```
/*
drop procedure if exists pro1;
delimiter $
create procedure pro1()
lbl1:begin
declare y varchar(10);
declare x varchar(10) default 'IET';
set y := 'Hello';
      select concat(y,' ' , x) as Message;
end lbl1 $
delimiter ;

*/


drop procedure if exists pro1;
delimiter $
create procedure pro1(x int, in y int)
lbl1:begin

select concat("Result is ", (x+y)) as "Output";

end lbl1 $
delimiter ;



/*

drop procedure if exists pro1;
delimiter $
create procedure pro1(x int, in y int, out o int)
lbl1:begin
      set o := x + y;
end lbl1 $
delimiter ;


/*

drop procedure if exists pro1;
delimiter $
create procedure pro1(x int, in y int, out o1 int, out o2 int)
lbl1:begin
      set o1 := x + y;
      set o2 := x * y;
end lbl1 $
delimiter ;



drop procedure if exists pro1;
delimiter $
```

```
create procedure pro1(inout x int)
lbl1:begin
      set x := x * x;
end lbl1 $
delimiter ;
```

```
drop procedure if exists pro1;
delimiter $
create procedure pro1(in _deptno int)
lbl1:begin
      select ename, deptno from emp where deptno = _deptno;
end lbl1 $
delimiter ;
```

```
drop procedure if exists pro1;
delimiter $
create procedure pro1(in _deptno int)
lbl1:begin
      declare x int;
      select distinct deptno into x from emp where deptno = _deptno;

      if x is not null  then
            select ename, deptno from emp where deptno = x;
      else
            select "Record not found.." as Message;

      end if;
end lbl1 $
delimiter ;
```

```
drop procedure if exists pro1;
delimiter $
create procedure pro1(in _deptno int , _dname varchar(12), _loc
varchar(20), _pwd  varchar(20))
lbl1:begin
 insert into dept values(_deptno, _dname, _loc, _pwd);
end lbl1 $
delimiter ;
```

```
drop procedure if exists pro1;
delimiter $
create procedure pro1()
lbl1:begin
declare exit handler for 1146 select "table not found" as "Message";
 insert into abc(deptno) values(2);
end lbl1 $
delimiter ;
```

```
drop procedure if exists pro1;
delimiter $
create procedure pro1(c1 varchar(100), tName varchar(2000))
lbl1:begin
      declare exit handler for 1146 select "table not found" as
"Message";
      set @t1 = concat("select ", c1, " from ", tName);
      prepare stat1 from @t1;
      execute stat1;

end lbl1 $
delimiter ;


drop procedure if exists pro1;
delimiter $
create procedure pro1()
begin
 declare x int default 0;
lbl1: loop
      select x;
      set x = x + 1;
      if x = 7 then
            leave lbl1;
      end if;
end loop lbl1;

end $
delimiter ;


drop procedure if exists pro1;
delimiter $
create procedure pro1()
begin
declare v1 int;
declare v2 varchar(20);
declare v3 varchar(20);
declare v4 varchar(20);
declare done tinyint default false;

declare  c1 cursor for select empno, ename, job, deptno from emp;
declare exit handler for not found set done=true;
open c1;
lbl1 :loop
      fetch c1 into v1, v2, v3, v4;
      if done then
            leave lbl1;
      else
            select v1, v2, v3, v4;
      end if;
end loop lbl1;
```

```
close c1;
end $
delimiter ;



/*

drop function if exists f1;
delimiter $
create function f1(x int , y int) returns int
begin
      return(ifnull(x,0) + ifnull(y,0));
end $

delimiter ;


drop function if exists f1;
delimiter $
create function f1() returns int
begin
declare x int default 0;
select max(deptno) + 1 into x from dept;
      return(x);
end $

delimiter ;




drop function if exists f1;
delimiter $
create function f1() returns int
begin
declare x int default 0;
      # select * from dept;   error
      return('Hello');
end $

delimiter ;




drop procedure if exists pro1;
delimiter $
create procedure pro1()
begin
set @x = 0;
      select deptno into @x from dept where deptno=10;
end $
```

```
delimiter ;


drop function if exists f1;
delimiter $
create function f1() returns int
begin
      call pro1();
      return(@x);
end $

delimiter ;




drop procedure if exists pro1;
delimiter $
create procedure pro1()
begin
declare v1 int;
declare v2 varchar(20);
declare v3 varchar(20);
declare v4 varchar(20);
declare done tinyint default false;

declare  c1 cursor for select empno, ename, job, deptno from emp;
declare exit handler for not found set done=true;
open c1;
lbl1 :loop
      fetch c1 into v1, v2, v3, v4;
      if done then
            select 'All records transfared' as message;
            leave lbl1;
      else
            insert into ab values( v1, v2, v3, v4);
      end if;
end loop lbl1;
close c1;
end $
delimiter ;




drop procedure if exists pro1;
delimiter $
create procedure pro1()
begin
      create table if not exists abc1(c1 int);
end $
delimiter ;
```

```
drop trigger if exists t1;
delimiter $
create trigger t1 before insert on dept for each row
begin
      insert into message values(1,'Record Inserted properly');
end $
delimiter ;



drop trigger if exists t1;
delimiter $
create trigger t1 before insert on dept for each row
begin
      insert into d values(new.deptno, new.dname, new.loc,
new.pwd,user(), curdate(), curtime());
end $
delimiter ;


#insert into dept values(41,'HRD','Pune','something');


drop trigger if exists t1;
delimiter $
create trigger t1 after delete on dept for each row
begin
      insert into d values(old.deptno, old.dname, old.loc,
old.pwd,user(), curdate(), curtime(),'Delete');
end $
delimiter ;

# delete from dept where deptno=40;




drop trigger if exists t1;
delimiter $
create trigger t1 after insert on dept for each row
begin
      if new.loc <> 'pune' then
            signal sqlstate '45000' set message_text = 'Record cannot be
inserted...';
      end if;
end $
delimiter ;



drop trigger if exists t1;
delimiter $
create trigger t1 before insert on message for each row
```

```
begin
declare x int default 0;
      select max(id) + 1  into x from message;

      set new.id = x;
end $
delimiter ;


drop trigger if exists t2;
delimiter $
create trigger t2 before update on dept for each row
begin
      insert into message values(default, concat("Old department name was
", old.dname, ' and new department name is ', new.dname));
end $
delimiter ;


drop trigger if exists t1;
delimiter $
create trigger t1 before insert on dept for each row
begin
      set new.dname = trim(upper(new.dname));
end $
delimiter ;


# insert into dept values(2,'pqr  ',1,1);



drop trigger if exists t1;
delimiter $
create trigger t1 before insert on dept for each row
begin
      declare x int default 0;
      select count(*) into x  from dept;
      if x > 6  then
            signal sqlstate '45000' set message_text='More records';
      end if;

end $
delimiter ;
*/
```