|  |
| --- |
|  |
| STUDENT ATTENDANCE MANAGEMENT SYSTEM |
| D.B.M.S Project |
|  |
| **Subhasree Bose** |
|  |

SECTION – B ROLL – 108

DEPARTMENT- C.S.E

**SCHEMA**

Student

Name, section, class roll number, enrollment number,email,course\_id

Course

Year, Department, Course\_name,Course\_id,Semester

Subject

Subject\_code, subject\_name,course\_id

Allocation

time day section course\_id lecture\_id

Lecturer

Name email course \_id lecturer\_id

Attendance

Periods, date ,course\_id ,enrollment number,lecture\_id,section,time

**NORMALIZED SCHEMA**

SQL> desc student

Name Null? Type

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

EROLL NOT NULL NUMBER

CROLL NOT NULL NUMBER

NAME VARCHAR2 (20)

EMAIL VARCHAR2(20)

COURSE\_ID VARCHAR2(10)

SECTION VARCHAR2(1)

SQL> desc course

Name Null? Type

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

ID NOT NULL VARCHAR2(10)

DEPT VARCHAR2(10)

YEAR NUMBER

SEMESTER NUMBER

COURSE\_NAME VARCHAR2(10)

SQL> desc subject

Name Null? Type

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

SUBJECT\_ID NOT NULL VARCHAR2(10)

NAME NOT NULL VARCHAR2(20)

COURSE\_ID VARCHAR2(10)

SQL> desc allocation

Name Null? Type

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

COURSE\_ID NOT NULL VARCHAR2(10)

SEC NOT NULL VARCHAR2(1)

DAY NOT NULL VARCHAR2(2)

PERIOD NOT NULL NUMBER

SUBJECT\_CODE VARCHAR2(10)

LECTURER\_ID NUMBER

SQL> desc attend

Name Null? Type

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

EROLL NOT NULL NUMBER

COURSE\_ID NOT NULL VARCHAR2(10)

ADATE NOT NULL DATE

PERIOD1 NUMBER

PERIOD2 NUMBER

PERIOD3 NUMBER

PERIOD4 NUMBER

PERIOD5 NUMBER

PERIOD6 NUMBER

SECTION VARCHAR2(1)

SQL> desc period

Name Null? Type

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

ID NOT NULL NUMBER

TIME\_IN VARCHAR2(10)

TIME\_OUT VARCHAR2(10)

**TABLE SCRIPTS**

Student table

create table student(eroll number primary key,croll number not null,name va

rchar2(20),email varchar2(20),course\_id varchar2(10),section varchar2(2) ,CONSTRAINT eustud UNIQUE (email));

alter table student add constraint fk\_student\_course foreign key student(course\_id) references course(id);

Course table

create table course(id varchar2(10) primary key,course\_name varchar2(10),dept varc

har2(10),year number,semester number);

Subject table

create table subject (subject\_id varchar2(10) primary key,name varchar2(10)

not null,course\_id varchar2(10));

alter table subject add constraint fk\_subject\_course foreign key subject(course\_id) references course(id);

Lecturer Table

create table lecturer( lecturer\_id number primary key,name varchar2(20) not

null,email varchar2(20),subject1 varchar2(10) ,subject2 varchar2(10) ,subject3 varchar2(10) CONSTRAINT eulect UNIQUE (email));

alter table lecturer add constraint fk\_lecturer\_subject1 foreign key lecturer(subject1) references subject(subject\_id);

alter table lecturer add constraint fk\_lecturer\_subject2 foreign key lecturer(subject2) references subject(subject\_id);

alter table lecturer add constraint fk\_lecturer\_subject3 foreign key lecturer(subject3) references subject(subject\_id);

Attendance table

create table attend (eroll number ,course\_id varchar2(10),adate date,period1 number,period2 number,period3 number,period4 number,period5 number,period6 number,section varchar2(2));

alter table attend add constraint pk\_attend primary key lecturer(eroll,course\_id,adate);

Period table

create table period(id number primary key,time\_in varchar2(6),time\_out varchar2(6));

Allocation table

create table allocation (course\_id varchar2(10),day varchar2(3),sec varchar2(2),subject\_code(varchar2(10),period number,lecturer\_id number);

alter table attend add constraint pk\_alloc primary key allocation(course\_id,period,day,sec);

alter table allocation add constraint fk\_allocation\_course foreign key allocation(course\_id) references course(id);

alter table allocation add constraint fk\_allocation\_subject foreign key allocation(subject\_code) references subject(subject\_id);

**STORED PROCEDURE**

GETTIG ATTENDANCE BY SUBJECT

create or replace procedure getDates(ss out sys\_refcursor)

is

begin

open ss for select distinct adate from attend order by adate;

end;

create or replace procedure testDynamicNumbers(ss out sys\_refcursor,dt varchar2,d varchar2, s varchar2) is

cursor c is select period from allocation where day=d and subject\_code = s;

array TNumberArray;

str\_query varchar2(500);

end\_query varchar2(500);

begin

open c;

loop

fetch c bulk collect into array;

exit when c%NotFound;

end loop;

close c;

str\_query:='select a.adate,a.eroll, b.name,';

end\_query:=' from attend a inner join student b on a.eroll=b.eroll where a.adate='''||dt||''' order by a.eroll';

if array is empty then

DBMS\_OUTPUT.put\_line('array empty');

else

for i in array.first..array.last

loop

DBMS\_OUTPUT.put\_line('v\_month\_va(i): '||array(i));

str\_query:=str\_query || 'a.Period'||array(i);

if i <> array.last then

str\_query:=str\_query||',';

end if;

end loop;

str\_query:=str\_query||end\_query;

DBMS\_OUTPUT.put\_line(str\_query);

open ss for str\_query;

end if;

end;

/

CALCULATING PERCENTAGE

create or replace function cal\_percentage(e number) return number

is

er number;

t number;

p number;

str\_query varchar2(250);

begin

str\_query:='select sum(decode(sign(period1),-1,0,1)+decode(sign(period2),-1,0,1)+decode(sign(period6),-1,0,1)+decode(sign(period3),-1,0,1)+decode(sign(period4),-1,0,1)+decode(sign(period5),-1,0,1)) from attend where eroll ='||e;

execute immediate str\_query into er;

str\_query:='select count(distinct adate) from attend';

execute immediate str\_query into t;

p:=(er/(t\*6))\*100;

return p;

end;

GETTING ATTENDANCE

create or replace procedure attendance (catCur OUT SYS\_REFCURSOR ,d varchar2,y number,s number,c varchar2,se varchar2) is

i varchar2(10);

str\_query varchar2(500);

begin

str\_query:='select id from course where ( dept='''||d ||''' and year='||y||' and semester ='||s ||' and course\_name = '''||c||''' )';

execute immediate str\_query into i;

str\_query:='select b.adate ,a.eroll,a.croll,a.name,b.period1,b.period2,b.period3,b.period4,b.period5,b.period6 from student a inner join attend b on a.eroll=b.eroll where b.course\_id='''||i||''' and b.section ='''||se||''' order by b.adate,a.croll';

open catcur for str\_query;end;

GETTING ATTENDANCE BY DATE

create or replace procedure attendance\_date (catCur OUT SYS\_REFCURSOR ,d varchar2,y number,s number,c varchar2,se varchar2,dt varchar2) is

i varchar2(10);

str\_query varchar2(250);

begin

str\_query:='select id from course where ( dept='''||d ||''' and year='||y||' and semester ='||s ||' and course\_name = '''||c||''' )';

execute immediate str\_query into i;

str\_query:='select b.adate ,a.eroll,a.croll,a.name,b.period1,b.period2,b.period3,b.period4,b.period5,b.period6 from student a inner join attend b on a.eroll=b.eroll where b.course\_id='''||i||''' and b.section ='''||se||''' and b.adate = '''|| dt || '''';

open catcur for str\_query;end;

TAKING ATTENDANCE

create or replace procedure attendance\_update(erol number,state number,period varchar2,atdate date,cid varchar2,se varchar2) is str\_query varchar2(200);flag number;begin str\_query:='select count(\*) from attend where adate = '''||atdate||''' and course\_id ='''||cid||''' and section = '''||se||'''';execute immediate str\_query into flag;dbms\_output.put\_line(flag);if flag =0 then

str\_query:='insert into attend(eroll,course\_id,adate,section,'||period||') values('||erol||','''||cid||''','''||atdate||''','''||se||''','||state||')';

else str\_query:='update attend set '||period||' = '||state||' where adate = '''||atdate||''' and course\_id ='''||cid||''' and section = '''||se||''' and eroll='||erol;

end if;execute immediate str\_query;end;

TRIGGER FOR GENERATING ROLL NUMBERS

create or replace trigger insert\_stud

before insert on student

for each row

declare

erol number;

crol number;

flag number;

begin

select nvl(max(eroll),100000) into erol from student;

select nvl(max(croll),0) into crol from student where course\_id=:new.course\_id;

:new.eroll:=erol + 1;

:new.croll:=crol+1;

if(:new.eroll<=10) then

:new.section:='A';

else

:new.section:='B';

end if;

end;

/

INSERTING STUDENT

create or replace procedure proc\_insert(n varchar2, e varchar2,d varchar2,y number,s

number,c varchar2)

is

i varchar2(10);

str\_query varchar2(200);

begin

str\_query:='select id from course where ( dept='''||d ||''' and year='||y||' and semester

='||s ||' and course\_name = '''||c||''' )';

execute immediate str\_query into i;

str\_query:='insert into student(name, email,course\_id) values('''||n||''' ,'''||

e||''','''||i||''')';

execute immediate str\_query ;

end;

/

STUDENT SCHEDULE

create or replace procedure proc\_student\_schedule(sscur OUT SYS\_REFCURSOR,d varchar2,y number,s number,c varchar2,se varchar2)

is

i varchar2(10);

str\_query varchar2(200);

begin

str\_query:='select id from course where ( dept='''||d ||''' and year='||y||' and semester ='||s ||' and course\_name = '''||c||''' )';

execute immediate str\_query into i;

str\_query:='select day,subject\_code,period,lecturer\_id from allocation where course\_id ='''||i||''' and sec = ''' ||se||'''';

open sscur for str\_query;

end;

TEACHER SCHEDULE

create or replace procedure proc\_teacher\_schedule(sscur OUT SYS\_REFCURSOR,e varchar2)

is

str\_query varchar2(200);

l number;

begin

str\_query:='select lecturer\_id from lecturer where email='''||e||'''';

execute immediate str\_query into l;

str\_query:='select day,subject\_code,period,sec from allocation where lecturer\_id = '||l ;

open sscur for str\_query;

end;

GETTING SUBJECT INFORMATION

create or replace procedure get\_subject(sscur OUT SYS\_REFCURSOR,s varchar2) is str\_query varchar2(200);begin

str\_query:='select a.name,b.course\_name,b.dept,b.year,b.semester from subject a inner join course b on a.course\_id=b.id where a.subject\_id='''||s||'''';open sscur for str\_query;end;