FINGERTIPS

SQL

Use Cases

SQL Use cases

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;

select First_Name 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;

select distinct First_Name 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) from Worker;

5 Write an SQL Query to Find the Position of The Alphabet (a) In the First Name Column for Amitabh From Worker Table.

select INSTR(First_Name,Binary'a') 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;

7 Write an SQL Query to Print the DEPARTMENT from Worker Table After Removing White Spaces from The Left Side.

```
select LTRIM(Department) from Worker;
```

8 Write an SQL Query That Fetches the Unique Values of DEPARTMENT from Worker Table and Prints Its Length

```
select distinct length (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;
```

12. Write an 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;
```

13 Write an SQL Query to Print Details for Workers with The First Name as Vipul And Satish From Worker Table

```
select * from worker where First_Name in ('Vipul',"Satish");
```

14 Write an SQL Query to Print Details of Workers Excluding First Names, Vipul And Satish From Worker Table.

```
select * from worker where First_Name not in ('Vipul',"Satish");
```

15 Write an SQL Query to Print Details of Workers with DEPARTMENT Name as Admin"

```
select * from worker where Department like "Admin%";
```

16 Write an SQL Query to Print Details of The Workers Whose FIRST_NAME Contains A

```
select * from worker where First_Name like "%a%";
```

17 Write an SQL Query to Print Details of The Workers Whose FIRST_NAME Ends With A

```
select * from worker where First_Name like "%a";
```

18 Write an SQL Query to Print Details of The Workers Whose FIRST_NAME Ends with H and Contains Six Alphabets

```
select * from worker where First_Name like'_____h';
```

19 Write an SQL Query to Print Details of The Workers Whose SALARY Lies Between 100000 And 500000.

```
select * from worker where salary between 100000 and 500000;
```

20 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;
```

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 >= 50000 And <= 100000

#with 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);
```

#Without Subquery

```
select CONCAT(FIRST_NAME, ' ', LAST_NAME) As Worker_Name, Salary
FROM worker where salary between 50000 and 100000;
```

23 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;
```

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');
```

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;
```

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

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

27 Write An SQL Query To Show Only Even Rows From A Table

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

28 Write An SQL Query To Clone (Copy)A New Table From Another Table

```
create table worker5 select * from worker;
```

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

```
select * from Worker
intersect
select * from Worker3;
```

30 Write An SQL Query To Fetch Records which are not available in other table

```
select * from Worker;

Minus

select * from worker3;
```

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

```
SELECT CURDATE();

SELECT NOW();
```

32 Write An SQL Query To Show The Top N (Say 5) Records Of A Table

```
select * from worker order by Salary desc limit 5;
```

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

```
SELECT Salary FROM Worker ORDER BY Salary DESC LIMIT 5,1;
```

34 Write An SQL Query To rename the column name WORKER_REF_ID to WORKER_RID in bonus table

```
ALTER TABLE bonus Change WORKER_REF_ID WORKER_RID int;
```

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.

# Max Salary

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

#Union only returns unique records

select * from worker

union

select * from worker3;

select Worker_ID,First_Name,Salary from worker

union

select Worker_ID,First_Name,Salary from worker3;

#where clause

select Worker_ID,First_Name,Salary from worker where Department="Admin"

union

select Worker_ID,First_Name,Salary from worker3 where Department="Admin";

38 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);

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;

42 Write an SQL Query to Show the Last Record from A Table

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

Select * from Worker order by Worker_ID DESC limit 1;

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 remove joining date from table.

ALTER TABLE Worker DROP COLUMN JOINING_DATE;

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

```
SELECT Max(FIRST_NAME), Max(Salary) from Worker group by department;
```

46 Write an SQL Query To change the LAST_NAME as Bhatt of worker_id =005.

```
update worker
set Last_Name="Bhatt"
where `Worker_ID`=005;
select * from worker;
```

47 Write an SQL Query To change the salary to 100000 where worker name is Satish

```
UPDATE Worker
SET SALARY= 100000
WHERE FIRST_NAME = Satish;
```

48 Write an SQL Query To delete the employee details where id is 003.

```
Delete from worker
Where worker_id = 003;
```

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 Add Newcolumn (int) in to worker table.

```
alter table worker add column Newcol2 int;
```

#If Clause/Case

51 Write an SQL Query To Add A class for 100000+ salary and B for others.

```
select *, if (SALARY > 100000, "A","B") as class from worker;
```

52. Write an SQL Query To View a Virtual Table based on the results of an SQL Statement.

```
create view HR_Department AS

select First_Name, Last_Name,Salary, Department

from Worker

where Department="HR";

select * from HR_Department;
```

53. The following MYSQL ensures that the "ID", "LastName", and "FirstName" columns will NOT accept NULL values when the "Persons" table is created:

```
CREATE TABLE Persons (

   ID int NOT NULL,

   LastName varchar(255) NOT NULL,

   FirstName varchar(255) NOT NULL,

   Age int

);
```

54. The following SQL creates a UNIQUE constraint on the "ID" column when the "Persons" table is created:

The UNIQUE constraint ensures that all values in a column are different

```
CREATE TABLE Persons (
   ID int NOT NULL,
   LastName varchar(255) NOT NULL,
   FirstName varchar(255),
   Age int,
   UNIQUE (ID)
);
```

55. The following SQL creates a PRIMARY KEY on the "ID" column when the "Persons" table is created:

```
CREATE TABLE Persons (
   ID int NOT NULL,
   LastName varchar(255) NOT NULL,
   FirstName varchar(255),
   Age int,
   PRIMARY KEY (ID)
);
```

56. The following SQL statement defines the "Personid" column to be an auto-increment primary key field in the "Persons" table:

Auto-increment allows a unique number to be generated automatically when a new record is inserted into a table.

```
CREATE TABLE Persons (
   Personid int NOT NULL AUTO_INCREMENT,
   LastName varchar(255) NOT NULL,
   FirstName varchar(255),
   Age int,
   PRIMARY KEY (Personid)
);
```