SQL 50 Interview Questions

SQL/50

23/12/22

Q1 Write a query to fetch "FIRST_NAME" form Worker table ~~in upper case~~ using alias name as <WORKER_NAME>.

~~→ SELECT UPPER (FIRST_NAME) FROM WORKER;~~

→ SELECT FIRST_NAME AS WORKER_NAME
FROM Worker;

O/P →

Worker_name
~
~
~
~

WORKER (id, firstname, lastname, salary, join date, department)
BONUS (wid, amount, Bonus_date)
TITLE (wrkid, title, affected FROM)
↓
SCHEMA

Q2. Write an SQL query to fetch "FIRST_NAME" from Worker table in Uppercase.

→ SELECT UPPER(FIRST_NAME) FROM WORKER;
→ post processing, cosmetics

Q3. Fetch unique values of department from WORKER TABLE

→ SELECT DISTINCT Department FROM Worker;

Same query how can we write using (group by)?

→ SELECT Department FROM Worker GROUP BY Department
when used with AGGregate at as distinct

Q4. Print the first ~~name~~ three characters of FIRST_NAME from Worker table

→ SELECT SUBSTRING (FIRSTNAME, 1, 3) FROM WORKER;
↓ ↑ ↓
Attribute start len

Q5. Find the position of the alphabet ('b') in the first name column 'Amitabh' from worker Table (2) (OM)

(INSTR) → position of string in another string

INSTR(Attribute, ~~pos~~ string)

→ SELECT INSTR(FIRST_NAME, 'B') FROM worker
WHERE FIRSTNAME = 'Amitabh'; → o/p → 6

(INSTR)

CASE insensitive (B==b)

Q6. Print the FIRST_NAME from worker table after removing white spaces from the right side.

→ SELECT RTRIM(FIRST_NAME) FROM WORKER;

Q7. Left whitespace I → Remove whitespaces.

→ SELECT LTRIM(FIRST_NAME) FROM worker;

Q8. Fetch the unique values of DEPARTMENT from worker table and prints its length.

→ SELECT DISTINCT department, LENGTH(department)
FROM WORKER;

Q9. Write an SQL query to print the firstname from worker table after replacing 'a' with 'A'

→ SELECT REPLACE(FIRSTNAME, 'a', 'A') FROM worker;

↓ string

Q10. Write an SQL query to print FIRST NAME and LAST NAME from WORKER TABLE into a single column NAME

→ SELECT CONCAT~~CAT~~(FIRSTNAME, LASTNAME) AS NAME
FROM WORKER;

↓ adds

↗ space(Imp)

SELECT CONCAT(FIRSTNAME, ' ', LASTNAME) AS FULLNAME
FROM WORKER;

Q11 Write an SQL query to print all Worker details from Worker table ordered by FIRST_NAME Ascending (3)

→ SELECT * FROM WORKER ORDER BY FIRST_NAME; (asc; not necessary)

Q12 Print all worker details from the worker table ordered by FIRST_NAME ascending and department descending.

→ SELECT * FROM WORKER WHERE ORDER BY FIRST NAME AND ~~DEPARTMENT~~ ORDER BY DEPARTMENT DESC. ✗

→ SELECT * FROM WORKER ORDER BY FISRTNAME, Department ASC DESC;

Q13 Print details of Workers with firstname 'Vipul' and 'Satish' from worker table.

→ SELECT FIRSTNAME FROM WORKER WHERE FIRSTNAME LIKE 'Vipul' ~~AND~~ OR 'Satish'; ✗

→ SELECT * FROM Worker WHERE FIRSTNAME IN ('Vipul', 'Satish');

Q14 print details of workers excluding first name Vipul and Satish.

→ SELECT * FROM Worker WHERE FIRSTNAME NOT IN ('Vipul', 'Satish');

Q15 print details of workers ~~exceeding~~ with department name as "Admin";

→ SELECT * FROM WORKERS WHERE DEPARTMENT = "Admin"; ✗

→ SELECT * FROM WORKERS WHERE department LIKE 'Admin';

'Admin%' wildcard

→ Administration

→ Administrator.

Q16 Details of workers whose first name contains 'a'. (4)

→ SELECT * FROM WORKER WHERE FIRSTNAME LIKE '%a%';

Q17 Details of workers whose first name ends with 'a'.

→ SELECT * FROM WORKER WHERE FIRSTNAME LIKE '%a';

Q18 Details of workers whose first name ends with 'h' and contains six alphabets.

→ SELECT * FROM WORKER WHERE FIRSTNAME LIKE '%h'
AND LENGTH (FIRSTNAME) = 6; ✗

→ SELECT * FROM WORKER WHERE FIRSTNAME LIKE '_____h';

Q19 Details of workers joined in Feb 2014.

→ SELECT * FROM WORKER WHERE JOINING_DATE
LIKE '2014-02%'; ✗

→ SELECT * FROM WORKER WHERE YEAR(JOININGDATE) = 2014
AND MONTH(JOININGDATE) = 02;

Q20 Details of workers whose salary lies between 100000 and 500000

→ SELECT * FROM WORKERS WHERE SALARY BETWEEN
100000 AND 500000

Q21 Count of employees working in the department 'Admin'

→ SELECT COUNT(department) WHERE department like
'Admin'; ✗

→ SELECT department, COUNT(*) FROM workers WHERE Department
LIKE 'ADMIN';

Q22 Fetch worker full name with salaries >= 50000 and <= 100000.

→ SELECT CONCAT(FIRSTNAME, ' ', LASTNAME) AS FULLNAME
WHERE SALARY BETWEEN 50000 AND 100000;

Q23 Fetch number of workers for each department in the descending order.

→ SELECT Department, COUNT(*) FROM Workers
~~ORDER~~ ORDER BY DEPARTMENT DESC; ✗

→ SELECT Department, count(worker_id) AS NUMBER FROM Worker
GROUP BY DEPARTMENT
ORDER BY COUNT(worker_id) DESC. ✓

Q24

→ SELECT Department, COUNT(worker_id) AS no_of_worker
GROUP BY Department
ORDER BY no_of_worker DESC;

Q24: details of workers who are also managers.

TABLES (worker, Bonus, TITLE)
↓ ↓
(wid) (wrf_id) JOIN.

→ SELECT * FROM worker as w
INNER JOIN ON Bonus as B WHERE
w.id = B.id ; ~~dep~~ worker_Title ~~is~~ LIKE 'manager';

(PK, FK)
Intersection

SELECT * FROM worker as w inner JOIN TITLE as t ON
WHERE w.id = ~~t.id~~ t.worker_id
WHERE t.title = 'manager';

SQL →

Q25: Write an SQL query to fetch number (more than 1) of same titles in the ORG of different types.

→ SELECT worker_title, count(*) as count from title GROUP BY worker_title HAVING COUNT(*) > 1;

Q26 Write SQL query to show only odd rows from a table

→ SELECT * FROM worker where MOD(WORKER_ID, 2) <> 0;

Q27 Write SQL query to show only even rows from table.

→ SELECT * FROM worker where (MOD(workerId, 2) = 0;

Q28 Write an SQL query to clone a new table from another table.

→ CREATE TABLE worker_clone LIKE worker;
INSERT INTO worker_clone SELECT * FROM worker.

Q29 Fetch intersecting records of two tables.

→ SELECT * FROM worker as w ~~OEI~~ ^ INNER JOIN ~~Cno~~ Project as p. ON w.id == p.id

→ SELECT worker.* from worker.
INNER JOIN ~~p1~~ worker_clone using (worker.id);

Q30 Show records from one table that another table doesn't have MINUS.

LEFT JOIN

→ SELECT ~~*~~ ~~FROM~~ worker.* ~~from~~ ^ worker_clone
USING (worker_id) WHERE worker_clone.worker_id is NULL

Q31 Show the current date and time

-- DUAL

→ Select curdate();
Select now();

Q32 Show the top n (say 5) records of a table order by descending salary.

→ SELECT * FROM WORKER ORDER BY SALARY DESC LIMIT 5;

Q33 ~~W~~ determine the nth ~~n~~ highest salary from table.

→ SELECT * FROM Worker ORDER BY Salary DESC LIMIT 4,1;

Q34 Determine the nth highest salary without using LIMIT keyword.

-- correlated subquery.

→ SELECT * FROM Worker W1
WHERE 4 = (SELECT COUNT (DISTINCT (W2.Salary))
FROM Worker W2;
WHERE W2.sal = W1.sal.

Q35 Fetch the list of employee with same salary.

→ SELECT W1.* FROM Worker W1, Worker W2
WHERE W1.salary = W2.Salary AND
W1.id <> W2.id;

Q36 Fetch second highest salary from the table using sub query

→ SELECT Salary FROM Workers ~~WHERE~~ LIMIT 1,1;

→ SELECT MAX(Salary) FROM Worker
WHERE SALARY NOT IN (SELECT max(Salary) from Worker);

Q37 to show one row twice in results from a table.

→ SELECT * FROM WORKER
UNION ALL
SELECT * FROM Worker ORDER BY worked_id

UNION - DISTINCT

Q38 ~~SEL~~ Show info of worker who dint get abonus.

SELECT worked_id from worker where workerid NOT IN (SELECT worker_ref_id from bonus);

Q39 Fetch first 50% of table

SELECT * FROM Worker WHERE worker_id <= (
SELECT COUNT (worker_id)/2 from Worker);

Q40 SQL query to print department where people < 4.

SELECT department FROM Worker as dep Count
GROUP BY department HAVING ~~COUNT~~ DISTINCT (worked_id) < 4;
dep count < 4;

Q41 Show all departments along with the ~~name~~ number of people in there

SELECT department FROM worker as dep Count GROUP BY
~~COUNT (department)~~ department;

Q42 Write a SQL query to fetch the last row of table.

SELECT * FROM Worker WHERE Worked_id =
(select max (worked_id) FROM
Worker;