

① Creating database : Creates a database.

General Syntax :

`create database <dbname>;`

Code :

`create database cse2302040022;`

② Showing databases: It shows the existing and newly created databases.

General Syntax & Executed Code :

`show databases;`

③ Using database : It allows to use/enter into a database.

General syntax :

`use <database name>;`

Executed Code :

`use cse2302040022;`

④ Showing tables: It shows the available tables in the database.

General syntax & Executed Code :

`show tables;`

⑤ Creating table : It allows to create a table with column names

General Syntax :

`create table <table name> (<column1> <datatype>, <column2> <datatype>, <column3> <datatype>);`

OUTPUT

① Query OK, 1 row affected

②

Database
cse2302040022
information-schema
mysql
performance-schema
sys

5 rows in set

③ Database changed

④ Empty set

⑤ Query OK, 0 rows affected.

Executed code:

```
create table studentdb (Name varchar(30), RollNo  
varchar(10), Gender varchar(10), Sem varchar(5), Branch  
varchar(10), EmailId varchar(20), City varchar(20), CGPA  
decimal (4,2));
```

⑥ Describing table: It describes the information about table.
General Syntax:

```
describe <table name>; / desc <table name>;
```

Executed Code:

```
describe studentdb;
```

⑦ Inserting rows: To insert rows into the table.

General syntax:

```
insert into <tablename> values (value1, "value2", ...);
```

Executed code:

```
insert into studentdb values ("Tribhuvan", "2302040022", "Male", "3rd",  
"CSE", "dabulu123@gmail.com", "Angeel", 9.74);
```

→ Like this other 8 rows are inserted.

⑧ Selecting ^{all} data: It allows to select data from the table and shows all the column data.

General syntax:

```
select * from <table name>;
```

Executed code:

```
select * from studentdb;
```


⑥

Field	Type	Null	Key	Default	Extra
Name	varchar(30)	YES		NULL	
RollNo	varchar(10) varchar(10)	YES		NULL	
Genders	varchar(10)	YES		NULL	
Sem	varchar(5)	YES		NULL	
Branch	varchar(10)	YES		NULL	
EmailId	varchar(20)	YES		NULL	
City	varchar(20)	YES		NULL	
CGPA	double(4,2)	YES		NULL	

8 rows in set

⑦ Query OK, 1 row affected

⑧

Name	Roll No	Genders	Sem	Branch	EmailId	City	CGPA
Dabulu	2302040022	Male	3rd	CSE	dabulu123@gmail.com	Angul	9.74
Rupesh	2302040030	Male	3rd	CSE	rupesh123@gmail.com	Chakrabharpur	7.88
Trisul	2302040002	Male	3rd	CSE	trisul123@gmail.com	Boskampurs	10.00
Anurag	2302040036	Male	3rd	CSE	anurag123@gmail.com	Sambalpur	9.85
Bikram	2302041046	Male	3rd	CSE	bikram123@gmail.com	Keonjhar	9.89
Ansuman	2302040019	Male	3rd	CSE	ansuman123@gmail.com	Rourkela	9.97
Unknown	2302040100	Female	3rd	CSE	unknown123@gmail.com	Bhubaneswar	6.45
Girl	2302040099	Female	3rd	CSE	girl123@gmail.com	Delhi	8.39

⑨ Selecting specific column : Allows to select column manually.

General Syntax :

select <column1>, <column2> from <tablename>;

Executed Code :

select name from studentdb;

⑩ Selecting with condition : Acts like filter to access rows

General Syntax :

select * from <tablename> where <columnname> <operator> value;

Executed Syntax :

select * from studentdb where CGPA = 10.00;

⑪

Executed Syntax :

select * from studentdb where CGPA < 7.00;

⑫ Executed Syntax :

select * from studentdb where CGPA > 10.00;

9

Name
Dabulu
Rupesh
Toisul
Anusag
Bikram
Anuman
Unknown
Girl

9 rows in set

10

Name	RollNo	Gender	Sem	Branch	EmailId	City	CGPA
Toisul	2302040002	Male	3rd	CSE	toisul123@gmail.com	Besbampus	10.00

1 row in set

11

Name	RollNo	Gender	Sem	Branch	EmailId	City	CGPA
Unknown	2302040100	Female	3rd	CSE	unknown123@gmail.com	Bhubaneswar	6.45

1 row in set

12 Empty set

⑬ Executed Syntax:

select Name from studenttbl where CGPA >= 9.89;

⑭ Executed Syntax:

select RollNo from studenttbl where CGPA <= 7.88;

⑮ Executed Syntax:

select Name from studenttbl where Gender <> "Male";

⑯ Executed syntax:

select Name from studenttbl where City in ("Anup",
"Beshampur");

OUTPUT

(13)

Name
Toisul
Bikram
Ansuman

3 rows in set

(14)

RollNo /
2302040030
2302004100

2 rows in set

(15)

Name
Unknown
Girl

2 rows in set

(16)

Name
Dabulu
Toisul

2 rows in set

(17) Executed syntax code:

```
select RollNo from studentdb where rnt (GPA > 7.00);
```

(18) Executed code:

```
select Name from studentdb whose CGPA between  
8.00 and 9.00;
```

(19) Executed code:

```
select Name from studentdb whose CGPA > 7.00 and  
CGPA < 8.00;
```

(20) Executed code:

```
select RollNo from studentdb whose CGPA = 10 or CGPA > 10;
```

(21) Executed Code:

```
select * from studentdb whose Name like 'A%';
```

OUTPUT

⑦

RollNo
230204000

1 row in set

⑧

Name
Girl

1 row in set

⑨

Name
Rupesh

1 row in set

⑩

Name
Trisul

1 row in set

⑪

Name	RollNo	Gender	Sem	Branch	EmailId	City	CGPA
Anusag	230204006	Male	3rd	CSE	anusag123@gmail.com	Sambalpur	9.85
Ansuman	230204007	Male	3rd	CSE	ansuman123@gmail.com	Raivakela	9.99

2 rows in set

(22) Executed Code using like and _ and % to filter :-

```
select Name where Name like ('_x%');
```

The above syntax filters & shows the Name column where the second letter is 'x'.

(23) Executed Code :

```
select Name where Name like '%a-';
```

Above code shows names with having second letter as 'a'.

(24) Executed Code :

```
select * where City like '-----';
```

Above code shows all columns, with having City length 8 and filters the rows

(25) like operators use with decimal datatype :-

Executed code :

```
select Name where CGPA like '(9%);
```

Above code shows names having first digit 9 in CGPA

(22) Executed Code using Like and _ and % to filter :-

```
select Name where Name like '_x%';
```

The above syntax filters & shows the Name column where the second letter is 'x'.

(23) Executed Code :

```
select Name where Name like '%a_';
```

Above code shows names with having second letter as 'a'.

(24) Executed Code :

```
select * where City like '-----';
```

Above code shows all columns, with having City length 8 and filters the rows

(25) Like operator use with decimal datatype :-

Executed code :

```
select Name * where CGPA like '9.9%';
```

Above code shows names having first digit 9 in CGPA