4. Exception

delimiter $

create procedure fine\_calculation(IN rno int(3), bname char(20))

begin

declare i\_date date;

declare fine\_amt int;

declare exit handler for sqlexeption select ‘Table not found’;

select dateofIssue into i\_date from borrower where rollin=rno and bname=bname;

select dateddiff(curdate(), i\_date) into diff;

if(diff>15 and diff<30) then

set fine\_amt = diff\*5;

insert into fine values(rno, curdate(), bname);

elseif(diff>30) then

set fine\_amt = 15\*5 + (diff-30)\*50;

insert into fie values(rno, curdate(), bname);

end if;

update borrower set status = ‘R’ where rollno=rno and bname = name;

end;

$

delimiter;

call fine\_calculation(3, 'DBMS');

5. Procedure

delimiter //

create procedure proc\_Grade(IN rno int, out gade varchar(25))

begin

declare m int;

select marks into m from stud\_marks where name = (select name from result where rol=rno);

if m>=990 and m<=1500 then

select 'Distinction' into grade;

update result set clas='Distiction' where roll=rno;

elseif m>=900 and m<=989 then

select 'First Class' into grade;

update result set clas='First Class' where roll=rno;

elseif m>=825 and m<=899 then

select 'Second Class' into grade;

update result set class='Second Class' where roll=rno;

else

select '\_ \_' into grade;

update result set clas='\_ \_' where roll=rno;

end if;

end;

//

delimiter //

create function func\_grade(rno int)

return varchar(25)

deterministic

begin

declare grade varchar(25);

call proc\_Grade(rno, grade)

return grade;

end;

//

6. Cusor

delimiter //

create procedure n1(IN rno1 int)

begin

declare rno2 int;

declare exit\_cond boolean;

declare c1 cursor for select rno from o\_rollcall where rno>rno1;

declare continue handler for not found set exit\_cond = TRUE;

open c1;

l1: loop

fetch c1 into rno2;

ifnot exist(select \* from n\_rollcall where rno=rno2) then

insert into n\_rollcall select \* from o\_rollcall where rno=rno2;

end if;

if exit\_cond then

close c1;

leave l1;

end if;

end loop l1;

end;

//

call n1(3)

select \* from n\_rollcall;

7. Trigger

delimiter //

create trigger tr\_insert\_stud

before insert

on stud

for each row

begin

insert into issued values(new.sid, new.bname)

//

delimiter //

create trigger tr\_del\_stud

after delete

on stud

for each row

begin

delete from issued where sid=old.sid;

//

delimiter //

create trigger tr\_up\_issued

after update

on stud

for each row

begin

update issued

set sid=new.sid where sid=old.sid

//