**CREATION OF STUDENT TABLE:**

create table student

(

sid varchar2(10),

sname varchar2(300),

dob date,

semno number(3),

fid varchar2(10),

cid varchar2(4),

pass varchar2(200),

constraint stud\_prim primary key(sid)

);

alter table student add constraint stud\_name check (sname = initcap(sname));

alter table student add constraint stud\_for1 foreign key(semno) references semester(semno);

alter table student add constraint stud\_for2 foreign key(cid) references course(cid);

alter table student add constraint stud\_for3 foreign key(fid) references faculty(fid);

**CREATION OF DEPARTMENT TABLE:**

create table depart

(

did varchar2(5),

dname varchar2(50),

constraint dep\_id check (did like 'D%'),

constraint dep\_dname check (dname = upper(dname)),

constraint dep\_prim primary key(did)

);

**CREATION OF FACULTY TABLE:**

create table faculty

(

fid varchar2(10),

fname varchar2(30),

tutor varchar2(10),

did varchar2(10),

password varchar2(20),

dob date,

constraint fac\_name check (fname = initcap(fname)),

constraint fac\_tutor foreign key(tutor) references course(cid),

constraint fac\_did foreign key(did) references depart(did),

constraint fac\_prim primary key(fid)

);

**CREATION OF COURSE TABLE:**

create table course

(

cid varchar2(10),

cname varchar2(30),

did varchar2(5),

constraint cour\_id check (cid like 'C%'),

constraint cour\_forky foreign key(did) references depart(did),

constraint cour\_prim primary key(cid,did)

);

**CREATION OF SEMESTER TABLE:**

create table semester

(

semno number(2),

startdate date,

enddate date,

constraint sem\_date check (enddate>startdate),

constraint sem\_prim primary key(semno)

);

**CREATION OF LEAVE TABLE:**

create table leave

(

leavno number(2),

sid varchar2(10),

fid varchar2(10),

doe date,

from\_date date,

from\_sess varchar2(20),

to\_dat date,

to\_sess varchar2(20),

leav\_type varchar2(20),

reason varchar2(500),

status varchar2(15),

constraint chk\_frmses check(from\_sess in ('AFTERNOON','FORENOON')),

constraint chk\_toses check(to\_sess in ('AFTERNOON','FORENOON')),

constraint date\_chk check(from\_date<=to\_dat),

constraint chk\_type check(leav\_type in ('Medical','Exception','Others')),

constraint chk\_status check(status in('confirmed','aborted')),

constraint leav\_prim primary key(sid ,leavno)

);

Alter table leave add semno number(2);

Alter table leave add constraint chk\_sem foreign key(semno) references semester(semno);

**CREATION OF LEAVE COUNT TABLE :**

create table leavecount

(

sid varchar2(10) primary key references student(sid),

count number(2) default 0

);

**INSERTION:**

|  |  |  |
| --- | --- | --- |
| Semno | Startdate | Enddate |
| 4 | 1-dec-2010 | 1-may-2011 |

|  |  |
| --- | --- |
| Did | Dname |
| D01 | DOMCA |

|  |  |  |
| --- | --- | --- |
| CID | Cname | Did |
| C01 | M.Sc SWE | D01 |
| C02 | M.Sc TCS | D01 |

**PROCEDURE FOR INSERTING IN FACULTY TABLE:**

create or replace procedure fac\_insert\_proc

(p\_fid in faculty.fid%type ,p\_fname in faculty.fname%type,

p\_tutor in faculty.tutor%type ,p\_did in faculty.did%type ,p\_dob in varchar2,p\_out out integer)

as

v\_fid faculty.fid%type;

v\_pass faculty.password%type;

v\_fname faculty.fname%type;

v\_did faculty.did%type;

v\_dob faculty.dob%type;

begin

v\_fname := initcap(p\_fname);

v\_dob := to\_date(p\_dob);

v\_pass := to\_char(v\_dob);

select fid into v\_fid from faculty where fid=p\_fid; p\_out:=0;

exception

when no\_data\_found then

select did into v\_did from depart where did = p\_did;

if v\_did is not NULL then

if p\_tutor is not NULL then

insert into faculty values(p\_fid,v\_fname,p\_tutor,p\_did,v\_pass,p\_dob);

p\_out:=1;

else

insert into faculty(fid,fname,did,password,dob) values(p\_fid,v\_fname,p\_did,v\_pass,p\_dob); p\_out:=1;

end if;

commit;

end if;

end;

**PROCEDURE FOR INSERTING IN STUDENT TABLE:**

create or replace procedure stud\_insert\_proc

(p\_sid in student.sid%type ,p\_sname in student.sname%type,p\_dob in varchar2,

p\_semno in student.semno%type ,p\_fid in student.fid%type ,p\_cid in student.cid%type)

as

v\_sid student.sid%type;

v\_pass student.pass%type;

v\_sname student.sname%type;

v\_cid student.cid%type;

v\_fid student.fid %type;

v\_sem student.semno%type;

begin

v\_sname := initcap(p\_sname);

v\_pass := p\_dob;

select sid into v\_sid from student where sid=p\_sid;

exception

when no\_data\_found then

select cid into v\_cid from course where cid=p\_cid;

if v\_cid is not NULL then

select fid into v\_fid from faculty where fid=p\_fid;

if v\_fid is not NULL then

select semno into v\_sem from semester where semno=p\_semno;

if v\_sem is not NULL then

insert into student values(p\_sid,v\_sname,p\_dob,p\_semno,p\_fid,p\_cid,v\_pass);

commit;

end if;

end if;

end if;

end;

**TRIGGER FOR INSERTING AFTER STUDENT:**

create or replace trigger stud\_leav\_count

after insert on student

for each row

begin

insert into leavecount(sid) values(:new.sid);

end;

**TRIGGER FOR UPDATING AFTER LEAVE:**

create or replace trigger leav\_insert

after insert on leave

for each row

begin

update leavecount set count = count + 1 where sid = :new.sid;

end;

**PROCEDURE FOR INSERTING LEAVE:**

create or replace PROCEDURE leav\_insert

(

p\_sid in leave.sid%type, p\_fromdate in VARCHAR2,

p\_fromsess in leave.from\_sess%type, p\_todate in VARCHAR2, p\_tosess in leave.to\_sess%type,

p\_leavtype in leave.leav\_type%type, p\_reason in leave.reason%type,p\_semno in leave.semno%type

)

as

v\_leaveno number(2);

v\_dat date;

v\_sid leave.sid%type;

v\_fid leave.fid%type;

v\_fromdate leave.from\_date%type;

v\_todate leave.to\_dat%type;

v\_fromsess leave.from\_sess%type;

v\_tosess leave.to\_sess%type;

v\_type leave.leav\_type%type;

v\_semdate date;

begin

v\_fromdate := to\_date(p\_fromdate);

v\_todate := to\_date(p\_todate);

v\_fromsess := upper(p\_fromsess);

v\_tosess := upper(p\_tosess);

v\_type := initcap(p\_leavtype);

if(v\_fromdate > v\_todate) then

goto label1;

end if;

select startdate into v\_semdate from semester where semno=p\_semno;

if(v\_semdate>v\_fromdate) then

goto label1;

end if;

select enddate into v\_semdate from semester where semno=p\_semno;

if(v\_semdate<v\_fromdate) then

goto label1;

end if;

select startdate into v\_semdate from semester where semno=p\_semno;

if(v\_semdate>v\_todate) then

goto label1;

end if;

select enddate into v\_semdate from semester where semno=p\_semno;

if(v\_semdate<v\_todate) then

goto label1;

end if;

if(v\_fromsess not in('AFTERNOON','FORENOON')) then

goto label1;

end if;

if(v\_tosess not in('AFTERNOON','FORENOON')) then

goto label1;

end if;

if(v\_type not in('Medical','Exception','Others')) then

goto label1;

end if;

select count into v\_leaveno from leavecount where sid=p\_sid;

v\_leaveno := v\_leaveno + 1;

select sysdate into v\_dat from dual;

select fid into v\_fid from student where sid= p\_sid;

select sid into v\_sid from student where sid= p\_sid;

if v\_sid is not NULL then

insert into leave(LEAVNO,SID,FID,DOE,FROM\_DATE,FROM\_SESS,TO\_DAT,TO\_SESS,LEAV\_TYPE,REASON)

values(v\_leaveno,p\_sid,v\_fid,v\_dat,v\_fromdate,v\_fromsess,v\_todate,v\_tosess,v\_type,p\_reason);

end if;

<<label1>> v\_leaveno :=1;

end;

**PROCEDURE FOR FETCHING SEMESTER NUMBER:**

Create or replace procedure semno\_fetch

(p\_semno out semester.semno%type, p\_start out semester.startdate%type, p\_end out semester.enddate%type)

As

V\_date date;

Begin

Select sysdate into v\_date from dual;

Select semno into p\_semno from semester where startdate<=v\_date and enddate>=sysdate;

Select startdate into p\_start from semester where semno = p\_semno;

Select enddate into p\_end from semester where semno = p\_semno;

End;

**PROCEDURE FOR FETCHING COURSE NAME:**

Create or replace procedure cour\_name

(p\_sid in student.sid%type,p\_cname out course.cname%type)

As

P\_cid student.cid%type;

Begin

Select cid into p\_cid from student where sid = p\_sid;

Select cname into p\_cname from course where cid = p\_cid;

End;

**PROCEDURE FOR FETCHING NAME:**

Create or replace procedure stud\_name

(v\_sid in student.sid%type,v\_name out student.sname%type)

As

begin

Select sname into v\_name from student where sid = v\_Sid;

End;

**PROCEDURE FOR CONFORMING LEAVE BY A TUTOR:**

Create or replace procedure con\_leav

( p\_fid in leave.fid%type,p\_cur in out sys\_refcursor)

As

begin

open p\_cur for select \* from leave where fid=p\_fid;

end;

**PROCEDURE FOR UPDATING CONFORMATION LEAVE BY A TUTOR:**

Create or replace procedure update\_leav

( p\_sid in leave.sid%type,p\_leavno in leave.leavno%type,p\_status in leave.status%type)

As

begin

update leave set status = p\_status where sid=p\_sid and leavno=p\_leavno;

end;

**PROCEDURE OF STUDENT POSTING LEAVE DETAILS:**

Create or replace procedure stud\_stud

(p\_sid in leave.sid%type, p\_cur in out sys\_refcursor)

As

Begin

Open p\_cur for select \* from leave where sid=p\_sid;

End;

**PROCEDURE FOR CHECKING PASSWORD -STUDENT:**

Create or replace procedure stud\_pass

(p\_sid in student.sid%type, p\_pass student.pass%type,p\_out out integer)

As

P\_rec student%rowtype;

begin

P\_out:=0;

Select \* into p\_rec from student where sid=p\_sid;

If(p\_rec.pass=p\_pass)then

P\_out:=1;

End if;

End;

**PROCEDURE FOR CHECKING PASSWORD\_FACULTY:**

Create or replace procedure fac\_pass

(p\_fid in faculty.fid%type, p\_pass faculty.password%type,p\_out out integer)

As

P\_rec faculty%rowtype;

begin

P\_out:=0;

Select \* into p\_rec from faculty where fid=p\_fid;

If(p\_rec.password=p\_pass)then

P\_out:=1;

End if;

End;