

**AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)**

**FACULTY OF SCIENCE & TECHNOLOGY**

**DEPARTMENT OF CSE**

**ADVANCE DATABASE MANAGEMENT SYSTEM**

**Summer 2021-2022**

**Section: B, Group name: Database Noobies**

**FINAL TERM PROJECT ON**

***AIUB portal system***

**Supervised By**

**Rezwan Ahmed**

**Submitted By**

|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **Contribution** |
| **1. Muhammad Shahriar Zaman** | **20-41840-1** | **All** |
|  |  |  |
|  |  |  |

Date of Submission: **August 28th, 2022**

**TABLE OF CONTENTS**

|  |
| --- |
| **TOPICS** |
| 1. **System Summary** |
| 1. **Table creation and data insertion into tables** |
| 1. **Procedures** |
| 1. **Views** |
| 1. **Triggers** |
|  |
|  |
|  |
|  |
|  |
|  |

1. **System Summary**

We have developed a database system which aims to replicate an online university portal. Here we have user types such as students, teachers, admin and officers.

Students will be able to see their information, results and apply for registering courses.

Teachers will be able to evaluate students, update their marks and grades

The admin will allocate courses for students who have applied for registration, store and update various attributes of the student account and also provide salary to the teachers.

**Table Creation and Value insertion**

**A. Creating a table for all students**

create table all\_students

(

name varchar(50) not null,

id varchar(15) not null primary key,

password varchar(30) not null,

program varchar(10),

cgpa float,

credits\_completed int,

admission\_date date,

graduation\_date date,

status varchar(20)

);

**B. Inserting records for students**

insert into all\_students values('Shahriar Zaman', '22-41840-1', '1234', 'CSE', 3.84, 16, to\_date('19-NOV-21','dd-mm-yyyy'), null, 'Studying');

insert into all\_students values('Tanjim Moula', '22-41841-1', '10203040', 'CSE', 3.75,16, to\_date('29-NOV-21','dd-mm-yyyy'), null, 'Studying');

insert into all\_students values('Ali Noor', '22-41842-1', '5655952', 'CSE', 3.84,16, to\_date('12-OCT-21','dd-mm-yyyy'), null, 'Studying');

insert into all\_students values('Abid Hasan', '22-41843-1', '675412', 'CSE', 3.84,16, to\_date('08-DEC-21','dd-mm-yyyy'), null, 'Studying');

insert into all\_students values('Tanjir Mehrab', '22-41844-1', 'e3sdAw2', 'CSE', 3.59,16, to\_date('12-OCT-21','dd-mm-yyyy'), null, 'Studying');

insert into all\_students values('Prokhor Roy', '22-41845-1', '5655952', 'CSE', 3.33,16, to\_date('12-OCT-21','dd-mm-yyyy'), null, 'Studying');

insert into all\_students values('Uma Bania Archi', '22-41846-1', '\_i\_j\_k\_', 'CSE', 2.78,16, to\_date('18-NOV-21','dd-mm-yyyy'), null, 'Studying');

insert into all\_students values('Dipanwita Saha', '22-41847-1', '\_Q\_E\_T\_', 'CSE', 3.33,16, to\_date('08-SEP-21','dd-mm-yyyy'), null, 'Studying');

insert into all\_students values('Mahmudul Hasan', '22-41848-1', 'gPLpR', 'CSE', 3.63,16, to\_date('02-DEC-21','dd-mm-yyyy'), null, 'Studying');

insert into all\_students values('Minhaz Ayon', '22-41849-1', '!@pNK-+', 'CSE', 3.52,16, to\_date('02-DEC-21','dd-mm-yyyy'), null, 'Studying');

**C. Creating a table for all teachers**

create table all\_teachers (

name varchar(50) not null,

id varchar(20) not null primary key,

password varchar(30),

domain varchar(10),

hire\_date date,

balance int,

salary int

);

**D. Inserting records for all teachers**

insert into all\_teachers values('Nazmul Hossain','2001-2044-1','CSC',to\_date('17-11-2018','dd-mm-yyyy'),30000,50000 );

insert into all\_teachers values('Manawar Uddin','1803-3042-2','ECO',to\_date('02-01-2004','dd-mm-yyyy'),30000,90000 );

insert into all\_teachers values('Nurul Kabir','5295-8197-1','PHY',to\_date('02-01-2011','dd-mm-yyyy'),30000,65000 );

insert into all\_teachers values('Rethwan Faiz','3903-7071-3','EEE',to\_date('12-09-2014','dd-mm-yyyy'),30000,65000 );

insert into all\_teachers values('Foysal Chowdhury','1567-2082-1','BAS',to\_date('12-10-2017','dd-mm-yyyy'),30000,50000 );

insert into all\_teachers values('Nadia Anam','4269-6381-2','COE',to\_date('01-11-2016','dd-mm-yyyy'),30000,50000 );

insert into all\_teachers values('Nafish Sarwar Islam','3166-5315-3','BAE',to\_date('11-03-2013','dd-mm-yyyy'),65000 );

insert into all\_teachers values('Mahfuzur Rahman','2546-7356-1','MAT',to\_date('04-03-2020','dd-mm-yyyy'),30000,50000 );

insert into all\_teachers values('Mosaddeq Rahman','1160-9391-3','CHM',to\_date('14-06-2017','dd-mm-yyyy'),30000,50000 );

insert into all\_teachers values('Dip Nandi','0216-3565-1','MGT',to\_date('14-06-2002','dd-mm-yyyy'),30000,90000 );

insert into all\_teachers values('Obaidul Islam','0378-4149-3','BBA',to\_date('16-01-2005','dd-mm-yyyy'),30000,90000 );

insert into all\_teachers values('Theotonius Gomes','2173-3743-2','ENG',to\_date('26-06-2009','dd-mm-yyyy'),30000,65000);

**E. Create a table for all the courses**

create table all\_course

(

course\_code varchar(10) not null primary key,

course\_name varchar(50) not null,

course\_prerequisite\_code varchar(10),

credit\_count int

);

**F. Inserting all the courses in this table**

insert into all\_course values('PHY1102', 'Physics-1 Lab', null, 1);

insert into all\_course values('ENG1101', 'English-1', null, 3);

insert into all\_course values('CSC1101', 'ICS', null, 1);

insert into all\_course values('CSC1102', 'C++', null, 3);

insert into all\_course values('CSC1103', 'C++ Lab', null, 1);

insert into all\_course values('CSC1204', 'Discrete Math', 'CSC1102', 3);

insert into all\_course values('MAT1205', 'Math-2', 'MAT1102', 3);

insert into all\_course values('CSC1205', 'Java', 'CSC1102', 3);

insert into all\_course values('MAT1102', 'Math-1', null, 3);

insert into all\_course values('PHY1101', 'Physics-1', null, 3);

insert into all\_course values('PHY1203', 'Physics-2', 'PHY1101', 3);

insert into all\_course values('PHY1204', 'Physics-2 Lab', 'PHY1102', 1);

insert into all\_course values('ENG1202', 'English-2', 'ENG1101', 3);

insert into all\_course values('COE2101', 'Introduction To Electrical Circuits', 'PHY1101', 3);

insert into all\_course values('COE2102', 'Introduction To Electrical Circuits Lab', 'PHY1102', 1);

insert into all\_course values('CHM1101', 'Chemistry', 'PHY1203', 3);

insert into all\_course values('MAT2101', 'Math-3', 'MAT1205', 3);

insert into all\_course values('CSC2108', 'Introduction To Database', 'CSC1205', 3);

insert into all\_course values('EEE2104', 'Electronic Devices Lab', 'COE2102', 1);

insert into all\_course values('BBA1102', 'Principles Of Accounting', 'MAT1205', 3);

insert into all\_course values('EEE2103', 'Electronic Devices', 'COE2101', 3);

insert into all\_course values('CSC2106', 'Data Structure', 'CSC1204', 3);

insert into all\_course values('CSC2107', 'Data Structure Lab', 'CSC1204', 1);

insert into all\_course values('BAE2101', 'CAD', null, 1);

insert into all\_course values('CSC2211', 'Algorithms', 'CSC2106', 3);

insert into all\_course values('MAT2202', 'Math-4', 'MAT2101', 3);

insert into all\_course values('CSC2210', 'C#', 'CSC2106', 3);

insert into all\_course values('CSC2209', 'OOAD', 'CSC2108', 3);

insert into all\_course values('BAS2101', 'Bangladesh Studies', 'CSC1101', 3);

insert into all\_course values('EEE3101', 'DLC', 'EEE2103', 3);

insert into all\_course values('EEE3102', 'DLC Lab', 'EEE2104', 1);

insert into all\_course values('MAT3103', 'Statistics', 'MAT2101', 3);

insert into all\_course values('CSC3113', 'Theory Of Computation', 'CSC2211', 3);

insert into all\_course values('ECO3150', 'Principles Of Economics', 'MAT3103', 2);

insert into all\_course values('ENG2103', 'Business Communication', 'BAS2101', 3);

insert into all\_course values('MAT3101', 'Math-5', 'MAT2202', 3);

insert into all\_course values('COE3103', 'Data Communication', 'EEE3101', 3);

insert into all\_course values('COE3104', 'Microprocessor And Embedded Systems', 'EEE3101', 3);

insert into all\_course values('CSC3112', 'Software Engineering', 'CSC2209', 3);

insert into all\_course values('CSC3217', 'Artificial Intelligence', 'CSC2211', 3);

insert into all\_course values('COE3206', 'Computer Networks', 'COE3103', 3);

insert into all\_course values('COE3205', 'Computer Organization And Architecture', 'COE3104', 3);

insert into all\_course values('CSC3214', 'Operating System', 'COE3104', 3);

insert into all\_course values('CSC3215', 'Web Technologies', 'CSC3112', 3);

insert into all\_course values('EEE2216', 'Engineering Ethics', 'CSC3112', 2);

insert into all\_course values('CSC3216', 'Compiler Design', 'CSC3113', 3);

insert into all\_course values('CSC4118', 'Computer Graphics', 'CSC2211', 3);

insert into all\_course values('MGT3202', 'Engineering Management', 'EEE2216', 3);

insert into all\_course values('CSC4197', 'Research Methodology', 'EEE2216', 3);

insert into all\_course values('CSC4299', 'Thesis', 'CSC4197', 3);

insert into all\_course values('CSC4296', 'Internship', 'CSC4299', 3);

insert into all\_course values('CSC4181', 'Advance Database Management System', 'CSC2108', 3);

insert into all\_course values('CSC4162', 'Programming in Python', 'CSC3215', 3);

insert into all\_course values('CSC4232', 'Machine Learning', 'CSC3217', 3);

insert into all\_course values('COE4128', 'Digital System Design', 'COE3205', 3);

insert into all\_course values('COE4232', 'Network Security', 'COE3103', 3);

**Creating University Report table**

create table university\_report

(

year int,

semester varchar(20),

start\_date date,

end\_date date,

income float,

expenditure float,

students\_admitted int,

teachers\_recruited int,

constraint semester\_alias primary key(semester,year)

);

**Creating logged in table**

create table loggedinTable

(

id varchar(30),

password varchar(30),

user\_type varchar(20)

);

**K. Creating table CSRT which will contain courses,students,results and teachers**

create table CSRT

(

serial varchar(7) not null primary key,

course\_code varchar(20) references all\_course(course\_code),

course\_name varchar(50),

teacher\_id varchar(20) references all\_teachers(ID),

teacher\_name varchar(50),

student\_id varchar(15) not null references all\_student(id),

student\_name varchar(50),

marks int,

grade varchar(4),

status varchar(30),

semester varchar(20) not null,

year int not null,

grade\_point float,

GPA float

);

**Procedures used in this project:**

**--This procedure ‘fresher\_registration’ does the registration for new students  
   
create or replace procedure fresher\_registration**

is

student\_name varchar(50);

student\_id varchar(20);

j number(10):=0;

u\_year number(5);

u\_semester varchar(15);

cursor c\_course is select \* from all\_courses where course\_prerequisite\_code is null;

cursor c\_student is select \* from all\_students where status='Started' and credits\_completed=0 order by id asc;

BEGIN

select year,semester into u\_year,u\_semester from university\_report where serial=(select max(serial) from university\_report);

for r\_course in c\_course

loop

for r\_student in c\_student

loop

j:=j+1;

insert into csrt values

(concat('C',to\_char(j)),r\_course.course\_code,r\_course.course\_name,null,null,r\_student.id,r\_student.name,0,null,'Running',u\_semester, u\_year,null,null );

end loop;

if r\_course.course\_name='CAD' then

exit;

end if;

end loop;

update all\_students set status='Studying' where status='Started';

end;

**--This procedure ‘allocate\_teachers’ allocates teachers to registered courses**

create or replace procedure allocate\_teachers

is

ccode varchar(4);

tname varchar(40);

tid varchar(40);

s varchar(8);

cursor cx is select \* from csrt where status='Running' and teacher\_id is null;

begin

for rx in cx

loop

ccode:=substr(rx.course\_code,1,3);

s:=rx.serial;

select id into tid from all\_teachers where domain=ccode;

select name into tname from all\_teachers where domain=ccode;

--dbms\_output.put\_line(ccode ||' '||tname||' '||tid||' '||s);

update csrt set teacher\_id=tid,teacher\_name=tname where serial=s;

end loop;

end;

**--This procedure ‘update\_marks’ updates marks,grades and grade points**

create or replace procedure update\_marks(s\_id all\_students.id%type,c\_code all\_courses.course\_code%type,s\_marks int)

is

t\_id all\_teachers.id%type;

sgrade csrt.grade%type;

sgrade\_point csrt.grade\_point%type;

begin

if s\_marks>=90 and s\_marks<=100 then

sgrade\_point:=4;

sgrade:='A+';

elsif s\_marks>=85 and s\_marks<90 then

sgrade\_point:=3.75;

sgrade:='A';

elsif s\_marks>=80 and s\_marks<85 then

sgrade\_point:=3.5;

sgrade:='B+';

elsif s\_marks>=75 and s\_marks<80 then

sgrade\_point:=3.25;

sgrade:='B';

elsif s\_marks>=70 and s\_marks<75 then

sgrade\_point:=3;

sgrade:='C+';

elsif s\_marks>=65 and s\_marks<70 then

sgrade\_point:=2.75;

sgrade:='C';

elsif s\_marks>=60 and s\_marks<65 then

sgrade\_point:=2.5;

sgrade:='D+';

elsif s\_marks>=50 and s\_marks<60 then

sgrade\_point:=2.25;

sgrade:='D';

elsif s\_marks<50 then

sgrade\_point:=0;

sgrade:='F';

else

dbms\_output.put\_line('Error');

end if;

select id into t\_id from loggedinTable;

update csrt set marks=s\_marks, grade\_point=sgrade\_point,grade=sgrade where teacher\_id=t\_id and student\_id=s\_id and course\_code=c\_code and status='Running';

end;

**--This procedure ‘gpa\_and\_credits\_completed’ updates gpa and credit count**

Create or replace procedure gpa\_and\_credits\_completed

is

cursor cs is select student\_id,sum(grade\_point) sgp,semester,year from csrt group by student\_id,semester,year;

sgpa float;

ssgp float;

scredit\_count int;

sid all\_student.id%type;

syear int;

ssemester csrt.semester%type;

begin

for rs in cs

loop

ssgp:=rs.sgp;

sid:=rs.student\_id;

ssemester:=rs.semester;

syear:=rs.year;

select sum(credit\_count) into scredit\_count from all\_course where course\_code in(select course\_code from csrt where student\_id=sid and semester=ssemester and year=syear);

sgpa:=round(ssgp/scredit\_count,2);

--dbms\_output.put\_line('ID: '||sid||' SGP: '||ssgp||' Semester: '||ssemester||' Year: '||syear||' GPA: '||sgpa||' credit\_count : '||scredit\_count);

update csrt set gpa=sgpa,status='Completed' where student\_id=sid and semester=ssemester and year=syear;

update all\_student set credits\_completed=credits\_completed+scredit\_count where id=sid;

end loop;

end;

**--This procedure ‘set\_cgpa’ updates cgpa**

Create or replace procedure set\_cgpa

is

sID all\_students.id%type;

cursor CA is select id from all\_students;

begin

for RA in CA

loop

--dbms\_output.put\_line(RA.id);

declare

nSGPA courses\_and\_students2.gpa%type;

nCGPA courses\_and\_students2.gpa%type;

nSID all\_students.id%type;

nCount number(4);

cursor CB is select student\_id, round((avg(grade\_point)),2) G\_P\_A,semester,year from courses\_and\_students2 where student\_id=RA.id and status='Completed' group by student\_id,semester,year;

begin

nCount:=0;

nSGPA:=0;

for RB in CB

loop

nSGPA:=nSGPA+RB.G\_P\_A;

nCount:=nCount+1;

end loop;

nCGPA:=nSGPA/nCount;

dbms\_output.put\_line('Student ID :'|| RA.id || ' CGPA :'||nCGPA);

update all\_students set CGPA=nCGPA where id=RA.id;

end;

end loop;

end;

**--This procedure ‘set\_loggedinTable\_null’ clears the loggedinTable**

create or replace procedure set\_loggedinTable\_null

is

begin

update loggedinTable set id=null,password=null,user\_type=null;

end;

**Views used in this project:**

create view csrt\_this\_teacher as select course\_code,course\_name,student\_id,student\_name,marks from csrt where teacher\_id=(select id from loggedIntable) and status='Running';

create view csrt\_this\_student as select course\_name,marks,nvl(grade,'Blank') grade,teacher\_name,semester,year from csrt where student\_id=(select id from loggedinTable) order by serial asc;

create view this\_semester\_report as select \* from university\_report where serial=(select max(serial) from university\_report );

create view admin\_viewing\_csrt as select student\_name,course\_name,teacher\_name,marks,grade,status,semester,year from csrt;

**Triggers used in this project:**

create or replace trigger expenditure\_recorder

before update on all\_teachers

for each row

begin

update this\_semester\_report set expenditure=expenditure+:new.balance-:old.balance;

end;

create or replace trigger marks\_value\_fixer

before update of marks on csrt

for each row

begin

if :new.marks>100 then

:new.marks:=100;

elsif :new.marks<0 then

:new.marks:=0;

end if;

end;

create or replace trigger result\_log

after update on csrt

begin

insert into task\_log values((select id from loggedinTable),'Updating results',sysdate,SYSTIMESTAMP);

end;

Our project can be downloaded from this link.

[https](https://drive.google.com/drive/folders/1GQ8-5bixjWSjXJiyIBUQADd1CXjRjEtz)://drive.google.com/drive/folders/1GQ8-5bixjWSjXJiyIBUQADd1CXjRjEtz