IMP queries

show procedure status where db='DB\_Name';

--------------------------------------------------------------------------------------------------------------------------------------

PL/SQL

**14 a.**

create table stud(RollNo int primary key, attendance int,status varchar(5));

insert into stud(RollNo,attendance) values(1,150),(2,200),(3,80),(4,70),(5,180);

select \* from stud;

**delimiter //**

**create procedure check\_att(in roll int)**

**begin**

**declare att int;**

**declare total int;**

**declare exit handler for not found select 'Data not found!!!' message;**

**set total=200;**

**select attendance into att from stud where RollNo=roll;**

**if ((att/total)\*100)>=75 then**

**update stud set status='ND' where RollNo=roll;**

**select 'Term Granted' Message;**

**else**

**update stud set status='D' where RollNo=roll;**

**select 'Term Not Granted' Message;**

**end if;**

**end;**

**//**

**delimiter ;**

14 b

create table account\_master(ID int primary key,Current\_balance int);

insert into account\_master values(1,10000),(2,5000),(3,60000);

delimiter //

create procedure withdraw(in acc\_id int,in amt int)

begin

declare bal int;

declare sp condition for sqlstate '45000';

select Current\_balance into bal from account\_master where ID=acc\_id;

if bal<amt then

signal sqlstate '45000'

set message\_text='NotEnoughBalance';

else

set bal = bal-amt;

update account\_master set Current\_balance=bal where ID=acc\_id;

end if;

end;

//

create procedure deposit(in acc\_id int,in amt int)

begin

declare bal int;

select current\_balance into bal from account\_master where ID=acc\_id;

update account\_master set current\_balance=bal+amt where ID=acc\_id;

end;

//

15A

delimiter //

create procedure check\_br(in uid int)

begin

declare temp\_bal int;

declare sp condition for sqlstate'45000';

select bal\_due into temp\_bal from client\_master where id=uid;

if temp\_bal<0 then

signal sqlstate '45000'

set message\_text='BR violated';

else

select 'BR not violated' Message;

end if;

end

//

Query OK, 0 rows affected (0.02 sec)

delimiter ;

15b

create procedure Calculate\_fine(in roll int)

begin

declare fine\_amt int;

declare no\_of\_days int;

declare issue\_date date;

select DateofIssue into issue\_date from borrower where Roll\_no=roll;

select datediff(curdate(),issue\_date) into no\_of\_days;

if no\_of\_days>15 and no\_of\_days<=30 then

set fine\_amt=no\_of\_days\*5;

elseif no\_of\_days>30 then

set fine\_amt=(no\_of\_days-30)\*50+30\*5;

else

set fine\_amt=0;

end if;

insert into fine values(roll,curdate(),fine\_amt);

update borrower set Status='R' where Roll\_no=roll;

end;

//

Cursor

15C

delimiter //

create procedure check\_salary()

begin

declare temp\_emp int;

declare temp\_dno int;

declare temp\_salary int;

declare avg\_salary int;

declare temp\_dno\_dept\_salary int;

declare ec boolean;

declare cur1 cursor for select avg(salary),dno from emp group by dno;

declare continue handler for not found set ec=true;

open cur1;

l1:loop

fetch cur1 into temp\_salary,temp\_dno;

insert into dept\_salary values(temp\_salary,temp\_dno);

if ec then

close cur1;

leave l1;

end if;

end loop l1;

end

//

16b

delimiter //

create procedure inc\_salary()

begin

declare temp\_salary int;

declare temp\_id int;

declare avg\_salary int;

declare exitcond boolean;

declare cur cursor for select emp\_id from salary;

declare cur2 cursor for select salary from salary;

declare continue handler for not found set exitcond=true;

select avg(salary) into avg\_salary from salary;

open cur;

open cur2;

l1:loop

fetch cur into temp\_id;

fetch cur2 into temp\_salary;

if(temp\_salary<avg\_salary) then

set temp\_salary=temp\_salary+temp\_salary\*0.1;

insert into incr\_salary values(temp\_id,temp\_salary);

end if;

if exitcond then

close cur;

close cur2;

leave l1;

end if;

end loop l1;

end

//

17C

delimiter //

create procedure check\_salary()

begin

declare temp\_emp int;

declare temp\_dno int;

declare temp\_salary int;

declare avg\_salary int;

declare temp\_dno\_dept\_salary int;

declare ec boolean;

declare cur1 cursor for select avg(salary),dno from emp group by dno;

declare continue handler for not found set ec=true;

open cur1;

l1:loop

fetch cur1 into temp\_salary,temp\_dno;

insert into dept\_salary values(temp\_salary,temp\_dno);

if ec then

close cur1;

leave l1;

end if;

end loop l1;

end

//

18A

delimiter //

create trigger after\_delete

after delete on client\_master

for each row

begin

insert into audit\_table

set action='DELETE',

id=old.id,

data=old.data;

end

//

delimiter //

create trigger after\_update

after update on client\_master

for each row

begin

insert into audit\_table

set action='UPDATE',

id=old.id,

data=old.data;

end

//

18B

delimiter //

create trigger after\_insert

after insert

on emp

for each row

begin

if(new.salary<50000) then

signal sqlstate '45000' set message\_text='Rejected!!!';

end if;

insert into tracking

set eno=new.eno,

salary=new.salary;

end

//