHERE clause is used to filter records.

It is used to extract only those records that fulfill a specified condition

```
SELECT column_name FROM table_name WHERE conditions;
```

```
SELECT name FROM classroom WHERE grade='A';
```

LIMIT Clause

The LIMIT clause is used to set an upper limit on the number of tuples returned by SQL.

```
SELECT column_name FROM table_name LIMIT 5;
```

ORDER BY Clause

The ORDER BY clause is used to sort the result-set in ascending (ASC) or descending order (DESC).

```
SELECT column_name FROM table_name  
ORDER BY column_name ASC;
```


String Functions

String functions are used to perform an operation on an input string and return an output string.

- **UPPER():** Converts the value of a field to uppercase.
- **LOWER():** Converts the value of a field to lowercase.
- **LENGTH():** Returns the length of the value in a text field.
- **SUBSTRING():** Extracts a substring from a string.
- **NOW():** Returns the current system date and time. (Note: While often used in the context of string functions for formatting, NOW() actually returns a date/time value, not a string directly.)
- **FORMAT():** Used to set the format of a field.
- **CONCAT():** Adds two or more strings together.
- **REPLACE():** Replaces all occurrences of a substring within a string with a new substring.
- **TRIM():** Removes leading and trailing spaces (or other specified characters) from a string.

Aggregate Functions

Aggregate functions perform a calculation on multiple values and return a single value. Aggregate functions are often used with the GROUP BY and SELECT statements.

- **COUNT():** Returns the number of values.
- **SUM():** Returns the sum of all values.
- **AVG():** Returns the average value.
- **MAX():** Returns the maximum value.
- **MIN():** Returns the minimum value.
- **ROUND():** Rounds a number to a specified number of decimal places.

GROUP BY

The GROUP BY statement group rows that have the same values into summary rows. It is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result-set by one or more columns.

```
SELECT column_name(s) FROM table_name GROUP BY  
column_name(s);
```

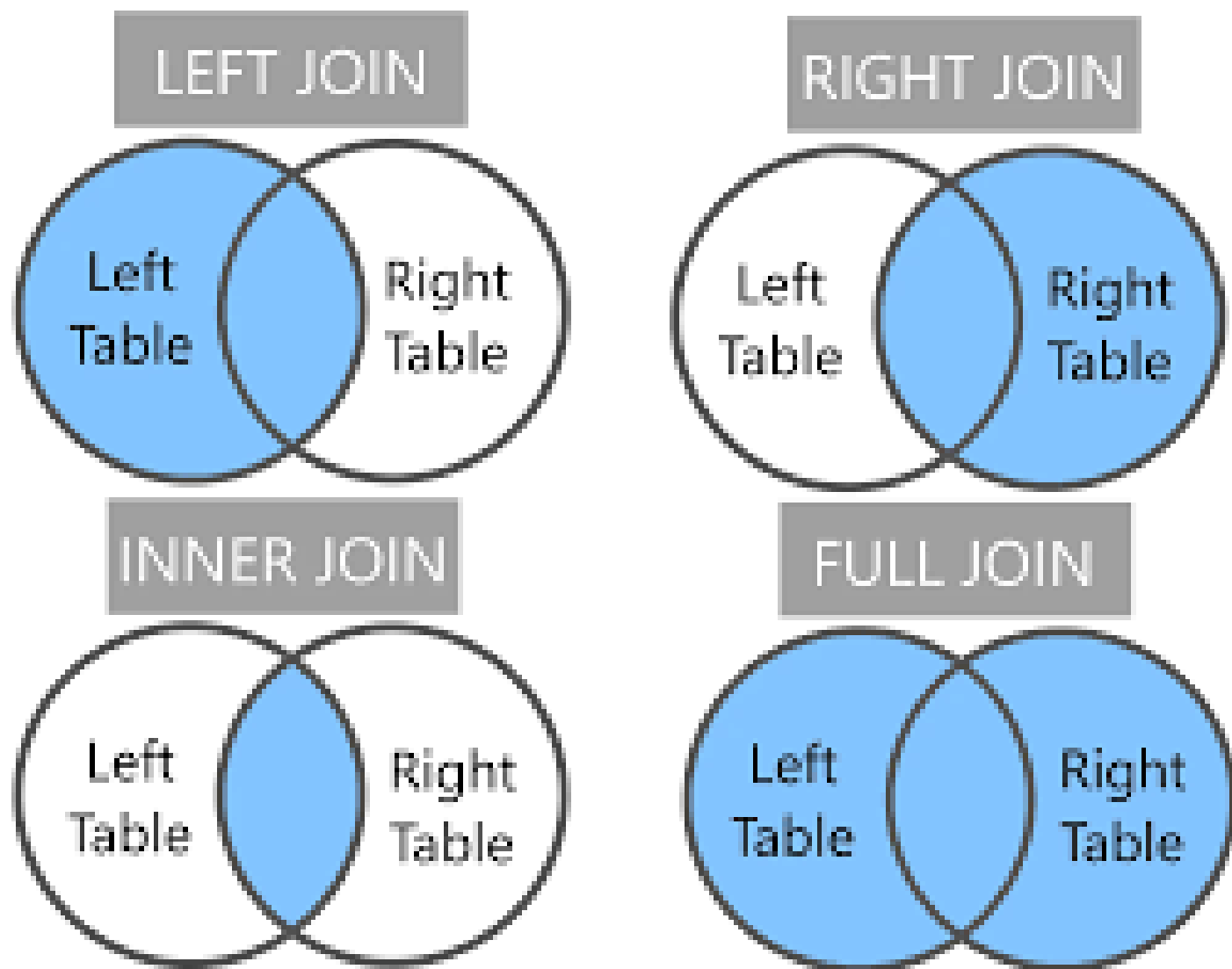
HAVING Clause

The HAVING clause is used to apply a filter on the result of GROUP BY based on the specified condition. The WHERE clause places conditions on the selected columns, whereas the HAVING clause places conditions on groups created by the GROUP BY clause

```
SELECT column_name(s) FROM table_name WHERE condition(s) GROUP  
BY column_name(s) HAVING condition(s)
```

SQL JOIN

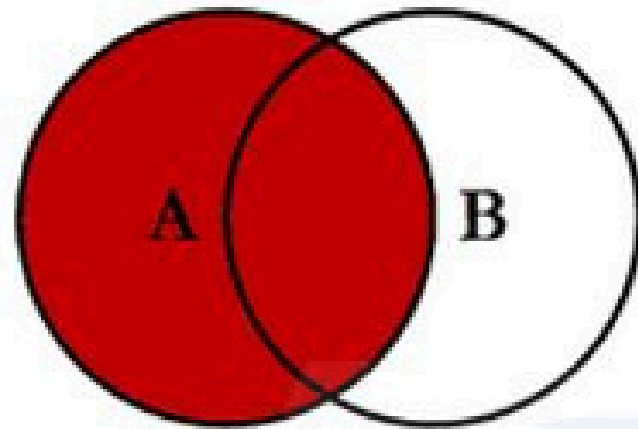
A JOIN clause is used to combine data from two or more tables, based on a related column between them.



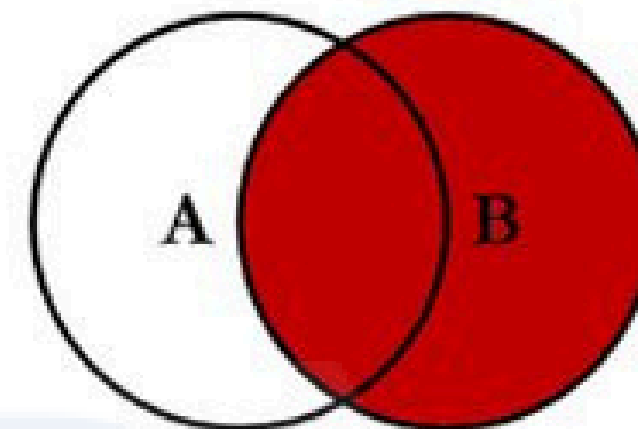
Which JOIN To Use

- **INNER JOIN:** Returns records that have matching values in both tables.
- **LEFT JOIN:** Returns all records from the left table, and the matched records from the right table.
- **RIGHT JOIN:** Returns all records from the right table, and the matched records from the left table.
- **FULL JOIN:** Returns all records when there is a match in either the left or right table.

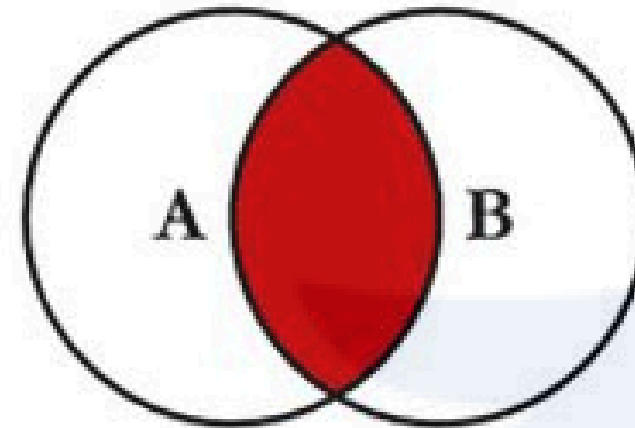
SQL JOINS



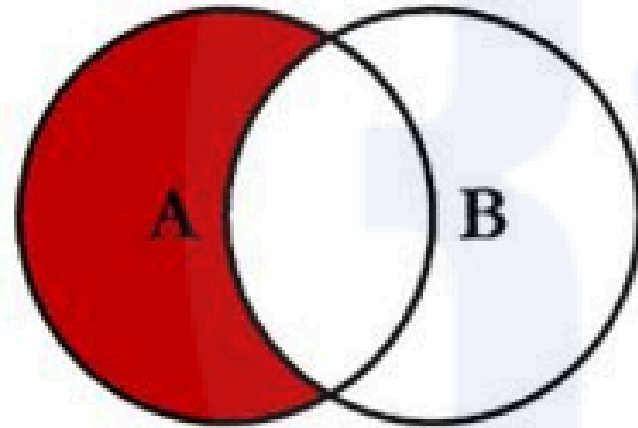
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



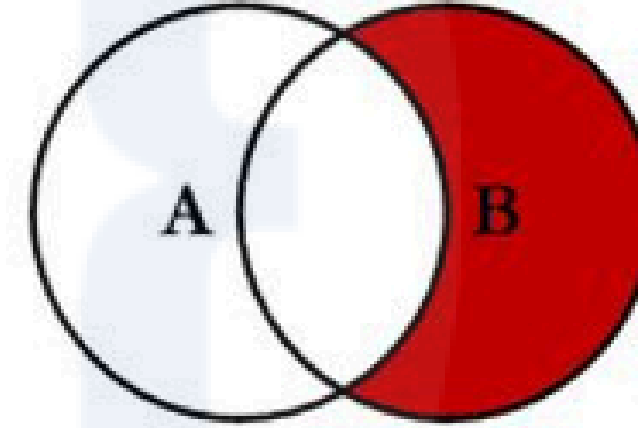
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



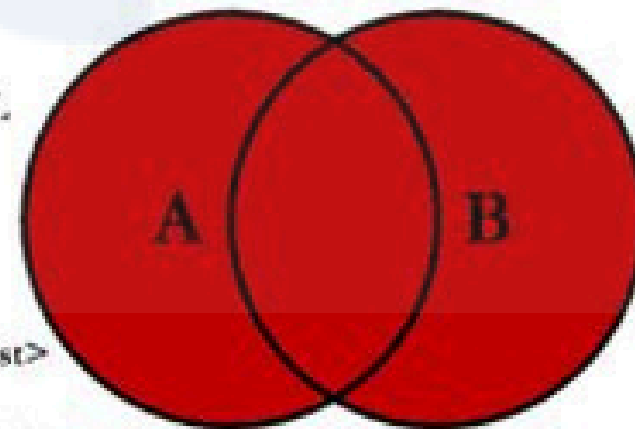
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



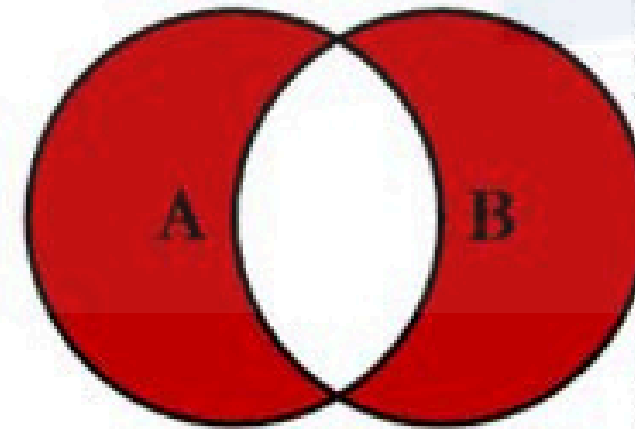
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

SELF JOIN

A self join is a regular join in which a table is joined to itself.SELF Joins are powerful for comparing values in a column of rows with the same table.

```
SELECT column_name(s)
FROM Table AS T1 JOIN
Table AS T2 ON
T1.col_name = T2.col_name
```



Empld	EmpName	ManagerId
1	Ram	3
2	Rahul	4
3	Ravi	2
4	Roshan	3

Find the name of respective managers for each of the employees?

```
SELECT T2.empname, T1.empname
FROM emp AS T1 JOIN emp AS T2
ON T1.empid = T2.manager_id
```

Important SQL Questions

-- Q-1. Write an SQL query to print all Worker details from the Worker table order by FIRST_NAME Ascending.

```
select * from worker ORDER by first_name;
```

-- Q-2. Write an SQL query to print details of the Workers whose FIRST_NAME contains 'a'.

```
select * from worker where first_name LIKE '%a%';
```

-- Q-3. Write an SQL query to print details of the Workers whose FIRST_NAME ends with 'h' and contains six alphabets.

```
select * from worker where first_name LIKE '____h';
```

-- Q-4. Write an SQL query to print details of the Workers whose SALARY lies between 100000 and 500000.

```
select * from worker where salary between 100000 AND 500000;
```

-- Q-5. Write an SQL query to print details of the Workers who have joined in Feb'2014.

```
select * from worker where YEAR(joining_date) = 2014 AND MONTH(joining_date) = 02;
```

-- Q-6. Write an SQL query to fetch the no. of workers for each department in the descending order.

```
select department, count(worker_id) AS no_of_worker from worker group by department  
ORDER BY no_of_worker desc;
```

-- Q-7. Write an SQL query to show the current date and time.

```
select curdate();
```

```
select now();
```

-- Q-8. Write an SQL query to show the top n (say 5) records of a table order by descending salary.

select * from worker order by salary desc LIMIT 5;

-- Q-9. Write an SQL query to determine the nth (say n=5) highest salary from a table.

select * from worker order by salary desc LIMIT 4,1;

-- Q-10. Write an SQL query to determine the 5th highest salary without using LIMIT keyword.

**select salary from worker w1
WHERE 4 = (
SELECT COUNT(DISTINCT (w2.salary))
from worker w2
where w2.salary >= w1.salary
);**

-- Q-11. Write an SQL query to fetch the list of employees with the same salary.

select w1.* from worker w1, worker w2 where w1.salary = w2.salary and w1.worker_id != w2.worker_id;

-- Q-12. Write an SQL query to fetch the departments that have less than 4 people in it.

select department, count(department) as depCount from worker group by department having depCount < 4;

-- Q-13. Write an SQL query to show all departments along with the number of people in there.

select department, count(department) as depCount from worker group by department;


-- Q-14. Write an SQL query to print the FIRST_NAME from Worker table after replacing 'a' with 'A'.

select REPLACE(first_name, 'a', 'A') from worker;

← Aditya Mishra

Cognizant Latest Interview Experience Q&A



 Package

4 Products



1 x Cognizant Interview Experience(Worst Case)

Digital Product



1 x Cognizant Interview Questions & Answer (IMP)

Digital Product



1 x Cognizant Latest Interview Experience(Jan - Feb)

Digital Product



1 x Cognizant Communication Round: Exact Ques.

Digital Product



THANKS FOR WATCHING