https://www.techbeamers.com/sql-query-questions-answers-for-practice/

WORKER_ID INT NOT NULL PRIMARY KEY AUTO_INCREMENT,→ my sql

WORKER_ID INT NOT NULL PRIMARY KEY **IDENTITY, → SQL server**

```
USE dkvORG;

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', 100000, '20140220 09:00:00', 'HR'),
		(002, 'Niharika', 'Verma', 80000, '20140611 09:00:00', 'Admin'),
		(003, 'Vishal', 'Singhal', 300000, '20140220 09:00:00', 'HR'),
		(004, 'Amitabh', 'Singh', 500000, '20140220 09:00:00', 'Admin'),
		(005, 'Vivek', 'Bhati', 500000, '20140611 09:00:00', 'Admin'),
		(006, 'Vipul', 'Diwan', 200000, '20140611 09:00:00', 'Account'),
		(007, 'Satish', 'Kumar', 75000, '20140120 09:00:00', 'Account'),
		(008, 'Geetika', 'Chauhan', 90000, '20140411 09:00:00', 'Admin');




	CREATE TABLE Bonus (
	WORKER_REF_ID INT,
	BONUS_AMOUNT INT,
	BONUS_DATE DATETIME);


	INSERT INTO Bonus
	(WORKER_REF_ID, BONUS_AMOUNT, BONUS_DATE) VALUES
		(001, 5000, '20160220'),
		(002, 3000, '20160611'),
		(003, 4000, '20160220'),
		(001, 4500, '20160220'),
		(002, 3500, '20160611');

	CREATE TABLE Title (
	WORKER_REF_ID INT,
	WORKER_TITLE CHAR(25),
	AFFECTED_FROM DATETIME,
);
```

```
INSERT INTO Title
      (WORKER_REF_ID, WORKER_TITLE, AFFECTED_FROM) VALUES
 (001, 'Manager', '20160220 00:00:00'),
 (002, 'Executive', '20160611 00:00:00'),
 (008, 'Executive', '20160611 00:00:00'),
 (005, 'Manager', '20160611 00:00:00'),
 (004, 'Asst. Manager', '20160611 00:00:00'),
 (007, 'Executive', '20160611 00:00:00'),
 (006, 'Lead', '20160611 00:00:00'),
 (003, 'Lead', '20160611 00:00:00');
```

Q-2. Write An SQL Query To Fetch “FIRST_NAME” values From Worker Table In Upper Case.

Ans.

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

Q-4. Write An SQL Query To Print The First Three Characters Of FIRST_NAME From Worker Table.

Ans.

```
Select substring(FIRST_NAME,1,3) from Worker;
```

Q-6. Write An SQL Query To Print The FIRST_NAME From Worker Table After Removing White Spaces From The Right Side.

Ans.

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

Q-7. Write An SQL Query To Print The DEPARTMENT From Worker Table After Removing White Spaces From The Left Side.

Ans.

```
Select LTRIM(DEPARTMENT) from Worker;
```

Q-8. Write An SQL Query That Fetches The Unique Values Of DEPARTMENT From Worker Table And Prints Its Length.

Ans.

```
Select distinct len(DEPARTMENT) from Worker;  //sql server
Select distinct length(DEPARTMENT) from Worker; //my sql
```

Q-9. Write An SQL Query To Print The FIRST_NAME From Worker Table After Replacing ‘a’ With ‘A’.

Ans.
The required query is:

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

Q-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.

Ans.
The required query is:

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

Q-14. Write An SQL Query To Print Details Of Workers Excluding First Names, “Vipul” And “Satish” From Worker Table.

Ans.
The required query is:

```
Select * from Worker where FIRST_NAME not in ('Vipul','Satish');
```

Q-16. Write An SQL Query To Print Details Of The Workers Whose FIRST_NAME Contains ‘A’.

Ans.
The required query is:

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

Starts with= ‘a%’

Ends with = ‘%a’

Contains=’%a%’

Q-18. Write An SQL Query To Print Details Of The Workers Whose FIRST_NAME Ends With ‘H’ And Contains Six Alphabets.

Ans.
The required query is:

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

Note-there are first 5 char as underscore.

Q-20. Write An SQL Query To Print Details Of The Workers Who Have Joined In Feb’2014.

Ans.
The required query is:

```
Select * from Worker where year(JOINING_DATE) = 2014 and
month(JOINING_DATE) = 2;
```

To include day of a date:

```
Select * from Worker where year(JOINING_DATE) = 2014 and month(JOINING_DATE) = 2 and
day(JOINING_DATE)=20;
```

Error for below query:

```
SELECT DEPARTMENT, COUNT(*) FROM worker WHERE DEPARTMENT = 'Admin'; //shows error
```

```
Error: Column 'worker.DEPARTMENT' is invalid in the select list because it is not
contained in either an aggregate function or the GROUP BY clause.
```

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

Q-23. Write An SQL Query To Fetch The No. Of Workers For Each Department In The Descending Order.

Ans.
The required query is:

```
SELECT DEPARTMENT, count(WORKER_ID) No_Of_Workers
```

```
FROM worker
GROUP BY DEPARTMENT
ORDER BY No_Of_Workers DESC;
```

Q-24. Write An SQL Query To Print Details Of The Workers Who Are Also Managers.

Ans.
The required query is:

```
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');
```

Should we put other than join condition in where clause or with join condition using AND as above.

Q-25. Write An SQL Query To Fetch Duplicate Records Having Matching Data In Some Fields Of A Table.

Ans.
The required query is:

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

Q-26. Write An SQL Query To Show Only Odd Rows From A Table.

```
SELECT * FROM Worker WHERE WORKER_ID%2= 1; //for odd rows

OR

SELECT * FROM Worker WHERE WORKER_ID%2!= 0; //for odd rows


SELECT * FROM Worker WHERE WORKER_ID%2= 0;  //for even rows
```

Q-28. Write An SQL Query To Clone A New Table From Another Table.

Ans.
The general query to clone a table with data is:

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

The general way to clone a table without information is:

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

Q-29. Write An SQL Query To Fetch Intersecting Records Of Two Tables.

Ans.
The required query is:

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

Q-30. Write An SQL Query To Show Records From One Table That Another Table Does Not Have.

Ans.
The required query is:

```
SELECT * FROM Worker
MINUS
SELECT * FROM Title;
```

Q-31. Write An SQL Query To Show The Current Date And Time.

Ans.
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;
```

Q-32. Write An SQL Query To Show The Top N (Say 10) Records Of A Table.

Ans.
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;
```

Q-33. Write An SQL Query To Determine The Nth (Say N=5) Highest Salary From A Table.

Ans.
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;
```

Q-34. Write An SQL Query To Determine The 5th Highest Salary Without Using TOP Or Limit Method.

Ans.
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.

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

Q-35. Write An SQL Query To Fetch The List Of Employees With The Same Salary.

Ans.
The required query is:

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

Q-37. Write An SQL Query To Show One Row Twice In Results From A Table.

Ans.
The required query is:

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

```
select FIRST_NAME, DEPARTMENT from Worker W1 where
W1.DEPARTMENT='HR';
```

Q-39. Write An SQL Query To Fetch The First 50% Records From A Table.

Ans.
The required query is:

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

or

```
SELECT top 50 percent * FROM Worker
```

Que : Select salary column and show double of salary present in table.

```
Select salary*2 from worker
```

Note: we can perform mathematical operation with column and among multiple columns in query itself.

```
select worker_id*salary from worker
```

```
select max(worker_id)-min(salary) from worker
```

```
select sum(worker_id+salary) from worker
```

distinct can be used with each aggregate function except min and max

Q-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;
```

Alias has to be in quotes (single or double) if it has spaces.

Q-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;
```

Q-42. Write An SQL Query To Show The Last Record From A Table.

Ans.
The following query will return the last record from the Worker table:

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

Q-43. Write An SQL Query To Fetch The First Row Of A Table.

Ans.
The required query is:

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

Q-45. Write An SQL Query To Print The Name Of Employees Having The Highest Salary In Each Department.

```
select FIRST_Name,Salary,Department from worker where salary in (select max(salary) from
worker group by department)

or
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;
```

find the all the worker ids whose salary is min

```
select worker_id from worker where salary in (select min(salary) from worker)
```

find workers id along with the min salary

```
select worker_id, Salary from worker where salary in (select min(salary) from worker)
```

or

```
select worker_id, (select min(salary) from worker ) as MinSal from worker where salary in
(select min(salary) from worker)   //not optimized query
```

Q-46. Write An SQL Query To Fetch Three Max Salaries From A Table.

```
Select distinct top 3  salary from worker order by salary DESC;
```

Or

```
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;
```

Q-47. Write An SQL Query To Fetch Three Min Salaries From A Table.

```
Select distinct top 3  salary from worker order by salary ASC;

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;
```

Q-50. Write An SQL Query To Fetch The Names Of Workers Who Earn The Highest Salary.

Ans.
The required query is:

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

-======================

to Replace a null value with a given value in query response:

```
select ISNULL(salary,0) from worker
```

Aggregate functions in SQL

Various Aggregate Functions

```
1) Count()
2) Sum()
3) Avg()
4) Min()
5) Max()
```

Now let us understand each Aggregate function with a example:

```
Id     Name     Salary
-----------------------
1       A        80
2       B        40
3       C        60
4       D        70
5       E        60
6       F        Null
```

Count():

Count(*): Returns total number of records .i.e 6.
Count(salary): Return number of Non Null values over the column salary. i.e 5.

Count(Distinct Salary): Return number of distinct Non Null values over the column salary .i.e 4

Sum():

sum(salary): Sum all Non Null values of Column salary i.e., 310
sum(Distinct salary): Sum of all distinct Non-Null values i.e., 250.

Avg():

Avg(salary) = Sum(salary) / count(salary) = 310/5
Avg(Distinct salary) = sum(Distinct salary) / Count(Distinct Salary) = 250/4

Min():

Min(salary): Minimum value in the salary column except NULL i.e., 40.
Max(salary): Maximum value in the salary i.e., 80.

```
select salary from worker order by salary
```

```
1.NULL
2.75000
3.90000
```

if salary is NULL then its is shown as first record when order by is ASC.

```
select TOP 1 salary from worker order by salary // =>NULL
```

```
select min(salary) from worker // ==>75000
```