# SQL(Structured Query Language)

02 August 2024 19:19

For database management system or DBMS There are special kind of languages called query language that can. be used to access and manipulate data from the database. The structured query language [SQL] Is the most popular query language used by major relational database management systems such as MySql, Oracle SQL-server, etc.

SQL is easy to learn as the statement comprise of descriptive English words and are not case sensitive. We can create and interact with the database using SQL easily. Benefit of using SQL is that we do not have to specify how to get the data from the database. Rather, we simply specify what is to be retrieved and SQL does the rest. Although called a query language, SQL can do much more besides Querying.

SQL provides statement for defining the structure of data, manipulating data in the database in various ways, depending on our requirement.

# Data types and constraints in MySQL

We know that database consists of one or more relations and each relation(table) is made up of attributes(columns). Each attribute has a data type. We can also specify constraints for each attribute of a relation.

### Data type of Attribute

Data type of an attribute indicates the type of data value that an attribute can have. It also decides the operations that can be performed on the data of that attribute. For example, arithmetic operations can be performed on numeric data, but not on character data. Commonly use data types in my SQL are numeric types, date and time types and string types(Varchar and char).

Data type	Description
CHAR(n)	Specifies character type data of length n where N could be any value
	from zero to 255. CHAR is of fixed length, means declaring CHAR(10)
	implies to reserve space for 10 characters. If data does not have 10
	characters, MySQL fills the remaining six characters with spaces
	padded on the right.
VARCHAR(N)	Specifies character type data of length, where N could be any value
	from zero to 65,535. But unlike CHAR, VARCHAR(N) is a variable
	length data type. That is declaring varchar(30) Means a maximum of
	30 characters can be stored, but the actual allocated bytes will
	depend on the length of entered string.
INT	Int specifies an integer value. Each int value occupies four bytes
	of storage. The range of unsigned values allowed in a 4 byte integer
	type, 0 to 4294967295. Four values larger than that we have to use
	BigInt which occupies eight bytes.
FLOAT	Holt's number with decimal points. Each float value occupies four
	bytes.
DATE	The DATE Type Is used for dates in "YYYY-MM-DD" format. YYYY is the
	4 digit year, MM is the 2digit month and DD is 2 digit date. The
	support range is '1000-01-01' to '9999-12-31'

# Constraints

Constraints are the certain types of restrictions on the data values that an attribute can have. They are used to ensure correctness of data. However, it is not mandatory to define constraints for each attribute of a table.

	Constraints	Description
╗	Constraints	Description
	NOT NULL	Engune that a column cannot have NULL value whome NULL means missing on
П	NOT NOLL	Ensure that a column cannot have NULL value, where NULL means missing or
		unknown or not applicable value.
		difference of flot applicable value.

$\ $	UNIQUE	Ensures that all the values of a column are distinct or unique.	
U	ONTÁGE	Ensures that all the values of a column are distinct or unique.	
1	DEFAULT	A default values specified for the column. If no value is provided.	_
	DEFAULT	A default values specified for the column. If no value is provided.	
I	DDTMADW WEW		_
	PRIMARY KEY	The column, which can uniquely identify each row or record in a table.	
H	FOREIGN KEY	The column, which refers to a value of an attribute defined as primary key	_
	FOREIGN KET		
Ħ		in another table.	Τ

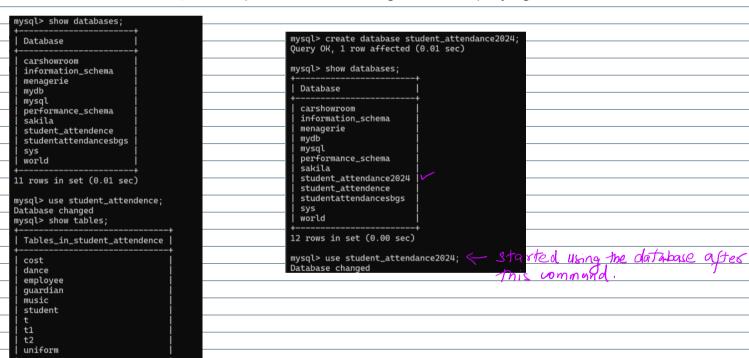
# SOL for Data Definition

In order to be able to store data, we need to first define the relationship schema. Defining a schema includes creating a relation and giving name to a relation identifying the attributes in a relation, deciding upon the data type for each attribute, and also specify the constraints as per requirements. Sometimes we may require to make changes to the relation schema also. SQL allows us to write statements for defining, modifying and deleting relation schemas. These are parts of data definition language, or DDL.

#### Create Database

# CREATE DATABASE databasename;

Add DBMS can manage multiple databases on one computer. Therefore, we need to select the database that we want to use. To know the names of existing database we use **the statement SHOW DATABASES**. From the listed databases, we can select the database to be used. Once the database is selected, we can proceed with creating tables or querying data.



# Create Table

10 rows in set (0.01 sec)

#### Syntax:

Create table tablename(
Attributename1 datatype constraint,
Attributename2 datatype constraint,

•

# AttributenameN datatype constraint,

It is important to observe the following points with respect to **create table** statement:

○ The number of columns in a table defines a degree of that relation, which is denoted

#### Attribute name specifies the name of the column in the table. o Data type specifies the type of data that an attribute can hold. o Constraint indicates the restriction imposed on the value of an attribute. By default, each attribute can take null value, except for the primary key. For example: mysql> CREATE TABLE STUDENT( mysql> CREATE TABLE STUDENT( -> ROLLNumber INT, -> SName VARCHAR(20), -> SDateOfBirth DATE, -> GUID CHAR(12), -> PRIMARY KEY (RollNumber)); Query OK, 0 rows affected (0.04 sec) mysql> show tables; | Tables\_in\_student\_attendance2024 | student 1 row in set (0.00 sec) of an already created table mysql> desc student; | Field Null | Key | Default | Extra | RollNumber int PRI NULL varchar(20) YES NULL **SName** SDateOfBirth YES NULL date char(12) NULL 4 rows in set (0.01 sec) mysql> create table guardian( -> guid char(12) -> gName varchar(20), -> gPhone int(10), -> gAddress varchar(30), -> PRIMARY KEY (guid)); Query OK, 0 rows affected, 1 warning (0.03 sec) mvsql> show tables: | Tables\_in\_student\_attendance2024 | guardian student 2 rows in set (0.01 sec) mysql> desc guardian; Field | Type | Null | Key | Default | Extra | char(12) guid gName varchar(20) NULL gPhone int YES NULL NULL varchar(30) gAddress 4 rows in set (0.01 sec) nysql> create table attendance( -> attendanceDate DATE, -> RollNumber int, -> attendanceStatus char(1), -> PRIMARY KEY(attendanceDate, RollNumber), -> FOREIGN KEY(RollNumber) REFERENCES Student(RollNumber) -> ON DELETE CASCADE); Query OK, 0 rows affected (0.06 sec) mysql> DESC attendance; | Field | Type Null | Key | Default | Extra | attendanceDate RollNumber NO PRI NULL NULL attendanceStatus | char(1) YES 3 rows in set (0.01 sec)

# Alter table

After reading a table, we may realize that we need to add or remove an attribute or to modify the data type of an existing attribute, or to add constraints in attribute. In all such cases, we need to change or alter the structure or schema of the table by using the

alter statement.

# A. Add primary key to a relation

```
mysql> alter table employee
-> add primary key(eid);
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> desc employee;
 Field
                                     | Null | Key | Default | Extra |
                  char(5) |
varchar(30) |
  eid
                                        NO
                                                  PRI
                                                           NULL
                                        YES
                                                           NULL
  ename
                   varchar(40)
   email
                                                           NULL
  mobilNum | int
4 rows in set (0.00 sec)
```

### B. Add foreign key to a relation

Once primary keys are added, the next step is to add foreign key to the relation, if any Following points needed to be observed while adding a foreign key to a relation:

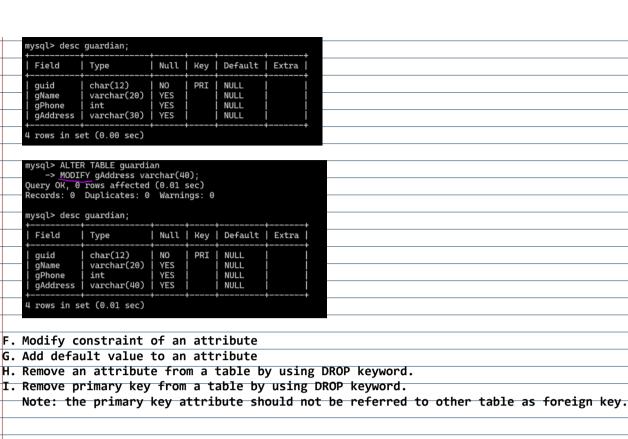
- o The referenced relation must be already created.
- The referenced attributes must be the part of the primary key of the reference to relation.
- o Data types and size of referenced and referencing attribute must be same.

mysql> desc department;								
Field	Туре	Null	Key	Default	Extra			
depid depName location	char(3) varchar(20) varchar(30)	YES YES YES		NULL NULL NULL				
++ 3 rows in set (0.00 sec)								
mysql> ALTER TABLE department -> ADD PRIMARY KEY(depid); Query OK, 0 rows affected (0.05 sec) Records: 0 Duplicates: 0 Warnings: 0 mysql> desc department;								
Field	Туре	Null	Key	Default	Extra			
depid depName location	char(3) varchar(20) varchar(30)	NO YES YES	PRI	NULL NULL NULL				
3 rows in se	et (0.01 sec)							
mysql> desc	employee;							
Field	Туре	Null	Key	Default	Extra			
eid   ename   email   mobilNum   depid	char(5) varchar(30) varchar(40) int char(3)	NO YES YES YES YES	PRI	NULL NULL NULL NULL NULL				
5 rows in se	 ≥t (0.00 sec)							

mysql> ALTER TABLE employee
-> ADD FOREIGN KEY(depid) REFERENCES department(depid);
Query OK, 0 rows affected (0.08 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc	employee;				
Field	Туре	Null	Key	Default	Extra
eid   ename   email   mobilNum   depid	char(5) varchar(30) varchar(40) int char(3)	NO YES YES YES YES	PRI MUL	NULL NULL NULL NULL	
5 rows in se	et (0.00 sec)				

- C. Add constraint UNIQUE to an existing attribute.
- D. Add an attribute to an existing table.
- E. Modify datatype of an attribute

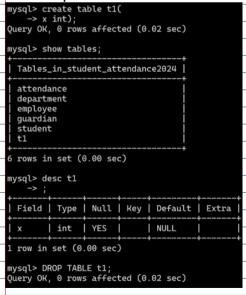


# DROP Statement

Sometimes a table in a database or a database itself needs to be removed. We can use a drop statement to remove a database or a table permanently from the system. However, one should be very cautious while using this statement as it cannot be undone.

# Syntax: DROP TABLE table\_name; Syntax: DROP DATABASE database\_name;

#### For example:



```
mysql> show tables;
  Tables_in_student_attendance2024 |
  attendance
  department
  employee
  guardian
  student
5 rows in set (0.00 sec)
Insertion of Records
INSERT INTO statement is used to insert new records in a table.
Syntax:
       INSERT INTO tableName
       VALUES(value1, value2, ... valueN);
Here value one corresponds to attribute one value 2 corresponds to attribute two, and so on.
Note that we need not to specify attribute name in the insert statement. If there are
exactly the same number of values in the insert statement, as the total number of attributes
in the table.
Caution: While populating records in a table with foreign key, ensure that records in
referenced table are already populated.
mysql> INSERT INTO guardian -> VALUES ('G0000000001',
                              'Amit Kumar', 987654101, 'Garia Garden');
 Query OK, 1 row affected (0.02 sec)
mysql> INSERT INTO guardian
  -> VALUES ('G0000000002', 'Sumi'
Query OK, 1 row affected (0.01 sec)
                              'Sumit Kumar', 987655101, 'Garia Garden');
mysql> INSERT INTO guardian
-> VALUES ('G0000000003', 'Ajay
Query OK, 1 row affected (0.01 sec)
                              'Ajay Parakh', 887653102, 'Garia Station Road');
Another Syntax:
INSERT INTO tableName(column1, column2,...columnN)
VALUES (value1, value2, ... valueN);
For example:
mysql> INSERT INTO guardian(guid, gName, gPhone, gAddress)
-> VALUES('G000000000004', 'Harshit Agarwal', 888999111, 'New Garia');
Query OK, 1 row affected (0.01 sec)
 mysql> select * from guardian;
   guid
                      gName
                                             gPhone
                                                           gAddress
   G00000000004
                      Harshit Agarwal
                                             888999111
                                                            New Garia
                                             987654101
                                                           Garia Garden
   G0000000001
                      Amit Kumar
                                             987655101
   G0000000002
                      Sumit Kumar
                                                            Garia Garden
   G000000003
                      Ajay Parakh
                                             887653102
                                                           Garia Station Road
 4 rows in set (0.00 sec)
                                                                                                                attributes
mysql> INSERT INTO guardian(guid, gAddress, gName, gPhone) -> VALUES('G00000000005','Maheshtalla', 'Nupur Verma', 999888777);
 -> VALUES('G000000000005','Mahes
Query OK, 1 row affected (0.01 sec)
 mysql> select * from guardian;
                gName
                                   gPhone
 quid
                                              gAddress
   G000000000004
                 Harshit Agarwal
                                    888999111
                                               New Garia
   G00000000005
                                    999888777
                                                Maheshtalla
                  Nupur Verma
   G0000000001
                  Amit Kumar
                                   987654101
                                               Garia Garden
                                    987655101
   G00000000000
                  Sumit Kumar
                                               Garia Garden
                                    887653102
                                               Garia Station Road
   G0000000003
                 Ajay Parakh
 5 rows in set (0.01 sec)
```

```
mysql> ALTER TABLE student
     -> ADD FOREIGN KEY(GUID) REFERENCES guardian(GUID);
Query OK, 1 row affected (0.09 sec)
Records: 1 Duplicates: 0 Warnings: 0
 mysql> desc student;
  Field
                          | Null | Key | Default | Extra |
  RollNumber
                                        NULL
                varchar(20)
  SName
                            YES
                                        NULL
  SDateOfBirth
               date
                            YES
                                        NULL
                char(12)
                                       NULL
 4 rows in set (0.01 sec)
 mysql> INSERT INTO student
-> VALUES(2, 'James Parakh', '2007-05-17', 'G000000000003');
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`student_attendance2024`.`student`, CONSTRAINT `student_ibfk_1` FOREIGN KEY (`GUID`) REFERENCES `guardian` (`guid`))
We can only provide values from parent table to child table:
-> VALUES(2, 'James Parakh', '2007-05-17', 'G0000000003');
Query OK, 1 row affected (0.01 sec)
mysql> select * from student;
| RollNumber | SName
                         | SDateOfBirth | GUID
          1 | Jack Parakh
                          2007-05-17
                                        G0000000003
                          2007-05-17
             James Parakh
                                        G0000000003
2 rows in set (0.00 sec)
SQL for Data Query
The SQL statement SELECT is used to retrieve data from the tables in a database and is also
called a query statement.
SELECT Statement
The SOL statement SELECT is used to retrieve data from the tables in a database, and the
output is also displayed in tabular form.
Syntax:
      SELECT attribute1, attribute2, attribute3,...
      FROM table name
      WHERE condition:
Here attribute1, attribute2,… Are the column names of table table name from which we want to
retrieve data. The FROM clause is always written with the select clause as it specifies the
name of the table from which data is to be retrieved. The WHERE clause is optional and is
used to retrieve data that meet the specified condition(s).
To select all data available in a table, we use the following select statement:
        select * from table_name;
   For example:
           select * from student;
    mysql> select * from student;
     RollNumber | SName
                             | SDateOfBirth | GUID
                              2007-05-17
2007-05-17
                 Jack Parakh
                                            G000000003
               James Parakh
                                            G0000000003
    2 rows in set (0.00 sec)
   NSERT INTO `student_attendance2024`.`department` (`depid`, `depName`, `location`) VALUES
   ('D03', 'Accounts', 'Garia');
   INSERT INTO `student_attendance2024`.`department` (`depid`, `depName`, `location`) VALUES
   ('D04', 'Marketing', 'Garia');
   INSERT INTO `student_attendance2024`.`department` (`depid`, `depName`, `location`) VALUES
   ('D05', 'Finance', 'Patuli');
   INSERT INTO `student attendance2024`.`department` (`depid`, `depName`,
                                                                                             `location`) VALUES
```

```
('D01', 'Supply', 'Tollygung');
    INSERT INTO `student_attendance2024`.`department` (`depid`, `depName`, `location`) VALUES
    ('D02', 'Maintenance', 'Tollygung');
   Retrieve selected columns
    Consider the table employee:
     mysql> select * from employee;
     | EmpNo | Ename
                         | Salary | Bonus | DeptId |
         101 | Aaliya
102 | Kritika
                            10000
                                             D<sub>0</sub>2
                                       123
566
565
                                             D01
D01
                            60000
               Shabbir
                            45000
                                             D04
D03
D02
D01
D05
         104
               Gurpreet
                            19000
                                       875
695
         105
               Joseph
                            34000
               Sanya
                            48000
15000
         106
               Vergese
Nachaobi
                                      NULL
         108
                            29000
                                      NULL
                                             D04
D02
D05
D02
                                      NULL
234
467
         109
               Daribha
                            42000
         101
110
               Aaliya
                            10000
         110 | Tanya
111 | Kapil
                            50000
                            15000
                                       763
    12 rows in set (0.00 sec)
A. Retrieve selected columns
       mysql> select Ename from employee;
                                                                    mysql> select Ename, salary from employee;
       Ename
                                                                                | salary |
                                                                      Ename
         Aaliya
                                                                                   10000
                                                                      Aaliva
         Kritika
                                                                      Kritika
                                                                                   60000
         Shabbir
                                                                      Shabbir
                                                                                   45000
                                                                                   19000
         Gurpreet
                                                                      Gurpreet
         Joseph
                                                                                   34000
                                                                      Joseph
                                                                                   48000
         Sanya
                                                                      Sanya
                                                                      Vergese
Nachaobi
         Vergese
                                                                                   15000
                                                                                   29000
42000
         Nachaobi
         Daribha
                                                                      Daribha
                                                                                   10000
50000
         Aaliya
                                                                      Aaliya
         Tanya
Kapil
                                                                      Tanya
Kapil
                                                                                   15000
       12 rows in set (0.00 sec)
                                                                    12 rows in set (0.00 sec)
      B. Renaming of columns
mysql> select Ename as Name, salary as Salary from employee;
                       | Salary |
             Name
             Aaliya
Kritika
                          10000
                          60000
             Shabbir
                          45000
             Gurpreet
                          19000
             Joseph
                          34000
             Sanya
                          48000
                          15000
             Vergese
Nachaobi
                          29000
             Daribha
                          42000
             Aaliya
                          10000
                          50000
             Tanya
             Kapil
                          15000
           12 rows in set (0.00 sec)
```

mysql> selec	t Ename	as Name,	(salary	* 12)	as	'Yearly	Salary'	from	employee	i
Name	Yearly	Salary								
Aaliya		120000								
Kritika		720000								
Shabbir		540000								
Gurpreet		228000								
Joseph		408000								
Sanya		576000								
Vergese		180000								
Nachaobi		348000								
Daribha		504000								
Aaliya		120000								
Tanya		600000								
Kapil		180000								
+		+								
12 rows in s	et (0.0	ec)								

# C. Distinct clause:

By default SQL shows all the data retrieved through query output. However, there can be duplicate values. The select statement, when combined with distinct clause, returns a record without repetition. For example, while. retrieving a department number from the employee relation, there can be a duplicate values, as many employees are assigned to the same department. To select unique department number for all the employees

we use 'distinct' as shown below: mysql> select \* from employee; | EmpNo | Ename | Salary | Bonus | DeptId | Aaliya Kritika 10000 60000 45000 19000 123 566 565 D01 D01 103 Shabbir 104 Gurpreet D<sub>0</sub>4 875 695 D03 D02 105 34000 Joseph 106 Sanya 48000 Vergese Nachaobi 107 15000 NULL D01 108 29000 NULL D05 D04 D02 D05 42000 10000 NULL 109 Daribha 101 Aaliva 234 Tanya Kapil 110 467 15000 12 rows in set (0.00 sec) mysql> select distinct DeptId from employee; D02 D01 D04 D05 5 rows in set (0.00 sec)

# D. WHERE Clause:

The wear clauses used to retrieve data that meet some specified condition. In the employee table, more than one employee can be working in one department.

```
nysql> select * from employee
   -> where salary > 20000 and deptid='D04';
  EmpNo | Ename
                      | Salary | Bonus | DeptId |
     109 | Daribha |
                          42000
                                      NULL | D04
1 row in set (0.00 sec)
mysql> select * from employee
-> where not ename='Aaliya';
  EmpNo | Ename
                        | Salary | Bonus | DeptId |
            Kritika
                            60000
                                         566
565
     103
                            45000
                                                D01
            Shabbir
     104
            Gurpreet
                            19000
                                                D04
                                                D03
D02
D01
D05
D04
            Joseph
Sanya
                                         875
     105
                            34000
     106
107
                            48000
                                        695
            Vergese
Nachaobi
                            15000
                                        NULL
     108
                            29000
                                        NULL
     109
            Daribha
                            42000
                                        NULL
            Tanya
Kapil
                            50000
15000
                                        467
763
                                                D05
     110
     111
                                                D02
10 rows in set (0.00 sec)
```

#### BETWEEN Clause

The 'between operator' defines the range of values in which the column value must fall into to make the condition true.

```
mysql> select ename as Name, deptid as 'Department ID' from employee -> where salary between 20000 and 50000
               | Department ID |
  Name
   Shabbir
                 D01
                 D03
   Joseph
   Sanya
                 D02
   Nachaobi
                 D05
  Daribha
                 D<sub>0</sub>4
   Tanya
                 D<sub>0</sub>5
6 rows in set (0.00 sec)
```

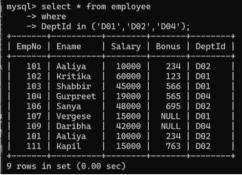
Printing records for selected departments using 'OR' operator.

```
mysql> select * from employee
     -> where
    -> DeptId = 'D01' OR DeptId = 'D02' OR DeptID = 'D04';
  EmpNo | Ename
                      | Salary | Bonus | DeptId
           Aaliya
Kritika
                         10000
                         60000
                                    123
     102
                                           D01
     103
           Shabbir
                         45000
                                           D01
     104
           Gurpreet
                         19000
                                    565
                                           D04
     106
           Sanya
                         48000
                                   695
NULL
                                           D02
                                          D01
D04
    107
           Vergese
Daribha
                         15000
                         42000
                                   NULL
     109
                                    234
763
     101
           Aaliya
                         10000
                                           D02
           Kapiĺ
9 rows in set (0.00 sec)
```

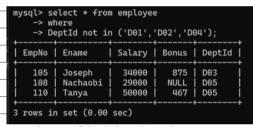
```
mysql> select * from employee
   -> NOT (DeptId = 'D01' OR DeptId = 'D02' OR DeptID = 'D04');
| EmpNo | Ename
                   | Salary | Bonus | DeptId |
   105
         Joseph
                      34000
                               875
                                      D03
         Nachaobi
   110
         Tanya
                      50000
                               467
                                     D05
3 rows in set (0.00 sec)
```

#### MEMBERSHIP operator IN

The 'IN' operator compares a value with a set of values and returns true if the value belongs to that set.



The members of 'D01', 'D02', 'D04' is selected for the output.



Not is applied in the above query to select the rows which are not the member of deptid('D01', 'D02', and 'D04').

# ORDER BY Clause

Order by clause is used to display data in an ordered form with respect to a specified column. By default, 'order by' displays records in ascending order of the specified columns values. To display the record in descending order. The DESC (means descending) keyword needs to be written with that column.

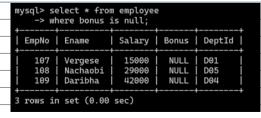
-> 0	<del>!</del>	<del>!</del>		
EmpNo	Ename +	Salary 	Bonus	DeptId +
101	Aaliya	10000	234	D02
107	Vergese	15000	NULL	D01
111	Kapil	15000	763	D02
104	Gurpreet	19000	565	D04
108	Nachaobi	29000	NULL	D05
105	Joseph	34000	875	D03
109	Daribha	42000	NULL	D04
103	Shabbir	45000	566	D01
106	Sanya	48000	695	D02
110	Tanya	50000	467	D05
	Kritika + in set (0.00		123 	D01 +
  11 rows	+	; ; ; sec) m employe		+ +
+ 11 rows mysql> s -> o	+in set (0.00	; ; ; sec) m employe		D01 
# 11 rows   mysql> s: -> o:   EmpNo   102	+ in set (0.00 elect * fro rder by sal	m employed ary desc;    Salary   60000	Bonus	+   DeptId 
mysql> s -> o +   EmpNo +   102   110	in set (0.00 elect * froi rder by sald +	m employed ary desc; 	Bonus 123 467	+   DeptId +   D01   D05
mysql> s -> o +   EmpNo +   102   110   106	in set (0.00 elect * fron rder by sala 	9 sec) m employedary desc;	Bonus 123 467 695	DeptId
mysql> s -> o EmpNo   102   110   106   103	in set (0.00 elect * froi rder by sala    Ename   Kritika   Tanya   Sanya   Shabbir	9 sec) m employed ary desc;   Salary   60000   50000   48000   45000	Bonus 123 467 695 566	DeptId  DeptId  Doble  Doble  Doble  Doble  Doble  Doble
mysql> s: -> o:   EmpNo   102   109	in set (0.00 elect * froi rder by sald Ename Kritika   Tanya   Sanya   Shabbir   Daribha	m employed ary desc;   Salary   60000   50000   48000   45000   42000	Bonus 123 467 695 566 NULL	DeptId  DoptId  DoptId
mysql> s -> o EmpNo 	in set (0.00 elect * frounder by salification of the salification	m employedary desc;   Salary   Salary   60000   50000   48000   45000   42000   34000	Bonus 123 467 695 566 NULL 875	DeptId  DeptId  D01  D05  D02  D01  D04  D04
mysql> s -> o EmpNo +   EmpNo +   102   110   106   103   109   109   105   108	in set (0.0) elect * froi rder by sali Ename Kritika Tanya Sanya Shabbir Daribha Joseph Nachaobi	m employed ary desc;   Salary   60000   50000   48000   42000   34000   29000	Bonus 123 467 695 566 NULL 875 NULL	DeptId  D01  D05  D02  D01  D04  D03  D05
mysql> s -> o EmpNo 1106 106 108 109 108 108 109	in set (0.00 elect * froi rder by sala tename tename Kritika Tanya Sanya Shabbir Daribha Joseph Nachaobi Gurpreet	m employedary desc;	Bonus 123 467 695 566 NULL 87 NULL 565	DeptId  DeptId
	in set (0.00 elect * froi rder by sald Ename Kritika Tanya Sanya Shabbir Daribha Joseph Nachaobi Gurpreet Vergese	m employedary desc; Salary Salary 60000 50000 48000 48000 42000 34000 29000 19000 15000	Bonus 123 467 695 566 NULL 875 NULL 565 NULL	DeptId DeptId D01 D05 D02 D01 D04 D03 D05 D04 D04
mysql> s -> o EmpNo 1106 106 108 109 108 108 109	in set (0.00 elect * froi rder by sala tename tename Kritika Tanya Sanya Shabbir Daribha Joseph Nachaobi Gurpreet	m employedary desc;	Bonus 123 467 695 566 NULL 87 NULL 565	DeptId  DeptId

	mysql> select * from employee -> order by ename desc;							
EmpNo	Ename	Salary	Bonus	DeptId				
107	Vergese	15000	NULL	D01				
110	Tanya	50000	467	D05				
103	Shabbir	45000	566	D01				
106	Sanya	48000	695	D02				
108	Nachaobi	29000	NULL	D05				
102	Kritika	60000	123	D01				
111	Kapil	15000	763	D02				
105	Joseph	34000	875	D03				
104	Gurpreet	19000	565	D04				
109	Daribha	42000	NULL	D04				
101	Aaliya	10000	234	D02				
+				·				
11 rows	in set (0.00	sec)						

# Handling NULL Values

SQL supports a special value called NULL to represent a missing or unknown value. For example, the bonus column in the employee table can have missing values for certain records. Hence, null is used to represent such unknown values. It is important to note that the null is different from zero. Also any arithmetic operation performed with null values gives null.

For example, 5+ NULL = NULL, because NULL is unknown hence the result is also unknown in order to check for NULL value in a column we use 'is null' operator.



	mysql> select * from employee -> where bonus is not null and deptid = 'D01';								
	EmpNo	Ename	Salary	Bonus	DeptId				
		Kritika Shabbir							
1	2 rows in	n set (0.00	sec)						

# Substring pattern matching

Many a times we come across a situation where we do not want to query by matching exact text or value. Rather, we are interested to find matching of only a few characters or values in column values. For example, to find out the name, starting with "T". or to find out a pin code, starting with "60". This is called substring pattern matching. We cannot match such pattern using equal to operator as we are not looking for exact match. SQL provides a "like" operator that can be used with. the wear clause to search for a specified pattern in a column.

The "LIKE" operator makes use of the following two wild card characters:

- %(percent) used to represent zero 1 or multiple characters.
- \_(underscore) He used to represent exactly a single character.

# Examples:



	mysql> select * from employee -> where ename like '%A%';								
EmpNo	Ename	Salary	Bonus	DeptId					
102	Kritika	60000	123	D01					
103	Shabbir	45000	566	D01					
106	Sanya	48000	695	D02					
108	Nachaobi	29000	NULL	D05					
109	Daribha	42000	NULL	D04					
110	Tanya	50000	467	D05					
111	Kapil	15000	763	D02					
101	Aaliya	10000	234	D02					
8 rows in	set (0.00	sec)	+	++					

# mysql> select \* from employee -> where ename like '%e%'; | EmpNo | Ename | Salary | Bonus | DeptId | | 104 | Gurpreet | 19000 | 565 | D04 | | 105 | Joseph | 34000 | 875 | D03 | | 107 | Vergese | 15000 | NULL | D01 | | 3 rows in set (0.00 sec) mysql> select \* from employee -> where ename like '%i%'; | EmpNo | Ename | Salary | Bonus | DeptId | | 102 | Kritika | 60000 | 123 | D01 | | 103 | Shabbir | 45000 | 566 | D01 | | 108 | Nachaobi | 29000 | NULL | D05 | | 109 | Daribha | 42000 | NULL | D04 | | 111 | Kapil | 15000 | 763 | D02 | | 101 | Aaliya | 10000 | 234 | D02 | | 6 rows in set (0.00 sec) mysql> select \* from employee -> where ename like '%e%' -> and salary > 40000; Empty set (0.00 sec) mysql> select \* from employee -> where ename like '%i%'

mysql> select * from employee -> where ename like '%e%' -> and salary > 40000; Empty set (0.00 sec)									
_> wh	mysql> select * from employee -> where ename like '%i%' -> and salary > 40000;								
EmpNo	Ename	Salary	Bonus	DeptId	į				
103	Kritika Shabbir Daribha	45000	123 566 NULL	D01					
3 rows in	set (0.00	sec)			_				

mysql> sel -> whe		om employe like '%se		
EmpNo	Ename	Salary	Bonus	DeptId
		34000 15000		
2 rows in	set (0.06	sec)		