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

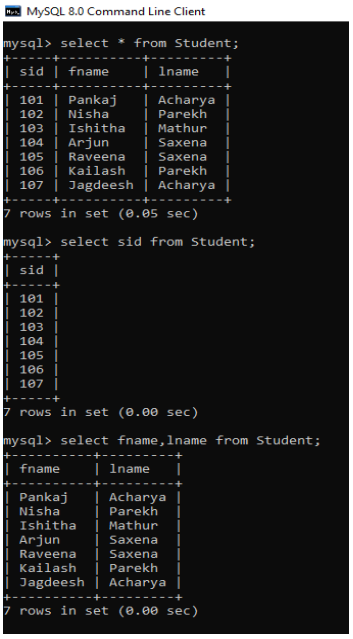

Fig-2: Display Student table with three columns-sid,fname and lname

3.DISTINCT:

-selects only distinct values of the given column

-does not select duplicate values

-select distinct colname from tablename

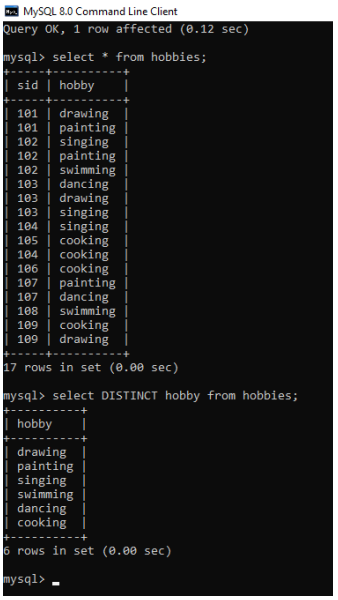

Fig-3.1: Display unique hobbies of students

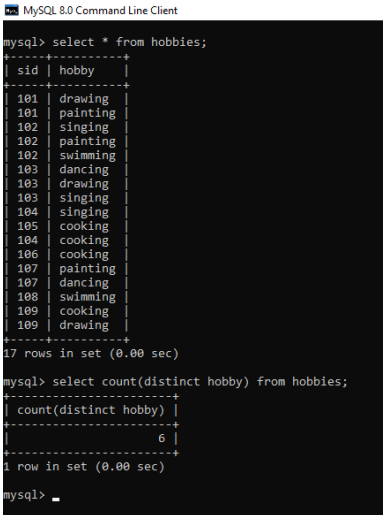

Fig-3.2: Count the number of unique records using count (distinct colname)

4.WHERE:

-select only the records which satisfy the required condition

-select col1 from tablename where condtion

-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

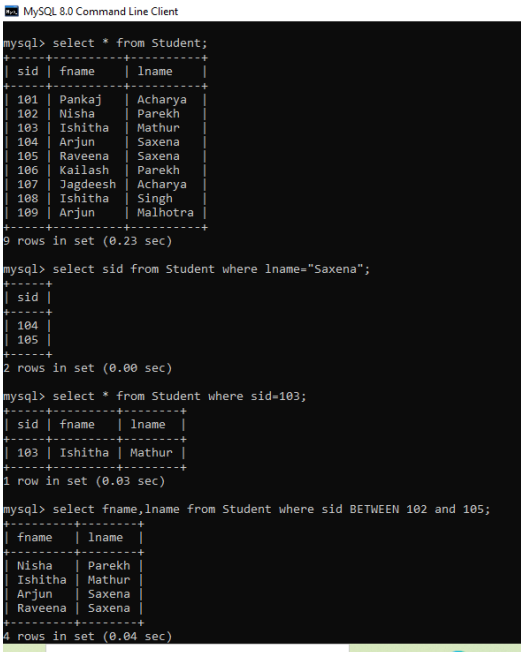

Fig-4.1: where clause with strings and numbers in the condition

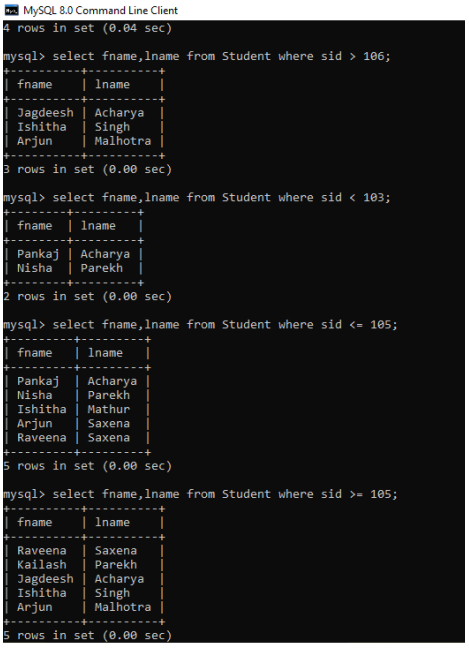

Fig-4.2: where clause with BETWEEN…AND and a few other operators

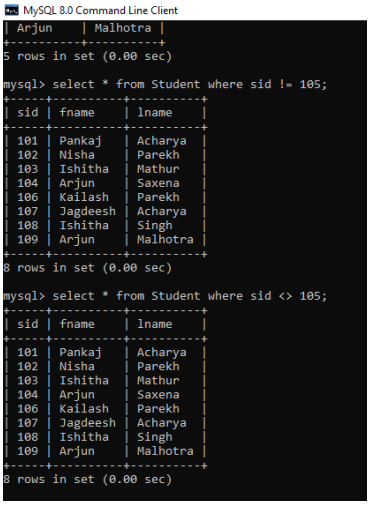

Fig-4.3: where clause with a few operators

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

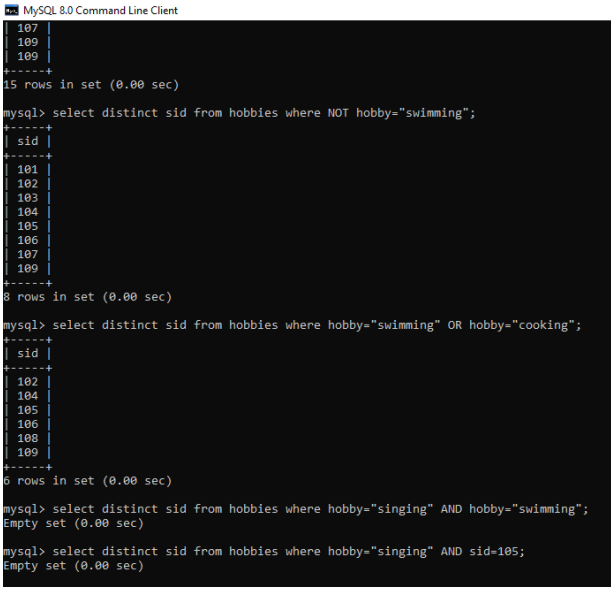

Fig-5: AND OR NOT

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

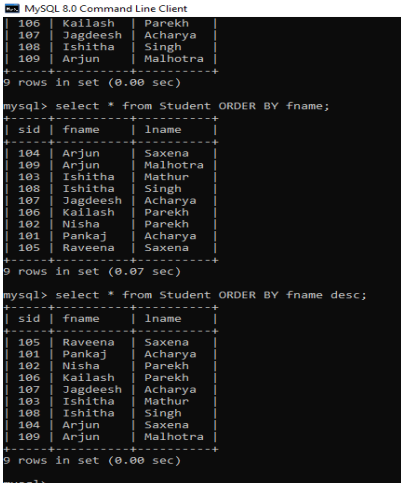

Fig-6.1: ORDER BY

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;

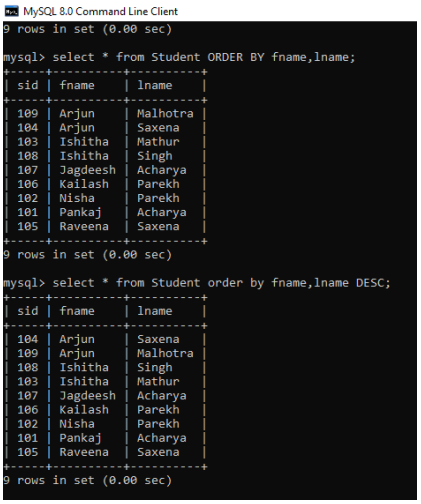

Fig-6.2: ORDER BY two columns. If entries in 1st column are same, then results are sorted based on the entries in the 2nd column, as mentioned in ascending or descending order

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>);

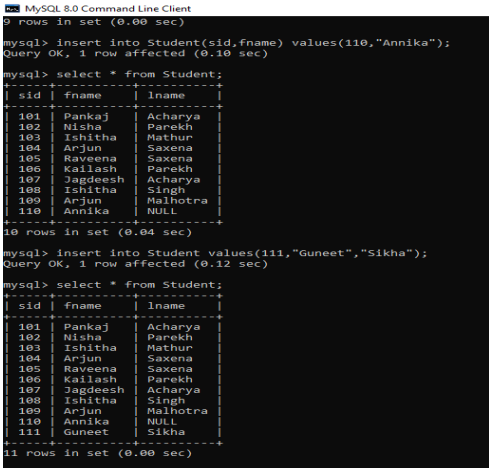

Fig-7: INSERT INTO STATEMENT

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.

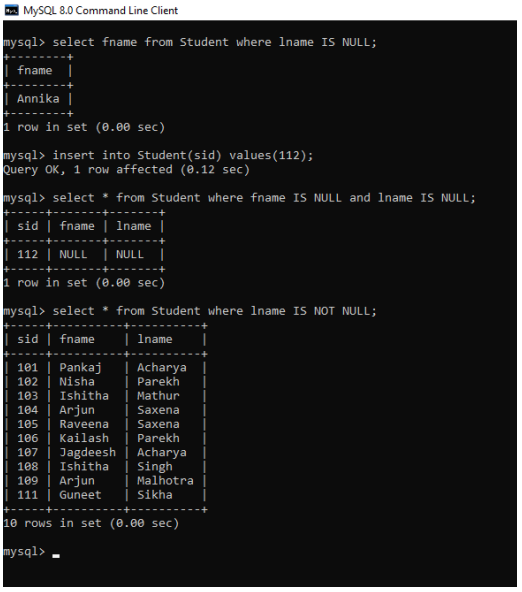

Fig-8 : IS NULL and IS NOT NULL operators

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!

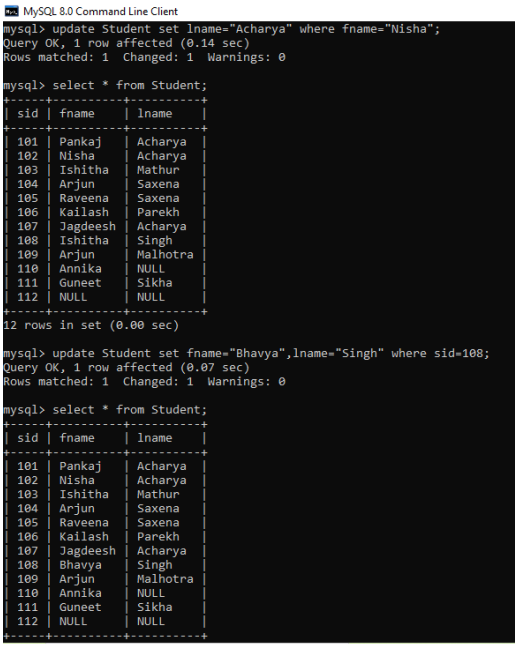

Fig-9 : UPDATE

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

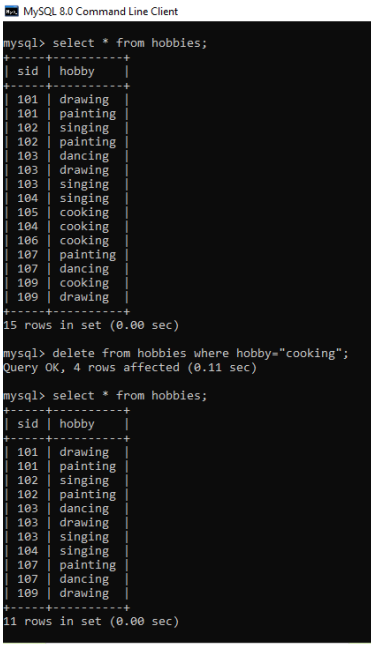

Fig-10 : DELETE

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

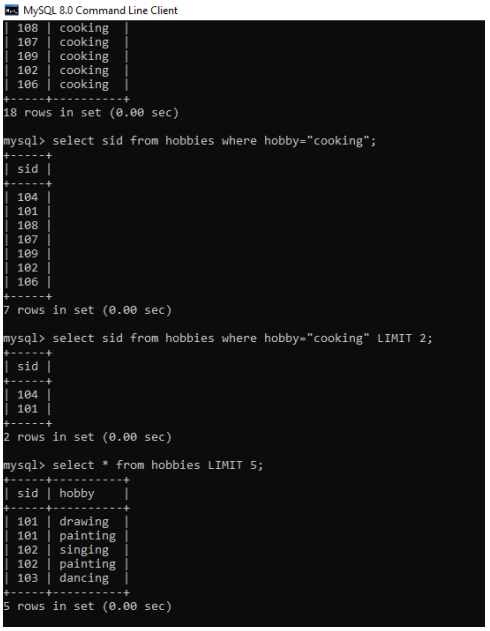

Fig-11 : LIMIT clause

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;*

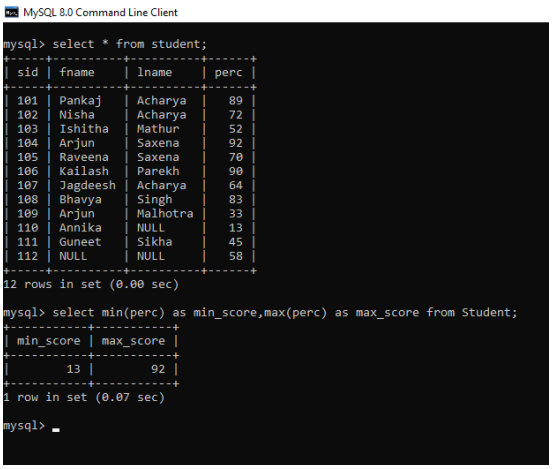

Fig-12: MIN () and MAX ()

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;

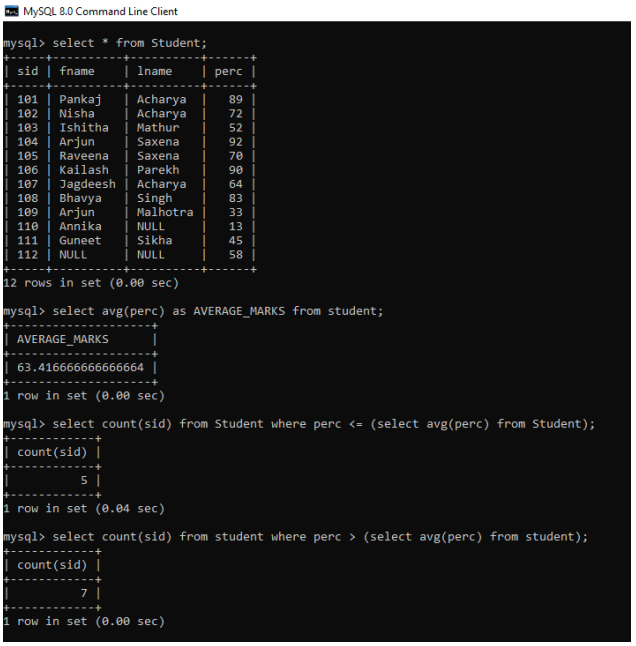

Fig-13.1: COUNT (), AVG ()

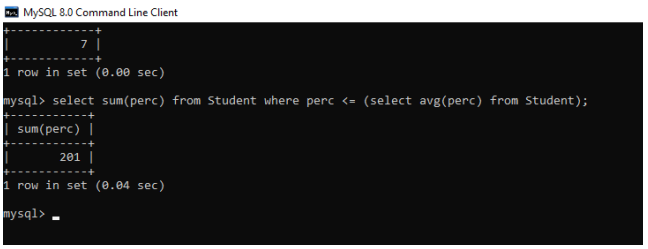

Fig-13.2: SUM ()

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

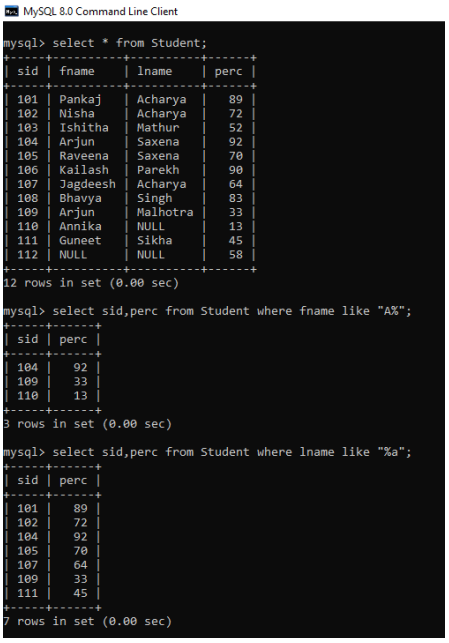

Fig-14: LIKE

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;

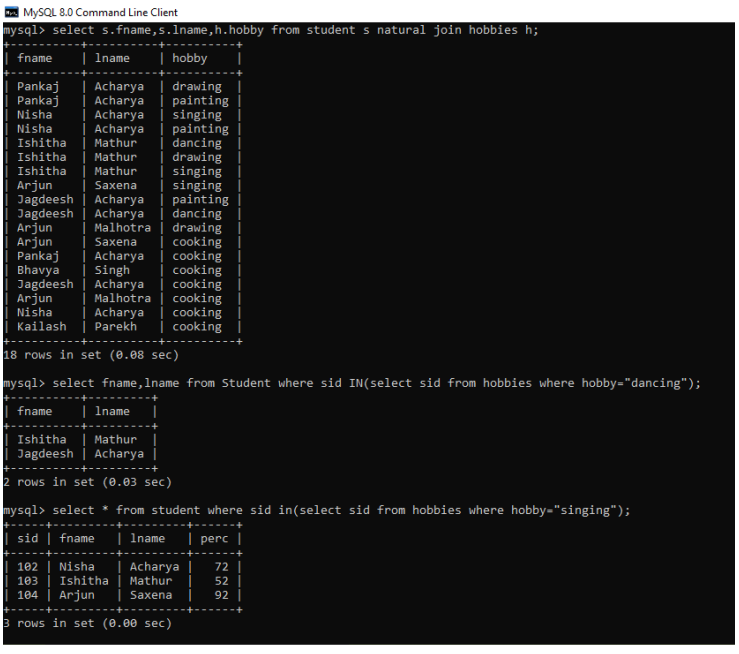

Fig-16.1: IN operator

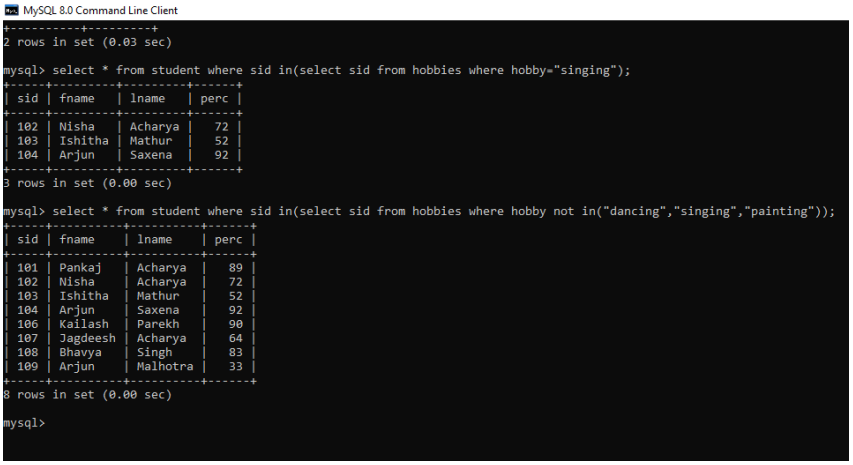

Fig-16.2: NOT IN

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;

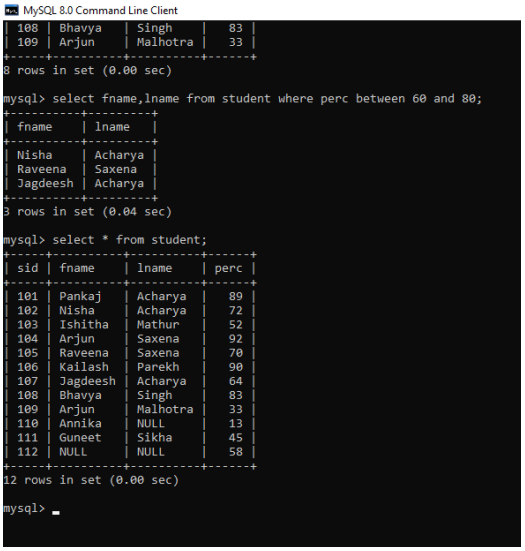

Fig-17.1: BETWEEN

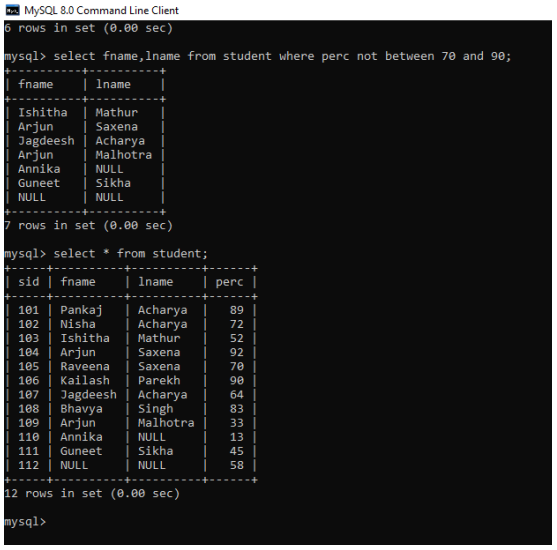

Fig-17.2: NOT BETWEEN

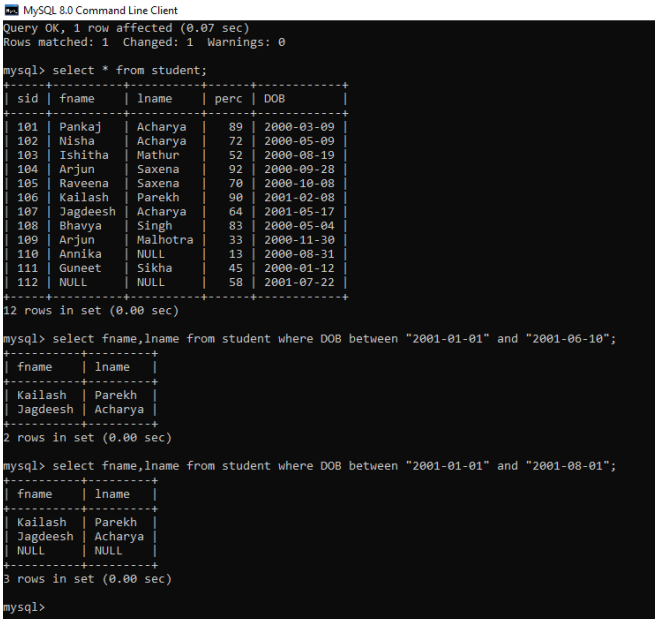

Fig-17.3: BETWEEN for dates

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;

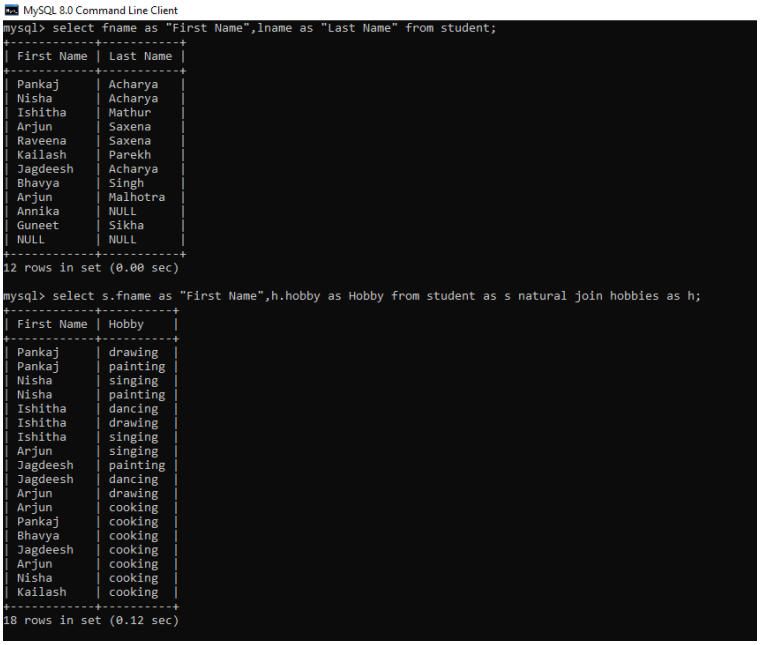

Fig-18: Aliasing for column names and table names

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:

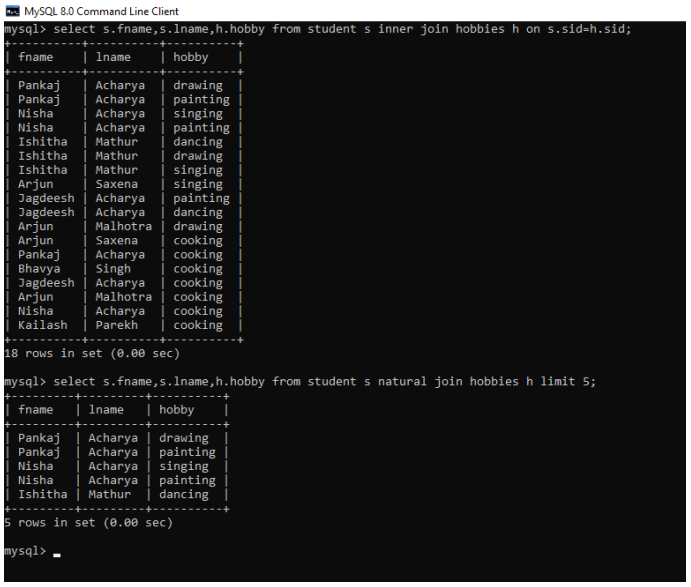

Fig-19.1 : Inner join

-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

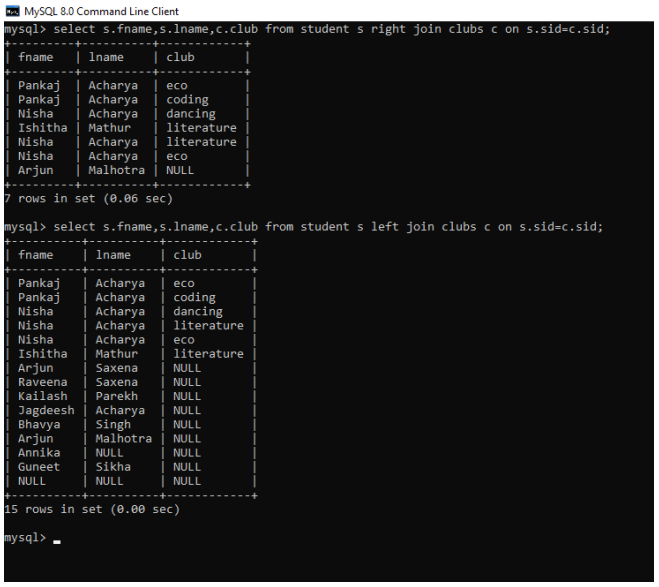

Fig-19.2 : Left and right outer join

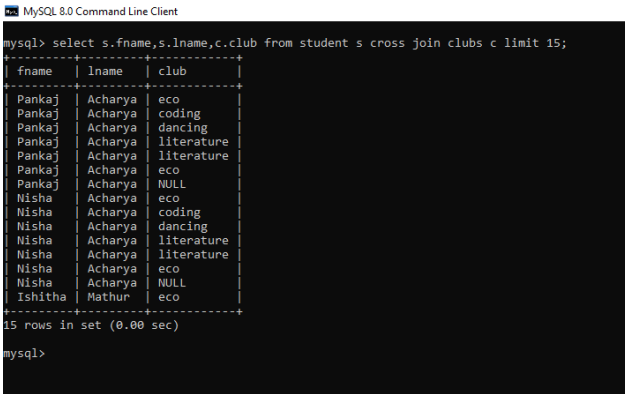

Fig-19.3 : CROSS join

5.SELF join:

-Two different aliases of the same table

-The table is joined with itself

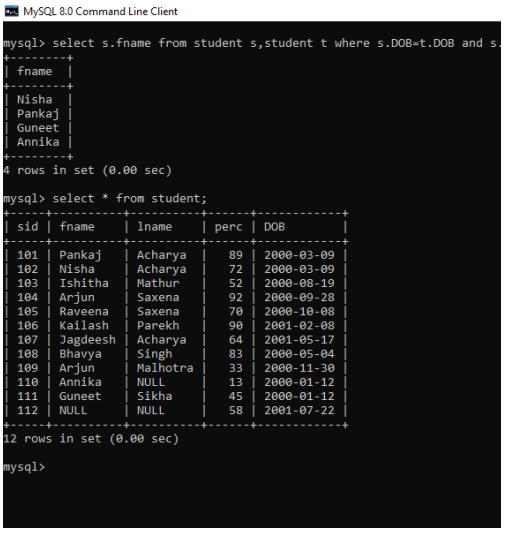

Fig-19.4 : Self join to find the students born on the same date(same DOB)

<u>20.UNION:</u>

-Union of two tables

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

Fig-20.1 : UNION

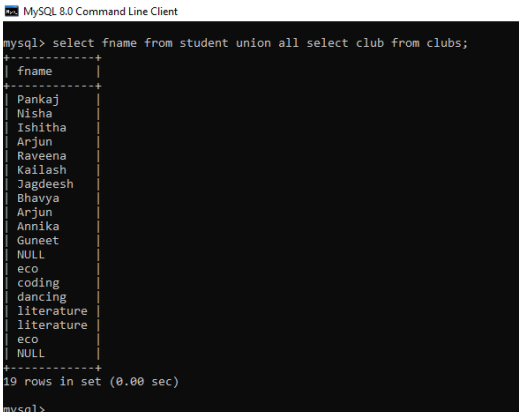

Fig-20.2 : UNION ALL

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)

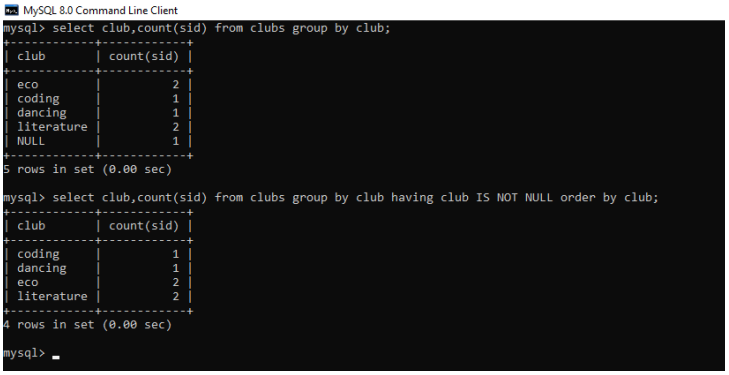

Fig-21 : GROUP BY

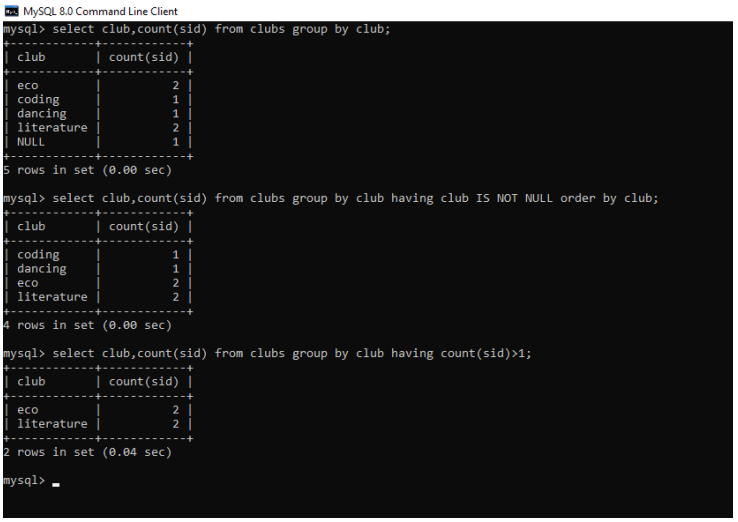

Fig-21.2 : HAVING with count

Fig-21.3: GROUP BY,HAVING and ORDER BY

22.EXISTS:

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

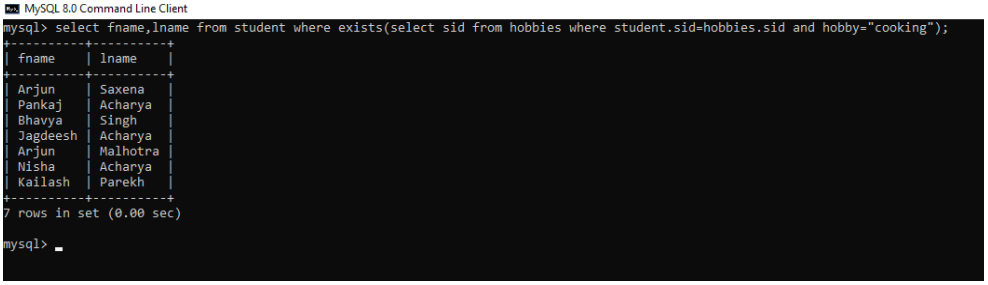

Fig-22: EXISTS

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

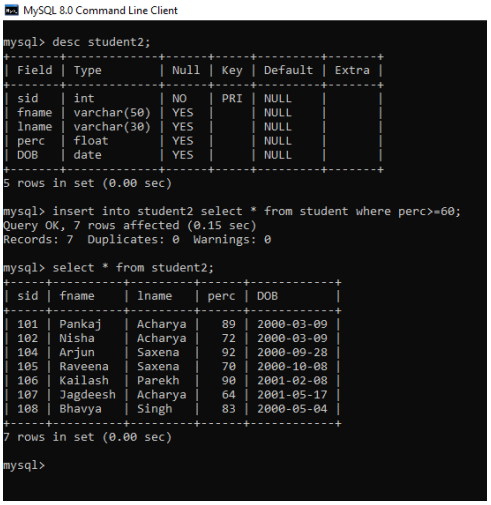

Fig-23.1 : INSERT INTO SELECT

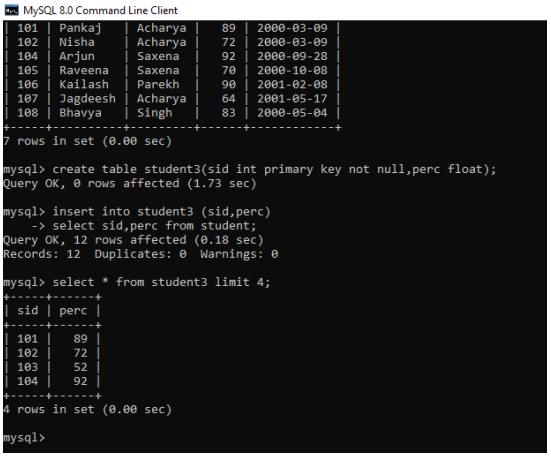

Fig-23.2: INSERT INTO SELECT (selected columns alone)

24.ANY,ALL:

-ANY-returns TRUE even if any of the subqueries is True

-ALL-returns TRUE only if all the subqueries are True

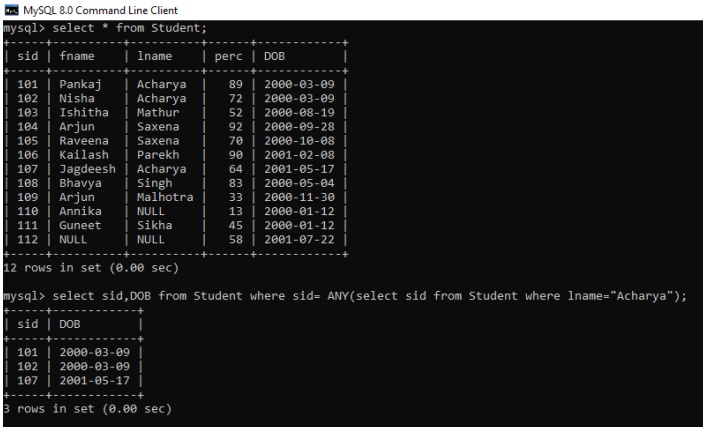

Fig-24 : ANY

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

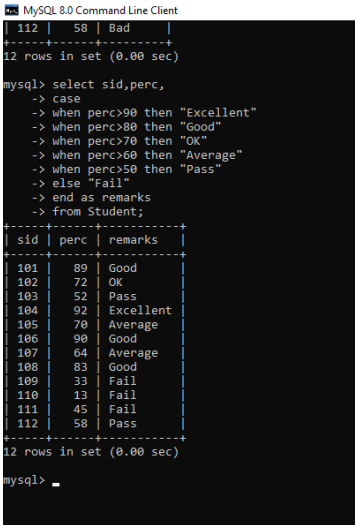

Fig-25.1 : CASE

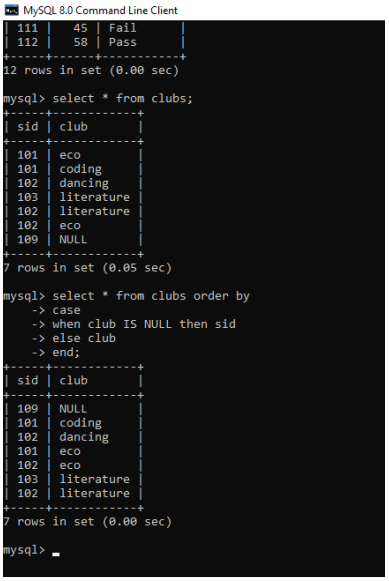

Fig-25.2 : CASE in ORDER BY

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.

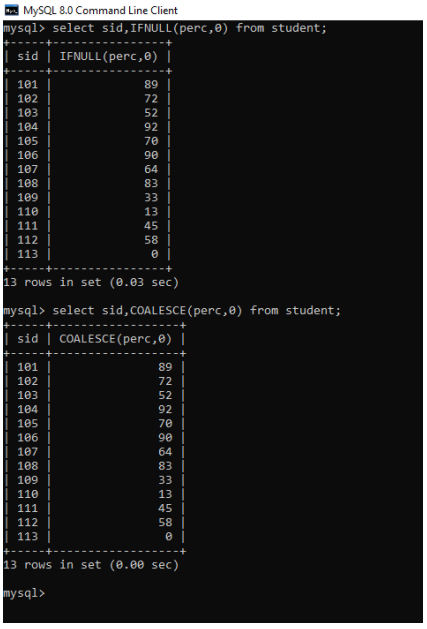

Fig-26 : IFNULL and COALESCE

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 | Arjun    | 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 | Arjun    | 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

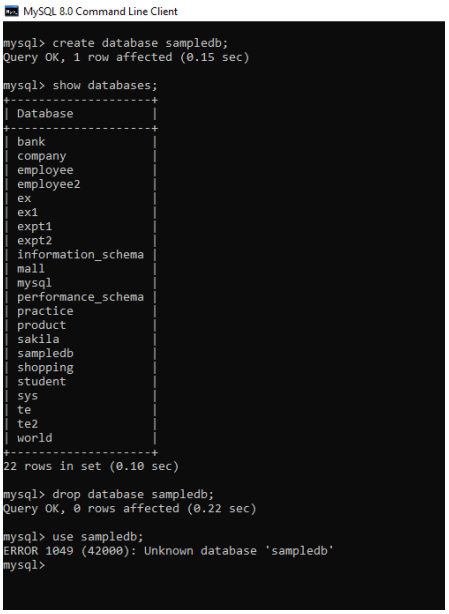

Fig-29 : Create and drop

30.Tables:

30.1 Create:

-creates a new table in the database

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

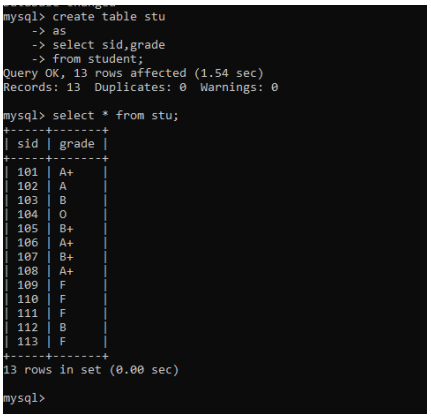

Fig-30.1.1 : Create a new table using another table

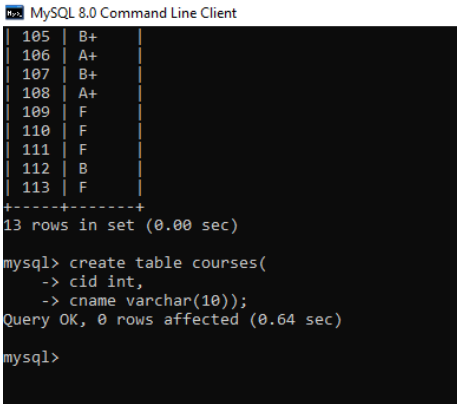

Fig-30.1.2 : Create a brand new table

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

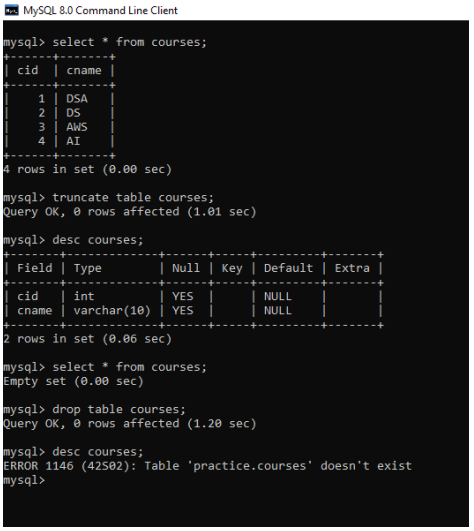

Fig-30.2.1 : DROP and TRUNCATE

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;

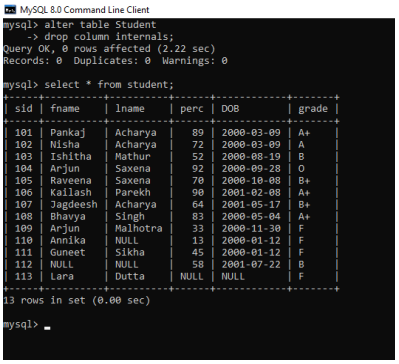

Fig-30.3.1 : Delete a column

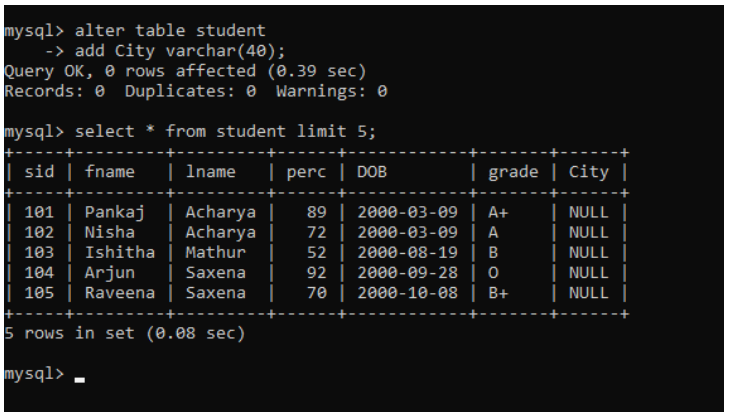

Fig-30.3.2 : Add a new column

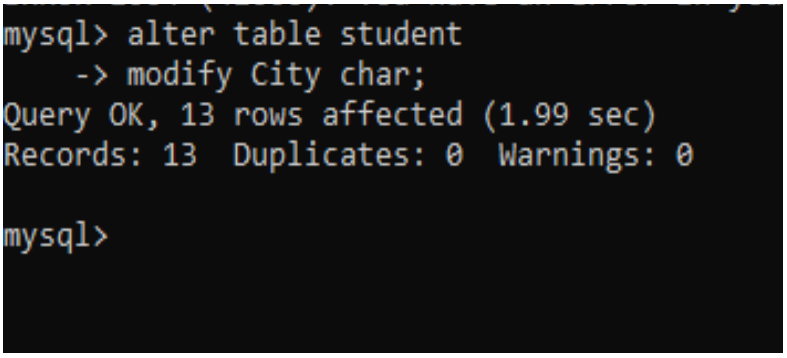

Fig-30.3.3 : Modify the data-type of a column

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)

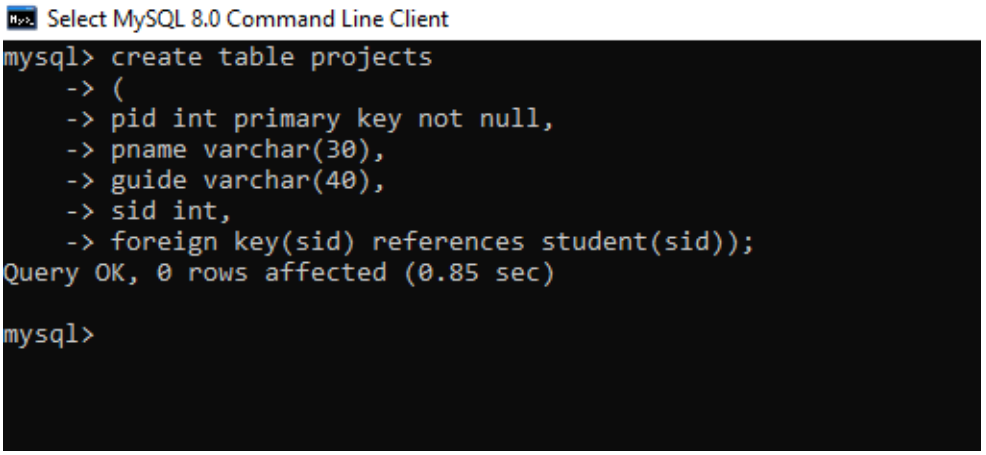

Fig-31.1 : Primary key,not null and foreign key

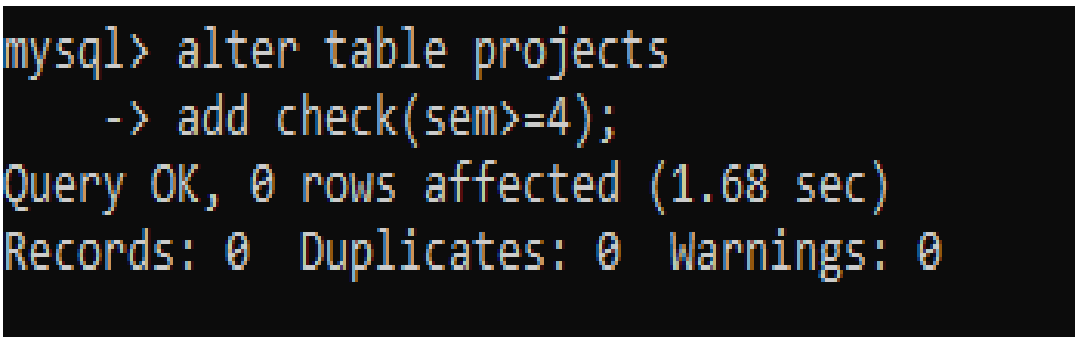

Fig-31.2 : CHECK constraint after adding sem column

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

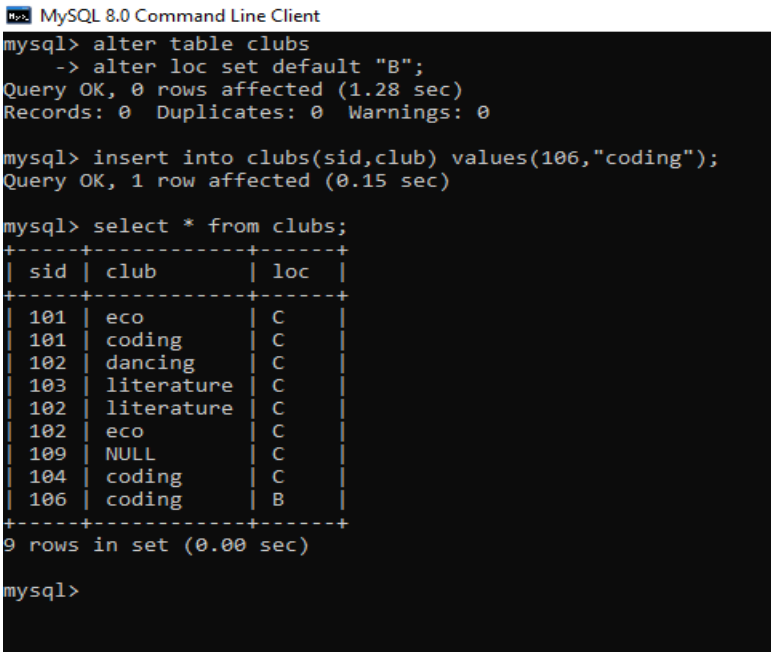

Fig-31.3 : DEFAULT Constraint

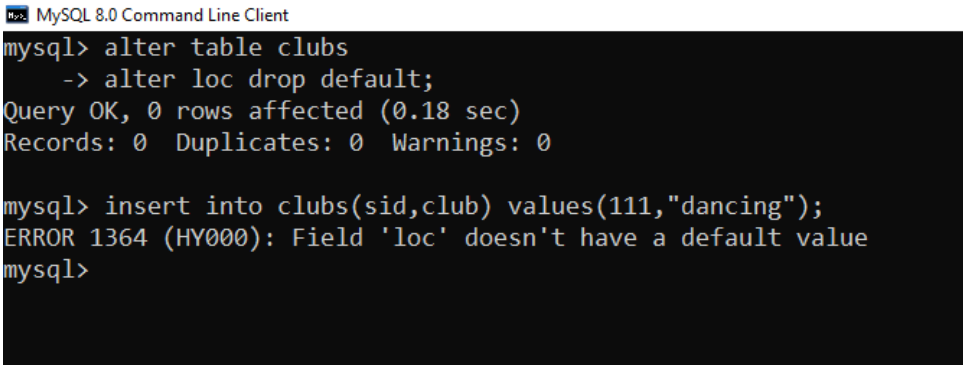

Fig-31.4 : Drop DEFAULT.After dropping default constraint,the column has no default value NOT EVEN NULL

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.

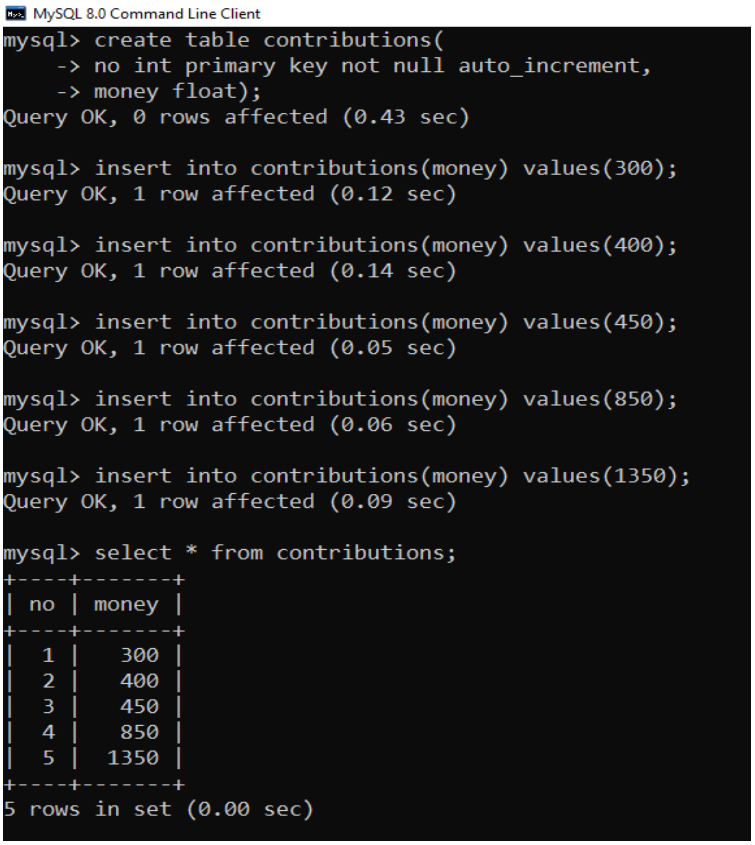

Fig-33: AUTO_INCREMENT

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

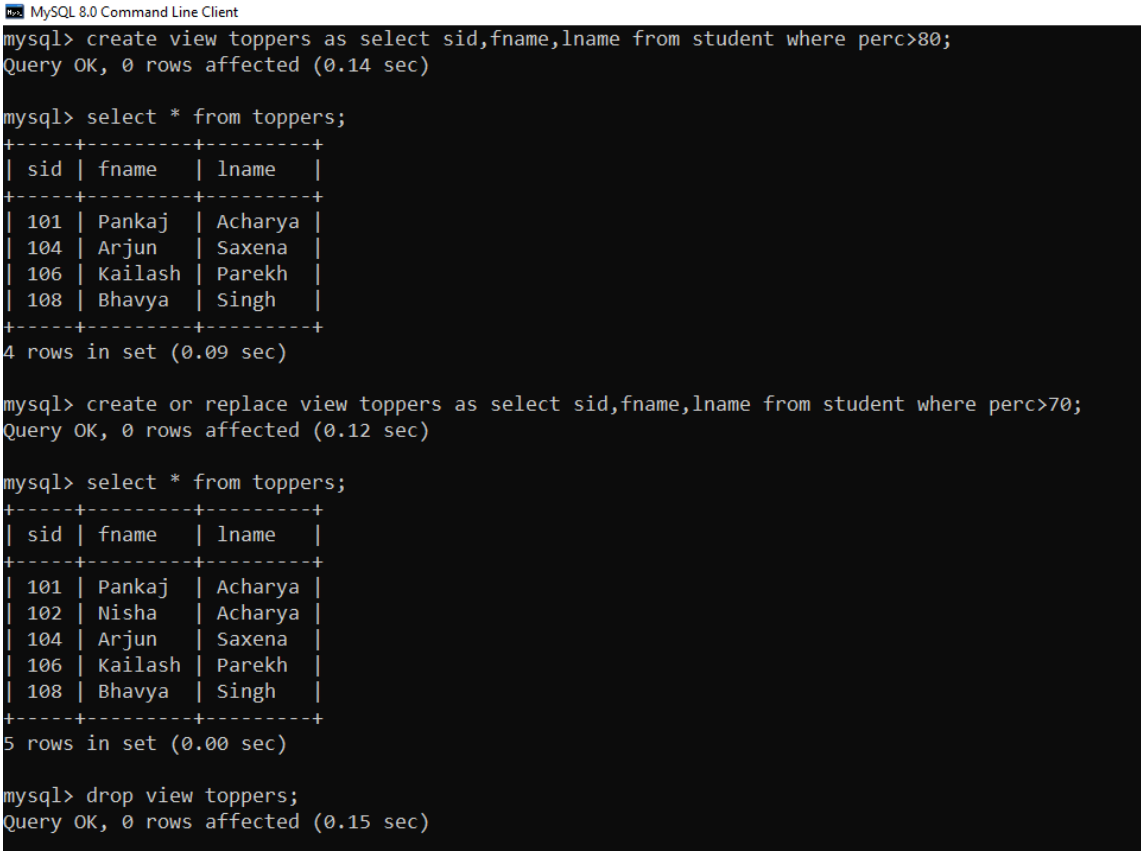

Fig-35 : View