SQL Tutorials

Top 50 SQL Queries for Interview

EmployeeDetails Table

EmpId	FullName	ManagerId	DateOfJoining	City
1	Praful Sharma	100	01/31/2019	Jhansi
2	Manglam Sen	105	01/30/2023	Kolkata
3	Mohit Agarwal	107	27/11/2022	New Delhi

EmployeeSalary Table

EmpId	Project	Salary	Variable
1	P1	8000	400
3	P2	7000	1000
4	P1	12000	0

Q1.

- Print all Records from EmployeeDetails Table ?

Answer :

```
Select * From EmployeeDetails ;
```

Q2.

- Print details of the Employee whose Employee Id is 1.

Answer :

```
Select * From EmployeeDetails where empid = 1 ;
```

Q3.

- Print details of the All Employees Whose Manager ID is 100 And Their City is Jhansi

Answer :

```
Select * From EmployeeDetails where managerid = 100 and city = 'Jhansi' ;
```

Q4.

- Print All Projects Available in EmployeeSalary Table

Answer :

```
SELECT DISTINCT(Project) FROM EmployeeSalary;
```

Q5.

- Fetch Count of Employees Working in P1 Project ?

Answer :

```
SELECT COUNT(*) FROM EmployeeSalary WHERE Project = 'P1';
```

Q6.

- Write an SQL query to find the maximum, minimum, and average salary of the employees.

Answer :

```
SELECT Max(Salary),
Min(Salary),
AVG(Salary)
FROM EmployeeSalary;
```

Q7.

- Write an SQL query to find the employee id whose salary lies in the range of 9000 and 15000.

Answer :

```
SELECT EmpId, Salary
FROM EmployeeSalary
WHERE Salary BETWEEN 9000 AND 15000;
```

Q8.

- Print All Employees Id Who live in Jhansi City or Their Manager Id is 100

Answer :

```
SELECT Empid
FROM EmployeeDetails
where city= 'Jhansi' or managerid = 100;
```

Q9.

- Write an SQL query to fetch all those employees who work on Projects other than P2.

Answer :

```
SELECT EmpId
FROM EmployeeSalary
WHERE NOT Project='P2';
```

Q9.

- Write an SQL query to fetch all those employees who work on Projects other than P2.

Answer :

```
SELECT EmpId
FROM EmployeeSalary
WHERE NOT Project='P2';
```

Or

```
SELECT EmpId
FROM EmployeeSalary
WHERE Project <> 'P2';
```

Q10.

- Write an SQL query to display the total salary of each employee adding the Salary with Variable value.

Answer :

```
SELECT EmpId,
Salary+Variable as TotalSalary
FROM EmployeeSalary;
```

Q11.

- Write an SQL query to display the Names of the Employee Where Second Letter of the Name is a.

Answer :

```
SELECT FullName
FROM EmployeeDetails
WHERE FullName LIKE '_a%';
```

Q12.

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

Answer :

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

Q13.

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

Answer :

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

Q14.

- Write an SQL query to fetch the EmpIds that are present in both the tables – ‘EmployeeDetails’ and ‘EmployeeSalary.

Answer :

```
SELECT EmpId FROM
EmployeeDetails
where EmpId IN
(SELECT EmpId FROM EmployeeSalary);
```

Q15.

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

Answer :

```
SELECT EmpId FROM
EmployeeDetails
where EmpId Not IN
(SELECT EmpId FROM EmployeeSalary);
```

Q16.

- Write an SQL query to fetch the employee’s full names and replace the space with ‘-’.

Answer :

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

Q17.

- Write an SQL query to display both the EmpId and ManagerId together.

Answer :

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

Q18.

- Write an SQL query to display both the EmpId and ManagerId together.

Answer :

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

Q19.

- Write a query to fetch only the first name(string before space) from the FullName column of the EmployeeDetails table.

Answer :

```
SELECT MID(FullName, 1, LOCATE(' ',FullName))
FROM EmployeeDetails;
```

Q20.

- Write an SQL query to uppercase the name of the employee and lowercase the city values.

Answer :

```
SELECT UPPER(FullName), LOWER(City)
FROM EmployeeDetails;
```

Q21.

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

Answer :

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

Q21.

- Write an SQL query to fetch employee names having a salary greater than or equal to 5000 and less than or equal to 10000.

Answer :

```
SELECT FullName
FROM EmployeeDetails
WHERE EmpId IN
(SELECT EmpId FROM EmployeeSalary
WHERE Salary BETWEEN 5000 AND 10000);
```

Q22.

- Write an SQL query to fetch all the Employee details from the EmployeeDetails table who joined in the Year 2022

Answer :

```
SELECT * FROM EmployeeDetails
WHERE YEAR(DateOfJoining) = '2022';
```

Q22.

- Write an SQL query to fetch all the Employee details from the EmployeeDetails table who joined in the Year 2022

Answer :

```
SELECT * FROM EmployeeDetails
WHERE YEAR(DateOfJoining) = '2022';
```

Q23.

- Write an SQL query to fetch all employee records from the EmployeeDetails table who have a salary record in the EmployeeSalary table.

Answer :

```
SELECT * FROM EmployeeDetails E
WHERE EXISTS
(SELECT * FROM EmployeeSalary S
WHERE  E.EmpId = S.EmpId);
```

Q24.

- Write an SQL query to fetch all employee records from the EmployeeDetails table who have a salary record in the EmployeeSalary table.

Answer :

```
SELECT * FROM EmployeeDetails E
WHERE EXISTS
(SELECT * FROM EmployeeSalary S
WHERE  E.EmpId = S.EmpId);
```

Q25.

- Write an SQL query to fetch the project-wise count of employees sorted by project's count in descending order.

Answer :

```
SELECT Project, count(EmpId) ProjectCount
FROM EmployeeSalary
GROUP BY Project
ORDER BY EmpProjectCount DESC;
```

Q26.

- Write an SQL query to fetch all the Employees who are also managers from the EmployeeDetails table.

Answer :

```
SELECT DISTINCT E.FullName
FROM EmployeeDetails E
INNER JOIN EmployeeDetails M
ON E.EmpID = M.ManagerID;
```

Q27.

- Write an SQL query to fetch records from EmployeeDetails Where Manager Id is Coming More than Once.

Answer :

```
SELECT * from employeedetails
WHERE ManagerId in (SELECT ManagerId
FROM EmployeeDetails
GROUP BY ManagerId
HAVING count(ManagerId)>1);
```

Q27.

- Write an SQL query to fetch records from EmployeeDetails Where Manager Id is Coming More than Once.

Answer :

```
SELECT * from employeedetails
WHERE ManagerId in (SELECT ManagerId
FROM EmployeeDetails
GROUP BY ManagerId
HAVING count(ManagerId)>1);
```

Q28.

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

Answer :

```
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;
```

Q29.

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

Answer :

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

Q30.

- Write an SQL query to create a new table with data and structure copied from another table.

Answer :

```
CREATE TABLE NewTable
SELECT * FROM EmployeeSalary;
```

Q31.

- Write an SQL query to fetch top n records.

Answer :

```
SELECT *
FROM EmployeeSalary
ORDER BY Salary DESC LIMIT 3;
```

Q31.

- Write SQL query to find the 3rd highest salary from a table without using the TOP/limit keyword.

Answer :

```
SELECT Salary
FROM EmployeeSalary Emp1
WHERE N-1 = (
            SELECT COUNT( DISTINCT ( Emp2.Salary ) )
            FROM EmployeeSalary Emp2
            WHERE Emp2.Salary > Emp1.Salary
)
```

Q31.

- Order Employee names Based On Alphabetical Order.

Answer :

```
SELECT Fullname from EmployeeDetails ORDER by Fullname;
```

Q32.

- Order Employee Names And Salary Based On Salary.

Answer :

```
SELECT Fullname , Salary From employeedetails E ,
employeesalary ES
Where E.EmpId = ES.EmpId ORDER by ES.Salary;
```

Q33.

- Print Total Salary Going from Each Project

Answer :

```
SELECT ES.Project , ES.Salary from employeesalary ES
GROUP by es.Project;
```

Q34.

- Print All Employee Details Whose Joining Date is Not in Last Year

Answer :

```
SELECT * FROM employeedetails
WHERE (DateOfJoining < CURRENT_DATE - INTERVAL 1 YEAR) ;
```

Q35.

- Print All Employee Who Gets Paid Above Average Salary

Answer :

```
SELECT * From employeedetails , employeesalary where
employeedetails.EmpId = employeesalary.EmpId
and employeesalary.Salary > (SELECT avg(salary) from
employeesalary);
```

Q36.

- Print All Employees Who is in the company for more than 4 years

Answer :

```
SELECT * from employeedetails WHERE year(CURRENT_DATE) - year(DateOfJoining) > 4;
```

Q37.

- Print All Employees With Total Number of years as Service.

Answer :

```
SELECT * , (year(CURRENT_DATE) - year(DateOfJoining)) as
'Service' from employeedetails ;
```

Q38.

- Print Total Employees in Each Project.

Answer :

```
Select employeesalary.Project,count(*) as 'Total Employees' from
employeesalary GROUP by employeesalary.project;
```

Q39.

- Return List of All Manager Order By Total number of employees managed by them.

Answer :

```
SELECT ManagerId , COUNT(*) as NumEmployees From Employeedetails
GROUP by ManagerId
ORDER by NumEmployees;
```

Q40.

- Return List of All Employee who are serving for more than 2 Years and not in Project P2 And P3

Answer :

```
SELECT * from employeesalary
where Project NOT IN('P2' , 'P3')
AND EmpId IN
(Select EmpId from employeedetails
where year(CURRENT_DATE) - year(dateofjoining) > 2);
```

Q41.

- Select Average Salary from Each Project

Answer :

```
SELECT AVG(Salary) , Project from employeesalary
GROUP by project;
```

Q41.

- Select Project with total Salary whose total employees salary sum is greater than the maximum of average salary project wise.

Answer :

```
SELECT Project , SUM(Salary)
From
employeesalary
GROUP by Project
HAVING sum(Salary) > (
SELECT max(Average.avgsalary) from (SELECT avg(Salary) as avgsalary
,project from employeesalary group by project) as Average);
```

Q42.

- Add new column role in EmployeeDetails

Answer :

```
ALTER TABLE Employeedetails
ADD Role varchar(255);
```

Q43.

- Update the value of Role if Salary + variable < 2000 then Analyst , Otherwise Sr Analyst.

Answer :

```
UPDATE employeedetails ed
INNER join employeesalary es on ed.EmpId = es.EmpId
set ed.Role = (
    CASE
                    WHEN es.Salary + es.Variable < 20000
                      THEN 'Analyst'
                    ELSE 'Sr Analyst’ END)
```

Q44.

- Produce the output as Name(Role)

Answer :

```
SELECT Concat(FullName , '(' , Role , ')') AS "EmployeeWithRole"
FROM employeedetails;
```

Q45.

- Display Total Number of Characters in Employee Name

Answer :

SELECT fullname , length(trim(FullName))-1 as Namelength from employeedetails

Q46.

- Display all details of employee whose Total salary will be more than 20000 after increasing salary by 20%

Answer :

```
SELECT * from employeedetails ed , employeesalary es WHERE ed.EmpId = es.EmpId And (es.Variable + es.Salary+es.Salary*0.2) > 20000 and es.Salary < 20000;
```

Q47.

- Display all Employees who joined in January

Answer :

```
SELECT * from employeedetails
where Monthname(DateOfJoining ) = 'january'
```

Q48.

- Return all manager id’s which are not Present in Employeedetails table as Empid

Answer :

```
SELECT ManagerId FROM employeedetails
WHERE ManagerId NOT IN
(SELECT EmpId from employeedetails) ;
```

Q49.

- Print Total Experience in Years Months Days Format

Answer :

```
SELECT
   CONCAT(TIMESTAMPDIFF( YEAR, DateOfJoining, CURDATE() ) , ' Years ' ,
   TIMESTAMPDIFF( MONTH, DateOfJoining, CURDATE() ) % 12 , ' Months ' ,
   FLOOR( TIMESTAMPDIFF( DAY, DateOfJoining, CURDATE() ) % 30 ) , ' Days ' )
as TotalExperience

   from employeedetails
```

Q49.

- Return Employees with even salary

Answer :

```
SELECT * from employeesalary where mod(salary,2) = 0;
```

Q50.

- Return Employees with 4 digit salary

Answer :

```
SELECT * from employeesalary WHERE length(employeesalary.Salary) =
4;
```

Q51.

- Return Employees who joined in last 11 months

Answer :

```
SELECT *
FROM employeedetails
WHERE dateofjoining >= CURDATE() - INTERVAL 11 MONTH;
```

Q52.

- Return Employees who did not join in January

Answer :

```
SELECT *
FROM employeedetails
WHERE monthname(dateofjoining) <> 'January'
```

Q53.

- Return Employees who either join on 12 December or 1 January

Answer :

```
SELECT *
FROM employeedetails
WHERE (MONTH(dateofjoining) = 12 AND DAY(dateofjoining) = 12)
   OR (MONTH(dateofjoining) = 1 AND DAY(dateofjoining) = 1);
```

Q54.

- Return Employees whose salary between min salary + 1000 and max salary - 1000

Answer :

```
SELECT *
FROM employeesalary ed
WHERE ed.Salary BETWEEN (SELECT min(salary) from employeesalary)+1000 and
(SELECT max(Salary) from employeesalary) - 10000;
```

Q55.

- Return Employees who work in P1 project and order them by salary

Answer :

```
SELECT * FROM employeedetails , employeesalary WHERE
employeedetails.EmpId = employeesalary.EmpId
and employeesalary.Project = 'P1'
ORDER by salary asc;
```

Q56.

- Print Average salary from each role

Answer :

```
SELECT employeedetails.role , AVG(salary) from employeesalary , employeedetails
WHERE employeesalary.EmpId = employeedetails.EmpId
GROUP by employeedetails.Role ;
```

Q57.

- Print Count of employees , minimum and maximum salary from each role

Answer :

```
SELECT employeedetails.role , min(salary) , max(salary) , count(employeedetails.EmpId) as EmpCount from employeesalary , employeedetails
WHERE employeesalary.EmpId = employeedetails.EmpId
GROUP by employeedetails.Role ;
```

Q. 51

- Employees who did not join in January