

## 1. 2nd Highest Salary

```
select max(sal) from emp
where sal not in (select max from emp);
```

For group by use an aggregate function at the beginning

## 2. Department wise highest Salary

Highest salary of each dpt

```
select max(sal),dept_no
from emp
group by dept_no;
```

Count no of emp present in each dpt

```
select count(*)
from emp
group by dept_no;
```

## 3. Display Alternate Records

```
select * from
(select emp_name,emp_id, rownum rn
from emp
order by rn)
where mod(rn,2)!=0
```

- used to display records sorted by rn in ascending order
- =0 for even
- orderby rownum to get displayed in ascending order of rn

```
1
2 select * from
3   (select empno, ename, sal, rownum rn
4     from emp
5     order by rn)
6   where mod (rn, 2) = 0;
```

EMPNO	ENAME	SAL	RN
7698	BLAKE	2850	2
7566	JONES	2975	4
7902	FORD	3000	6

[Download CSV](#)  
3 rows selected.

## 4. Display Duplicate of a Column

```
select ename,count(*)
from emp
group by ename
having count(*)>1;
```

- first get the frequency of count values
- you have to use group by to group the count based on number of employees
- you cannot use where on using group by
- use having to filter based on condns instead of where

## 5. Pattern matching in SQL

a) Display Empname starts with 'M'

```
select ename from emp
where ename like 'M%';
```

b) Empname ends with 'N'

```
select ename from emp
where ename like '%N';
```

c) Empname having 'M' in any position in their name

```
select ename from emp
where ename like '%M%';
```

d) Empname not having 'M' anywhere

```
select ename from emp
where ename not like '%M%';
```

## 6. Pattern searching in SQL- 2

a) Name having exactly 4 letters

```
select ename from emp
where ename like '____';
```

b) Name with i) 2nd letter as 'i' and ii) 4th letter as 'm'

i)

```
select ename from emp
where ename like '_I%';
```

ii)

```
select ename from emp
where ename like '___M%';
```

c) Display emp names whose hire date is in the month of december

-- month has the first 3 letters

```
select hiredate, ename from emp
where hiredate like '%DEC%';
```

d) Empname having exactly 2 'L's

```
select ename from emp
where ename like '%LL%';
```

e) Empname starting with J and ending with S

```
select ename from emp
where ename like 'J%S';
```

## 7. Display nth row in SQL

-- 1st method

```
select * from emp
where rownum <= 4
minus
select * from emp
where rownum <= 3
```

-- MINUS operator in SQL to subtract one result set from another result set to evaluate the result set difference

```
-- Other method  
SELECT * FROM  
(select rownum r students.* FROM students)  
where r in (2,3,7);
```

```
-- where r=4
```

## 10 Nth highest salary

```
select * from ( select distinct empname,sal order by sal desc)  
where rownum<=3  
minus  
select * from ( select distinct empname,sal order by sal desc)  
where rownum<=2
```

## 8. union vs uninonall

*The columns used in all the select statements must have the following*

- ✓ the **same number of columns**
- ✓ Similar or **compatible data types**
- ✓ **same logical order**

### Selecting more than one column in the select query

- ✓ Whenever **more than one column is specified in the select clause** then the combination of all the columns **considered that is if both the values in the row are same then only this considered as a duplicate value**

```
select cityname,postalcode from tabl1  
union  
select cityname,postalcode from tabl2
```

## 9 Get first and last row of the table

```
select * from (select rownum r , ename,sal from emp )  
Where r=1 or r=(select count(*) from emp )
```

## 10 Nth highest salary

```
select * from ( select distinct empname,sal order by sal desc)  
where rownum<=3  
minus  
select * from ( select distinct empname,sal order by sal desc)  
where rownum<=2
```

**1. Write an SQL query to fetch the employees whose name begins with any two characters, followed by a text “hn” and ending with any sequence of characters.**

Ans. For this question, we can create an SQL query using like operator with ‘\_’ and ‘%’ wild card characters, where ‘\_’ matches a single character and ‘%’ matches ‘0 or multiple characters’.

```
SELECT FullName  
FROM EmployeeDetails  
WHERE FullName LIKE ‘__hn%’;
```

**2. Write an SQL query to fetch all the EmpIds which are present in either of the tables – ‘EmployeeDetails’ and ‘EmployeeSalary’.**

Ans. In order to get unique employee ids from both the tables, we can use Union clause which can combine the results of the two SQL queries and return unique rows.

```
SELECT EmpId FROM EmployeeDetails  
UNION  
SELECT EmpId FROM EmployeeSalary;
```

**Write an SQL query to fetch common records between two tables.**

Ans. SQL Server – Using INTERSECT operator-

```
SELECT * FROM EmployeeSalary  
INTERSECT  
SELECT * FROM ManagerSalary;
```

MySQL – Since MySQL doesn't have INTERSECT operator so we can use the sub query-

## Subqueries

```
SELECT *  
FROM EmployeeSalary  
WHERE EmpId IN  
(SELECT EmpId from ManagerSalary);
```

**Write an SQL query to fetch the EmpIds that are present in EmployeeDetails but not in EmployeeSalary.**

Ans. Using sub query-  
SELECT EmpId FROM

EmployeeDetails  
where Empld Not IN  
(SELECT Empld FROM EmployeeSalary);

**Ques.16. Write an SQL query to fetch the employee full names and replace the space with '-'.**

Ans. Using 'Replace' function-

```
SELECT REPLACE(FullName, ' ', '-')  
FROM EmployeeDetails;
```

**Write an SQL query to fetch the position of a given character(s) in a field.**

Ans. Using 'Instr' function-

```
SELECT INSTR(FullName, 'Snow')  
FROM EmployeeDetails;
```

**Ques.18. Write an SQL query to display both the Empld and ManagerId together.**

Ans. Here we can use the CONCAT command.

```
SELECT CONCAT(Empld, ManagerId) as NewId  
FROM EmployeeDetails;
```

**Write an SQL query to update the employee names by removing leading and trailing spaces.**

Ans. Using the 'Update' command with the 'LTRIM' and 'RTRIM' function.

```
UPDATE EmployeeDetails  
SET FullName = LTRIM(RTRIM(FullName));
```

**Write an SQL query to fetch only odd rows from the table.**

Ans. In case we have an auto-increment field e.g. EmpId then we can simply use the below query-

```
SELECT * FROM EmployeeDetails  
WHERE MOD (EmpId, 2) <> 0;
```

In case we don't have such a field then we can use the below queries.

Using Row\_number in SQL server and checking that the remainder when divided by 2 is 1-

```
SELECT E.EmpId, E.Project, E.Salary  
FROM (  
    SELECT *, Row_Number() OVER(ORDER BY EmpId) AS RowNumber  
    FROM EmployeeSalary  
) E  
WHERE E.RowNumber % 2 = 1;
```

**Write an SQL query to fetch only even rows from the table.**

Ans. In case we have an auto-increment field e.g. EmpId then we can simply use the below query-

```
SELECT * FROM EmployeeDetails  
WHERE MOD (EmpId, 2) = 0;
```

**Write an SQL query to fetch top n records?**

Ans. In MySQL using [LIMIT](#)-

```
SELECT *  
FROM EmployeeSalary  
ORDER BY Salary DESC LIMIT N;
```



In SQL server using **TOP command**-

```
SELECT TOP N *  
FROM EmployeeSalary  
ORDER BY Salary DESC;
```

**Write an SQL query to print all Worker details from the Worker table order by FIRST\_NAME Ascending and DEPARTMENT Descending.**

Ans.

The required query is:

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

**Write an SQL query to print details of the Workers who have joined in Feb'2014.**

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

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

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

**Write an SQL query to fetch worker names with salaries >= 50000 and <= 100000. (Use Subquery)**

```
SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) As Worker_Name, Salary  
FROM worker  
WHERE WORKER_ID IN  
(SELECT WORKER_ID FROM worker  
WHERE Salary BETWEEN 50000 AND 100000);
```

**Write an 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;
```

**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;
```

**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;
```

The general way to clone a table without information is:

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

An alternate way to clone a table (for MySQL) without is:

```
CREATE TABLE WorkerClone LIKE Worker;
```

```
Search for "3" in string "W3Schools.com", and return position:
```

```
SELECT INSTR("W3Schools.com", "3") AS MatchPosition;
```

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

```
Select INSTR(FIRST_NAME, BINARY'a') from Worker where FIRST_NAME = 'Amitabh';
```

Write an SQL query to print details for Workers with the first name as “Vipul” and “Satish” from the Worker table.

```
Select * from Worker where FIRST_NAME in ('Vipul','Satish');
```

## What is AUTO\_INCREMENT?

- ❖ AUTO\_INCREMENT is used in SQL to automatically generate a unique number whenever a new record is inserted into a table.
- ❖ Since the primary key is unique for each record, we add this primary field as the AUTO\_INCREMENT field so that it is incremented when a new record is inserted.
- ❖ The AUTO-INCREMENT value is by default starts from 1 and incremented by 1 whenever a new record is inserted.

Syntax:

```
CREATE TABLE Employee(  
Employee_id int NOT NULL AUTO-INCREMENT,  
Employee_name varchar(255) NOT NULL,  
Employee_designation varchar(255)  
Age int,  
PRIMARY KEY (Employee_id)  
)
```



**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  
);
```

(or)

```
Select * from (select distinct salary,emp_id,emp_name order by salary desc )  
Where rownum<=n  
Minus  
Select * from (select distinct salary,emp_id,emp_name order by salary desc )  
Where rownum<=n-1
```

### **Write a query to get the current date.**

You can write a query as follows in SQL Server:

```
SELECT GETDATE () ;
```

You can write a query as follows in [MySQL](#):

```
SELECT SYSDATE () ;
```

**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;
```

---

**Write a query to retrieve the top N records.**

You can write a query using the TOP command in SQL Server:

```
SELECT TOP N *  
FROM EmployeePosition  
ORDER BY Salary DESC;
```

You can also create a query using the LIMIT command in MySQL:

```
SELECT *  
FROM EmpPosition
```

**ORDER BY Salary DESC LIMIT N;**

**Create an SQL query to fetch EmpPostion and the total salary paid for each employee position.**

The query for this request is:

**SELECT EmpPosition, SUM(Salary)**

**from EmployeePosition**

**GROUP BY EmpPosition;**

**Write a query to fetch 50% records from the EmployeeInfo table**

The query for this request is:

**SELECT \***

**FROM EmployeeInfo WHERE**

**EmpID <= (SELECT COUNT(EmpID)/2 from EmployeeInfo);**