

We at The Data Monk hold the vision to make sure everyone in the IT industry has an equal stand to work in an open domain such as analytics. Analytics is one domain where there is no formal under-graduation degree and which is achievable to anyone and everyone in the World.

We are a team of 30+ mentors who have worked in various product-based companies in India and abroad, and we have come up with this idea to provide study materials directed to help you crack any analytics interview.

Every one of us has been interviewing for at least the last 6 to 8 years for different positions like Data Scientist, Data Analysts, Business Analysts, Product Analysts, Data Engineers, and other senior roles. We understand the gap between having good knowledge and converting an interview to a top product-based company.

Rest assured that if you follow our different mediums like our blog cum questions-answer portal [www.TheDataMonk.com](http://www.TheDataMonk.com), our youtube channel - [The Data Monk](#), and our e-books, then you will have a very strong candidature in whichever interview you participate in.

There are many blogs that provide free study materials or questions on different analytical tools and technologies, but we concentrate mostly on the questions which are asked in an interview. We have a set of 100+ books which are available both on Amazon and on [The Data Monk e-shop page](#)

We would recommend you to explore our website, youtube channel, and e-books to understand the type of questions covered in our articles. We went for the question-answer approach both on our website as well as our e-books just because we feel that the best way to go from beginner to advance level is by practicing a lot of questions on the topic.

We have launched a series of 50 e-books on our website on all the popular as well as niche topics. Our range of material ranges from SQL, Python, and Machine Learning algorithms to ANN, CNN, PCA, etc.

We are constantly working on our product and will keep on updating it. It is very necessary to go through all the questions present in this book.

Give a rating to the book on Amazon, do provide your feedback and if you want to help us grow then please subscribe to our Youtube channel.

# SQL Interview Questions

Level : Intermediate

**Q 1. For the Given table Write a SQL Statement for find the resultant table with manager name , Employee name and their cities for the manager and Employee who belongs to same city.**

**Table 1 : Manager**

Manager_id	Name	City	Commission
4001	Rajesh	New york	0.15
4002	Mahesh	paris	0.20
4005	Umesh	London	0.35
4004	Raju	paris	0.40
4006	Rahul	Rome	0.78
4008	Vikas	San jose	0.80

**Table 2: Employee**

Employee_id	Name	City	Manager_id
3001	Nick	New york	4001
3002	hockky	New york	4002
3005	Justein	London	4006

3004	Julian	paris	4008
3006	Fabian	Berlin	4004
3008	Jeff	Moscow	4009

A1. Query :

```

Select Manager.Name as "Manager_Name" , Employee.Name as
"Employee_Name",Employee.City
From Managerrr , Employeee
Where Manager.City = Employee.City ;

```

Manager_Name	Employee_Name	City
Rajesh	Nick	New york
Rajesh	hockky	New york
Umesh	justein	London
Mahesh	julian	paris
Raju	julian	paris

**Q2. Answer the following Questions with the use of given tables.**

**Table : EmployeeDetails**

Emp_id	Full Name	Manager_id	Joining Date	City
121	Neha	321	01/31/2014	Toronto
321	Supriya	986	01/30/2015	California
421	Isha	876	27/11/2016	New Delhi

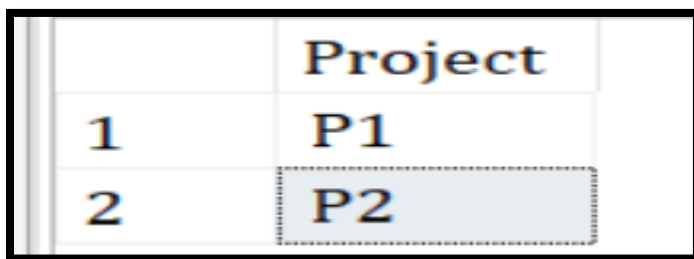
**Table 2: EmployeeSalary**

Emp_id	Project	Salary	Variable
121	P1	8000	500
321	P2	10000	1000
421	P1	12000	0

**1 ) Write an Sql query to fetch all the different projects available in the EmployeeSalary Table.**

A1. To Solve this problem we can use DISTINCT keyword. which will give the total number of distinct elements from the table.

Query : **Select DISTINCT**(Project) **from** EmployeeSalary



The screenshot shows a table with two columns. The first column contains the numbers 1 and 2, representing row indices. The second column is titled 'Project' and contains the values 'P1' and 'P2'. The row containing 'P2' is highlighted with a blue background, indicating it is the current row being viewed or edited.

	Project
1	P1
2	P2

**2) Write an SQL query to fetch all the Employees having salary range between 8000 and 12000.**

A2.

Query: `Select * from EmployeeSalary where Salary BETWEEN 8000 and 12000`

EmpId	Project	Salary	Variable
121	P1	8000	500
321	P2	10000	1000

**3) Write an SQL Query to fetch the Employee name start with 's'.**

A3. For Fetching this information we can use LIKE operator of SQL, Basically LIKE operator is use for Search a given pattern in Columns.

There are two wildcards used with the LIKE operator

1. % represents 0,1,2 or multiple characters.
2. \_ (underscore) represents single character.

Query :

`Select FullName from EmployeeDetails where FullName LIKE '_p%'`

Output:

	FullName
1	Supriya

4) Write an SQL query to fetch all those Empids that are present in both tables.

A4. Query: **Select EmpId From**  
**EmployeeDetails**  
**where EmpId IN**  
**(Select EmpId From EmployeeSalary);**

Output: first table is EmployeeSalary and Second one is EmployeeDetails table and the last table is common EmpIds in both tables.

	EmpId	Project	Salary	Variable
1	121	P1	8000	500
2	321	P2	10000	1000
3	421	P1	12000	0

	EmpId	FullName	ManagerId	Joining Date	City
1	121	Neha	321	2014-01-31	Toronto
2	321	Supriya	986	2015-01-30	California
3	421	Isha	876	2016-11-27	New Delhi

	EmpId
1	121
2	321
3	421

**5) Write an SQL query to check the total number of occurrences of a character 'a' in a FullName Field.**

A5. Query :

```
Select FullName, Len(FullName) - Len(Replace(FullName, 'a', '')) From EmployeeDetails;
```

Output:

	FullName	(No column name)
1	Neha	1
2	Supriya	1
3	Isha	1

**6. Write an Sql query to fetch all Employee records from EmployeeDetails who have Salary record in Employee Salary Table.**

A 6. Query: 

```
Select * From EmployeeDetails e
Where Exists
(Select * From EmployeeSalary s
Where e.EmpId = s.EmpId);
```

Output: This query will fetch records of those Employees who have also salary record in Employee Salary table.

EmpId	FullName	ManagerId	Joining Date	City
121	Neha	321	2014-01-31	Toronto
321	Supriya	986	2015-01-30	California
421	Isha	876	2016-11-27	New Delhi

7). Write an Sql Query to fetch project wise count of Employees which will be sorted by Projects count in descending order.

A7. Query : `Select Project, count(EmpId) ProjectCount  
FROM EmployeeSalary  
GROUP BY Project  
ORDER BY ProjectCount DESC;`

Output :

	Project	ProjectCount
1	P1	2
2	P2	1



**Q3. Write a queries of the all types of joins for the given table.**

A3. Table 1: Student Details

Roll_no	Name	Address	Phone	Age
1	Raju	Uttarakhand	XXXXXXXXXX	20
2	Raj	Dehradun	XXXXXXXXXX	18
3	Abhishek	Herbertpur	XXXXXXXXXX	21
4	Pulkit	Siliguri	XXXXXXXXXX	22
5	Rajat	Rajpur	XXXXXXXXXX	23
6	Aayush	Rudrapur	XXXXXXXXXX	20
7	Palak	premnagar	XXXXXXXXXX	21

Table 2 :

Course_Id	Roll_no
1	1
2	2
2	3
3	4
1	5
4	9
5	10

## Inner Join:

Query : **Select** Student\_IDs.Course\_Id, Student\_Details.Name,  
Student\_Details.Age **From** Student\_Details  
**INNER JOIN** Student\_IDs **on** Student\_Details.ROLL\_NO = Student\_IDs.Roll\_no

Output : The query will show the Name and Age of students enrolled in different- different Courses.

	Course_Id	Name	Age
1	1	Raju	20
2	2	Raj	18
3	2	Abhishek	21
4	3	Pulkit	22
5	1	Rajat	23

## Left Join:

Query : **Select** Student\_Details.Name, Student\_IDs.Course\_Id  
**From** Student\_Details **LEFT JOIN** Student\_IDs **On**  
Student\_Details.Roll\_no = Student\_IDs.Roll\_no;

Output : The query will show the Name, Course Id if that student in any course. Otherwise it will give Null.

	Name	Course_Id
1	Raju	1
2	Raj	2
3	Abhishek	2
4	Pulkit	3
5	Rajat	1
6	Aayush	NULL
7	Palak	NULL

## Right Join :

Query : **Select** Student\_Details.Name,Student\_IDs.Course\_Id  
**From** Student\_Details **RIGHT JOIN** Student\_IDs **on**  
Student\_Details.Roll\_no = Student\_IDs.Roll\_no;

Output: The query will fetch the Name, Course Id if that student enroll in any course. Otherwise it will give Null.

	Name	Course_Id
1	Raju	1
2	Raj	2
3	Abhishek	2
4	Pulkit	3
5	Rajat	1
6	NULL	4
7	NULL	5

## Full Join :

Query : **Select** Student\_Details.Name, Student\_IDs.Course\_Id  
**From** Student\_Details Full Join Student\_IDs **On**  
Student\_Details.Roll\_no = Student\_IDs.Roll\_no;

**Output :** The query will fetch the Name and Course Id from both Tables.

	Name	Course_Id
1	Raju	1
2	Raj	2
3	Abhishek	2
4	Pulkit	3
5	Rajat	1
6	Aayush	NULL
7	Palak	NULL
8	NULL	4
9	NULL	5

**Q 4. Write an Sql query to know which Manager is working for which Employee.**

**Table 1 : Manager**

Manager_id	Name	City	Commission
4001	Rajesh	New york	0.15
4002	Mahesh	paris	0.20
4005	Umesh	London	0.35
4004	Raju	paris	0.40
4006	Rahul	Rome	0.78
4008	Vikas	San jose	0.80

**Table 2: Employee**

Employee_id	Name	City	Manager_id
3001	Nick	New york	4001
3002	hockky	New york	4002
3005	Justein	London	4006
3004	Julian	paris	4008
3006	Fabian	Berlin	4004
3008	Jeff	Moscow	4009

A4. We can use inner join for this question because we have Manager\_id Column is common in both the Tables.

Query : `SELECT a.Name AS "Employee Name",  
a.city, b.name AS "Manager Name", b.Commission  
FROM Employee a  
INNER JOIN Manager b  
ON a.Manager_id=b.Manager_id;`

Output :

The above query will fetch the Employee name, manager name and respective cities, Commission of each employee from both the tables.

Employee Name	city	Manager Name	Commision
Nick	New york	Rajesh	12
hockky	New york	Mahesh	14
Fabian	Berlin	Raju	18
justein	London	Rahul	78
julian	paris	Vikas	45

**Q 5. Write an sql query to fetch Employee Names and salary records. Display the Employee Details ( Names ) Even if Salary Records are present or not for the Employee.**

**Given Table:**

**1. Employee\_Details:**

Emp_id	Full Name	Manager_id	Joining Date	City
121	Neha	321	01/31/2014	Toranto
321	Supriya	986	01/30/2015	California
421	Isha	876	27/11/2016	New Delhi

**2. Employee\_Salary:**

Emp_id	Project	Salary
121	P1	8000
321	P2	10000
421	P1	NULL

A5 . In this Case we want to fetch the Employee Details(Name ) Even if Salary Records are present or not. So, we can use Left join.

Query : **SELECT** E.FullName, S.Salary  
**FROM** Employee\_Details E  
**LEFT JOIN**  
Employee\_Salary S  
**ON** E.EmpId = S.EmpId;

Output : The query will fetch the Employee Name and Salary.

	FullName	Salary
1	Neha	8000
2	Supriya	12000
3	Isha	NULL

#### Q 6. Write a Query for join 3 Tables.

A6. Let Suppose we have 3 Tables, Table A, Table B, Table C. We can use 2 join clauses .

Query :  
**SELECT** column1, column2  
**FROM** TableA



JOIN TableB ON TableA.Column3 = TableB.Column3  
JOIN TableC ON TableA.Column4 = TableC.Column4;

**Q 7. Write an Sql query for fetch only Even rows from the table.**

Given Table :

Emp_id	Project	Salary
121	P1	8000
321	P2	10000
421	P1	NULL
111	P2	1600

Query : **SELECT** E.EmpId, E.Project, E.Salary  
          **FROM** ( **SELECT** \*, **Row\_Number**() **OVER**(**ORDER BY** EmpId) **AS**  
RowNumber **FROM** SalaryEmployee) E  
**WHERE** E.RowNumber % 2 = 0;

Output :

	EmpId	Project	Salary
1	121	P1	8000
2	421	P1	NULL

**Q 8. For the given Table Answer the following questions.**

**Table 1 : Salesman**

Salesman_id	Name	City	Commission
4001	Rajesh	New york	15
4002	Mahesh	paris	20
4005	Umesh	London	35
4004	Raju	paris	40
4006	Rahul	Rome	78
4008	Vikas	San jose	80

**Table 2: Orders**

Order_no.	Pur. amount	Cust. Id	Salesman_id
3001	175	5005	4002
3002	150	5003	4002
3005	175	5004	4001
3004	187	5002	4006
3006	190	5001	4004
3008	198	5006	4008

**Q 1. Write an Sql query to display all the orders from the given orders table issued by 'Mahesh'.**

A1. The Query will fetch the Salesman\_id of the given Salesman..Then because Salesman\_id is common in both tables . So, with the use of Salesman\_id query will fetch all the orders of the Salesman.

Query : `Select * from Orders where Salesman_Id= (Select Salesman_Id from Salesman where Name= 'Mahesh');`

Output :

Order_no.	Pur. amount	Cust. Id	Salesman_Id
3001	175	5005	4002
3002	150	5003	4002

**Q 2. Write an Sql query to display all the orders of the Salesman who belongs to 'New York'.**

A2. Same as Above question the query will fetch the Salesman\_id of the given Salesman and then with the use of Salesman\_id query will fetch all the orders of the Salesman.

Query : `Select * from Orders where Salesman_Id =(Select Salesman_Id from Salesman where City= 'New York');`

Output:

	Order_no.	Pur. amount	Cust. Id	Salesman_Id
1	3005	175	5004	4001

**Q3. Write an Sql query to fetch all the orders issued against the salesman who may works for customer whose id is 5002.**

**A3. Query :** `Select * From Orders  
Where Salesman_id =  
(Select Distinct Salesman_id  
From Orders  
Where [Cust. Id] =5002);`

Output :

Order_no.	Pur. amount	Cust. Id	Salesman_Id
3004	187	5002	4006

**Q 9. For the given Table Answer the following questions.**

**Table 1 : Salesman**

Salesman_id	Name	City	Commission
4001	Rajesh	New york	15
4002	Mahesh	paris	20
4005	Umesh	London	35
4004	Raju	paris	40
4006	Rahul	Rome	78
4008	Vikas	San jose	80

**Table 2 : Customer**

Customer_id	Cust_Name	City	Grade	Salesman_id
3002	Nick	New york	100	4002
3001	hockky	New york	200	4001
3003	Justein	London	300	4005
3004	Julian	paris	100	4008
3005	Fabian	Berlin	200	4006
3006	Jeff	Moscow	100	4002
3007	Josh	Berlin	300	4001

**Q 1. Write a query to display the Commision, Salesman name whose Customers are servicing in 'paris'.**

A1. The query will firstly fetch the Salesman\_id from the Customer table. And then (because Salesman column is common in both tables )with the help of this id we can fetch the Commision of the Salesman.

Query: `Select Commisison , Name from Salesman where Salesman_id =(Select Salesman_Id from Customer where City = 'paris')`

Output:

	Commisison	Name
1	80	Vikas

**Q 2. With the help of Customer table write an Sql query to Count the Customers with grades above New york's average.**

A2. The given question itself contains two questions:

1. Firstly we have to calculate the New york's grade average which is  $(100+200)/2 = 150$ .
2. Then Compare the grade of each column with the New York's grade.

Query : `Select * from Customer  
Select Grade, Count (*) as 'Count'  
From Customer  
group by Grade having Grade >  
(Select AVG(Grade)  
From Customer Where City = 'New York');`

Output:

	Grade	Count
1	200	2
2	300	2

**Q 3. Write an Sql query to find the names of all the salesman who had more than one customer.**

A3. Query : `Select Salesman_id, Name  
From Salesman a Where 1 < (SELECT COUNT(*)  
From Customer  
Where Salesman_id=a.Salesman_id);`

Output:

	Salesman_id	Name
1	4001	Rajesh
2	4002	Mahesh

**Q10. In the Given tables apply group by Clause in Single Column and Multiple Columns.**

A10.

1. Group by Single Column:

**Table : EmployeeInfo**

Sn. No.	Name	Salary	Age
1	Rajat	40000	20
2	Ritesh	50000	18
3	Monika	30000	21
4	Ritesh	20000	22
5	Rajat	10000	23

Query: **Select \* from EmployeeInfo**  
**Select Name, SUM(Salary) From EmployeeInfo**  
**Group by Name;**

Output : The query will group by the salary according to their name.



	Name	(No column name)
1	Monika	30000
2	Rajat	50000
3	Ritesh	70000

## 2 . Group by Multiple Columns:

Table : Student

Subject	Year	Name
Maths	1	Nick
Maths	1	hockky
Maths	2	Justein
Maths	2	Julian
Hindi	1	Fabian
Hindi	1	Jeff
Hindi	1	Josh

Query: **Select Subject, Year, Count(\*) as 'No. of Students'**  
**From Student Group by Subject, Year;**

Output: The query will group by Subject, year columns.

	Subject	Year	No. of Students
1	Hindi	1	3
2	Maths	1	2
3	Maths	2	2

**Q 11. Write an Sql query to find total purchase amount of all the orders.**

**Table 1: Orders**

Order_no.	Pur. amount	Cust. Id	Salesman_id
3001	175	5005	4002
3002	150	5003	4002
3005	175	5004	4001
3004	187	5002	4006
3006	190	5001	4004
3008	198	5006	4008

Query: **Select SUM** (amount) **as** ' **Total Purchase Amount**' **From** Orders;

Output:

	Total Purchase Amount
1	1075

**Q 12. Write a SQL query to find the average Commision amount of the Salesmans.**

**Table : Salesman**

Salesman_id	Name	City	Commission
4001	Rajesh	New york	15
4002	Mahesh	paris	20
4005	Umesh	London	35
4004	Raju	paris	40
4006	Rahul	Rome	78
4008	Vikas	San jose	80

Query : Select Avg (Commisison) as ' Average Commisison'  
From Salesman;

Output:

	Average Commisison
1	44

**Q13. Write a Sql Query to find highest purchase ordered by each customer with their ID and highest purchase amount.**

Table :

Order_no.	Pur. amount	Customerid	Salesman_id
3001	175	5005	4002
3002	150	5003	4002
3005	175	5004	4001
3004	187	5002	4006
3006	190	5001	4004
3008	198	5006	4008

Query :

```
Select Customerid,MAX(amount) as 'Max purchase Amount'
FROM Orders
GROUP BY Customerid
```

Output:

	Customerid	Max purchase Amount
1	5001	190
2	5002	187
3	5003	150
4	5004	175
5	5005	175
6	5006	198

**Q14. Answer the Following Questions on the basis of the given table :  
Teacher**

A14. Table : Teacher

Teacher_id	TName	Tsal	DOJ
101	Neha	15000	12-10-80
102	Supriya	20000	10-02-90
103	Abhishek	25000	23-07-93
104	Rajat	30000	19-09-97
105	Isha	35000	09-07-00

**1 . Write a Query to fetch the details of the Teacher who joins after 1990.**

A1. Query : `Select * from Teacher where DOJ > '10-02-1990'`

**Output:**

Teacher_id	TName	Tsal	DOJ
103	Abhshek	25000	1993-07-23
104	Rajat	30000	1997-09-19
105	Isha	35000	2000-07-09

**2 . Write a query to display the name of Teachers whose name starts from 'A'.**

A2. Query : `Select * from Teacher where TName like 'A%';`

Output : The query will fetch the name of Teachers whose name starts from 'A'.

	Teacher_id	TName	Tsal	DOJ
1	103	Abhshek	25000	1993-07-23

**3 . Write a query to display the details of Teachers whose Salary is greater than 25000 and name ends with 'a'.**

A3. Query : `Select * from Teacher where Tsal > 25000 and TName like '%a'.`

Output : The query will fetch the details of Teacher whose Salary is greater than 25000 and name ends with 'a'. we can use LIKE clause for both type (starts with and ends with) of questions.

For “Starts with” type of questions : Like ‘a%’

For “ ends with” type of questions : Like ‘% a’

	Teacher_id	TName	Tsal	DOJ
1	105	Isha	35000	2000-07-09

4 . Write a query to fetch all records in increasing number of name.

A4. Query : `Select * from Teacher order by TName`

Output : By default Order by clause arranges in Asscending order if we want to arrange the column attributes in descending order ,then we can use ' Desc' keyword with the order by clause.

	Teacher_id	TName	Tsal	DOJ
1	103	Abhshek	25000	1993-07-23
2	105	Isha	35000	2000-07-09
3	101	Neha	15000	1980-10-12
4	104	Rajat	30000	1997-09-19
5	102	Supriya	20000	1990-02-10



**5 . Write an Sql query to display the details of faculty who earns maximum.**

A5. Query : `Select * from Teacher where Tsal = (Select max(Tsal) from Teacher);`

	Teacher_id	TName	Tsal	DOJ
1	105	Isha	35000	2000-07-09

**6 . Display “Annual Salary” of all faculties. (Given salary is monthly)**

A6. Query : `Select Tsal * 12 as 'Annual Salary' from Teacher;`  
Output :

	Annual Salary
1	180000
2	240000
3	300000
4	360000
5	420000

**Q 15 . Answer the Following Questions on the basis of the given table :  
Teacher.**

A15. Table – Product

P_id	PName	Price	Qty
101	P1	15000	23
102	P2	20000	30
103	P3	25000	24
104	P4	30000	27
105	P5	35000	28

1 . Write an Sql query to count distinct price from product.

A1. Query : `Select count(distinct(price)) from product.`

Output : 5

2 . Write an Sql query to Calculate the Sum of those product whose Quality is greater than 20.

A2. Query : `Select sum(Price) from Product where Qty > 20;`

Output : 125000

3 . Write a query to fetch all records in increasing number of Qty.

A3. Query : `Select * from Product order by Qty`

Output :

	P_id	PName	Price	Qty
1	101	P1	15000	23
2	103	P3	25000	24
3	104	P4	30000	27
4	105	P5	35000	28
5	102	P2	20000	30

**Q16. Write an Sql query to perform Cross Join operation.**

**Table 1: Employee**

Id	Employee_Name	Department_id
1	Homer Simp	4
2	Ned	1
3	Barney	5
4	Clancy	3
5	Moe	NULL

**Table 2 : Department**

id	Employee_Name
1	Sales
2	Engineering

<b>3</b>	Management
<b>4</b>	Customer Service
<b>5</b>	Research Development

A16. Query : **SELECT \* FROM** Employee **CROSS JOIN** Department;  
Output :

	Id	Employee_Name	Department_id	id	Employee_Name
1	1	Homer Simp	4	1	Sales
2	2	Ned	1	1	Sales
3	3	Barney	5	1	Sales
4	4	Clancy	3	1	Sales
5	5	Moe	2	1	Sales
6	1	Homer Simp	4	2	Engineering
7	2	Ned	1	2	Engineering
8	3	Barney	5	2	Engineering
9	4	Clancy	3	2	Engineering
10	5	Moe	2	2	Engineering
11	1	Homer Simp	4	3	Management
12	2	Ned	1	3	Management
13	3	Barney	5	3	Management
14	4	Clancy	3	3	Management
15	5	Moe	2	3	Management
16	1	Homer Simp	4	4	Customer Servi...
17	2	Ned	1	4	Customer Servi...
18	3	Barney	5	4	Customer Servi...
19	4	Clancy	3	4	Customer Servi...
20	5	Moe	2	4	Customer Servi...
21	1	Homer Simp	4	5	Research devlo...
22	2	Ned	1	5	Research devlo...
23	3	Barney	5	5	Research devlo...
24	4	Clancy	3	5	Research devlo...
25	5	Moe	2	5	Research devlo...

Cross join does not use 'on' or 'using' when it is being declared. Cross join is different from other types of join clause. In cross join the size of result set is the number of rows in the first table multiplied by the number of rows in the second table. This type of result is also called Cartesian product of the two tables

**Q17 . Write an Sql query to perform Inner Join operation.**

A17. Table 1: Employee

Id	Employee_Name	Department_id
1	Homer Simp	4
2	Ned	1
3	Barney	5
4	Clancy	3
5	Moe	NULL

**Table 2 : Department**

id	Employee_Name
1	Sales
2	Engineering
3	Management
4	Customer Service
5	Research Development

Query : **SELECT \* FROM** Employee emp **Inner JOIN** Department dep  
**ON** emp.Id = dep.id;

Output :

Results		Messages			
	Id	Employee_Name	Department_id	id	Employee_Name
1	1	Homer Simp	4	1	Sales
2	2	Ned	1	2	Engineering
3	3	Barney	5	3	Management
4	4	Clancy	3	4	Customer Service
5	5	Moe	2	5	Research development

Query executed successfully

**Q18 . Write an Sql query to fetch the details of all the Salesman and Customer Located in 'New York'.**

**A18. Table 1: Salesman**

Salesman_id	Name	City	Commission
4001	Rajesh	New york	15
4002	Mahesh	paris	20
4005	Umesh	London	35
4004	Raju	paris	40
4006	Rahul	Rome	78
4008	Vikas	San jose	80

**Table 2: Customer**

Customer_id	Cust_Name	City	Grade	Salesman_id
3002	Nick	New york	100	4002
3001	hockky	New york	200	4001
3003	Justein	London	300	4005
3004	Julian	paris	100	4008
3005	Fabian	Berlin	200	4006
3006	Jeff	Moscow	100	4002
3007	Josh	Berlin	300	4001

Query : **Select** salesman\_id "ID", name, 'Salesman' **From** salesman  
**Where** city='New york' **UNION** (**Select** Cust\_id "ID", cust\_name,  
'Customer' **From** customer **Where** city='New york')

Output :

ID	name	(No column name)
3001	hockky	Customer
3002	Nick	Customer
4001	Rajesh	Salesman

**Q19. Write an Sql query to display Distinct Salesman , Customer and their cities.**

**A19. Table 1: Salesman**

Salesman_id	Name	City	Commission
4001	Rajesh	New york	15
4002	Mahesh	paris	20
4005	Umesh	London	35
4004	Raju	paris	40
4006	Rahul	Rome	78
4008	Vikas	San jose	80

**Table 2: Customer**

Customer_id	Cust_Name	City	Grade	Salesman_id
3002	Nick	New york	100	4002
3001	hockky	New york	200	4001
3003	Justein	London	300	4005
3004	Julian	paris	100	4008
3005	Fabian	Berlin	200	4006
3006	Jeff	Moscow	100	4002
3007	Josh	Berlin	300	4001

Query : **Select** salesman\_id, city **From** Customer **Union**  
(**Select** salesman\_id, city **From** Salesman)

Output : Salesman Id is common in both tables (Customer , Salesman).



	salesman_id	city
1	4001	Berlin
2	4001	New york
3	4002	Moscow
4	4002	New York
5	4002	Paris
6	4004	paris
7	4005	London
8	4006	Berlin
9	4006	Rome
10	4008	paris
11	4008	San jose

**Q20. Write an Sql query to make a report of those Salesman who do not have customers in their cities as well as who do.**

**A20. Table 1: Salesman**

Salesman_id	Name	City	Commission
4001	Rajesh	New york	15
4002	Mahesh	paris	20
4005	Umesh	London	35
4004	Raju	paris	40
4006	Rahul	Rome	78
4008	Vikas	San jose	80

**Table 2: Customer**

Customer_id	Cust_Name	City	Grade	Salesman_id
3002	Nick	New york	100	4002
3001	hockky	New york	200	4001
3003	Justein	London	300	4005
3004	Julian	paris	100	4008
3005	Fabian	Berlin	200	4006
3006	Jeff	Moscow	100	4002
3007	Josh	Berlin	300	4001

Query : **SELECT** Salesman.Salesman\_id, Name, Cust\_Name, Commisison  
**FROM** Salesman, Customer  
**WHERE** Salesman.City = Customer.City  
**UNION**  
(**SELECT** Salesman\_id, Name, 'No Match', Commisison  
**FROM** Salesman  
**WHERE** NOT City = ANY  
(**SELECT** City **FROM** Customer))  
**ORDER BY** 2 **DESC**

Output: The query will fetch the Name and Salesman\_id of those Salesman who do not have customers in their cities and also fetch the Salesman who have Customer in their cities.

	Salesman_id	Name	Cust_Name	Commisison
1	4008	Vikas	No Match	80
2	4005	Umesh	Justein	35
3	4004	Raju	Julian	40
4	4001	Rajesh	hockky	15
5	4001	Rajesh	Nick	15
6	4006	Rahul	No Match	78
7	4002	Mahesh	Julian	20

**Q21. Write an Sql query to create a union of two Sql queries that displays the names, ratings and Cities of all customers. Those with a rating of 200 or greater will also have the words High Rating, while the others will have the words Low Rating.**

**Table 1: Customer**

Customer_id	Cust_Name	City	Grade	Salesman_id
3002	Nick	New york	100	4002
3001	hockky	New york	200	4001
3003	Justein	London	300	4005
3004	Julian	paris	100	4008
3005	Fabian	Berlin	200	4006
3006	Jeff	Moscow	100	4002
3007	Josh	Berlin	300	4001

Query : **Select** Cust\_Id, City, Grade, 'High Rating'  
**From** Customer  
**Where** Grade >= 200  
**Union**  
(**Select** Cust\_Id , City, Grade, 'Low Rating' **From** Customer  
**Where** Grade < 200)

Output : The above query will categorize the Grade in high Rating and Low Rating if the given Grade is greater than 200 then it is highest Rated Customer. And if Grade is less than 200 it will categorize as Low Rating Customer.

	Cust_Id	City	Grade	(No column name)
1	3001	New York	200	High Rating
2	3002	New York	100	Low Rating
3	3003	London	300	High Rating
4	3004	paris	100	Low Rating
5	3005	Berlin	200	High Rating
6	3006	Moscow	100	Low Rating
7	3007	Berlin	300	High Rating

**Q22. Write an Sql query that displays the name and id of each salesman and each customer with more than one current order.**

**Table 1: Customer**

Customer_id	Cust_Name	City	Grade	Salesman_id
3002	Nick	New york	100	4002
3001	hockky	New york	200	4001
3003	Justein	London	300	4005
3004	Julian	paris	100	4008
3005	Fabian	Berlin	200	4006
3006	Jeff	Moscow	100	4002
3007	Josh	Berlin	300	4001

**Table 2: Salesman**

Salesman_id	Name	City	Commission
4001	Rajesh	New york	15
4002	Mahesh	paris	20
4005	Umesh	London	35
4004	Raju	paris	40
4006	Rahul	Rome	78
4008	Vikas	San jose	80

**Table 3: Orders**

Order_no.	Pur. amount	Cust. Id	Salesman_id
3001	175	5005	4002
3002	150	5003	4002
3005	175	5004	4001
3004	187	5002	4006
3006	190	5001	4004
3008	198	5006	4008

Query : **Select** Cust\_Id, Cust\_Name  
**From** Customer a  
**Where** 1 < (**Select Count** (\*)  
**From** Orders b **Where** a.Cust\_Id = b.Customerid)  
**Union**  
(**Select** Salesman\_id, Name  
**From** Salesman a  
**Where** 1 < (**Select Count**(\*)  
**From** Orders b  
**Where** a.Salesman\_id = b.Salesman\_Id))  
**Order by** 2

Output : The query will fetch the Name and id of salesman and customers having more than one orders.

	Cust_Id	Cust_Name
1	4002	Mahesh

**Q 23. Write an Sql query to fetch the Second Highest Salary of the Employee.**

**Table : Employee**

EmployeeNo	Salary
1234	5000
2345	1222
1235	6000
4000	8000
6532	10000
9999	12000
1233	15000

Query : **Select MAX(Salary) as salary**  
**From Employee**  
**Where Salary < (SELECT MAX(Salary) from Employeee);**

Output :



	salary
1	12000

**Q24. Write an Sql query to fetch the nth Salary of the Employee.**

A24. Table : Employee

EmployeeNo	Salary
1234	5000
2345	1222
1235	6000
4000	8000
6532	10000
9999	12000
1233	15000

Query : **Select** salary  
**From** Employee A  
**Where** n-1 = (**Select** **count**(1)  
**From** Employee B  
**Where** B.salary>A.salary)

Where n=2,3,4,5.. etc

Let Suppose we want to calculate 4<sup>th</sup> highest Salary then value of n will be 4.

And the output Salary is :

	salary
1	8000

**Q25. Write an Sql query to find the department wise highest Salary of the Employees.**

A25. Table : Salary

DeptNo.	Salary	Employee Name
1	1000	Neha
1	2000	Supriya
3	5000	Hema
1	3000	Isha
1	5000	Anjali
2	4000	Parul
3	6000	Silki

Query : **Select** DeptNo , **MAX**(Salary)  
**from** Salary **group by** DeptNo

Ouput : The above query will fetch the department wise highest salary.

	DeptNo	(No column name)
1	1	5000
2	2	4000
3	3	6000

**Q 26. Write an Sql to display the alternate records of the Table.**

**Table : Orders**

Order_no.	Pur. amount	Cust. Id	Salesman_id
1	175	5005	4002
2	150	5003	4002
3	175	5004	4001
4	187	5002	4006
5	190	5001	4004
6	198	5006	4008

Query : Select \* from Orders where mod (Order\_no. , 2) = 1 order by Order\_no. desc.

**Q 27. Write an SQL query to find duplicate records in the table.**

**A27.**

Name
Neha
Neha
Supriya
Supriya
Isha
Supriya

Query : **Select** Name , **COUNT** (\*) **from** Name **group by** Name **having** **count**(\*)>1

Output : The query will fetch only duplicate records from the table.

	Name	(No column name)
1	Neha	2
2	Supriya	3

**Q28. Write an Sql query to find intersection two tables .**

**A28. Table 1: Sample 1**

City	Country
hydrabaad	india
london	uk
texas	usa
texas	usa
hydrabad	india
chamoli	uk

**Table 2: Sample 2**

City	Country
hydrabaad	india
berlin	uk
texasi	usa
texasi	usa
hydrabad	india
champawat	uk

Query : `select * from Sample 1  
intersect  
select * from Sample 2`

Output :

	City	Country
1	hydrabad	india

**Q 29. Write an Sql query to perform UNION and UNION ALL operations on the given table.**

A29.

Sample\_1 :

City	Country
hydrabad	india
london	uk
texas	usa
texas	usa
hydrabad	india
chamoli	uk

Sample\_ 2:

City	Country
------	---------

hydrabad	india
berlin	uk
texas	usa
texasi	usa
hydrabad	india
champawat	uk

**Union Command :** Union Operation Removes the duplicates rows from the result.

Query : `Select * from Sample_1  
UNION  
Select * from Sample_2`

Output :

	City	Country
1	berlin	uk
2	chamoli	uk
3	champawat	uk
4	hydrabad	india
5	london	uk
6	texas	usa
7	texasi	usa

**Union All Operation :** Union All Command will not eliminate the duplicate rows from the result and gives the all rows as a output.

Query : `Select * from Sample_1`  
`Union ALL`  
`Select * from Sample_2`

Output :

	City	Country
1	hydrabad	india
2	london	uk
3	texas	usa
4	texas	usa
5	hydrabad	india
6	chamoli	uk
7	hydrabad	india
8	berlin	uk
9	texas	usa
10	texasi	usa
11	hydrabad	india
12	champa...	uk

**Q30. Write an Sql query to perform Intersection on the given tables.**

A30. Sample\_1 :

City	Country
hydrabad	india
london	uk



texas	usa
texas	usa
hydrabad	india
chamoli	uk

Sample\_ 2:

City	Country
hydrabad	india
berlin	uk
texas	usa
texasi	usa
hydrabad	india
champawat	uk

Query : **Select \* from Sample\_1**  
**Intersect**  
**Select \* from Sample\_2**

Output :

	City	Country
1	hydrabad	india
2	texas	usa

**Q 31. Write an Sql query to fetch the Third Highest Salary of the Employee.**

**Table : Employee**

EmployeeNo	Salary
1234	5000
2345	1222
1235	6000
4000	8000
6532	10000
9999	12000
1233	15000

Query : **Select MAX(Salary) as salary**  
**From Employee**  
**Where Salary < (SELECT MAX(Salary) from Employeee);**

Output : 10000

**Q32. Write an Sql query to find the department wise lowest Salary of the Employees.**

A25. Table : Salary

DeptNo.	Salary	Employee Name
1	1000	Neha
1	2000	Supriya
3	5000	Hema
1	3000	Isha
1	5000	Anjali
2	4000	Parul
3	6000	Silki

Query : `Select DeptNo , MIN(Salary)  
from Salary group by DeptNo`

Ouput : The above query will fetch the department wise lowest salary.

## Summary

SQL stands for Structured query Language. Sql is a special purpose language that let's you access and manipulate the databases. The best way to learn anything is by practice and try lot of questions.

So without taking your much time let me tell you the Summary of this complete Article. Basically In this Article we have covered intermediate level questions of Sql which are asked a lot in interviews.

So, In this Article we have covered the following topic questions

- Different types of Clause ( group by, order by, Like)
- Basic operations on the Tables(Union ,Union all, Intersection)
- Different types of joins(Inner join, left join, right join, Full Join)

- Cross Join
- Different types of Aggregate Functions (Max, Min, Count)
- Subqueries

As we know how much SQL is important for interviews. In this Article we have practiced many questions in which we have tried different manipulations using different different clauses on the different different databases.