

~\Desktop\Interview Preparation\DBMS\50_questions_sql.sql











```
1  -- worker.* means all columns from worker table.
2
3  -- Q-1. Write an SQL query to fetch "FIRST_NAME" from Worker table using the alias name as <
   WORKER_NAME>.
4  select FIRST_NAME AS worker_name from worker;
5
6  -- Q-2. Write an SQL query to fetch "FIRST_NAME" from Worker table in upper case.
7  select UPPER(FIRST_NAME) FROM worker;
8
9  -- Q-3. Write an SQL query to fetch unique values of DEPARTMENT from Worker table.
10 SELECT distinct DEPARTMENT FROM DEPARTMENT; //HW GROUP BY
11
12 -- Q-4. Write an SQL query to print the first three characters of FIRST_NAME from Worker
   table.
13 SELECT substring(FIRST_NAME, 1, 3) FROM Worker;
14
15 -- Q-5. Write an SQL query to find the position of the alphabet ('b') in the first name
   column 'Amitabh' from Worker table.
16 select INSTR(first_name, 'b') from worker where first_name="Amitabh";
17
18 -- Q-6. Write an SQL query to print the FIRST_NAME from Worker table after removing white
   spaces from the right side.
19 select RTRIM(first_name) from worker;
20
21 -- Q-7. Write an SQL query to print the DEPARTMENT from Worker table after removing white
   spaces from the left side.
22 select LTRIM(first_name) from worker;
23
24 -- Q-8. Write an SQL query that fetches the unique values of DEPARTMENT from Worker table
   and prints its length.
25 select distinct department, LENGTH(department) from worker;
26
27 -- Q-9. Write an SQL query to print the FIRST_NAME from Worker table after replacing 'a'
   with 'A'.
28 select REPLACE(first_name, 'a', 'A') from worker;
29
30 -- Q-10. Write an SQL query to print the FIRST_NAME and LAST_NAME from Worker table into a
   single column COMPLETE_NAME.
31 -- A space char should separate them.
32 select CONCAT(first_name, ' ', LAST_NAME) as COMPLETE_NAME from worker; //vimp
33
34 -- Q-11. Write an SQL query to print all Worker details from the Worker table order by
   FIRST_NAME Ascending.
35 select * from worker ORDER by first_name ASC; //important.
36
37 -- Q-12. Write an SQL query to print all Worker details from the Worker table order by
   FIRST_NAME Ascending and DEPARTMENT Descending.
38 -- FIRST_NAME Ascending and DEPARTMENT Descending.
39 select * from worker ORDER by first , department DESC;
40
41 -- Q-13. Write an SQL query to print details for Workers with the first name as "Vipul" and
   "Satish" from Worker table.
42 select * from worker where first_name IN("vipul", "satish");
43
44 -- Q-14. Write an SQL query to print details of workers excluding first names, "Vipul" and
   "Satish" from Worker table.
45 select * from worker where first_name NOT IN("vipul", "satish");
46
47 -- Q-15. Write an SQL query to print details of Workers with DEPARTMENT name as "Admin*".
48 select * from worker where first_name LIKE "Admin%";
49
```

```

50 -- Q-16. Write an SQL query to print details of the Workers whose FIRST_NAME contains 'a'.
51 select * from worker where first_name LIKE "%a%";
52
53
54 -- Q-17. Write an SQL query to print details of the Workers whose FIRST_NAME ends with 'a'.
55 select * from worker where first_name LIKE "%a";
56
57 -- Q-18. Write an SQL query to print details of the Workers whose FIRST_NAME ends with 'h'
and contains six alphabets.
58 select * from worker where first_name LIKE "_____h";
59
60 -- Q-19. Write an SQL query to print details of the Workers whose SALARY lies between 100000
and 500000.
61 select * from worker where SALARY between 100000 and 500000;
62
63 -- Q-20. Write an SQL query to print details of the Workers who have joined in Feb'2014.
64 select * from worker where JOINING_DATE="Feb 2024";           //this is wrong.
65 select * from worker where YEAR(JOINGING_DATE)=2024 AND MONTH(JOINGING_DATE)=02;
//very important(right).
66
67 -- Q-21. Write an SQL query to fetch the count of employees working in the department
'Admin'.
68 select COUNT(employees) from worker where department="Admin";
69 -- or
70 select department, COUNT(*) from worker where department="Admin";
71
72 -- Q-22. Write an SQL query to fetch worker full names with salaries >= 50000 and <= 100000.
73 select CONCAT(first_name, ' ', LAST_NAME) from worker where salary between 50000 and 100000;
//important.
74
75
76 -- Q-23. Write an SQL query to fetch the no. of workers for each department in the
descending order.
77 select COUNT(worker_id), department from worker GROUP By department ORDER BY
COUNT(worker_id); //in output of this, first count and then corresponding department name
side by side will be printed.
78 select department, COUNT(worker_id) from worker GROUP BY department ORDER BY
COUNT(worker_id); //here first department name then corresponding count side by side will be
printed
79
80 -- Q-24. Write an SQL query to print details of the Workers who are also Managers.  🚩🚩
81 select w.* from worker as w inner join title as t on w.worker_id=t.worker_ref_id where
worker_title="Manager"; //due to w.* details of worker table will be printed only.
82
83 -- Q-25. Write an SQL query to fetch number (more than 1) of same titles in the ORG of
different types.
84 select worker_title , COUNT(*) as count from title group by worker_title having count>1;
//important //does not understood.
85
86 -- Q-26. Write an SQL query to show only odd rows from a table.
87 -- select * from worker where MOD(worker_id, 2)!=0;
-- 🚩important.
88 or
89 select * from worker where MOD(worker_id, 2)<>0;
90
91
92 -- Q-27. Write an SQL query to show only even rows from a table.
93 select * from worker where MOD(worker_id, 2)=0;           -- 🌟
94
95 -- Q-28. Write an SQL query to clone a new table from another table.
96 create Table clone_table LIKE worker -- 🚩very important //creates same
table schema for clone_table like worker but dont fill entries automatically.
97

```

```

98 insert INTO clone_table select * from worker;      --  //this inserting entries.
99
100 -- Q-29. Write an SQL query to fetch intersecting records of two tables.
101 select worker.* from worker inner join clone_table using (worker_id);      --  "using" is
    same as on worker.worker_id=clone_table.worker_id;
102
103 -- Q-30. Write an SQL query to show records from one table that another table does not have.
104 -- MINUS
105 select worker.* from worker left join clone_table using (worker_id) where
    clone_table.worker_id is NULL; --  means all entries of left table which does not matches
    with right table.
106
107 -- Q-31. Write an SQL query to show the current date and time.      
108 -- DUAL
109 select curdate();          //important
110 select now();
111
112 -- Q-32. Write an SQL query to show the top n (say 5) records of a table order by descending
    salary.
113 select * from worker order by salary desc limit 5;      //this will perform query on only
    top 5 elements now.
114
115 -- Q-33. Write an SQL query to determine the nth (say n=5) highest salary from a table.
116 select * from worker order by salary desc limit 4, 1;      --  means 1st entry after 4th
    entry.(limit 4, 2 means 2nd entry after 4th one).
117
118 -- Q-34. Write an SQL query to determine the 5th highest salary without using LIMIT keyword.
119 SELECT DISTINCT salary FROM employees e1 WHERE 5 = (SELECT COUNT(DISTINCT salary) FROM
    employees e2 WHERE e2.salary > e1.salary ); -- easy just think.
120
121 -- Q-35. Write an SQL query to fetch the list of employees with the same salary.
122 select w1.* from worker w1, worker w2 where w1.salary = w2.salary and w1.worker_id !=
    w2.worker_id; --  when we use , bet'n two tables it is cross join.
123
124 -- Q-36. Write an SQL query to show the second highest salary from a table using sub-query.
125 select max(salary) from worker      -- 
126 where salary not in (select max(salary) from worker);
127
128 -- Q-37. Write an SQL query to show one row twice in results from a table.      
129 select * from worker
130 UNION ALL
131 select * from worker ORDER BY worker_id;
132
133 --Q.38 Write an SQL query to show all except top 5 workers having highest salaries.
134 select * from worker order by salary ASC offset 5 ROWS      --  this will
    return table by skipping last 5 records from o/p table.
135      --offset is used to skip the rows from the end of
    resultant table
136      --using subquery :
137 SELECT DISTINCT salary FROM employees e1 WHERE 5 < (SELECT COUNT(DISTINCT salary) FROM
    employees e2 WHERE e2.salary > e1.salary); --important
138
139
140 -- Q-38. Write an SQL query to list worker_id who does not get bonus.
141 select worker_id from worker where worker_id not in (select worker_ref_id from bonus);
    --important
142
143 -- Q-39. Write an SQL query to fetch the first 50% records from a table.
144 select * from worker where worker_id <= ( select count(worker_id)/2 from worker);
145
146 -- Q-40. Write an SQL query to fetch the departments that have less than 4 people in it.
147 select department, count(department) as depCount from worker group by department having
    depCount < 4; -- 

```

```
148
149 -- Q-41. Write an SQL query to show all departments along with the number of people in
    there.
150 select department, count(department) as depCount from worker group by department;
151
152 -- Q-42. Write an SQL query to show the last record from a table.
153 select * from worker where worker_id = (select max(worker_id) from worker);
154
155 -- Q-43. Write an SQL query to fetch the first row of a table.
156 select * from worker where worker_id = (select min(worker_id) from worker);
157
158 -- Q-44. Write an SQL query to fetch the last five records from a table.
159 (select * from worker order by worker_id desc limit 5) order by worker_id; --
160
161 -- Q-45. Write an SQL query to print the name of employees having the highest salary in each
    department.
162 select w.department, w.first_name, w.salary from
163     (select max(salary) as maxsal, department from worker group by department) temp
164 inner join worker w on temp.department = w.department and temp.maxsal = w.salary;
165
166 -- Q-46. Write an SQL query to fetch three max salaries from a table using co-related
    subquery
167 select distinct salary from worker w1
168 where 3 >= (select count(distinct salary) from worker w2 where w1.salary <= w2.salary) order
    by w1.salary desc;
169 -- DRY RUN AFTER REVISING THE CORELATED SUBQUERY CONCEPT FROM LEC-9.
170 select distinct salary from worker order by salary desc limit 3;
171
172 -- Q-47. Write an SQL query to fetch three min salaries from a table using co-related
    subquery
173 select distinct salary from worker w1
174 where 3 >= (select count(distinct salary) from worker w2 where w1.salary >= w2.salary) order
    by w1.salary desc;
175
176 -- Q-48. Write an SQL query to fetch nth max salaries from a table.
177 select distinct salary from worker w1
178 where n = (select count(distinct salary) from worker w2 where w1.salary <= w2.salary) order
    by w1.salary desc;
179
180 -- Q-49. Write an SQL query to fetch departments along with the total salaries paid for each
    of them.
181 select department , sum(salary) as depSal from worker group by department order by depSal
    desc;
182
183 -- Q-50. Write an SQL query to fetch the names of workers who earn the highest salary.
184 select first_name, salary from worker where salary = (select max(Salary) from worker);
185
186 --Q-51. Department Highest Salary for Each Department.
187 select dp.name as Department, em.name as Employee, em.salary as Salary from Employee as em
    inner join Department
188 as dp on em.departmentId=dp.id where (em.departmentId, em.salary) IN (select departmentId,
    Max(salary) from
189 Employee group by departmentId);
```