TOP 30 SQL QUERIES

IN HINDI

Biplab Parida [www.linkedin.com/in/biplabparida]

CodingGlitz [YouTube]

Execute all the queries in serial order and practice all the important 30 SQL queries as discussed in SQL Queries Interview Questions and answers. (***More than 1 Lakh views on YouTube***)

Subscribe our channel:

https://youtube.com/channel/UCaoCTb7FoUYOo9R8r_X9YHg?sub_confirmation=1

Note: This will execute perfectly in ORACLE.

To execute this for other databases like MySQL and SQL Server small changes in syntax might be needed.

Please Subscribe our channel and share our videos with anyone interested in learning SQL.

Checkout other important videos:

PLSQL Tutorial Playlist: https://youtube.com/playlist?list=PLKVxK2jv1oA-LwzbMlOFZS-CgbucH5OhQ

SQL Interview Questions and answers Playlist: sql interview questions and answers in Hindi - YouTube

Top 50 SQL Concepts Interview Questions and Answers : TOP 50 SQL Interview Questions And Answers

SQL JOINS : JOINS IN SQL with examples

SUB QUERY : Subquery in SQL || SQL Subquery tutorial with examples || SQL Tutorial for Beginners

FOR **MYSQL** use **VARCHAR** and **INT** in place of **VARCHAR2** and **NUMBER.**

# Create computer table used for JOINS problem.

```
CREATE TABLE COMPUTER (
COMPID NUMBER (10) PRIMARY KEY,
BRAND VARCHAR2(50),
COMPMODEL VARCHAR2(50),
MANUFACTUREDATE DATE);
```

# Create EMPLOYEE table

```
	CREATE TABLE EMPLOYEE (
	EMPID NUMBER (5) PRIMARY KEY,
	FIRSTNAME VARCHAR2(50),
	LASTNAME VARCHAR2(50),
	SALARY NUMBER (8),
	EMAILID VARCHAR2 (50),
	MANAGERID NUMBER (5),
	DATEOFJOINING DATE,
	DEPT VARCHAR2 (10),
	COMPID NUMBER (10),
	CONSTRAINT FK_COMPID FOREIGN KEY (COMPID) REFERENCES COMPUTER(COMPID));
```

# Insert data into COMPUTER table

```
	INSERT INTO COMPUTER VALUES (1001,'Lenovo','T480','12-JUN-19');
	INSERT INTO COMPUTER VALUES (1002,'Lenovo','T490','24-AUG-20');
	INSERT INTO COMPUTER VALUES (1003,'SONY','SQ112','01-DEC-19');
	INSERT INTO COMPUTER VALUES (1004,'SONY','SX1001','21-DEC-20');
```

#Insert data into EMPLOYEE table

```
INSERT INTO EMPLOYEE VALUES (1,'NANDA','KUMAR',50000, ‘NANDA@GMAIL.COM',NULL,'15-JUN-12','IT',1001);

INSERT INTO EMPLOYEE VALUES (2,'BIPLAB','PARIDA',30000,  'BPARIDA@YAHOO.COM',1,'21-DEC-15','IT',1001);

INSERT INTO EMPLOYEE VALUES (3,'DISHA','PATEL',50000,'DISHAP@GMAIL.COM',NULL,'21-AUG-13','HR',NULL);

INSERT INTO EMPLOYEE VALUES (4,'SIBA','PRASAD',90000,'SIBA@GMAIL.COM',3,'01-JUN-20','HR',1002);
INSERT INTO EMPLOYEE VALUES (5,'ANUSHKA','SHARMA', 20000, 'SHARMAA@GMAIL.COM',1,'01-MAR-21','IT', NULL);

INSERT INTO EMPLOYEE VALUES (6,'SOMNATH','MAHARANA', 65000, 'SMAHA@GMAIL.COM',3,'07-MAY-19','IT',1003);
```

#Create table to practice deleting duplicate records

```
        CREATE TABLE DUPLICATE (

        EMPID NUMBER (5) PRIMARY KEY,

        FIRSTNAME VARCHAR2 (50),

        LASTNAME VARCHAR2 (50),

        SALARY NUMBER (8),

        EMAILID VARCHAR2 (50),

        MANAGERID NUMBER (5),

        DATEOFJOINING DATE

        );
```

```
INSERT INTO DUPLICATE VALUES (1,'NANDA','KUMAR',50000, 'NANDA@GMAIL.COM', NULL,'15-JUN-12');

INSERT INTO DUPLICATE VALUES (2,'BIPLAB','PARIDA',30000,'BPARIDA@YAHOO.COM',1,'21-DEC-15');

INSERT INTO DUPLICATE VALUES (3,'SIBA','PRASAD',90000,'SIBA@GMAIL.COM',3,'01-JUN-20');

INSERT INTO DUPLICATE VALUES (4,'ANUSHKA','SHARMA', 20000, 'SHARMAA@GMAIL.COM',1,'01-MAR-21');

INSERT INTO DUPLICATE VALUES (5,'BIPLAB','PARIDA', 30000, 'BPARIDA@YAHOO.COM',1,'21-DEC-15');
```

# Query to view data from tables

```
select * from computer;

select * from employee;

select * from duplicate;
```

1. SQL Query to update DateOfJoining to 15-jul-2012 for empid =1.

   A. UPDATE EMPLOYEE SET DATEOFJOINING = ’15-JUL-2012’ WHERE EMPID =1;

2. SQL Query to select all student name where age is greater than 22

   A. SELECT * FROM STUDENT WHERE AGE > 22;

3. SQL Query to Find all employee with Salary between 40000 and 80000?

   A. SELECT * FROM EMPLOYEE WHERE SALARY BETWEEN 40000 AND 80000;
   B. SELECT * FROM EMPLOYEE WHERE SALARY >=40000 AND SALARY <=80000

4. SQL Query to display full name?

   A. SELECT CONCAT (FIRSTNAME, LASTNAME) FROM EMPLOYEE;
   B. SELECT FIRSTNAME || ‘ ‘ || LASTNAME FROM EMPLOYEE;

5. SQL Query to find name of employee beginning with S?

   A. SELECT * FROM EMPLOYEE WHERE FIRSTNAME LIKE ‘S%’;

6. Write a query to fetch details of employees whose firstname ends with an alphabet ‘A’ and contains exactly five alphabets ?

   A. SELECT * FROM EMPLOYEE WHERE FIRSTNAME LIKE ‘____A’;

7. Write a query to fetch details of all employees excluding few Employees :

   A. SELECT * FROM EMPLOYEE WHERE FIRSTNAME NOT IN (‘BIPLAB’,’DISHA’);

8. SQL query to display the current date ?

ORACLE:

A. SELECT SYSDATE, SYSTIMESTAMP, CURRENT_DATE, CURRENT_TIMESTAMP FROM DUAL; [ORACLE]

MYSQL:

A. SELECT CURDATE ( ); or SELECT CURRENT_DATE();

9. SQL Query to get day of last day of the previous month:

ORACLE:

A. SELECT TO_CHAR(LAST_DAY(ADD_MONTHS(SYSDATE-1)),’DAY’) FROM DUAL; [ORACLE]

MYSQL:

A. SELECT DAYNAME (LAST_DAY(DATE_ADD(CURDATE ()-1, INTERVAL -1 MONTH)))

10. Write an SQL query to fetch the employee FIRST names and replace the A with ‘@’:

A. SELECT REPLACE (FIRSTNAME,’A’,’@’) FROM EMPLOYEE;

11. Write an SQL query to fetch the domain from an email address:

ORACLE:

A. SELECT SUBSTR (EMAILID, INSTR(EMAILID,’@’) +1 FROM EMPLOYEE;

MYSQL:

A. SELECT SUBSTRING (EMAILID, INSTR(EMAILID,"@")+1) FROM EMPLOYEE;

12. Write an SQL query to update the employee names by removing leading and trailing spaces :

UPDATE EMPLOYEE SET FIRSTNAME = TRIM(FIRSTNAME);

13. Write an SQL query to fetch all the Employees details from Employee table who joined in the Year 2020:

ORACLE:

SELECT * FROM EMPLOYEE WHERE DATEOFJOINING BETWEEN ‘1-JAN-2020’ AND ’31-DEC-2020’;

```
SELECT * FROM EMPLOYEE WHERE TO_CHAR(DATEOFJOINING,’YYYY’)=2020;
```

MYSQL:

```
SELECT * FROM EMPLOYEE WHERE YEAR (DATEOFJOINING)=2020;
```

14. Write an SQL query to fetch only odd rows / Even rows from the table :

```
SELECT *
FROM EMPLOYEE
WHERE mod(EMPID,2) = 0;
```

FOR **SQL SERVER** % CAN BE USED INSTEAD OF MOD FUNCTION

15. Write an SQL query to create a new table with data and structure copied from another table:

```
CREATE TABLE EMP AS (SELECT * FROM EMPLOYEE);
```

16. Write an SQL query to create an empty table with the same structure as some other table :

```
CREATE TABLE EMP2 AS (SELECT * FORM EMPLOYEE WHERE 1=2);
```

17. Write an SQL query to fetch top 3 HIGHEST salaries :

```
SELECT SALARY FROM
(SELECT DISTINCT SALARY FROM EMPLOYEE ORDER BY SALARY DESC)
WHERE ROWNUM < 4;
```

For MYSQL:

USE LIMIT 3 INSTEAD OF ROWNUM<4

18. Find the first employee and last employee from employee table :

First Employee:

```
SELECT * FROM EMPLOYEE WHERE EMPID = (SELECT MIN(EMPID) FROM EMPLOYEE);
```

Last Employee:

```
SELECT * FROM EMPLOYEE WHERE EMPID = (SELECT MAX(EMPID) FROM EMPLOYEE);
```

19. List the ways to get the count of records in a table:

```
SELECT COUNT (*) FROM EMPLOYEE;
SELECT COUNT (EMPID) FROM EMPLOYEE;

SELECT COUNT (1) FROM EMPLOYEE;
```

20. Write a query to fetch the department-wise count of employees sorted by department's count in ascending order:

```
SELECT DEPT, COUNT (*) FROM EMPLOYEE GROUP BY DEPT ORDER BY COUNT(*);
```

21. Write a query to retrieve Departments who have less than 4 employees working in it :

```
SELECT DEPT, COUNT (*) FROM EMPLOYEE GROUP BY DEPT HAVING COUNT(*) < 4;
```

22. Write a query to retrieve Department wise Maximum salary:

```
SELECT DEPT, MAX(SALARY) FROM EMPLOYEE GROUP BY DEPT;
```

23. Write a query to Employee earning maximum salary in his department :

```
SELECT * FROM EMPLOYEE E1 JOIN (

SELECT DEPT, MAX(SALARY) SAL FROM EMPLOYEE GROUP BY DEPT) E2

ON E1.DEPT = E2.DEPT AND E1.SALARY = E2.SAL;
```

24. Write an SQL query to fetch the first 50% records from a table:

ORACLE:

```
SELECT * FROM EMPLOYEE WHERE ROWNUM <= (SELECT COUNT(*) FROM EMPLOYEE) / 2;
```

MYSQL:

```
SELECT * FROM Customers LIMIT (select COUNT(*)/2 from Customers);
```

25. Query to fetch details of employees not having computer:

```
SELECT * FROM EMPLOYEE WHERE COMPID IS NULL;
```

26. Query to fetch employee details along with the computer details who have been assigned with a computer :

```
SELECT * FROM EMPLOYEE E JOIN COMPUTER C ON E.COMPID = C.COMPID;
```

27. Fetch all employee details along with the computer name assigned to them:

ORACLE:

```
SELECT E.EMPID, E.FIRSTNAME || ‘ ’ || E.LASTNAME, NVL (C.BRAND, ‘NOT ASSIGNED’)
FROM EMPLOYEE E LEFT JOIN COMPUTER C ON E.COMPID = C.COMPID;
```

MYSQL : IFNULL () FUNCTION CAN BE USED INSTEAD OF NVL

28. Fetch all Computer Details along with employee name using it :

```
SELECT C.BRAND, C.COMPMODEL, E.FIRSTNAME FROM EMPLOYEE E RIGHT JOIN
COMPUTER C ON E.COMPID = C.COMPID;
```

29. Delete duplicate records from a table:

```
DELETE FROM TABLE WHERE rowid NOT IN (
SELECT MIN(rowid)  FROM TABLE  GROUP BY column1, column2, ...);
                    OR

DELETE t1 FROM TABLE  t1
JOIN (   SELECT column1, column2, ..., MIN(id) AS min_id
  FROM TABLE  GROUP BY column1, column2, ...
) t2
ON t1.column1 = t2.column1 AND t1.column2 = t2.column2 AND ...
WHERE t1.id > t2.min_id;
```

30. Find Nth Highest salary :

A.

```
SELECT E1.SALARY,COUNT (DISTINCT E2.SALARY)

FROM EMPLOYEE E1 JOIN EMPLOYEE E2 ON E1.SALARY<=E2.SALARY

GROUP BY E1.SALARY HAVING  COUNT(DISTINCT E2.SALARY) =N;
```

B.

```
SELECT  DISTINCT SALARY FROM EMPLOYEE E1
WHERE N = (SELECT COUNT(DISTINCT SALARY)
FROM EMPLOYEE E2 WHERE E2.SALARY > E1.SALARY);
```

C.

```
SELECT * FROM (SELECT SALARY, (DENSE_RANK() OVER (ORDER BY SALARY DESC))  R
FROM EMPLOYEE )WHERE R = N;
```