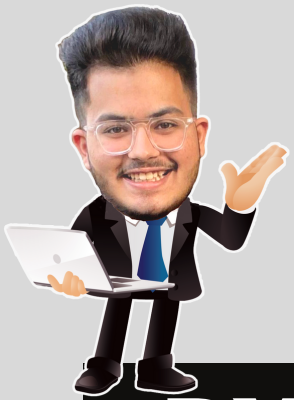
MOST ASKED

SQL

INTERVIEW QUESTIONS



BY SIDDHARTH SINGH

SAMPLE TABLE: WORKER

WORKER_ID	FIRST NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	Siddharth	Singh	80000	2019-03-20 09:00:00	HR
2	Lavesh	Ahir	300000	2019-07-11 09:00:00	Admin
3	Abhishek	Midha	500000	2019-03-20 09:00:00	HR
4	Rahul	Mahar	200000	2019-03-20 09:00:00	Admin
5	Saurabh	Madavi	90000	2019-07-11 09:00:00	Admin
6	Aman	Nain	75000	2019-07-11 09:00:00	Account
7	Vaibhav	Varshney	100000	2019-02-20 09:00:00	Account
8	Farhaan	Majied	500000	2019-05-11 09:00:00	Admin

SAMPLE TABLE: TITLE

WORKER_REF_ID	WORKER_TITLE	AFFECTED FROM
1	Manager	2021-02-20 00:00:00
2	Executive	2021-06-11 00:00:00
8	Executive	2021-06-11 00:00:00
5	Manager	2021-06-11 00:00:00
4	Asst. Manager	2021-06-11 00:00:00
7	Executive	2021-06-11 00:00:00
6	Lead	2021-06-11 00:00:00
3	Lead	2021-06-11 00:00:00

SAMPLE TABLE: Job grades

GRADE_LEVEL	LOWEST_SAL	HIGHEST_SAL
A	10000	75999
B	76000	80999
C	81000	99999
D	100000	199999
E	200000	299999
F	300000	600000

CREATE

Q 1 Write a SQL query to create WORKER Table

```
CREATE TABLE Worker (
 WORKER_ID INT NOT NULL PRIMARY KEY,
 FIRST_NAME CHAR(25),
 LAST_NAME CHAR(25),
 SALARY INT(15),
 JOINING_DATE DATETIME,
 DEPARTMENT CHAR(25)
);
```

INSERT

Q 2 Write a SQL Query to insert above values in WORKER Table

```
INSERT INTO Worker
 (WORKER_ID, FIRST_NAME, LAST_NAME, SALARY, JOINING_DATE,
DEPARTMENT) VALUES
(1, 'Siddharth', 'Singh', 80000, '2019-03-20 09:00:00', 'HR')
(2, 'Lavesh', 'Ahir', 300000, '2019-07-11 09:00:00', 'Admin')
(3, 'Abhishek', 'Midha', 500000, '2019-03-20 09:00:00', 'HR')
(4, 'Rahul', 'Mahar',200000, '2019-03-20 09:00:00', 'Admin')
(5, 'Saurabh', 'Madavi', 90000, '2019-07-11 09:00:00', 'Admin')
(6, 'Aman', 'Nain', 75000, '2019-07-11 09:00:00','Account')
(7, 'Vaibhav', 'Varshney', 100000, '2019-02-20 09:00:00', 'Account')
(8, 'Farhaan', 'Majied', 500000, '2019-05-11 09:00:00', 'Admin');
```

FORIEGN KEY

Q 3 Write a SQL Query to create table Title which has WORKER_REF_ID as foreign key

```
CREATE TABLE Title (
 WORKER_REF_ID INT,
 WORKER_TITLE CHAR(25),
 AFFECTED_FROM DATETIME,
 FOREIGN KEY (WORKER_REF_ID)
  REFERENCES Worker(WORKER_ID)
      ON DELETE CASCADE
);
```

CREATE

Q 4 Write a SQL query to clone a new table WorkCopy from another table.

The general query to clone a table with data is:
SELECT * INTO WorkerCopy FROM Worker;

The general way to clone a table without information is:
SELECT * INTO WorkerCopy FROM Worker WHERE 1 = 0;

An alternate way to clone a table (for MySQL) without is:
CREATE TABLE WorkerCopy LIKE Worker;

ALIAS

Q 5 Write a SQL query to fetch “FIRST_NAME” from Worker table using the alias name as <WORKER_NAME>.

Select FIRST_NAME AS WORKER_NAME from Worker;

UPPER

Q 6 Write a SQL query to fetch “FIRST_NAME” from Worker table in upper case.

```
Select upper(FIRST_NAME) from Worker;
```

DISTINCT

Q 7 Write an SQL query to fetch unique values of DEPARTMENT from Worker table.

```
Select distinct DEPARTMENT from Worker;
```

REPLACE

Q 8 Write a SQL query to print the FIRST_NAME from Worker table after replacing ‘a’ with ‘A’.

```
Select REPLACE(FIRST_NAME,'a','A') from Worker;
```

CONCAT

Q 9 Write a SQL query to print the FIRST_NAME and LAST_NAME from Worker table into a single column COMPLETE_NAME. A space char should separate them.

```
Select CONCAT(FIRST_NAME, ' ', LAST_NAME) AS
'COMPLETE_NAME' from Worker;
```

ORDER BY

Q 10 Write a SQL query to print all Worker details from the Worker table order by FIRST_NAME Ascending.

```
Select * from Worker order by FIRST_NAME asc;
```

ORDER BY

Q 11 Write a SQL query to print all Worker details from the Worker table order by FIRST_NAME Ascending and DEPARTMENT Descending.

```
Select * from Worker order by FIRST_NAME asc,DEPARTMENT desc;
```

IN

Q 12 Write a SQL query to print details for Workers with the first name as “Rahul” and “Lavesh” from Worker table.

```
Select * from Worker where FIRST_NAME in ('Rahul','Lavesh');
```

NOT IN

Q 13 Write a SQL query to print details of workers excluding first names, “Rahul” and “Lavesh” from Worker table.

```
Select * from Worker where FIRST_NAME  not in ('Rahul','Lavesh');
```

LIKE %

Q 14 Write a SQL query to print details of the Workers whose FIRST_NAME starts with ‘S’.

```
Select * from Worker where FIRST_NAME like 'S%';
```

LIKE %

Q 15 Write a SQL query to print details of the Workers whose FIRST_NAME contains ‘a’.

```
Select * from Worker where FIRST_NAME like '%a%';
```

LIKE %

Q 16 Write a SQL query to print details of the Workers whose FIRST_NAME ends with 'n’.

```
Select * from Worker where FIRST_NAME like '%n';
```

LIKE _

Q 17 Write a SQL query to print details of the Workers whose FIRST_NAME ends with ‘l’ and contains five alphabets.

```
Select * from Worker where FIRST_NAME like '____l';
```

BETWEEN

Q 18 Write a SQL query to print details of the Workers whose SALARY lies between 100000 and 500000.

Select * from Worker where SALARY between 100000 and 500000;

DATE

Q 19 Write a SQL query to print details of the Workers who have joined in Mar'2019.

Select * from Worker where year(JOINING_DATE) = 2019 and month(JOINING_DATE) = 3;

DATE

Q 20 Write a SQL query to show the current date and time.

Following MySQL query returns the current date:
SELECT CURDATE();

Following MySQL query returns the current date and time:
SELECT NOW();

Following SQL Server query returns the current date and time:
SELECT getdate();

Following Oracle query returns the current date and time:
SELECT SYSDATE FROM DUAL;

COUNT

Q 21 Write a SQL query to fetch the count of employees working in the department ‘Admin’.

```
SELECT COUNT(*) FROM worker WHERE DEPARTMENT = 'Admin';
```

GROUP BY

Q 22 Write a SQL query to fetch the no. of workers for each department in the descending order.

```
SELECT DEPARTMENT, count(WORKER_ID) No_Of_Workers
FROM worker
GROUP BY DEPARTMENT
ORDER BY No_Of_Workers DESC;
```

GROUP BY

Q 23 Write a SQL query to fetch departments along with the total salaries paid for each of them.

```
SELECT DEPARTMENT, sum(Salary) from worker group by
DEPARTMENT;
```

GROUP BY

Q 24 Write a SQL query to show all departments along with the number of people in there.

```
SELECT DEPARTMENT, COUNT(DEPARTMENT) as 'Number of
Workers' FROM Worker GROUP BY DEPARTMENT;
```

HAVING

Q 25 Write a SQL query to fetch the departments that have less than five people in it.

SELECT DEPARTMENT, COUNT(WORKER_ID) as 'Number of Workers' FROM Worker GROUP BY DEPARTMENT HAVING COUNT(WORKER_ID) < 5;

UNION ALL

Q 26 Write a SQL query to show one row twice in results from a table with department 'HR'.

```
select FIRST_NAME, DEPARTMENT from worker W where W.DEPARTMENT='HR'
union all
select FIRST_NAME, DEPARTMENT from Worker W1 where W1.DEPARTMENT='HR';
```

NON CORRELATED SUBQUERY

Q 27 Write a SQL query to fetch the names of workers who earn the highest salary.

SELECT FIRST_NAME, SALARY from Worker WHERE SALARY=(SELECT max(SALARY) from Worker);

NON CORRELATED SUBQUERY

Q 28 Write a SQL query to show the second highest salary from a table.

```
Select max(Salary) from Worker
where Salary not in (Select max(Salary) from Worker);
```

LIMIT / TOP

Q 29 Write a SQL query to show the top n (say 10) records of a table.

Following MySQL query will return the top n records using the LIMIT method:
SELECT * FROM Worker ORDER BY Salary DESC LIMIT 10;

Following SQL Server query will return the top n records using the TOP command:
SELECT TOP 10 * FROM Worker ORDER BY Salary DESC;

Following Oracle query will return the top n records with the help of ROWNUM:
SELECT * FROM (SELECT * FROM Worker ORDER BY Salary DESC)
WHERE ROWNUM <= 10;

LIMIT / TOP

Q 30 Write a SQL query to determine the nth (say n=3) highest salary from a table.

The following MySQL query returns the nth highest salary:

SELECT Salary FROM Worker ORDER BY Salary DESC LIMIT n-1,1;
The following SQL Server query returns the nth highest salary:

```
SELECT TOP 1 Salary
FROM (
 SELECT DISTINCT TOP n Salary
 FROM Worker
 ORDER BY Salary DESC
 )
ORDER BY Salary ASC;
```

CORRELATED SUBQUERY:

Q 31 Write a SQL query to determine the 3rd highest salary without using TOP or limit method.

The following query is using the correlated subquery to return the 3rd highest salary:

```
SELECT Salary
FROM Worker W1
WHERE 2 = (
 SELECT COUNT( DISTINCT ( W2.Salary ) )
 FROM Worker W2
 WHERE W2.Salary >= W1.Salary
 );
```

Use the following generic method to find nth highest salary without using TOP or limit.

```
SELECT Salary
FROM Worker W1
WHERE n-1 = (
 SELECT COUNT( DISTINCT ( W2.Salary ) )
 FROM Worker W2
 WHERE W2.Salary >= W1.Salary
 );
```

CORRELATED SUBQUERY:

Q 32 Write a SQL query to fetch three max salaries from a table.

```
SELECT distinct Salary from worker a WHERE 3 >= (SELECT
count(distinct Salary) from worker b WHERE a.Salary <= b.Salary)
order by a.Salary desc;
```

CORRELATED SUBQUERY:

Q 33 Write a SQL query to fetch n max salaries from a table.

SELECT distinct Salary from worker a WHERE n >= (SELECT count(distinct Salary) from worker b WHERE a.Salary <= b.Salary) order by a.Salary desc;

CORRELATED SUBQUERY:

Q 34 Write a SQL query to fetch three min salaries from a table.

SELECT distinct Salary from worker a WHERE 3 >= (SELECT count(distinct Salary) from worker b WHERE a.Salary >= b.Salary) order by a.Salary asc;

CROSS JOIN

Q 35 Write a SQL query to fetch the list of employees with the same salary.

```
Select distinct W.WORKER_ID, W.FIRST_NAME, W.Salary
from Worker W, Worker W1
where W.Salary = W1.Salary
and W.WORKER_ID != W1.WORKER_ID;
```

INNER JOIN

Q 36 Write a SQL query to print details of the Workers who are also Managers.

```
SELECT DISTINCT W.FIRST_NAME, T.WORKER_TITLE
FROM Worker W
INNER JOIN Title T
ON W.WORKER_ID = T.WORKER_REF_ID
AND T.WORKER_TITLE in ('Manager');
```

INNER JOIN

Q 37 Write a SQL query to find the first name, last name, salary, and job grade for all employees.

```
SELECT W.FIRST_NAME, W.LAST_NAME, W.Salary, J.grade_level
 FROM Worker W JOIN job_grades J
 ON W.salary BETWEEN J.lowest_sal AND J.highest_sal;
```

INNER JOIN

Q 38 Write a SQL query to compute the average salary of Workers for each job title.

```
SELECT WORKER_TITLE, AVG(Salary)
FROM Worker
NATURAL JOIN Title
GROUP BY WORKER_TITLE;
```

INNER JOIN

Q 39 Write a SQL query to print the name of employees having the highest salary in each department.

```
SELECT t.DEPARTMENT,t.FIRST_NAME,t.Salary from
(SELECT max(Salary) as TotalSalary,DEPARTMENT from Worker
group by DEPARTMENT) as TempNew
Inner Join
Worker t on TempNew.DEPARTMENT=t.DEPARTMENT
 and TempNew.TotalSalary=t.Salary;
```

VIEW

Q 40 Write a SQL query to create a view for those Worker belonging to the Department 'HR'

```
CREATE VIEW HRStaff
AS SELECT *
FROM Worker
WHERE DEPARTMENT = 'HR';
```

For Further Practice:

HackerRank **LINK**