

## /\*50 Database Questions (Create a database with name ORG)\*/

```
CREATE DATABASE ORG;
```

```
--SHOW DATABASES;
```

```
USE ORG;
```

```
CREATE TABLE Worker
```

```
(  
WORKER_ID INT NOT NULL PRIMARY KEY,  
FIRST_NAME CHAR(25),  
LAST_NAME CHAR(25),  
SALARY INT,  
JOINING_DATE DATETIME,  
DEPARTMENT CHAR(25)  
)
```

```
INSERT INTO Worker(WORKER_ID, FIRST_NAME, LAST_NAME, SALARY, JOINING_DATE, DEPARTMENT)  
VALUES(001, 'Monika', 'Arora', 1000, '2021-12-10', 'HR'),  
(002, 'Niharika', 'Verma', 8000, '2011-06-14 ', 'Admin'),  
(003, 'Vishal', 'Singhal', 3000, '2020-02-14 ', 'HR'),  
(004, 'Amitabh', 'Singh', 5000, '2020-02-14 ', 'Admin'),  
(005, 'Vivek', 'Bhati', 5000, '2011-06-10 ', 'Admin'),  
(006, 'Vipul', 'Diwan', 2000, '2014-06-11 ', 'Account'),  
(007, 'Satish', 'Kumar', 7500, '2020-01-20 ', 'Account'),  
(008, 'Geetika', 'Chauhan', 9000, '2011-04-11 ', 'Admin');
```

```
CREATE TABLE Bonus (  
WORKER_REF_ID INT,  
BONUS_AMOUNT INT,  
BONUS_DATE DATETIME,  
FOREIGN KEY (WORKER_REF_ID)  
REFERENCES Worker(WORKER_ID)  
ON DELETE CASCADE  
);
```

```
INSERT INTO Bonus  
(WORKER_REF_ID, BONUS_AMOUNT, BONUS_DATE)  
VALUES  
(001, 5000, '2016-02-20'),  
(002, 3000, '2016-06-11'),  
(003, 4000, '2016-02-20'),  
(001, 4500, '2016-02-20'),  
(002, 3500, '2016-06-11');
```

```
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  
);
```

```
INSERT INTO Title  
(WORKER_REF_ID, WORKER_TITLE, AFFECTED_FROM) VALUES  
(001, 'Manager', '2016-02-20 00:00:00'),  
(002, 'Executive', '2016-06-11 00:00:00'),  
(008, 'Executive', '2016-06-11 00:00:00'),  
(005, 'Manager', '2016-06-11 00:00:00'),  
(004, 'Asst. Manager', '2016-06-11 00:00:00'),  
(007, 'Executive', '2016-06-11 00:00:00'),  
(006, 'Lead', '2016-06-11 00:00:00'),  
(003, 'Lead', '2016-06-11 00:00:00');
```

```

-----

select *from Worker
select *from Bonus
select *from Title

/*1)Write an SQL query to fetch "FIRST_NAME" from Worker table using the alias
name as <WORKER_NAME>. */
Select FIRST_NAME AS WORKER_NAME from Worker;

/*2)Write an SQL query to fetch "FIRST_NAME" from Worker table in upper case. */
Select upper(FIRST_NAME) from Worker;

/*3)Write an SQL query to fetch unique values of DEPARTMENT from Worker table. */
Select distinct DEPARTMENT from Worker;

/*4) Write an SQL query to print the first three characters of FIRST_NAME from
Worker table.*/
Select substring(FIRST_NAME,1,3) as First_Name from Worker;

/*5)Write an SQL query to find the position of the alphabet ('a') in the first name
column 'Amitabh' from Worker table.*/

Select CHARINDEX('a',FIRST_NAME) from Worker where FIRST_NAME = 'Amitabh';

/*6) Write an SQL query to print the FIRST_NAME from Worker table after removing
white spaces from the right side.*/

Select RTRIM(FIRST_NAME) from Worker;
select *from Worker

/*7)Write an SQL query to print the DEPARTMENT from Worker table after removing
white spaces from the left side.*/

Select LTRIM(FIRST_NAME) from Worker;
select *from Worker

/*8)Write an SQL query that fetches the unique values of DEPARTMENT from Worker
table and prints its length.*/

Select distinct len(DEPARTMENT), DEPARTMENT from Worker;

/*9)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;

/*10)Write an 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;

/*11)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 asc;

/*21)Write an SQL query to fetch the count of employees working in the department
'Admin'.*/

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

/*22) Write an SQL query to fetch worker names with salaries >= 5000 and <= 10000*/

SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) As Worker_Name, Salary FROM worker
WHERE WORKER_ID IN(SELECT WORKER_ID FROM worker WHERE Salary BETWEEN 5000 AND 10000);

```

/\*23) 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_Workers'  
FROM worker  
GROUP BY DEPARTMENT  
ORDER BY No_Of_Workers DESC, DEPARTMENT ASC;
```

/\*24) Write an 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');
```

--OR--

```
SELECT DISTINCT Worker.FIRST_NAME, Title.WORKER_TITLE  
FROM Worker  
INNER JOIN Title  
ON Worker.WORKER_ID = Title.WORKER_REF_ID  
AND Title.WORKER_TITLE in ('MANAGER');
```

/\*25) Write an SQL query to fetch duplicate records having matching data in some fields of a table. \*/

```
SELECT WORKER_TITLE, AFFECTED_FROM, COUNT(*)  
FROM Title  
GROUP BY WORKER_TITLE, AFFECTED_FROM  
HAVING COUNT(*) > 1;  
select *from Title
```

/\*26) Write an SQL query to show only odd rows from a table.\*/

```
SELECT * FROM Worker WHERE (WORKER_ID % 2) != 0;
```

--OR

```
SELECT * FROM Worker WHERE (WORKER_ID % 2) <> 0;
```

/\*27) Write an SQL query to show only even rows from a table.\*/

```
SELECT * FROM Worker WHERE (WORKER_ID % 2) = 0;
```

/\*28) Write an SQL query to clone a new table from another table.\*/

--The general query to clone a table with data is:

```
SELECT * INTO WorkerClone FROM Worker;  
SELECT *FROM WorkerClone
```

--The general way to clone a table without information is:

```
SELECT * INTO WorkerClone1 FROM Worker WHERE 1 = 0;  
SELECT *FROM WorkerClone1
```

/\*29) Write an SQL query to fetch intersecting records of two tables.\*/

```
SELECT * FROM Worker INTERSECT SELECT * FROM WorkerClone;
```

/\*30) Write an SQL query to show records from one table that another table does not have.\*/

```
SELECT * FROM Worker  
WHERE JOINING_DATE not in (SELECT c.AFFECTED_FROM from Title c);  
select *from Title
```

```
/*31) Write an SQL query to show the current date and time.*/
```

```
SELECT getdate() AS 'CURRENT DATE AND TIME';
```

```
/*32) Write an SQL query to show the top n (say n=10) records of a table.*/
```

```
SELECT TOP 10 * FROM Worker ORDER BY Salary DESC;
```

```
/*33) Write an SQL query to determine the nth (say n=5) highest salary from a table.*/
```

```
SELECT FIRST_NAME, MAX(Salary) as 'Salary'  
FROM Worker group by FIRST_NAME,SALARY ORDER BY Salary DESC;
```

```
-----  
SELECT * FROM Worker ORDER BY SALARY DESC  
SELECT MAX(SALARY) FROM Worker  
WHERE SALARY<(SELECT MAX(SALARY) FROM Worker)
```

```
-----  
SELECT * FROM Worker ORDER BY SALARY DESC  
SELECT DISTINCT TOP 2 SALARY  
FROM Worker  
ORDER BY SALARY DESC
```

```
-----  
SELECT SALARY,  
DENSE_RANK() OVER (ORDER BY SALARY DESC) AS SALARY_RANK  
FROM Worker
```

```
-----  
SELECT *FROM [DBO].Worker ORDER BY SALARY DESC  
GO  
WITH RESULT AS  
(  
    SELECT SALARY,  
    DENSE_RANK() OVER (ORDER BY SALARY DESC) AS DENSERANK  
    FROM Worker  
)  
SELECT TOP 1 SALARY  
FROM RESULT  
WHERE DENSERANK = 3
```

```
/*34) Write an SQL query to determine the 5th highest salary without using TOP or  
limit method.*/
```

```
--The following query is using the correlated subquery to return the 5th highest salary:
```

```
SELECT Salary  
FROM Worker W1  
WHERE 4 = (  
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.
```

```
declare @n int = 3  
SELECT Salary  
FROM Worker W1  
WHERE @n-1 = (SELECT COUNT( DISTINCT ( W2.Salary ) )  
FROM Worker W2  
WHERE W2.Salary >= W1.Salary  
);
```

```
/*35) Write an 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;
```

/\*36) Write an 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);
```

/\*37) Write an SQL query to show one row twice in results from a table.\*/

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

/\*38) Write an SQL query to fetch intersecting records of two tables.\*/

```
(SELECT * FROM Worker)
INTERSECT
(SELECT * FROM WorkerClone);
```

/\*39) Write an SQL query to fetch the first 50% records from a table.\*/

```
SELECT *
FROM WORKER
WHERE WORKER_ID <= (SELECT count(WORKER_ID)/2 from Worker);
```

/\*40) Write an 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;
```

/\*41) Write an 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;
```

/\* Write an SQL query to show the last record from a table.\*/

```
Select * from Worker where WORKER_ID = (SELECT max(WORKER_ID) from Worker);
```

/\*43) Write an SQL query to fetch the first row of a table.\*/

```
Select * from Worker where WORKER_ID = (SELECT min(WORKER_ID) from Worker);
```

/\*44) Write an SQL query to fetch the last five records from a table.\*/

```
SELECT * FROM Worker WHERE WORKER_ID <=5
UNION
SELECT * FROM (SELECT * FROM Worker W) AS W1
WHERE W1.WORKER_ID <=5;
```

/\*45) Write an 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;
```

/\*46) Write an 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;
```

/\*47) Write an 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 desc;
```

/\*48) Write an SQL query to fetch nth(say n=5) max salaries from a table.\*/

```
SELECT distinct Salary from worker a WHERE 5 >= (SELECT count(distinct Salary) from  
worker  
b WHERE a.Salary <= b.Salary) order by a.Salary desc;
```

/\*49) Write an SQL query to fetch departments along with the total salaries paid for each of them.\*/

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

/\*50) Write an 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);
```