

## Top Leetcode SQL Problems

1. [\*\*175. Combine Two Tables\*\*](#)
2. [\*\*176. Second Highest Salary\*\*](#)
3. [\*\*177. Nth Highest Salary\*\*](#)
4. [\*\*178. Rank Scores\*\*](#)
5. [\*\*181. Employees Earning More Than Their Manager\*\*](#)
6. [\*\*182. Duplicate Emails\*\*](#)
7. [\*\*183. Customers Who Never Order\*\*](#)
8. [\*\*184. Department Highest Salary\*\*](#)
9. [\*\*197. Rising Temperature\*\*](#)
10. [\*\*584. Find Customer Referee\*\*](#)
11. [\*\*586. Customer Placing the Largest Number of Orders\*\*](#)
12. [\*\*595. Big Countries\*\*](#)
13. [\*\*596. Classes More Than 5 Students\*\*](#)
14. [\*\*607. Sales Person\*\*](#)
15. [\*\*608. Tree Node\*\*](#)
16. [\*\*620. Not Boring Movies\*\*](#)
17. [\*\*626. Exchange Seats\*\*](#)
18. [\*\*627. Swap Salary\*\*](#)
19. [\*\*1050. Actors and Directors Who Cooperated At Least Three Times\*\*](#)
20. [\*\*1084. Sales Analysis III\*\*](#)
21. [\*\*1141. User Activity for the Past 30 Days I\*\*](#)
22. [\*\*1148. Article Views I\*\*](#)
23. [\*\*1158. Market Analysis I\*\*](#)
24. [\*\*1179. Reformat Department Table\*\*](#)
25. [\*\*1393. Capital Gain/Loss\*\*](#)
26. [\*\*1407. Top Travellers\*\*](#)
27. [\*\*1484. Group Sold Products By The Date\*\*](#)
28. [\*\*1527. Patients With a Condition\*\*](#)
29. [\*\*1587. Bank Account Summary II\*\*](#)

30. [1667. Fix Names in a Table](#)
31. [1693. Daily Leads and Partners](#)
32. [1729. Find Followers Count](#)
33. [1741. Find Total Time Spent by Each Employee](#)
34. [1757. Recyclable and Low Fat Products](#)
35. [1873. Calculate Special Bonus](#)
36. [1890. The Latest Login in 2020](#)
37. [185. Department Top Three Salaries \(Hard\)](#)

SQL

(Structured Query Language)

## Introduction :-

In June 1970 Dr. E.F. Codd published the paper, "A Relational Model of Data for Large Shared Data Banks" in the Association of Computer Machinery (ACM) journal.

Using Codd's model the language, Structured English query language was developed by IBM Corporation in San Jose Research Center. Later on it was named as SQL officially. And afterwards many other companies like Oracle, Microsoft works on it.

- SQL is domain specific language.
- SQL is declarative language. It means we only we define what to do not how to do. C is a procedural language where we define how to do as well as what to do.
- DDL, DCL, DML, TCL
- Keys and constraints (Primary key, foreign key)
- Operators (Like, Between, In, Not In, Exist, NOT EXISTS, Conditional)
- Clauses (Distinct, ORDER BY, Having, group By from)
- Aggregate Functions (sum, Avg, count) etc.

- Joins and Nested Anonyms
- PL SQL :-

It is procedural query language - where  
we can define how to do as well.  
( Triggers, function, Procedures )  
cursor

## SQL Commands:-

### Data Definition Language (DDL):-

DDL commands deals with the schema definition of a table.

- Create → Create table
- Alter → Alter table
- Drop → drop table
- Truncate
- Rename → Rename a table

### Data Manipulation Language (DML):-

DML commands deals with the data where we can retrieve or we can insert data or we can change the data.

- Select
- Insert
- Update
- Delete

### Data Control Language (DCL):-

DCL commands deals with giving the permission or authorization on the data.

- Grant - to give privilege
- Revoke - to take back privilege

## Transaction Control Language:- (TCL)

JCL TCL commands are used for controlling the transactions.

- Commit
- Roll Back
- Save Point

## Constraints:-

- Primary Key
- Foreign Key
- Check
- Unique
- Default
- Not Null
- Index

These constraints are used for applying restriction on the data.

Start XAMPP server:-

go to directory

xampp/mysql/bin

Enter command

mysql -u root -p

Show Databases:-

\$ show databases;

List down all available databases.

Select database to work:-

\$ use database-name;

To show available tables in the databases :-

\$ show tables;

Schema of the table:-

\$ desc table-name;

Create Table command:-

Syntax

CREATE <sup>table</sup> table\_name (

column1\_name datatype,

column2\_name datatype,

column3\_name datatype,

column4\_name datatype,

) ;

Ex.

CREATE table student (

RollNo int,

name varchar(20),

Age int,

course varchar(20),

) ;

column with default value:-

State varchar(50) Default 'Rajasthan',

SQL Data Types :- Data types are important features of any programming language.

SQL has following data types:-

- Character String :- '5', 'Python'
- Numeric :- 48, 10.34, 0.001, -55, 2.5E2
- Boolean :- TRUE, FALSE, UNKNOWN
- Datetime :- Date '2016-05-14'  
TIME '04:12:00'  
TIMESTAMP '2016-05-14 04:12:00'
- Interval :- INTERVAL '15-3'

Fixed Length String vs Variable length String:-

Name CHAR (15)

CHAR is used to define fixed length strings and  
VARCHAR is used for variable length strings.

Alter command:-

Alter command is used for changing the structure or schema of a table.

- Add column(s)
- Remove column(s)
- Modify datatype
- Modify data type length
- Add constraints
- Remove constraints
- Rename column name | table name

Current table

Create table student (

RollNo int(11),

name varchar(50),

Age int(11),

City varchar(50),

Email varchar(50),

?

Add a column:-

Want to add a column last\_name gender

alter table student add column (gender char(1))

define multiple if needed  
column

Drop  
Remove ↓ column :-

Drop column gender

alter table student drop column gender;

Modify datatype | length :-

Changing name column length to 100 and  
column name to Name

alter table student modify column Name varchar(100);

only changing datatype for its length.

Retain column ↓ <sup>table</sup> Name :-

City to district

alter table student rename stud;

(changing table name from student to stud)

Change column name City to district :-

Change

alter table student change column city district varchar(100);

Add constraints:-

Making Roll No. as a primary key

alter table student add primary key (Roll No);

Drop primary key from table:-

alter table student drop primary key;

Update command:-

Update command is used for manipulating data of a table. It is DML command.

Syntax:-

Update <table name> set <column1> = <value1>, <column2> = <value2>, ... where <condition>;

If we do not mention any condition then update command affects all the rows.

Ex. Changing age to 24 where Roll No is 1

Update student set age=24 where roll no=1;

## Delete vs Drop vs Truncate:-

Delete	Drop	Truncate
DDL	DQL	DDL
used to delete one or more tuples from the table	It is used to delete a table	Used to delete all tuples from the table
Rollback possible	Rollback Not possible	Rollback Not possible
Syntax:- delete from table-name;  delete from table-name where condition;	drop table-table-name;	truncate table-table-name;
Create log file	Does not Create log file	Does not Create log file
Slow		fast

Insert data into table:-

Inserting data into student table:-

insert into student(RollNo, Name, Age, District, Email, State)  
values (7, 'Nisha', 21, 'Jaipur', 'nishi@gmail.com', 'Rajasthan'), (1, 'Rishabh', 20, 'Delhi', 'rishi123@gmail.com', 'Delhi')

## SQL Constraints:-

SQL constraints are used to specify rules for the data in a table.

Constraints are used to limit the type of the data that can go into the table. This ensures the accuracy and reliability of the data in the table. If there will be any violation between the constraints and the data action, the action will be aborted.

- NOT NULL :- Ensures that a column cannot have NULL value.
- UNIQUE :- Ensures that all values in a column are different.
- PRIMARY KEY :- Not NULL and UNIQUE
- FOREIGN KEY :- Prevents actions that would destroy links between tables.
- CHECK :- Ensures that the values in a column satisfies a specific condition.
- DEFAULT :- Sets a default value for a column if no values is specified.
- CREATE INDEX :- Used to create and retrieve data from the database very quickly.

## SQL operators:-

### i) Arithmetic operators:-

+ , - , \* , %

Used for arithmetic operators

### ii) Comparison operators:-

Returns true or false

= , > , < , >= , <= , <> , !=  
↑  
Not Equal

### iii) Logical operators:-

AND, OR, NOT

## Special operators:-

operator	Description	operator on
IN	The IN operator checks a value within a set of values separated by commas and retrieve the rows from the table which are matching	Any set of values of the same data types

In operator is combination of many  
emp OR operators.

eid	ename	dept	salary
1	RAM	HR	10000
2	Amrit	MRKT	20000
3	Ravi	HR	30000
4	Nitin	MRKT	40000
5	Vayu	IT	50000

Query:- display employee name which belongs  
to HR or MRKT department.

Select ename from emp where dept in ('HR', 'MRKT')

Between :-

The SQL between operator tests an expression against a range. The range consists of a beginning, followed by an AND keyword and an end expression.

Query:- display employee name whose salary is between 20000 to 45000.

Select ename from emp where salary between 10000 and 40000;

Output

ename
Amrit
Ravi
Nitin

**ANY:-** ANY compares a value to each value in a list or results from a query and evaluates to true if the result of an inner query contains at least one row.

\*\* Any must be preceded by comparison operators. ( $>$ ,  $<$ ,  $\neq$ ,  $\leq$ ,  $\geq$ )

**Query :-**

Display employee names who has salary lesser than any other employee.

select ename from emp where salary < any(select salary from emp);

**Output :-**

ename
Ram
Amrit
Ravi
Nitin

**ALL:-** ALL is used to select all records of a select statement. It compares a value to every value in a list or result from a query.

It must be preceded by any comparison operator. It returns true only if condition satisfy true with every value

It is combination of AND Operators.

Query return employee names whose salary is greater than or equal to all other employees

Select emame from emp where salary  $\geq$  All (select salary from emp)

out put

ename
Vikram

Some :-

All and some operators give same.

Not EXISTS

Exists Operator:- It verifies whether a subquery returns any items or not.  
It is used in correlated subquery.

Find the details of employee if he is working in at least one project

Select \* from emp where eid exists (select eid from project where emp.eid = project.eid)

Where Clause:-

In a select statement where clause is optional. We can use select statement without where clause. But if we want to filter data in select statement then we can use where clause.

Syntax

select \* from table-name where (condition operator value)

Aggregate Functions:-

Count :-

The SQL count function returns the number of rows in a table satisfying the criteria specified in the where clause. It sets the number of rows or non NULL column values.

Q) Display number of employee whose salary is greater than 20000.

select count(salary) from emp where salary > 20000;

Use of distinct in count

count (distinct salary)

Sum :-

sum() function returns the sum of all selected column.

Avg :-

Display sum of all employees salary.

```
select sum(salary) from emp;
```

Avg :-

The SQL Avg function calculates the average value of a column of numeric type. It returns the average of all non NULL values.

```
select avg(salary) from emp;
```

Max() :- Returns maximum value out of all values of a given column.

```
select max(salary) as 'Max Salary' from emp;
```

Min() :-

Returns minimum value out of all values of a given column.

```
select min(salary) as 'Min Salary' from emp;
```

## Subqueries :-

- Write a SQL query to display maximum salary from emp table.

Select max(salary) from emp;

- Write a query to display employee name who is taking maximum salary.

salary,ename from emp where salary = (select max(salary)  
from emp);

Output

ename
salary

- Write a query to display second highest salary from emp table.

Logic :- logic is first exclude the maximum salary and then take maximum out of remaining data

\*\* Write a subquery to find second highest salary from emp table extends it to kth largest as well.

Approach 1:-

Select max(salary) from emp where salary in  
Select salary from emp where salary <= (Select  
max(salary) from emp)

Approach 2:-

for Kth Largest as well

Select ifnull((Select salary from emp order by  
salary desc limit 1,1), null) as SecondHighest  
Salary;

If null :-

if null takes to expression it return  
first expression if it is not null otherwise  
it returns second expression.

Select ifnull(expression1, expression2);

Select ifnull(5, null)  
Output 5

Select ifnull(null, 8)  
Output 8

Select ifnull(null, null)  
Output null

## Declare a Variable :-

We can create our own variable in SQL

### Syntax

```
DECLARE my-var int;  
          ↑  
        datatype
```

### Set a Variable :-

```
SET my-var = value;
```

### Group By clause:-

- The usage of SQL Group By clause is, to divide the rows in a table into smaller groups.
- The Group By clause is used with the SQL Select statement.
- The grouping can happen after retrieves the rows from a table.
- When some rows are retrieved from a grouped result against some condition, that is possible with Having clause.
- The Where clause is used to retrieve rows based on a certain condition, but it can not be applied to grouped result.

### Syntax

```
Select dept, sum(salary) from emp group by dept;
```

We can only write dept name in select statement.

## Having Clause :-

Having clause specifies a search condition for a group or an aggregate. Having is usually used in group by clause. But we can use having clause without group by clause as well.

### Having with Group By :-

#### • Query :-

write a query which displays the dept takes having more than 2 employees.

select dept from emp group by dept having count(\*) > 2;

### Having with Where :-

select dept,ename from emp where dept = 'IT'  
having ename = 'SHIVAM';

\*\* attribute mentioned in select statement is mandatory if we use same attribute for having clause.

Simple Having :-

Select deptname from emp having dept = 'IT';

- Print employee names whose working in a department having number of employees  $> 2$ .

Select ename from emp where dept in (Select dept from emp group by dept having count(\*) > 2);

Query:-

Write a query to print the person name and salary who is taking highest salary in every department.

Select ename, salary, dept from emp where salary in (Select max(salary) from emp group by dept);

DateDiff() :- Date diff function returns the difference in number of days for given arguments.

dateDiff(date1, date2);

select dateDiff('2016-05-06', '2015-05-04');

Output 2

IS NULL Condition :-

IS NOT NULL

On null values we cannot use operator.  
Because null value represents nothing.

We use ISNULL and IS NOT NULL to check the value is null or not

Syntax

expression IS NULL;

expression IS NOT NULL;

## Correlated Subquery :-

- It is a subquery that uses values from outer query.
- Top down approach
- Exists / Not Exists mainly we use here.

Find all employee detail who work in a department.

Select \* from emp where exists (select eid from dept where emp.eid = dept.eid);

Here we have two table where some attribute with foreign key relation.

### Nested Subquery

• Bottom Up

Display employee name  
who works in only department.

Selectename from emp where

emp.eid in (select eid from dept);

Selectename from emp

where exists (select eid from

dept where emp.eid = dept.eid);

### Correlated Subquery

Top Down

Cross Product +  
condition

Select <sup>emp.</sup>ename from  
emp, dept where  
emp.eid = dept.eid;

'Find N<sup>th</sup> Highest Salary using SQL :-

Correlated query :-

select id, salary from emp e1 where  
N-1 = (select count(distinct salary) from  
emp e2 where e2.salary >  
e1.salary)

Using order By :-

select distinct salary from emp ~~where~~ order by salary desc  
limit N-1, 1

## CASE Statement:-

The case statement goes through the conditions and returns the value where <sup>first</sup> condition met.

### CASE

```
WHEN Condition1 THEN Result1  
WHEN Condition2 THEN Result2  
WHEN Condition3 THEN Result3  
:  
ELSE Result  
END;
```

## length() function:-

### Syntax

```
length(string);
```

return the length of the string.

## substr() function:-

### Syntax

```
substr(str, start pos, No. of char)
```

## lower() :-

```
Syntax lower(string);
```

## upper() :-

```
Syntax upper(string);
```

## concat() :-

```
Syntax concat(str1, str2)  
to concat two strings.
```

if() function :-

Syntax

`if( condition, value-if-true, value-if-false)`

Group-concat() :- It is a function used to concatenate multiple strings. It is extended form of concat() function.

Syntax

`Group-concat([distilct] [of-name] [order by clause]  
[SEPARATOR string-val]);`

LIKE pattern:-

Like is a operator to match a pattern into a string.

We use to wildcard character - % & \_ %.

% for zero or multiple character

- for exactly one character

### Syntax

Select col1, col2 from table-name where  
column like pattern;

'%q%' end with q

'q%' starts with q

'\_q%' second character is q

## ENUM in SQL:-

An ENUM is a string object whose value is decided from a set of permitted values that are explicitly defined at the time of execution.

### Syntax

```
Create table table_name(
```

```
    id int not null,
```

```
    priority ENUM('high', 'medium', 'low'),
```

```
);
```

### Access

```
priority = 'Value')
```

```
priority = 1
```

Join :-

Joins are basically other form of nested queries.

To apply joins on two tables there should be some common attribute between both the tables.

Joins are cross Product + some condition

Types of Joins :-

1. Cross Product Joins

2. Natural Join

3. Conditional Join

4. Equi Join

5. Self Join

6. Outer Join

Left Outer

Right Outer

Full Outer

Display employee names whose are working in HR Department.

E.No	name	Address	DNo.	d-name	emp-No
1	Ram	Delhi	d1	HR	1
2	Vayu	Chd	d2	II	2
3	Ravi	Jaipur	d3	Sales	3
4	Nitin	Noida	d4	Marketing	4

## Natural Join :-

Natural Join works on those tables when tables have a common attribute and we want to equate the values of that common attribute.

- Find the employees name whose working in a department.

Emp			Dept		
eid	Name	Address	dno	dname	e-id
1	Ram	Delhi	d1	HR	1
2	Vayu	Jaipur	d2	MARKI	2
3	Nith	Ajmer	d3	IT	3
4	Ravi	Bhopal			

- Select name from emp, dept where Emp.eid=Dept.e-id

1	Ram	d1	1
1	Ram	d2	2
1	Ram	d3	3
2	Vayu	d1	1
2	Vayu	d2	2
2	Vayu	d3	3
3	Nith	d1	1
3	Nith	d2	2
3	Nith	d3	3
4	Ravi	d1	1
4	Ravi	d2	2
4	Ravi	d3	3

## Natural Join query

select name from emp Natural join dept

## Self Join:-

when we cross product the same table then it is called self join.

AXA

- Find student-id who is enrolled in at least two course.

study

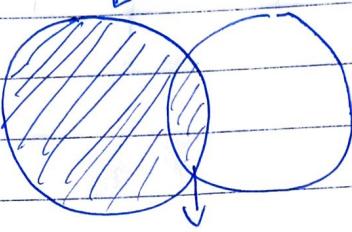
S-id	C-id	since
S1	C1	2016
S2	C2	2017
S1	C2	2017

Select s-id from study as T<sub>1</sub>, study as T<sub>2</sub>  
 where T<sub>1</sub>.s-id = T<sub>2</sub>.s-id and  
 $T_1.c\_id \leq T_2.c\_id$ ;

Equi Join:- Equi Join is almost similar to natural join. But in equi join we can equate to different column of both the tables as well.

Left outer Join :-

Left outer Join



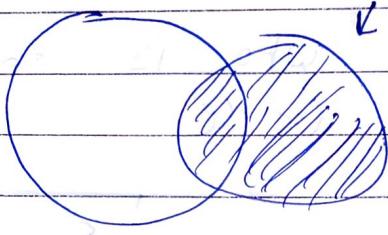
Natural Join

Select col1, col2 from table1 left outer join table2 on

Right outer Join :-

attr1 = attr2;

Right outer Join

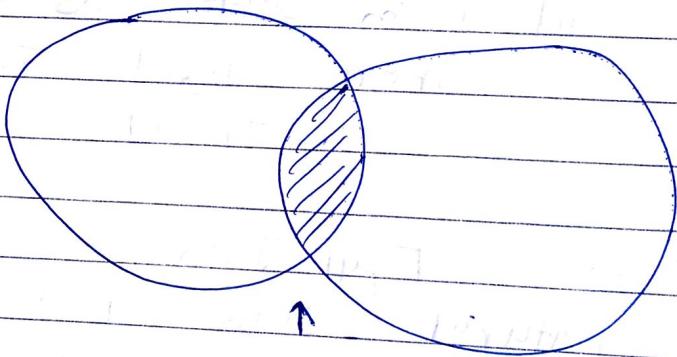


Select col1, col2 from table1 right outer join

table2 on attr1 = attr2;

Inner Join :-

Inner Join



Inner join gives matching record of two table on the basis of given attribute.

Select col1, col2 from table1 inner join table2  
on table1.attr1 = table2.attr2;

## GETDATE () :-

This function is used to get the current date of the system.

### Syntax

```
select getdate();
```

## ROW-NUMBER() :-

It is a function which is used to give a sequence number to rows.

given sequence is like this way (1,2,3,4,5,6,7,8)

Write a SQL query to order the rows by salary in Desc order and assign a row number to every tuple.

```
select *, row_number() over (order by age) as RowNo from student;
```

## RANK() function :-

Rank function is used to rank the tuples using some column and order.

In case of tie it gives the same rank and skips the next rank.

Write a SQL query to give rank to the rows of a table.

Based on salary in DESC order.

```
select *, rank() over (order by salary) as [RANK]  
from empf
```

Write same query with partition by Dept Name.

```
select *, rank over (partition by dept order  
by salary) as [RK]  
from empf
```

DENSE-RANK() :-

dense-rank, in case of tie it gives the same rank but it doesn't skip any rank it goes in same continuous sequence.

Salary Rank

5	1
5	1
4	2
3	3
2	4
2	4

```
select *, dense_rank() over (order by salary)  
as [DenseRank] from empf
```

Select \*, dense\_rank() over (partition by dept order  
by salary) as DensRank  
from emp;

How to Create a View:-

CREATE VIEW view-name as  
Select column-lists from table-name  
where condition;

View is treated as a table.

We can perform Select, Update statement  
on views as well.

SQL stored Procedures:-

A stored procedure is a prepared  
SQL code that we can save, so that  
code can be used over and over again.

We can pass parameter also to a stored  
procedure