

1.Introduction:

-Structured Query language

-standard language to deal with RELATIONAL databases (RDBMS)

RDBMS: Store data in tables (in rows and columns)

-SQL keywords are not case-sensitive (insert and INSERT are the same)

-separate SQL queries that can be executed in one call to the server using separator (;) so; can be used to terminate an SQL query

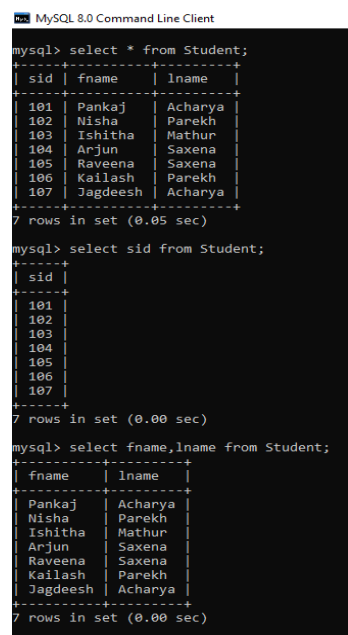
2.SELECT:

-select all the columns:

Select * from tablename

-select particular columns:

Select col1, col2 from tablename



```
mysql> select * from Student;
+----+-----+-----+
| sid | fname | lname |
+----+-----+-----+
| 101 | Pankaj | Acharya |
| 102 | Nisha | Parekh |
| 103 | Ishitha | Mathur |
| 104 | Arjun | Saxena |
| 105 | Raveena | Saxena |
| 106 | Kailash | Parekh |
| 107 | Jagdeesh | Acharya |
+----+-----+-----+
7 rows in set (0.05 sec)

mysql> select sid from Student;
+----+
| sid |
+----+
| 101 |
| 102 |
| 103 |
| 104 |
| 105 |
| 106 |
| 107 |
+----+
7 rows in set (0.00 sec)

mysql> select fname,lname from Student;
+-----+-----+
| fname | lname |
+-----+-----+
| Pankaj | Acharya |
| Nisha | Parekh |
| Ishitha | Mathur |
| Arjun | Saxena |
| Raveena | Saxena |
| Kailash | Parekh |
| Jagdeesh | Acharya |
+-----+-----+
7 rows in set (0.00 sec)
```

Fig-2: Display Student table with three columns-sid,fname and lname

3.DISTINCT:

- selects only distinct values of the given column
- does not select duplicate values
- select distinct colname from tablename*

```
MySQL 8.0 Command Line Client
Query OK, 1 row affected (0.12 sec)

mysql> select * from hobbies;
+----+-----+
| sid | hobby |
+----+-----+
| 101 | drawing |
| 101 | painting |
| 102 | singing |
| 102 | painting |
| 102 | swimming |
| 103 | dancing |
| 103 | drawing |
| 103 | singing |
| 104 | singing |
| 105 | cooking |
| 104 | cooking |
| 106 | cooking |
| 107 | painting |
| 107 | dancing |
| 108 | swimming |
| 109 | cooking |
| 109 | drawing |
+----+-----+
17 rows in set (0.00 sec)

mysql> select DISTINCT hobby from hobbies;
+-----+
| hobby |
+-----+
| drawing |
| painting |
| singing |
| swimming |
| dancing |
| cooking |
+-----+
6 rows in set (0.00 sec)

mysql>
```

Fig-3.1: Display unique hobbies of students

```
MySQL 8.0 Command Line Client

mysql> select * from hobbies;
+----+-----+
| sid | hobby |
+----+-----+
| 101 | drawing |
| 101 | painting |
| 102 | singing |
| 102 | painting |
| 102 | swimming |
| 103 | dancing |
| 103 | drawing |
| 103 | singing |
| 104 | singing |
| 105 | cooking |
| 104 | cooking |
| 106 | cooking |
| 107 | painting |
| 107 | dancing |
| 108 | swimming |
| 109 | cooking |
| 109 | drawing |
+----+-----+
17 rows in set (0.00 sec)

mysql> select count(distinct hobby) from hobbies;
+-----+
| count(distinct hobby) |
+-----+
| 6 |
+-----+
1 row in set (0.00 sec)

mysql>
```

Fig-3.2: Count the number of unique records using count (distinct colname)

4.WHERE:

- select only the records which satisfy the required condition
- select col1 from tablename where condition*
- The where clause can be used with Update, Insert etc., as well
- condition can be a number or a string.

Select col from tablename where col="abc"

Select col from tablename where col=123

-Operators which can be used with where clause:

=,>,<,<=,>=,!= or <>,BETWEEN....AND,IN,LIKE

```
MySQL 8.0 Command Line Client
mysql> select * from Student;
+----+-----+-----+
| sid | fname | lname |
+----+-----+-----+
| 101 | Pankaj | Acharya |
| 102 | Nisha | Parekh |
| 103 | Ishitha | Mathur |
| 104 | Arjun | Saxena |
| 105 | Raveena | Saxena |
| 106 | Kailash | Parekh |
| 107 | Jagdeesh | Acharya |
| 108 | Ishitha | Singh |
| 109 | Arjun | Malhotra |
+----+-----+-----+
9 rows in set (0.23 sec)

mysql> select sid from Student where lname="Saxena";
+----+
| sid |
+----+
| 104 |
| 105 |
+----+
2 rows in set (0.00 sec)

mysql> select * from Student where sid=103;
+----+-----+-----+
| sid | fname | lname |
+----+-----+-----+
| 103 | Ishitha | Mathur |
+----+-----+-----+
1 row in set (0.03 sec)

mysql> select fname,lname from Student where sid BETWEEN 102 and 105;
+-----+-----+
| fname | lname |
+-----+-----+
| Nisha | Parekh |
| Ishitha | Mathur |
| Arjun | Saxena |
| Raveena | Saxena |
+-----+-----+
4 rows in set (0.04 sec)
```

Fig-4.1: where clause with strings and numbers in the condition

```
MySQL 8.0 Command Line Client
4 rows in set (0.04 sec)

mysql> select fname,lname from Student where sid > 106;
+-----+-----+
| fname | lname |
+-----+-----+
| Jagdeesh | Acharya |
| Ishitha | Singh |
| Arjun | Malhotra |
+-----+-----+
3 rows in set (0.00 sec)

mysql> select fname,lname from Student where sid < 103;
+-----+-----+
| fname | lname |
+-----+-----+
| Pankaj | Acharya |
| Nisha | Parekh |
+-----+-----+
2 rows in set (0.00 sec)

mysql> select fname,lname from Student where sid <= 105;
+-----+-----+
| fname | lname |
+-----+-----+
| Pankaj | Acharya |
| Nisha | Parekh |
| Ishitha | Mathur |
| Arjun | Saxena |
| Raveena | Saxena |
+-----+-----+
5 rows in set (0.00 sec)

mysql> select fname,lname from Student where sid >= 105;
+-----+-----+
| fname | lname |
+-----+-----+
| Raveena | Saxena |
| Kailash | Parekh |
| Jagdeesh | Acharya |
| Ishitha | Singh |
| Arjun | Malhotra |
+-----+-----+
5 rows in set (0.00 sec)
```

Fig-4.2: where clause with BETWEEN...AND and a few other operators

```

MySQL 8.0 Command Line Client
+----+-----+-----+
| sid | fname | lname |
+----+-----+-----+
| 101 | Pankaj | Acharya |
| 102 | Nisha | Parekh |
| 103 | Ishitha | Mathur |
| 104 | Arjun | Saxena |
| 106 | Kailash | Parekh |
| 107 | Jagdeesh | Acharya |
| 108 | Ishitha | Singh |
| 109 | Arjun | Malhotra |
+----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from Student where sid != 105;
+----+-----+-----+
| sid | fname | lname |
+----+-----+-----+
| 101 | Pankaj | Acharya |
| 102 | Nisha | Parekh |
| 103 | Ishitha | Mathur |
| 104 | Arjun | Saxena |
| 106 | Kailash | Parekh |
| 107 | Jagdeesh | Acharya |
| 108 | Ishitha | Singh |
| 109 | Arjun | Malhotra |
+----+-----+-----+
8 rows in set (0.00 sec)

mysql> select * from Student where sid <> 105;
+----+-----+-----+
| sid | fname | lname |
+----+-----+-----+
| 101 | Pankaj | Acharya |
| 102 | Nisha | Parekh |
| 103 | Ishitha | Mathur |
| 104 | Arjun | Saxena |
| 106 | Kailash | Parekh |
| 107 | Jagdeesh | Acharya |
| 108 | Ishitha | Singh |
| 109 | Arjun | Malhotra |
+----+-----+-----+
8 rows in set (0.00 sec)

```

Fig-4.3: where clause with a few operators

5.AND, OR, NOT:

- when we have multiple conditions, combine them using AND, OR , NOT and use in the where clause
- OR: True even if one of the conditions is True
- AND: True only if all the conditions are True
- NOT: Negate the condition

```

MySQL 8.0 Command Line Client
+----+
| 107 |
| 109 |
| 109 |
+----+
15 rows in set (0.00 sec)

mysql> select distinct sid from hobbies where NOT hobby="swimming";
+----+
| sid |
+----+
| 101 |
| 102 |
| 103 |
| 104 |
| 105 |
| 106 |
| 107 |
| 109 |
+----+
8 rows in set (0.00 sec)

mysql> select distinct sid from hobbies where hobby="swimming" OR hobby="cooking";
+----+
| sid |
+----+
| 102 |
| 104 |
| 105 |
| 106 |
| 108 |
| 109 |
+----+
6 rows in set (0.00 sec)

mysql> select distinct sid from hobbies where hobby="singing" AND hobby="swimming";
Empty set (0.00 sec)

mysql> select distinct sid from hobbies where hobby="singing" AND sid=105;
Empty set (0.00 sec)

```

Fig-5: AND OR NOT

6.ORDER BY:

- sort the results in ascending order(default) or descending order
- We can order by a single column or by several columns.
- select colname from tablename ORDER BY colname-sorted output in ascending order
- select colname from tablename ORDER BY colname DESC-get sorted output in descending order

-sort strings and numbers

```
MySQL 8.0 Command Line Client
+----+-----+-----+
| 106 | Kailash | Parekh |
| 107 | Jagdeesh | Acharya |
| 108 | Ishitha | Singh |
| 109 | Arjun | Malhotra |
+----+-----+-----+
9 rows in set (0.00 sec)

mysql> select * from Student ORDER BY fname;
+----+-----+-----+
| sid | fname | lname |
+----+-----+-----+
| 104 | Arjun | Saxena |
| 109 | Arjun | Malhotra |
| 103 | Ishitha | Mathur |
| 108 | Ishitha | Singh |
| 107 | Jagdeesh | Acharya |
| 106 | Kailash | Parekh |
| 102 | Nisha | Parekh |
| 101 | Pankaj | Acharya |
| 105 | Raveena | Saxena |
+----+-----+-----+
9 rows in set (0.07 sec)

mysql> select * from Student ORDER BY fname desc;
+----+-----+-----+
| sid | fname | lname |
+----+-----+-----+
| 105 | Raveena | Saxena |
| 101 | Pankaj | Acharya |
| 102 | Nisha | Parekh |
| 106 | Kailash | Parekh |
| 107 | Jagdeesh | Acharya |
| 103 | Ishitha | Mathur |
| 108 | Ishitha | Singh |
| 104 | Arjun | Saxena |
| 109 | Arjun | Malhotra |
+----+-----+-----+
9 rows in set (0.00 sec)
```

Fig-6.1: ORDER BY

Suppose if two entries match, then in the results be matched in the order in which the rows were inserted. Instead, we can mention a second column:

-select colname from tablename ORDER BY col1, col2;

-select colname from tablename ORDER BY col1 ASC, col2 DESC;

```
MySQL 8.0 Command Line Client
9 rows in set (0.00 sec)

mysql> select * from Student ORDER BY fname, lname;
+----+-----+-----+
| sid | fname | lname |
+----+-----+-----+
| 109 | Arjun | Malhotra |
| 104 | Arjun | Saxena |
| 103 | Ishitha | Mathur |
| 108 | Ishitha | Singh |
| 107 | Jagdeesh | Acharya |
| 106 | Kailash | Parekh |
| 102 | Nisha | Parekh |
| 101 | Pankaj | Acharya |
| 105 | Raveena | Saxena |
+----+-----+-----+
9 rows in set (0.00 sec)

mysql> select * from Student order by fname, lname DESC;
+----+-----+-----+
| sid | fname | lname |
+----+-----+-----+
| 104 | Arjun | Saxena |
| 109 | Arjun | Malhotra |
| 108 | Ishitha | Singh |
| 103 | Ishitha | Mathur |
| 107 | Jagdeesh | Acharya |
| 106 | Kailash | Parekh |
| 102 | Nisha | Parekh |
| 101 | Pankaj | Acharya |
| 105 | Raveena | Saxena |
+----+-----+-----+
9 rows in set (0.00 sec)
```

Fig-6.2: ORDER BY two columns. If entries in 1st column are same, then results are sorted based on the entries in the 2nd column, as mentioned in ascending or descending order

7.INSERT INTO:

-insert new rows into the table

-insert into specific columns(2) or insert to all columns(1)

(1)INSERT INTO TABLENAME VALUES (<same order as that of the table>);

(2)INSERT INTO TABLENAME (<list of colnames>) VALUES (<list of values in the same order of the columns as mentioned>);

```

MySQL 8.0 Command Line Client
0 rows in set (0.00 sec)

mysql> insert into Student(sid,fname) values(110,"Annika");
Query OK, 1 row affected (0.10 sec)

mysql> select * from Student;
+----+-----+-----+
| sid | fname | lname |
+----+-----+-----+
| 101 | Pankaj | Acharya |
| 102 | Nisha | Parekh |
| 103 | Ishitha | Mathur |
| 104 | Arjun | Saxena |
| 105 | Raveena | Saxena |
| 106 | Kallash | Parekh |
| 107 | Jagdeesh | Acharya |
| 108 | Ishitha | Singh |
| 109 | Arjun | Malhotra |
| 110 | Annika | NULL |
+----+-----+-----+
10 rows in set (0.04 sec)

mysql> insert into Student values(111,"Guneet","Sikha");
Query OK, 1 row affected (0.12 sec)

mysql> select * from Student;
+----+-----+-----+
| sid | fname | lname |
+----+-----+-----+
| 101 | Pankaj | Acharya |
| 102 | Nisha | Parekh |
| 103 | Ishitha | Mathur |
| 104 | Arjun | Saxena |
| 105 | Raveena | Saxena |
| 106 | Kallash | Parekh |
| 107 | Jagdeesh | Acharya |
| 108 | Ishitha | Singh |
| 109 | Arjun | Malhotra |
| 110 | Annika | NULL |
| 111 | Guneet | Sikha |
+----+-----+-----+
11 rows in set (0.00 sec)

```

Fig-7: INSERT INTO STATEMENT

8.NULL:

-If a field (column value for an entry) in a table is optional, it can be ignored while inserting a new value or updating. This field has no value and is called NULL field.

-IS NULL and IS NOT NULL-operators used to check if a column has any NULL values.

```

MySQL 8.0 Command Line Client

mysql> select fname from Student where lname IS NULL;
+-----+
| fname |
+-----+
| Annika |
+-----+
1 row in set (0.00 sec)

mysql> insert into Student(sid) values(112);
Query OK, 1 row affected (0.12 sec)

mysql> select * from Student where fname IS NULL and lname IS NULL;
+----+-----+-----+
| sid | fname | lname |
+----+-----+-----+
| 112 | NULL | NULL |
+----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from Student where lname IS NOT NULL;
+----+-----+-----+
| sid | fname | lname |
+----+-----+-----+
| 101 | Pankaj | Acharya |
| 102 | Nisha | Parekh |
| 103 | Ishitha | Mathur |
| 104 | Arjun | Saxena |
| 105 | Raveena | Saxena |
| 106 | Kallash | Parekh |
| 107 | Jagdeesh | Acharya |
| 108 | Ishitha | Singh |
| 109 | Arjun | Malhotra |
| 111 | Guneet | Sikha |
+----+-----+-----+
10 rows in set (0.00 sec)

mysql>

```

Fig-8 : IS NULL and IS NOT NULL operators

9.UPDATE :

-update an existing record in the table

-*update tablename set colname=newValue where condition;*

-*update tablename set col1=val1,col2=val2 where condition;*

-if where clause is ignored then all the records will be updated!

```

MySQL 8.0 Command Line Client
mysql> update Student set lname="Acharya" where fname="Nisha";
Query OK, 1 row affected (0.14 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from Student;
+----+-----+-----+
| sid | fname | lname |
+----+-----+-----+
| 101 | Pankaj | Acharya |
| 102 | Nisha | Acharya |
| 103 | Ishitha | Mathur |
| 104 | Arjun | Saxena |
| 105 | Raveena | Saxena |
| 106 | Kailash | Parekh |
| 107 | Jagdeesh | Acharya |
| 108 | Ishitha | Singh |
| 109 | Arjun | Malhotra |
| 110 | Annika | NULL |
| 111 | Guneet | Sikha |
| 112 | NULL | NULL |
+----+-----+-----+
12 rows in set (0.00 sec)

mysql> update Student set fname="Bhavya",lname="Singh" where sid=108;
Query OK, 1 row affected (0.07 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from Student;
+----+-----+-----+
| sid | fname | lname |
+----+-----+-----+
| 101 | Pankaj | Acharya |
| 102 | Nisha | Acharya |
| 103 | Ishitha | Mathur |
| 104 | Arjun | Saxena |
| 105 | Raveena | Saxena |
| 106 | Kailash | Parekh |
| 107 | Jagdeesh | Acharya |
| 108 | Bhavya | Singh |
| 109 | Arjun | Malhotra |
| 110 | Annika | NULL |
| 111 | Guneet | Sikha |
| 112 | NULL | NULL |
+----+-----+-----+

```

Fig-9 : UPDATE

10.DELETE:

-Delete existing records from the table

-*delete from tablename where condition;* -delete specific records satisfying a particular condition from the table

-*delete from tablename;* -delete the entire table(delete all the records from the table)

-if where clause is ignored, the entire table will be deleted

```

MySQL 8.0 Command Line Client
mysql> select * from hobbies;
+----+-----+
| sid | hobby |
+----+-----+
| 101 | drawing |
| 101 | painting |
| 102 | singing |
| 102 | painting |
| 103 | dancing |
| 103 | drawing |
| 103 | singing |
| 104 | singing |
| 105 | cooking |
| 104 | cooking |
| 106 | cooking |
| 107 | painting |
| 107 | dancing |
| 109 | cooking |
| 109 | drawing |
+----+-----+
15 rows in set (0.00 sec)

mysql> delete from hobbies where hobby="cooking";
Query OK, 4 rows affected (0.11 sec)

mysql> select * from hobbies;
+----+-----+
| sid | hobby |
+----+-----+
| 101 | drawing |
| 101 | painting |
| 102 | singing |
| 102 | painting |
| 103 | dancing |
| 103 | drawing |
| 103 | singing |
| 104 | singing |
| 107 | painting |
| 107 | dancing |
| 109 | drawing |
+----+-----+
11 rows in set (0.00 sec)

```

Fig-10 : DELETE

11.LIMIT:

- Specify the number of records to return
- Returning a large number of records when dealing with larger databases impacts performance.
- LIMIT 4 returns only 4 records
- select * from tablename where condition LIMIT number;*

Number-no. of records to be returned

```
MySQL 8.0 Command Line Client
+----+-----+
| 108 | cooking |
| 107 | cooking |
| 109 | cooking |
| 102 | cooking |
| 106 | cooking |
+----+-----+
18 rows in set (0.00 sec)

mysql> select sid from hobbies where hobby="cooking";
+----+
| sid |
+----+
| 104 |
| 101 |
| 108 |
| 107 |
| 109 |
| 102 |
| 106 |
+----+
7 rows in set (0.00 sec)

mysql> select sid from hobbies where hobby="cooking" LIMIT 2;
+----+
| sid |
+----+
| 104 |
| 101 |
+----+
2 rows in set (0.00 sec)

mysql> select * from hobbies LIMIT 5;
+----+-----+
| sid | hobby |
+----+-----+
| 101 | drawing |
| 101 | painting |
| 102 | singing |
| 102 | painting |
| 103 | dancing |
+----+-----+
5 rows in set (0.00 sec)
```

Fig-11 : LIMIT clause

12.MIN() and MAX():

- MIN()-find minimum value of a column-*Select MIN(colname) from tablename;*
- MAX()-find maximum value of a column-*Select MAX(colname) from tablename;*

```
MySQL 8.0 Command Line Client

mysql> select * from student;
+----+-----+-----+-----+
| sid | fname | lname | perc |
+----+-----+-----+-----+
| 101 | Pankaj | Acharya | 89 |
| 102 | Nisha | Acharya | 72 |
| 103 | Ishltha | Mathur | 52 |
| 104 | Anjun | Saxena | 92 |
| 105 | Raveena | Saxena | 70 |
| 106 | Kailash | Parekh | 90 |
| 107 | Jagdeesh | Acharya | 64 |
| 108 | Bhavya | Singh | 83 |
| 109 | Anjun | Malhotra | 33 |
| 110 | Annika | NULL | 13 |
| 111 | Guneeet | Sikha | 45 |
| 112 | NULL | NULL | 58 |
+----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql> select min(perc) as min_score,max(perc) as max_score from Student;
+-----+-----+
| min_score | max_score |
+-----+-----+
| 13 | 92 |
+-----+-----+
1 row in set (0.07 sec)

mysql>
```

Fig-12: MIN () and MAX ()

13.COUNT(),AVG(),SUM():

-COUNT()-get the number of records satisfying the condition

-select count(colname) from tablename where condition;

-AVG()-average of all the values of the given column

-select avg(colname) from tablename where cond;

-SUM()-get the sum of all the values of the column

-select sum(col) from tablename where cond;

```
MySQL 8.0 Command Line Client
mysql> select * from Student;
+----+-----+-----+-----+
| sid | fname | lname | perc |
+----+-----+-----+-----+
| 101 | Pankaj | Acharya | 89 |
| 102 | Nisha | Acharya | 72 |
| 103 | Ishitha | Mathur | 52 |
| 104 | Arjun | Saxena | 92 |
| 105 | Raveena | Saxena | 70 |
| 106 | Kailash | Parekh | 90 |
| 107 | Jagdeesh | Acharya | 64 |
| 108 | Bhavya | Singh | 83 |
| 109 | Arjun | Malhotra | 33 |
| 110 | Annika | NULL | 13 |
| 111 | Guneet | Sikha | 45 |
| 112 | NULL | NULL | 58 |
+----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql> select avg(perc) as AVERAGE_MARKS from student;
+-----+
| AVERAGE_MARKS |
+-----+
| 63.416666666666664 |
+-----+
1 row in set (0.00 sec)

mysql> select count(sid) from Student where perc <= (select avg(perc) from Student);
+-----+
| count(sid) |
+-----+
| 5 |
+-----+
1 row in set (0.04 sec)

mysql> select count(sid) from student where perc > (select avg(perc) from student);
+-----+
| count(sid) |
+-----+
| 7 |
+-----+
1 row in set (0.00 sec)
```

Fig-13.1: COUNT (), AVG ()

```
MySQL 8.0 Command Line Client
+-----+
| 7 |
+-----+
1 row in set (0.00 sec)

mysql> select sum(perc) from Student where perc <= (select avg(perc) from Student);
+-----+
| sum(perc) |
+-----+
| 201 |
+-----+
1 row in set (0.04 sec)

mysql>
```

Fig-13.2: SUM ()

14.LIKE:

-search for a specified pattern

% zero or more characters

_ only one character

-select colname from tablename where colname LIKE "pattern";

-Eg: A% starts with A,%A ends with A,%TH% TH should be anywhere in the string

```

MySQL 8.0 Command Line Client

mysql> select * from Student;
+----+-----+-----+-----+
| sid | fname | lname | perc |
+----+-----+-----+-----+
| 101 | Pankaj | Acharya | 89 |
| 102 | Nisha | Acharya | 72 |
| 103 | Ishitha | Mathur | 52 |
| 104 | Arjun | Saxena | 92 |
| 105 | Raveena | Saxena | 70 |
| 106 | Kailash | Parekh | 90 |
| 107 | Jagdeesh | Acharya | 64 |
| 108 | Bhavya | Singh | 83 |
| 109 | Arjun | Malhotra | 33 |
| 110 | Annika | NULL | 13 |
| 111 | Guneet | Sikha | 45 |
| 112 | NULL | NULL | 58 |
+----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql> select sid,perc from Student where fname like "A%";
+----+-----+
| sid | perc |
+----+-----+
| 104 | 92 |
| 109 | 33 |
| 110 | 13 |
+----+-----+
3 rows in set (0.00 sec)

mysql> select sid,perc from Student where lname like "%a";
+----+-----+
| sid | perc |
+----+-----+
| 101 | 89 |
| 102 | 72 |
| 104 | 92 |
| 105 | 70 |
| 107 | 64 |
| 109 | 33 |
| 111 | 45 |
+----+-----+
7 rows in set (0.00 sec)

```

Fig-14: LIKE

15.Wildcard characters:

- Substitute one or more characters in a string
- used with LIKE
- Wildcard characters are % and _

16.IN:

- Specify multiple values in a where clause-Equivalent to combining multiple ORs
- select colname from tablename where condition IN sub-query;*

```

MySQL 8.0 Command Line Client

mysql> select s.fname,s.lname,h.hobby from student s natural join hobbies h;
+-----+-----+-----+
| fname | lname | hobby |
+-----+-----+-----+
| Pankaj | Acharya | drawing |
| Pankaj | Acharya | painting |
| Nisha | Acharya | singing |
| Nisha | Acharya | painting |
| Ishitha | Mathur | dancing |
| Ishitha | Mathur | drawing |
| Ishitha | Mathur | singing |
| Arjun | Saxena | singing |
| Jagdeesh | Acharya | painting |
| Jagdeesh | Acharya | dancing |
| Arjun | Malhotra | drawing |
| Arjun | Saxena | cooking |
| Pankaj | Acharya | cooking |
| Bhavya | Singh | cooking |
| Jagdeesh | Acharya | cooking |
| Arjun | Malhotra | cooking |
| Nisha | Acharya | cooking |
| Kailash | Parekh | cooking |
+-----+-----+-----+
18 rows in set (0.08 sec)

mysql> select fname,lname from Student where sid IN(select sid from hobbies where hobby="dancing");
+-----+-----+
| fname | lname |
+-----+-----+
| Ishitha | Mathur |
| Jagdeesh | Acharya |
+-----+-----+
2 rows in set (0.03 sec)

mysql> select * from student where sid in(select sid from hobbies where hobby="singing");
+----+-----+-----+-----+
| sid | fname | lname | perc |
+----+-----+-----+-----+
| 102 | Nisha | Acharya | 72 |
| 103 | Ishitha | Mathur | 52 |
| 104 | Arjun | Saxena | 92 |
+----+-----+-----+-----+
3 rows in set (0.00 sec)

```

Fig-16.1: IN operator

```
MySQL 8.0 Command Line Client
2 rows in set (0.03 sec)

mysql> select * from student where sid in(select sid from hobbies where hobby="singing");
+----+-----+-----+-----+
| sid | fname | lname | perc |
+----+-----+-----+-----+
| 102 | Nisha | Acharya | 72 |
| 103 | Ishitha | Mathur | 52 |
| 104 | Arjun | Saxena | 92 |
+----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from student where sid in(select sid from hobbies where hobby not in("dancing","singing","painting"));
+----+-----+-----+-----+
| sid | fname | lname | perc |
+----+-----+-----+-----+
| 101 | Pankaj | Acharya | 89 |
| 102 | Nisha | Acharya | 72 |
| 103 | Ishitha | Mathur | 52 |
| 104 | Arjun | Saxena | 92 |
| 106 | Kailash | Parekh | 90 |
| 107 | Jagdeesh | Acharya | 64 |
| 108 | Bhavya | Singh | 83 |
| 109 | Arjun | Malhotra | 33 |
+----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql>
```

Fig-16.2: NOT IN

17.BETWEEN:

- Select values lying in the given range (inclusive of start and end)
- Values can be text/numbers/dates
- select colname from tablename where colname between lowerbound and upperbound;

```
MySQL 8.0 Command Line Client
+----+-----+-----+-----+
| 108 | Bhavya | Singh | 83 |
| 109 | Arjun | Malhotra | 33 |
+----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select fname,lname from student where perc between 60 and 80;
+-----+-----+
| fname | lname |
+-----+-----+
| Nisha | Acharya |
| Raveena | Saxena |
| Jagdeesh | Acharya |
+-----+-----+
3 rows in set (0.04 sec)

mysql> select * from student;
+----+-----+-----+-----+
| sid | fname | lname | perc |
+----+-----+-----+-----+
| 101 | Pankaj | Acharya | 89 |
| 102 | Nisha | Acharya | 72 |
| 103 | Ishitha | Mathur | 52 |
| 104 | Arjun | Saxena | 92 |
| 105 | Raveena | Saxena | 70 |
| 106 | Kailash | Parekh | 90 |
| 107 | Jagdeesh | Acharya | 64 |
| 108 | Bhavya | Singh | 83 |
| 109 | Arjun | Malhotra | 33 |
| 110 | Annika | NULL | 13 |
| 111 | Guneet | Sikha | 45 |
| 112 | NULL | NULL | 58 |
+----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql>
```

Fig-17.1: BETWEEN

```

MySQL 8.0 Command Line Client
0 rows in set (0.00 sec)

mysql> select fname,lname from student where perc not between 70 and 90;
+-----+-----+
| fname | lname |
+-----+-----+
| Ishitha | Mathur |
| Arjun | Saxena |
| Jagdeesh | Acharya |
| Arjun | Malhotra |
| Annika | NULL |
| Guneet | Sikha |
| NULL | NULL |
+-----+-----+
7 rows in set (0.00 sec)

mysql> select * from student;
+----+-----+-----+-----+
| sid | fname | lname | perc |
+----+-----+-----+-----+
| 101 | Pankaj | Acharya | 89 |
| 102 | Nisha | Acharya | 72 |
| 103 | Ishitha | Mathur | 52 |
| 104 | Arjun | Saxena | 92 |
| 105 | Raveena | Saxena | 70 |
| 106 | Kailash | Parekh | 90 |
| 107 | Jagdeesh | Acharya | 64 |
| 108 | Bhavya | Singh | 83 |
| 109 | Arjun | Malhotra | 33 |
| 110 | Annika | NULL | 13 |
| 111 | Guneet | Sikha | 45 |
| 112 | NULL | NULL | 58 |
+----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql>

```

Fig-17.2: NOT BETWEEN

```

MySQL 8.0 Command Line Client
Query OK, 1 row affected (0.07 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from student;
+----+-----+-----+-----+-----+
| sid | fname | lname | perc | DOB |
+----+-----+-----+-----+-----+
| 101 | Pankaj | Acharya | 89 | 2000-03-09 |
| 102 | Nisha | Acharya | 72 | 2000-05-09 |
| 103 | Ishitha | Mathur | 52 | 2000-08-19 |
| 104 | Arjun | Saxena | 92 | 2000-09-28 |
| 105 | Raveena | Saxena | 70 | 2000-10-08 |
| 106 | Kailash | Parekh | 90 | 2001-02-08 |
| 107 | Jagdeesh | Acharya | 64 | 2001-05-17 |
| 108 | Bhavya | Singh | 83 | 2000-05-04 |
| 109 | Arjun | Malhotra | 33 | 2000-11-30 |
| 110 | Annika | NULL | 13 | 2000-08-31 |
| 111 | Guneet | Sikha | 45 | 2000-01-12 |
| 112 | NULL | NULL | 58 | 2001-07-22 |
+----+-----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql> select fname,lname from student where DOB between "2001-01-01" and "2001-06-10";
+-----+-----+
| fname | lname |
+-----+-----+
| Kailash | Parekh |
| Jagdeesh | Acharya |
+-----+-----+
2 rows in set (0.00 sec)

mysql> select fname,lname from student where DOB between "2001-01-01" and "2001-08-01";
+-----+-----+
| fname | lname |
+-----+-----+
| Kailash | Parekh |
| Jagdeesh | Acharya |
| NULL | NULL |
+-----+-----+
3 rows in set (0.00 sec)

mysql>

```

Fig-17.3: BETWEEN for dates

18.ALIASES:

-AS keyword

-Temporary name for a table, column etc.,

-To make column names more readable,when more than 1 column is involved in a query

-Aliasing for column:

-select colname as COL from tablename where cond;

-Aliasing for table:

-select colname from tablename as TABLENAME where cond;

```
MySQL 8.0 Command Line Client
mysql> select fname as "First Name",lname as "Last Name" from student;
+-----+-----+
| First Name | Last Name |
+-----+-----+
| Pankaj     | Acharya  |
| Nisha     | Acharya  |
| Ishitha    | Mathur   |
| Arjun      | Saxena   |
| Raveena    | Saxena   |
| Kailash    | Parekh   |
| Jagdeesh   | Acharya  |
| Bhavya     | Singh    |
| Arjun      | Malhotra |
| Annika     | NULL     |
| Guneet     | Sikha    |
| NULL      | NULL     |
+-----+-----+
12 rows in set (0.00 sec)

mysql> select s.fname as "First Name",h.hobby as Hobby from student as s natural join hobbies as h;
+-----+-----+
| First Name | Hobby    |
+-----+-----+
| Pankaj     | drawing  |
| Pankaj     | painting |
| Nisha     | singing  |
| Nisha     | painting |
| Ishitha    | dancing  |
| Ishitha    | drawing  |
| Ishitha    | singing  |
| Arjun      | singing  |
| Jagdeesh   | painting |
| Jagdeesh   | dancing  |
| Arjun      | drawing  |
| Arjun      | cooking  |
| Pankaj     | cooking  |
| Bhavya     | cooking  |
| Jagdeesh   | cooking  |
| Arjun      | cooking  |
| Nisha     | cooking  |
| Kailash    | cooking  |
+-----+-----+
18 rows in set (0.12 sec)
```

Fig-18: Aliasing for column names and table names

19.JOIN:

-Combine two tables based on a related column between them

-Types of joins:-1.Inner(Equi join)

2.Left outer join

3.Right outer join

4.Cross join

5.Self join

1.Inner join:

```

MySQL 8.0 Command Line Client
mysql> select s.fname,s.lname,h.hobby from student s inner join hobbies h on s.sid=h.sid;
+-----+-----+-----+
| fname | lname | hobby |
+-----+-----+-----+
| Pankaj | Acharya | drawing |
| Pankaj | Acharya | painting |
| Nisha | Acharya | singing |
| Nisha | Acharya | painting |
| Ishitha | Mathur | dancing |
| Ishitha | Mathur | drawing |
| Ishitha | Mathur | singing |
| Arjun | Saxena | singing |
| Jagdeesh | Acharya | painting |
| Jagdeesh | Acharya | dancing |
| Arjun | Malhotra | drawing |
| Arjun | Saxena | cooking |
| Pankaj | Acharya | cooking |
| Bhavya | Singh | cooking |
| Jagdeesh | Acharya | cooking |
| Arjun | Malhotra | cooking |
| Nisha | Acharya | cooking |
| Kailash | Parekh | cooking |
+-----+-----+-----+
18 rows in set (0.00 sec)

mysql> select s.fname,s.lname,h.hobby from student s natural join hobbies h limit 5;
+-----+-----+-----+
| fname | lname | hobby |
+-----+-----+-----+
| Pankaj | Acharya | drawing |
| Pankaj | Acharya | painting |
| Nisha | Acharya | singing |
| Nisha | Acharya | painting |
| Ishitha | Mathur | dancing |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> _

```

Fig-19.1 : Inner join

-select t1.col1,t2.col2 from tablename1 t1 inner join tablename2 t2 on t1.commonattribute=t2.commonattribute;

2.Left outer join:

-All the entries in left table including NULL will be fetched

3.Right outer join:

-All the entries in the right table including NULL will be fetched

4.Cross join:

-All the possibilities will be returned

```

MySQL 8.0 Command Line Client
mysql> select s.fname,s.lname,c.club from student s right join clubs c on s.sid=c.sid;
+-----+-----+-----+
| fname | lname | club |
+-----+-----+-----+
| Pankaj | Acharya | eco |
| Pankaj | Acharya | coding |
| Nisha | Acharya | dancing |
| Ishitha | Mathur | literature |
| Nisha | Acharya | literature |
| Nisha | Acharya | eco |
| Arjun | Malhotra | NULL |
+-----+-----+-----+
7 rows in set (0.06 sec)

mysql> select s.fname,s.lname,c.club from student s left join clubs c on s.sid=c.sid;
+-----+-----+-----+
| fname | lname | club |
+-----+-----+-----+
| Pankaj | Acharya | eco |
| Pankaj | Acharya | coding |
| Nisha | Acharya | dancing |
| Nisha | Acharya | literature |
| Nisha | Acharya | eco |
| Ishitha | Mathur | literature |
| Arjun | Saxena | NULL |
| Raveena | Saxena | NULL |
| Kailash | Parekh | NULL |
| Jagdeesh | Acharya | NULL |
| Bhavya | Singh | NULL |
| Arjun | Malhotra | NULL |
| Annika | NULL | NULL |
| Guneeet | Sikha | NULL |
| NULL | NULL | NULL |
+-----+-----+-----+
15 rows in set (0.00 sec)

mysql> _

```

Fig-19.2 : Left and right outer join

```

MySQL 8.0 Command Line Client

mysql> select s.fname,s.lname,c.club from student s cross join clubs c limit 15;
+-----+-----+-----+
| fname | lname | club |
+-----+-----+-----+
| Pankaj | Acharya | eco |
| Pankaj | Acharya | coding |
| Pankaj | Acharya | dancing |
| Pankaj | Acharya | literature |
| Pankaj | Acharya | literature |
| Pankaj | Acharya | eco |
| Pankaj | Acharya | NULL |
| Nisha | Acharya | eco |
| Nisha | Acharya | coding |
| Nisha | Acharya | dancing |
| Nisha | Acharya | literature |
| Nisha | Acharya | literature |
| Nisha | Acharya | eco |
| Nisha | Acharya | NULL |
| Ishitha | Mathur | eco |
+-----+-----+-----+
15 rows in set (0.00 sec)

mysql>

```

Fig-19.3 : CROSS join

5.SELF join:

- Two different aliases of the same table
- The table is joined with itself

```

MySQL 8.0 Command Line Client

mysql> select s.fname from student s,student t where s.DOB=t.DOB and s.
+-----+
| fname |
+-----+
| Nisha |
| Pankaj |
| Gunet |
| Annika |
+-----+
4 rows in set (0.00 sec)

mysql> select * from student;
+----+-----+-----+-----+-----+
| sid | fname | lname | perc | DOB |
+----+-----+-----+-----+-----+
| 101 | Pankaj | Acharya | 89 | 2000-03-09 |
| 102 | Nisha | Acharya | 72 | 2000-03-09 |
| 103 | Ishitha | Mathur | 52 | 2000-08-19 |
| 104 | Arjun | Saxena | 92 | 2000-09-28 |
| 105 | Raveena | Saxena | 70 | 2000-10-08 |
| 106 | Kailash | Parekh | 90 | 2001-02-08 |
| 107 | Jagdeesh | Acharya | 64 | 2001-05-17 |
| 108 | Bhavya | Singh | 83 | 2000-05-04 |
| 109 | Arjun | Malhotra | 33 | 2000-11-30 |
| 110 | Annika | NULL | 13 | 2000-01-12 |
| 111 | Gunet | Sikha | 45 | 2000-01-12 |
| 112 | NULL | NULL | 58 | 2001-07-22 |
+----+-----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql>

```

Fig-19.4 : Self join to find the students born on the same date(same DOB)

20.UNION:

- Union of two tables
- Number of columns to be fetched from both the tables should be same

```

us at line 1
mysql> select fname from student union select club from clubs;
+-----+
| fname |
+-----+
| Pankaj |
| Nisha |
| Ishitha |
| Arjun |
| Raveena |
| Kailash |
| Jagdeesh |
| Bhavya |
| Annika |
| Gunet |
| NULL |
| eco |
| coding |
| dancing |
| literature |
+-----+
15 rows in set (0.00 sec)

```

Fig-20.1 : UNION

```
MySQL 8.0 Command Line Client
mysql> select fname from student union all select club from clubs;
+-----+
| fname |
+-----+
| Pankaj |
| Nisha |
| Ishitha |
| Arjun |
| Raveena |
| Kailash |
| Jagdeesh |
| Bhavya |
| Arjun |
| Annika |
| Guneet |
| NULL |
| eco |
| coding |
| dancing |
| literature |
| literature |
| eco |
| NULL |
+-----+
19 rows in set (0.00 sec)

mysql>
```

Fig-20.2 : UNION ALL

21.GROUP BY...HAVING:

- Group rows having same values
- Used with COUNT, AVG, MIN, MAX, SUM (aggregate functions)
- Having clause can be used only with GROUP BY and only with select(not with update,insert,delete like where clause)

```
MySQL 8.0 Command Line Client
mysql> select club,count(sid) from clubs group by club;
+-----+-----+
| club | count(sid) |
+-----+-----+
| eco | 2 |
| coding | 1 |
| dancing | 1 |
| literature | 2 |
| NULL | 1 |
+-----+-----+
5 rows in set (0.00 sec)

mysql> select club,count(sid) from clubs group by club having club IS NOT NULL order by club;
+-----+-----+
| club | count(sid) |
+-----+-----+
| coding | 1 |
| dancing | 1 |
| eco | 2 |
| literature | 2 |
+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

Fig-21 : GROUP BY


```

MySQL 8.0 Command Line Client
mysql> select club,count(sid) from clubs group by club;
+-----+-----+
| club      | count(sid) |
+-----+-----+
| eco       | 2          |
| coding    | 1          |
| dancing   | 1          |
| literature | 2          |
| NULL      | 1          |
+-----+-----+
5 rows in set (0.00 sec)

mysql> select club,count(sid) from clubs group by club having club IS NOT NULL order by club;
+-----+-----+
| club      | count(sid) |
+-----+-----+
| coding    | 1          |
| dancing   | 1          |
| eco       | 2          |
| literature | 2          |
+-----+-----+
4 rows in set (0.00 sec)

mysql> select club,count(sid) from clubs group by club having count(sid)>1;
+-----+-----+
| club      | count(sid) |
+-----+-----+
| eco       | 2          |
| literature | 2          |
+-----+-----+
2 rows in set (0.04 sec)

mysql> _

```

Fig-21.2 : HAVING with count

```

mysql> select club as CLUB,count(sid) as COUNT from clubs group by club having count(sid)>1 order by club desc;
+-----+-----+
| CLUB      | COUNT |
+-----+-----+
| literature | 2      |
| eco       | 2      |
+-----+-----+
2 rows in set (0.00 sec)

mysql> _

```

Fig-21.3: GROUP BY,HAVING and ORDER BY

22.EXISTS:

-Returns TRUE if the sub-query returns atleast one record

```

MySQL 8.0 Command Line Client
mysql> select fname,lname from student where exists(select sid from hobbies where student.sid=hobbies.sid and hobby="cooking");
+-----+-----+
| fname | lname |
+-----+-----+
| Arjun  | Saxena |
| Pankaj | Acharya |
| Bhavya | Singh  |
| Jagdeesh | Acharya |
| Arjun  | Malhotra |
| Nisha  | Acharya |
| Kailash | Parekh |
+-----+-----+
7 rows in set (0.00 sec)

mysql> _

```

Fig-22: EXISTS

23.INSERT INTO SELECT:

-Copies data from one table and inserts into another

*Insert into table2 select * from table1 where cond;*

-Data in table1 wont be removed/affected(ONLY COPIED)

- Source and destination columns should have the same datatypes
- Copy all or some selected columns from table1 and paste them in table2

```

MySQL 8.0 Command Line Client
mysql> desc student2;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| sid   | int  | NO   | PRI | NULL    |       |
| fname | varchar(50) | YES |     | NULL    |       |
| lname | varchar(30) | YES |     | NULL    |       |
| perc  | float | YES  |     | NULL    |       |
| DOB   | date  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> insert into student2 select * from student where perc>=60;
Query OK, 7 rows affected (0.15 sec)
Records: 7 Duplicates: 0 Warnings: 0

mysql> select * from student2;
+-----+-----+-----+-----+-----+
| sid | fname | lname | perc | DOB |
+-----+-----+-----+-----+-----+
| 101 | Pankaj | Acharya | 89 | 2000-03-09 |
| 102 | Nisha | Acharya | 72 | 2000-03-09 |
| 104 | Anjun | Saxena | 92 | 2000-09-28 |
| 105 | Raveena | Saxena | 70 | 2000-10-08 |
| 106 | Kailash | Parekh | 90 | 2001-02-08 |
| 107 | Jagdeesh | Acharya | 64 | 2001-05-17 |
| 108 | Bhavya | Singh | 83 | 2000-05-04 |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql>

```

Fig-23.1 : INSERT INTO SELECT

```

MySQL 8.0 Command Line Client
+-----+-----+-----+-----+-----+
| 101 | Pankaj | Acharya | 89 | 2000-03-09 |
| 102 | Nisha | Acharya | 72 | 2000-03-09 |
| 104 | Anjun | Saxena | 92 | 2000-09-28 |
| 105 | Raveena | Saxena | 70 | 2000-10-08 |
| 106 | Kailash | Parekh | 90 | 2001-02-08 |
| 107 | Jagdeesh | Acharya | 64 | 2001-05-17 |
| 108 | Bhavya | Singh | 83 | 2000-05-04 |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> create table student3(sid int primary key not null,perc float);
Query OK, 0 rows affected (1.73 sec)

mysql> insert into student3 (sid,perc)
-> select sid,perc from student;
Query OK, 12 rows affected (0.18 sec)
Records: 12 Duplicates: 0 Warnings: 0

mysql> select * from student3 limit 4;
+-----+-----+
| sid | perc |
+-----+-----+
| 101 | 89 |
| 102 | 72 |
| 103 | 52 |
| 104 | 92 |
+-----+-----+
4 rows in set (0.00 sec)

mysql>

```

Fig-23.2: INSERT INTO SELECT (selected columns alone)

24.ANY,ALL:

- ANY-returns TRUE even if any of the subqueries is True
- ALL-returns TRUE only if all the subqueries are True

```

MySQL 8.0 Command Line Client
mysql> select * from Student;
+----+-----+-----+-----+-----+
| sid | fname | lname | perc | DOB |
+----+-----+-----+-----+-----+
| 101 | Pankaj | Acharya | 89 | 2000-03-09 |
| 102 | Nisha | Acharya | 72 | 2000-03-09 |
| 103 | Ishitha | Mathur | 52 | 2000-08-19 |
| 104 | Arjun | Saxena | 92 | 2000-09-28 |
| 105 | Raveena | Saxena | 70 | 2000-10-08 |
| 106 | Kailash | Parekh | 90 | 2001-02-08 |
| 107 | Jagdeesh | Acharya | 64 | 2001-05-17 |
| 108 | Bhavya | Singh | 83 | 2000-05-04 |
| 109 | Arjun | Malhotra | 33 | 2000-11-30 |
| 110 | Annika | NULL | 13 | 2000-01-12 |
| 111 | Guneet | Sikha | 45 | 2000-01-12 |
| 112 | NULL | NULL | 58 | 2001-07-22 |
+----+-----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql> select sid,DOB from Student where sid= ANY(select sid from Student where lname="Acharya");
+----+-----+
| sid | DOB |
+----+-----+
| 101 | 2000-03-09 |
| 102 | 2000-03-09 |
| 107 | 2001-05-17 |
+----+-----+
3 rows in set (0.00 sec)

```

Fig-24 : ANY

25.CASE:

-Like switch case with default(ELSE)

CASE

WHEN CONDITION-1 THEN RES-1

WHEN CONDITION-2 THEN RES-2

....

WHEN CONDITION-N THEN RES-N

ELSE DEFAULT-RESULT

END;

-Checks conditions-when a condition is met, corresponding result is displayed and stopped

-If none of the conditions are met,default result is returned

-If ELSE clause is not mentioned,NULL is returned

```

MySQL 8.0 Command Line Client
+----+-----+-----+
| 112 | 58 | Bad |
+----+-----+-----+
12 rows in set (0.00 sec)

mysql> select sid,perc,
-> case
-> when perc>90 then "Excellent"
-> when perc>80 then "Good"
-> when perc>70 then "OK"
-> when perc>60 then "Average"
-> when perc>50 then "Pass"
-> else "Fail"
-> end as remarks
-> from Student;
+----+-----+-----+
| sid | perc | remarks |
+----+-----+-----+
| 101 | 89 | Good |
| 102 | 72 | OK |
| 103 | 52 | Pass |
| 104 | 92 | Excellent |
| 105 | 70 | Average |
| 106 | 90 | Good |
| 107 | 64 | Average |
| 108 | 83 | Good |
| 109 | 33 | Fail |
| 110 | 13 | Fail |
| 111 | 45 | Fail |
| 112 | 58 | Pass |
+----+-----+-----+
12 rows in set (0.00 sec)

mysql>

```

Fig-25.1 : CASE

```

MySQL 8.0 Command Line Client
+----+-----+-----+
| 111 | 45 | Fail |
| 112 | 58 | Pass |
+----+-----+-----+
12 rows in set (0.00 sec)

mysql> select * from clubs;
+----+-----+
| sid | club |
+----+-----+
| 101 | eco |
| 101 | coding |
| 102 | dancing |
| 103 | literature |
| 102 | literature |
| 102 | eco |
| 109 | NULL |
+----+-----+
7 rows in set (0.05 sec)

mysql> select * from clubs order by
-> case
-> when club IS NULL then sid
-> else club
-> end;
+----+-----+
| sid | club |
+----+-----+
| 109 | NULL |
| 101 | coding |
| 102 | dancing |
| 101 | eco |
| 102 | eco |
| 103 | literature |
| 102 | literature |
+----+-----+
7 rows in set (0.00 sec)

mysql>

```

Fig-25.2 : CASE in ORDER BY

26.NULL FUNCTIONS-IFNULL() and COALESCE():

-If col is NULL take a value

IFNULL(colname,value if col is NULL)

-IFNULL() or COALESCE() functions(NULL functions) are used for this.

```

MySQL 8.0 Command Line Client
mysql> select sid,IFNULL(perc,0) from student;
+----+-----+
| sid | IFNULL(perc,0) |
+----+-----+
| 101 | 89 |
| 102 | 72 |
| 103 | 52 |
| 104 | 92 |
| 105 | 70 |
| 106 | 90 |
| 107 | 64 |
| 108 | 83 |
| 109 | 33 |
| 110 | 13 |
| 111 | 45 |
| 112 | 58 |
| 113 | 0 |
+----+-----+
13 rows in set (0.03 sec)

mysql> select sid,COALESCE(perc,0) from student;
+----+-----+
| sid | COALESCE(perc,0) |
+----+-----+
| 101 | 89 |
| 102 | 72 |
| 103 | 52 |
| 104 | 92 |
| 105 | 70 |
| 106 | 90 |
| 107 | 64 |
| 108 | 83 |
| 109 | 33 |
| 110 | 13 |
| 111 | 45 |
| 112 | 58 |
| 113 | 0 |
+----+-----+
13 rows in set (0.00 sec)

mysql>

```

Fig-26 : IFNULL and COALESCE

27.COMMENTS:

-- single-line comments

/*

*/ multi-line comment

Everything inside the comments will not be executed

28.Operators supported by MySQL:

-,+,,*,% Arithmetic

^,&,| Bitwise

AND,OR,NOT,LIKE,EXISTS,ANY,ALL Logical

+=,-=,*=,|=,/= Compound

```
mysql> update Student
    -> set internals=0.4*perc;
Query OK, 12 rows affected (0.11 sec)
Rows matched: 13  Changed: 12  Warnings: 0

mysql> select * from student;
```

sid	fname	lname	perc	DOB	grade	internals
101	Pankaj	Acharya	89	2000-03-09	A+	35.6
102	Nisha	Acharya	72	2000-03-09	A	28.8
103	Ishitha	Mathur	52	2000-08-19	B	20.8
104	Anjun	Saxena	92	2000-09-28	O	36.8
105	Raveena	Saxena	70	2000-10-08	B+	28
106	Kailash	Parekh	90	2001-02-08	A+	36
107	Jagdeesh	Acharya	64	2001-05-17	B+	25.6
108	Bhavya	Singh	83	2000-05-04	A+	33.2
109	Anjun	Malhotra	33	2000-11-30	F	13.2
110	Annika	NULL	13	2000-01-12	F	5.2
111	Guneet	Sikha	45	2000-01-12	F	18
112	NULL	NULL	58	2001-07-22	B	23.2
113	Lara	Dutta	NULL	NULL	F	NULL

```
13 rows in set (0.00 sec)

mysql> _
```

Fig-28 : Arithmetic operator in update

29.Database commands:

-Create a new database:

Create database database-name;

-Delete a database

Drop database database-name;

-Create and drop commands need ADMIN PRIVILEGE

```
MySQL 8.0 Command Line Client

mysql> create database sampledb;
Query OK, 1 row affected (0.15 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| bank      |
| company   |
| employee  |
| employee2 |
| ex        |
| ex1       |
| expt1     |
| expt2     |
| information_schema |
| mall      |
| mysql     |
| performance_schema |
| practice  |
| product   |
| sakila    |
| sampledb  |
| shopping  |
| student   |
| sys       |
| te        |
| te2       |
| world     |
+-----+
22 rows in set (0.10 sec)

mysql> drop database sampledb;
Query OK, 0 rows affected (0.22 sec)

mysql> use sampledb;
ERROR 1049 (42000): Unknown database 'sampledb'

mysql>
```

Fig-29 : Create and drop

30.Tables:

30.1 Create:

-creates a new table in the database

Create table table2 as <required columns from table1> create table2 using table1

```
mysql> create table stu
-> as
-> select sid,grade
-> from student;
Query OK, 13 rows affected (1.54 sec)
Records: 13 Duplicates: 0 Warnings: 0

mysql> select * from stu;
+----+-----+
| sid | grade |
+----+-----+
| 101 | A+    |
| 102 | A     |
| 103 | B     |
| 104 | O     |
| 105 | B+    |
| 106 | A+    |
| 107 | B+    |
| 108 | A+    |
| 109 | F     |
| 110 | F     |
| 111 | F     |
| 112 | 0     |
| 113 | F     |
+----+-----+
13 rows in set (0.00 sec)

mysql>
```

Fig-30.1.1 : Create a new table using another table

```

MySQL 8.0 Command Line Client

105 | B+ |
106 | A+ |
107 | B+ |
108 | A+ |
109 | F |
110 | F |
111 | F |
112 | B |
113 | F |
+-----+
13 rows in set (0.00 sec)

mysql> create table courses(
-> cid int,
-> cname varchar(10));
Query OK, 0 rows affected (0.64 sec)

mysql>

```

Fig-30.1.2 : Create a brand new table

30.2 Drop and Truncate

- DROP-drop a table completely (both table and all the records in the table will be deleted)
- TRUNCATE-Delete the records of a table but not the table. Table still exists in memory but all the data in it is gone

```

MySQL 8.0 Command Line Client

mysql> select * from courses;
+----+-----+
| cid | cname |
+----+-----+
| 1   | DSA   |
| 2   | DS    |
| 3   | AWS   |
| 4   | AI    |
+----+-----+
4 rows in set (0.00 sec)

mysql> truncate table courses;
Query OK, 0 rows affected (1.01 sec)

mysql> desc courses;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| cid   | int  | YES  |     | NULL    |       |
| cname | varchar(10) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.06 sec)

mysql> select * from courses;
Empty set (0.00 sec)

mysql> drop table courses;
Query OK, 0 rows affected (1.20 sec)

mysql> desc courses;
ERROR 1146 (42502): Table 'practice.courses' doesn't exist
mysql>

```

Fig-30.2.1 : DROP and TRUNCATE

30.3 ALTER:

- Add a new column, modify the data-type of an existing column, delete a column
- Delete a column

Alter table tablename drop column colname;

- Modify the datatype of a column

Alter table tablename modify colname destination-data-type;

- Add a new column

Alter table tablename add colname2 datatype2;

```

MySQL 8.0 Command Line Client
mysql> alter table Student
-> drop column internals;
Query OK, 0 rows affected (2.22 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> select * from student;
+----+-----+-----+-----+-----+-----+
| sid | fname | lname | perc | DOB      | grade |
+----+-----+-----+-----+-----+-----+
| 101 | Pankaj | Acharya | 89 | 2000-03-09 | A+ |
| 102 | Nisha | Acharya | 72 | 2000-03-09 | A |
| 103 | Ishitha | Mathur | 52 | 2000-08-19 | B |
| 104 | Arjun | Saxena | 92 | 2000-09-28 | O |
| 105 | Raveena | Saxena | 70 | 2000-10-08 | B+ |
| 106 | Kallash | Parekh | 90 | 2001-02-08 | A+ |
| 107 | Jagdeesh | Acharya | 64 | 2001-05-17 | B+ |
| 108 | Bhavya | Singh | 83 | 2000-05-04 | A+ |
| 109 | Arjun | Malhotra | 33 | 2000-11-30 | F |
| 110 | Annika | NULL | 13 | 2000-01-12 | F |
| 111 | Guneet | Sika | 45 | 2000-01-12 | F |
| 112 | NULL | NULL | 58 | 2001-07-22 | B |
| 113 | Lara | Dutta | NULL | NULL | F |
+----+-----+-----+-----+-----+-----+
13 rows in set (0.00 sec)

mysql>

```

Fig-30.3.1 : Delete a column

```

mysql> alter table student
-> add City varchar(40);
Query OK, 0 rows affected (0.39 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> select * from student limit 5;
+----+-----+-----+-----+-----+-----+-----+
| sid | fname | lname | perc | DOB      | grade | City |
+----+-----+-----+-----+-----+-----+-----+
| 101 | Pankaj | Acharya | 89 | 2000-03-09 | A+ | NULL |
| 102 | Nisha | Acharya | 72 | 2000-03-09 | A | NULL |
| 103 | Ishitha | Mathur | 52 | 2000-08-19 | B | NULL |
| 104 | Arjun | Saxena | 92 | 2000-09-28 | O | NULL |
| 105 | Raveena | Saxena | 70 | 2000-10-08 | B+ | NULL |
+----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.08 sec)

mysql>

```

Fig-30.3.2 : Add a new column

```

mysql> alter table student
-> modify City char;
Query OK, 13 rows affected (1.99 sec)
Records: 13 Duplicates: 0 Warnings: 0

mysql>

```

Fig-30.3.3 : Modify the data-type of a column

31.Constraints:

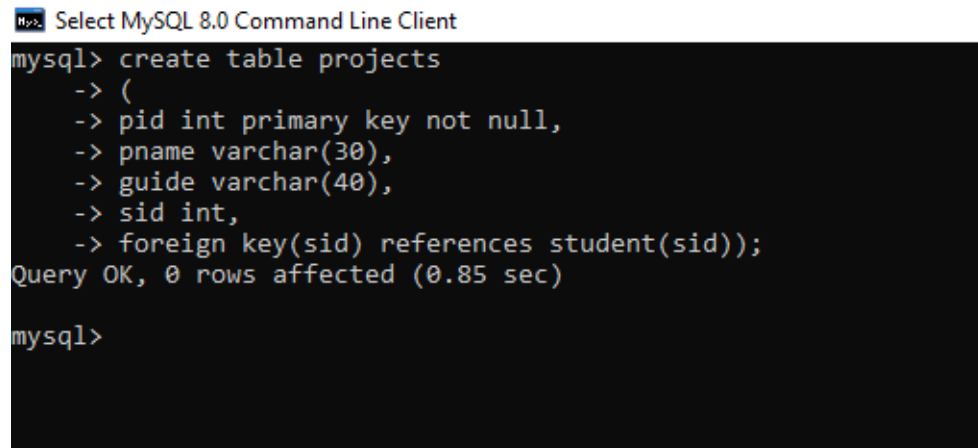
- Primary key
- NOT NULL
- Foreign key
- CHECK-limit the range of values in a column
- DEFAULT-set a default value for a column(this value is used if no value is mentioned)
- CREATE INDEX
- AUTO INCREMENT

-can be used with alter,modify and drop statements also

Alter table tablename drop primary key;

Alter table tablename drop foreign key

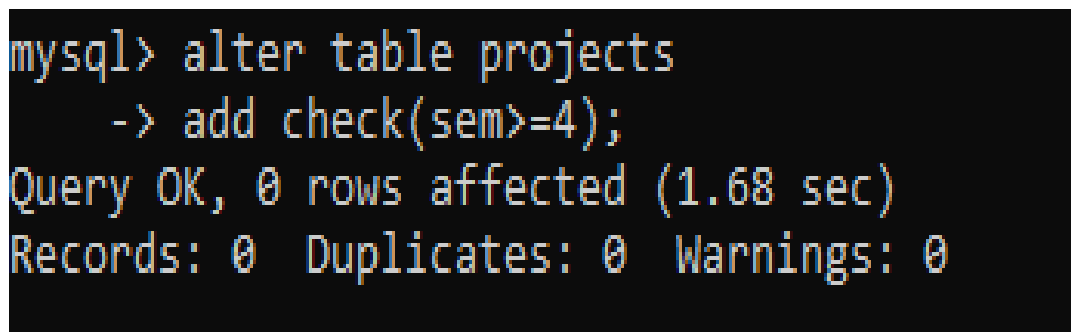
Alter table tablename add foreign key(keyname) references tablename(keyname in the table)



```
Select MySQL 8.0 Command Line Client
mysql> create table projects
-> (
-> pid int primary key not null,
-> pname varchar(30),
-> guide varchar(40),
-> sid int,
-> foreign key(sid) references student(sid));
Query OK, 0 rows affected (0.85 sec)

mysql>
```

Fig-31.1 : Primary key,not null and foreign key



```
mysql> alter table projects
-> add check(sem>=4);
Query OK, 0 rows affected (1.68 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Fig-31.2 : CHECK constraint after adding sem column

Add constraint constraint-name check(sem>=4) to name a constraint

Alter table table-name drop constraint constraint-name

Naming a constraint will be useful while dropping it.

CURRENT_DATE()->gives today's date

```

MySQL 8.0 Command Line Client
mysql> alter table clubs
-> alter loc set default "B";
Query OK, 0 rows affected (1.28 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> insert into clubs(sid,club) values(106,"coding");
Query OK, 1 row affected (0.15 sec)

mysql> select * from clubs;
+-----+-----+-----+
| sid | club      | loc |
+-----+-----+-----+
| 101 | eco       | C   |
| 101 | coding    | C   |
| 102 | dancing   | C   |
| 103 | literature | C   |
| 102 | literature | C   |
| 102 | eco       | C   |
| 109 | NULL      | C   |
| 104 | coding    | C   |
| 106 | coding    | B   |
+-----+-----+-----+
9 rows in set (0.00 sec)

mysql>

```

Fig-31.3 : DEFAULT Constraint

```

MySQL 8.0 Command Line Client
mysql> alter table clubs
-> alter loc drop default;
Query OK, 0 rows affected (0.18 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> insert into clubs(sid,club) values(111,"dancing");
ERROR 1364 (HY000): Field 'loc' doesn't have a default value
mysql>

```

Fig-31.4 : Drop DEFAULT. After dropping default constraint, the column has no default value NOT EVEN NULL

32.CREATE INDEX:

-Create index for a column/a set of columns of a table

-Create UNIQUE index does not allow duplicates

Create index index_name on table_name(col1,col2,col3...,coln);

Alter table tablename drop index indexname;

-Easier to fetch data using index

33. AUTO INCREMENT:

-auto_increment-automatically increase value in col by 1 everytime a new record is inserted

Alter table tablename auto_increment=100;

Auto increment starts from 100 and goes like 101,102,103 etc.,->default start value=1.Here,default start value will be 100.

```
MySQL 8.0 Command Line Client
mysql> create table contributions(
    -> no int primary key not null auto_increment,
    -> money float);
Query OK, 0 rows affected (0.43 sec)

mysql> insert into contributions(money) values(300);
Query OK, 1 row affected (0.12 sec)

mysql> insert into contributions(money) values(400);
Query OK, 1 row affected (0.14 sec)

mysql> insert into contributions(money) values(450);
Query OK, 1 row affected (0.05 sec)

mysql> insert into contributions(money) values(850);
Query OK, 1 row affected (0.06 sec)

mysql> insert into contributions(money) values(1350);
Query OK, 1 row affected (0.09 sec)

mysql> select * from contributions;
+----+-----+
| no | money |
+----+-----+
| 1  | 300   |
| 2  | 400   |
| 3  | 450   |
| 4  | 850   |
| 5  | 1350  |
+----+-----+
5 rows in set (0.00 sec)
```

Fig-33: AUTO_INCREMENT

34.DATE data types:

-DATE YYYY-MM-DD

-DATETIME YYYY-MM-DD HH:MM: SS

-TIMESTAMP YYYY-MM-DD HH:MM: SS

-YEAR YYYY or YY

-When a new column is created in a table,date datatype is set it.

35.VIEWS:

- In SQL, a view is a virtual table based on the result-set of an SQL statement.

-Always shows up-to-date data

```
MySQL 8.0 Command Line Client
mysql> create view toppers as select sid,fname,lname from student where perc>80;
Query OK, 0 rows affected (0.14 sec)

mysql> select * from toppers;
+-----+-----+-----+
| sid | fname | lname |
+-----+-----+-----+
| 101 | Pankaj | Acharya |
| 104 | Arjun | Saxena |
| 106 | Kailash | Parekh |
| 108 | Bhavya | Singh |
+-----+-----+-----+
4 rows in set (0.09 sec)

mysql> create or replace view toppers as select sid,fname,lname from student where perc>70;
Query OK, 0 rows affected (0.12 sec)

mysql> select * from toppers;
+-----+-----+-----+
| sid | fname | lname |
+-----+-----+-----+
| 101 | Pankaj | Acharya |
| 102 | Nisha | Acharya |
| 104 | Arjun | Saxena |
| 106 | Kailash | Parekh |
| 108 | Bhavya | Singh |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> drop view toppers;
Query OK, 0 rows affected (0.15 sec)
```

Fig-35 : View