```
 use dbmslab;
Database changed
mysql> show tables;
+-------------------+
| Tables_in_dbmslab |
+-------------------+
| author            |
| bank_account      |
| bank_account1     |
| book              |
| cus               |
| customers         |
| orders            |
| products          |
| stu               |
| stud              |
| studdetails       |
| student           |
| users             |
+-------------------+
13 rows in set (0.02 sec)

mysql> select * from student
    -> ;
+---------+-------+------+------+
| name    | marks | dept | id   |
+---------+-------+------+------+
| kelavii |    99 | cse  |    1 |
| roja    |    95 | cse  |    2 |
| saras   |    94 | ece  |    3 |
+---------+-------+------+------+
3 rows in set (0.00 sec)

mysql> create trigger marks_update_trigger
    -> before insert
    -> on student
    -> for each row
    -> set new.marks = new.marks+6;
Query OK, 0 rows affected (0.03 sec)

mysql> insert into student values ('raghul',94,'ece',4);
Query OK, 1 row affected (0.02 sec)
```

```
mysql> select * from student;
+---------+-------+------+------+
| name    | marks | dept | id   |
+---------+-------+------+------+
| kelavii |    99 | cse  |    1 |
| roja    |    95 | cse  |    2 |
| saras   |    94 | ece  |    3 |
| raghul  |   100 | ece  |    4 |
+---------+-------+------+------+
4 rows in set (0.00 sec)

 create table final_marks (tot_marks int);
Query OK, 0 rows affected (0.03 sec)

mysql> create trigger final_marks
    -> after insert
    -> on student
    -> for each row
    -> insert into final_marks values(new.marks);
Query OK, 0 rows affected (0.01 sec)

mysql> insert into student values('raja',100,'mett',5);
Query OK, 1 row affected (0.01 sec)

mysql> select * from student;
+---------+-------+------+------+
| name    | marks | dept | id   |
+---------+-------+------+------+
| kelavii |    99 | cse  |    1 |
| roja    |    95 | cse  |    2 |
| saras   |    94 | ece  |    3 |
| raghul  |   100 | ece  |    4 |
| raja    |   106 | mett |    5 |
+---------+-------+------+------+
5 rows in set (0.00 sec)

mysql> select * from final_marks;
+-----------+
| tot_marks |
+-----------+
|       106 |
+-----------+
1 row in set (0.00 sec)
```

```
mysql> create table student1 (id int,name varchar(10),class int,email varchar(20));
Query OK, 0 rows affected (0.04 sec)

mysql> create table student_log (user varchar(20),description varchar(40));
Query OK, 0 rows affected (0.03 sec)

 delimiter $$
mysql> create trigger after_update_studentInfo
    -> after update
    -> on student1
    -> for each row
    -> begin
    ->  INSERT INTO student_log VALUES (USER(), CONCAT('update stu rec', OLD.name,
'previous class', OLD.class, 'present class:', NEW.class));
    -> end $$
Query OK, 0 rows affected (0.02 sec)
```

```
/*DDL COMMANDS*/

/*1.CREATE*/

create database dbmslab;

create table employee(
                              emp_no varchar(5),
                              emp_name varchar(20),
          designation varchar(20),
          salary varchar(20)
          );

desc employee;

create table employ(
                              emp_no varchar(5),
                              emp_name varchar(20),
          designation varchar(20),
    salary varchar(20)
    ) as select * from employee;

desc employ;

create table empp(
            emp_no varchar(5),
            emp_name varchar(20)
    ) as select emp_no,emp_name from employee;

desc empp;

/*2.ALTER*/

/*ADD*/

alter table employee add(qualification varchar(20));

alter table employee add(doj date, dob date);

/*MODIFY*/

alter table employee modify emp_no varchar(7);

alter table employee modify emp_no varchar(7), modify emp_name varchar(21);

/*DROP*/

alter table employee drop column qualification;

alter table employee drop doj, drop dob;

/*RENAME*/

alter table employee rename column emp_no to employee_no;

/*3.DROP*/

drop table employ;

/*4.TRUNCATE*/

truncate table employee;
```

```
/*5.RENAME*/

rename table employee to emp1;

/*6.DESCRIBE*/

desc emp1;

/*--------------------------EXP: NO: 02---------------------------------*/

/*DOMAIN INTEGRITY CONSTRAINTS
          1.NOT NULL
   2.CHECK CONSTRAINTS
*/

/*NOT NULL*/

create table student (
                              id varchar(10),
          fname char(20) not null,
          dept varchar(20),
          age int(5),
          salary int(30),
          location varchar(30)
          );

desc student;

insert into student values(01,'Raghul','cse',20,150000,'cuddalore');

select * from student;

alter table student add column gender varchar(2);

/*CHECK CONSTRAINT*/

alter table student add constraint ch_gender check(gender in ('M','F'));

insert into student values(02,'Rajeesh','ece',18,200000,'salem','M');

insert into student values(01,'Rajeesh','ece',18,200000,'sa','M');
```

```
/*----ENTITY INTEGRITY CONSTRAINT----*/

/*UNIQUE*/

alter table student modify location varchar(30) unique;

alter table student add constraint un_id unique(id);

delete from student where gender = 'j';

/*PRIMARY KEY*/

alter table student add constraint pk_id primary key(id);

/*REFERENTIAL INTEGRITY CONSTRAINT*/

CREATE TABLE customers (
    customer_id INT PRIMARY KEY,
    customer_name VARCHAR(50)
);

CREATE TABLE orders (
    order_id INT PRIMARY KEY,
    order_date DATE,
    customer_id INT,
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);

show databases;

use dbmslab;

SELECT DATABASE();


insert into customers values(01,'raghul'),(02,'sanmathi'),(03,'roja');
insert into orders values (091,'2023-11-13',01);
insert into orders values (092,'2023-11-13',02);

select * from customers;

select * from orders;
```

```
CREATE TABLE cus (
    customer_id INT ,
    customer_name VARCHAR(50),
    constraint PK_cus PRIMARY KEY(customer_id)
);

alter table cus disable constraint primary key;


/*--------------------------EXP: NO: 03---------------------------------*/

/*TCL*/

create table stu (fname varchar(10), id int);

select database();

use dbmslab;

insert into stu values('raghul',01);
insert into stu values('sanmathi',01);

select * from stu;

commit;

savepoint s5;

insert into stu values('roja',02);

rollback to s5;

-- Create a table for demonstration
CREATE TABLE bank_account1 (
    account_id INT PRIMARY KEY,
    balance DECIMAL(10, 2)
);

-- Insert initial data
INSERT INTO bank_account VALUES (1, 1000.00);

-- Start a transaction
BEGIN;
```

```
-- Savepoint before making changes
SAVEPOINT start_transaction;

-- Update balance
UPDATE bank_account1 SET balance = balance - 500.00 WHERE account_id = 1;

-- Check the balance
SELECT * FROM bank_account WHERE account_id = 1;

-- Commit the transaction
COMMIT;

-- Try to rollback to the savepoint (This will throw an error as the transaction is already
committed)
ROLLBACK TO start_transaction;

/*DCL*/

-- Create tables
CREATE TABLE users (
   user_id INT PRIMARY KEY,
   username VARCHAR(50) UNIQUE NOT NULL,
   password VARCHAR(50) NOT NULL
);

CREATE TABLE products (
   product_id INT PRIMARY KEY,
   product_name VARCHAR(100) NOT NULL,
   price DECIMAL(10, 2) NOT NULL
);

-- Create roles
CREATE ROLE sales_role;
CREATE ROLE admin_role;

-- Create users
CREATE USER sales_user IDENTIFIED BY 'abc123';
CREATE USER admin_user IDENTIFIED BY 'ramkrish7890';

-- Assign roles to users
GRANT sales_role TO sales_user;
GRANT admin_role TO admin_user;
```

```
-- Grant privileges
GRANT SELECT ON products TO sales_role;

GRANT INSERT, UPDATE, DELETE ON products TO admin_role;

select * from products;


-- Revoke privileges
REVOKE INSERT ON products FROM sales_role;

-- View granted privileges
SELECT * FROM information_schema.table_privileges WHERE table_name = 'products';

use dbmslab;
/*--------------------------EXP: NO: 04---------------------------------*/

-- DATA FUNCTION
-- addmonth
select timestampadd(month,2,sysdate()) from dual;

-- lastday
select last_day("2023-01-20") from dual;

-- difference
select timestampdiff(month,'2023-01-20','2023-05-03') from dual;

-- next day
select date_add(curdate(),interval 1 day) as tomorrow_date from dual;

-- NUMERICAL FUNCTION

-- abs
select abs(-29.5) from dual;

-- ceil
select ceil(134.78) from dual;

-- floor
select floor(29.87) from dual;

-- exp
```

```sql
select exp(5) from dual;

-- power
select power(5,2) from dual;

-- mod
select mod(13,3) from dual;

-- round
select round(5.8497463) from dual;

-- truncate
select truncate(2.79345,3) from dual;

-- sqrt
select sqrt(16) from dual;

-- CHARACTER FUNCTION

-- lower
select lower('HELLO') from dual;

-- upper
select upper('raghul') from dual;

-- ltrim
select trim(leading 'cse' from 'cseit') as trimmed;

-- rtrim
select trim(trailing 'it' from 'cseit') as trimmed;

-- replace
select replace('tamilnadu','t','T') from dual;

-- substr
select substr('information',3,4) from dual;

-- 4.CONVERTION FUNCTION

-- to char
select date_format('2023-05-03','%d %m %y') from dual;

-- to date
```

```sql
select str_to_date('15 may 2023','%d %m %y') from dual; -- ///null

-- 5.MISCELLANOUS FUNCTION

-- uid
select uuid() as uuid_value from dual;

-- user
select user() from dual;

-- null
select ifnull(null,'database');

-- vsize no of bytes
select length('college') as lengthstring from dual;

-- II. GROUP FUNCTION

-- avg
select avg(balance) from bank_account;

-- max
select max(balance) from bank_account;

-- min
select min(balance) from bank_account;

-- sum
select sum(balance) from bank_account;

-- III. COUNT FUNCTION

-- count(*)
select  count(*) from bank_account;

-- count(columns)
select  count(balance) from bank_account;

-- count(distinct_column)
select  count(distinct balance) from bank_account;

-- group by clause
select max(balance), account_id from bank_account group by account_id;
```

```sql
-- having clause
select max(balance), account_id from bank_account group by account_id having account_id=2;

-- IV. SPECIAL OPERATOR

SHOW TABLES;

SELECT * FROM CUSTOMERS;

-- like -- not like
select * from customers where customer_name like 'r%';

-- between not between
select * from bank_account where balance between 100 and 300;

-- in not in
select * from customers where customer_name in ('raghul','sanmathi');

-- any
select * from bank_account where balance > any (select * from bank_account where balance >
200);

/*------------------------EXP: NO: 05 SET OPERATIONS------------------------------*/

-- union

show tables;
use dbmslab;

CREATE TABLE student (
   name varchar(10),
   marks int
);

CREATE TABLE stud (
   name varchar(10),
   marks int
);

insert into student values('raghul',90),('sanmathi',99),('roja',95);
insert into stud values('raghul',90),('raja',99),('pugal',95);
```

```
select * from student;
select * from stud;

-- union
select name from student union select name from stud;

-- union all
select name from student union all select name from stud;

-- instersect
select student.name from student inner join stud where student.name = stud.name;

-- minus left/right join
select name from student left join stud using(name) where stud.name is null;
select name from stud right join student using(name) where student.name is null; -- //error

/*--------------------------EXP: NO: 06 IMPLEMENTATION OF VIEWS-----------------------------------*/

create view v_student as select * from student;

desc v_student;

select * from v_student;

select name from v_student where marks = 99;

insert into v_student values('saras',94);

update v_student set name = 'kelavii' where marks = 99;

delete from v_student where name = 'raghul';

drop view v_student;
```