-- A001 기초 정보 관리 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- 하위 선택지로 ‘1. 과정, 2. 과목, 3강의실, 4.교재’

-- A001-1 과정 조회, 추가, 수정, 삭제

-- A001-1-1 과정 조회

select

\*

from tblBasicCourseInfo

order by seqbasiccourseinfo;

-- A001-1-2 과정 추가 - Procedure

create or replace procedure procAddBasicCourseInfo(

pname varchar2, -- 과정명

pperiod number, -- 과정기간(단위 : 일)

pinfo varchar2 -- 과정소개

)

is

begin

insert into tblBasicCourseInfo (seqBasicCourseInfo, name, period, info)

values (seqBasicCourseInfo.nextVal, pname, pperiod, pinfo);

commit;

exception

when others then

rollback;

end;

-- A001-1-3 과정 수정 - Procedure

create or replace procedure procUpdateBasicCourseInfo(

pseqbasiccourseinfo number, -- 수정할 과정번호

pname varchar2, -- 수정할 과정명

pperiod number, -- 수정할 과정기간(단위 : 일)

pinfo varchar2 -- 수정할 과정소개

)

is

begin

update tblBasicCourseInfo

set name = pname,

period = pperiod,

info = pinfo

where seqBasicCourseInfo = pseqbasiccourseinfo;

commit;

exception

when others then

rollback;

end;

-- A001-1-4 과정 삭제 - Procedure

create or replace procedure procDeleteBasicCourseInfo(

pseqbasiccourseinfo number -- 삭제할 과정번호

)

is

begin

delete from tblBasicCourseInfo

where seqBasicCourseInfo = pseqbasiccourseinfo;

commit;

exception

when others then

rollback;

end;

begin

procDeleteBasicCourseInfo(1);

end;

select \* from tblBasicCourseInfo;

-- A001-2-1 과목 조회 - View

select \* from tblBasicSubject order by seqBasicSubject;

select \* from tblBook order by seqBook;

create or replace view vwBasicSubject

as

select

bs.seqbasicsubject,

bs.name,

bs.info,

b.name as book

from

tblBasicSubject bs

inner join tblBook b

on bs.seqBook = b.seqBook

order by bs.seqBasicSubject;

-- A001-2-2 과목 추가 - Procedure

create or replace procedure procAddBasicSubject(

pname varchar2, -- 과목명

pinfo varchar2, -- 과목소개

pseqbook number -- 교재번호

)

is

begin

insert into tblBasicSubject (seqBasicSubject, seqBook, name, info)

values (seqBasicSubject.nextVal, pseqBook, pname, pinfo);

commit;

exception

when others then

rollback;

end;

-- A001-2-3 과목 수정 - Procedure

create or replace procedure procUpdateBasicSubject(

pseqBasicSubject number, -- 과목번호

pseqbook number, -- 교재번호

pname varchar2, -- 과목명

pinfo varchar2 -- 과목소개

)

is

begin

update tblBasicSubject

set seqbook = pseqbook,

name = pname,

info = pinfo

where seqBasicSubject = pseqBasicSubject;

commit;

exception

when others then

rollback;

end;

-- 확인

begin

procupdatebasicsubject(1,1,'연습','연습');

end;

select \* from tblBasicSubject;

-- A001-2-4 과목 삭제

create or replace procedure procDeleteBasicSubject (

pseqBasicSubject number

)

is

begin

delete from tblBasicSubject

where seqBasicSubject = pseqBasicSubject;

commit;

exception

when others then

rollback;

end;

begin

procDeleteBasicSubject(2);

end;

select \* from tblBasicSubject;

-- A001-3-1 강의실 조회

select

\*

from tblRoom

order by seqRoom;

-- A001-3-2 강의실 추가 - Procedure

create or replace procedure procAddRoom(

pname varchar2, -- 강의실명

ppeople number -- 수용인원

)

is

begin

insert into tblRoom (seqRoom, name, people)

values (seqRoom.nextVal, pname, ppeople);

commit;

exception

when others then

rollback;

end;

-- A001-3-3 강의실 수정 - Procedure

create or replace procedure procUpdateRoom(

pseqroom number, -- 수정할 강의실번호

pname varchar2, -- 수정할 강의실명

ppeople number -- 수정할 수용인원

)

is

begin

update tblRoom

set name = pname,

people = ppeople

where seqRoom = pseqRoom;

commit;

exception

when others then

rollback;

end;

-- A001-3-4 강의실 삭제 - Procedure

create or replace procedure procDeleteRoom(

pseqroom number -- 삭제할 강의실번호

)

is

begin

delete from tblRoom

where seqRoom = pseqroom;

commit;

exception

when others then

rollback;

end;

begin

procDeleteRoom(1);

end;

seleCt \* from tblROOM;

-- A001-4-1 교재 조회

select

\*

from tblBook

order by seqBook;

-- A001-4-2 교재 추가 - Procedure

create or replace procedure procAddBook(

pname varchar2, -- 책이름

ppublisher varchar2,-- 출판사

pwriter varchar2 -- 저자

)

is

begin

insert into tblBook (seqBook, name, publisher, writer)

values (seqBook.nextVal, pname, ppublisher, pwriter);

commit;

exception

when others then

rollback;

end;

-- A001-4-3 교재 수정 - Procedure

create or replace procedure procUpdateBook(

pname varchar2, -- 수정할 책이름

ppublisher varchar2,-- 수정할 출판사

pwriter varchar2 -- 수정할 저자

)

is

begin

update tblBook

set name = pname,

publisher = ppublisher,

writer = pwriter

where seqBook = pseqbook;

commit;

exception

when others then

rollback;

end;

-- A001-4-4 교재 삭제 - Procedure

create or replace procedure procDeleteBook(

pseqbook number -- 삭제할 교재번호

)

is

begin

delete from tblBook

where seqBook = pseqbook;

commit;

exception

when others then

rollback;

end;

--A003 교사 계정 관리

-- 1. 전체 교사 목록 출력 - 교사번호, 이름, 주민번호, 전화번호, 가능 과목

create or replace view vwTeacher

as

select

t.seqTeacher as seqTeacher,

t.name as name,

t.ssn as ssn,

t.tel as tel,

bs.name as availableSubject

from tblTeacher t

inner join tblAvailableSubject avs

on t.seqTeacher = avs.seqTeacher

inner join tblBasicSubject bs

on avs.seqBasicSubject = bs.seqBasicSubject

order by t.seqTeacher;

select \* from vwTeacher;

-- 2.1. 교사 등록

create or replace procedure procAddTeacher(

pname varchar2,

pssn varchar2,

ptel varchar2,

pseqTeacher out number

)

is

vid tblTeacher.id%type;

begin

select id into vid from tblTeacher where seqTeacher = (select max(seqTeacher) from tblTeacher where id is not null); -- 마지막 id 가져오기

vid := 'T' || ltrim(to\_char(to\_number(substr(vid, 2)) + 1, '000000')); -- 마지막 id에서 +1 -> 새로 추가될 교사의 id -> format : T000000

insert into tblTeacher (seqTeacher, name, id, ssn, tel)

values (seqTeacher.nextVal, pname, vid, pssn, ptel); -- 교사 테이블에 등록

select max(seqTeacher) into pseqTeacher from tblTeacher; -- 추가된 교사번호 반환 -> 가능 과목 등록에 사용

commit;

exception

when others then

rollback;

end procAddTeacher;

-- 2.2. 가능 과목 등록

create or replace procedure procAddAvailableSubject(

pseqTeacher number,

pseqBasicSubject number

)

is

begin

if pseqBasicSubject = 0 then -- 가능 과목 0 입력한 경우

return;

end if;

insert into tblAvailableSubject (seqAvailableSubject, seqTeacher, seqBasicSubject)

values (seqAvailableSubject.nextVal, pseqTeacher, pseqBasicSubject); -- 가능 과목 테이블에 등록

commit;

exception

when others then

rollback;

end procAddAvailableSubject;

-- 2.3. 교사 및 가능 과목 등록 실행 예

declare

vseqTeacher number;

begin

procAddTeacher('홍길동', '800407-1131592', '010-1234-5678', vseqTeacher); -- 이름, 주민번호, 전화번호, 교사번호(out)

procAddAvailableSubject(vseqTeacher, 1); -- 교사번호, 과목번호

procAddAvailableSubject(vseqTeacher, 2);

procAddAvailableSubject(vseqTeacher, 0);

end;

-- 3.1. 특정 교사 정보 조회

--배정된 개설 과목명, 개설 과목기간(시작, 끝), 과정명, 개설 과정기간(시작, 끝), 교재명, 강의실, 강의진행여부

create or replace view vwSpecificTeacher

as

select

t.seqTeacher as seqTeacher,

t.name as name,

bs.name as subject,

os.startDate as subjectStartDate,

os.endDate as subjectEndDate,

bc.name as course,

oc.startDate as courseStartDate,

oc.endDate as courseEndDate,

b.name as book,

r.name as room,

case

when oc.startDate > sysdate then '강의전'

when oc.startDate < sysdate and oc.endDate > sysdate then '강의중'

when oc.endDate < sysdate then '강의종료'

end as lectureState

from tblTeacher t

inner join tblAssignTeacher at

on t.seqTeacher = at.seqTeacher

inner join tblOpenCourse oc

on at.seqOpenCourse = oc.seqOpenCourse

inner join tblOpenSubject os

on oc.seqOpenCourse = os.seqOpenCourse

inner join tblBasicSubject bs

on os.seqBasicSubject = bs.seqBasicSubject

inner join tblBasicCourseInfo bc

on oc.seqBasicCourseInfo = bc.seqBasicCourseInfo

inner join tblBook b

on bs.seqBook = b.seqBook

inner join tblRoom r

on oc.seqRoom = r.seqRoom

order by t.seqTeacher, os.startDate;

-- 3.2. 특정 교사 조회 예

select

\*

from vwSpecificTeacher

where seqTeacher = 1;

-- 4.1. 교사 정보 수정

select \* from vwTeacher;

create or replace procedure procUpdateTeacher(

pseqTeacher number,

pname varchar2,

pssn varchar2,

ptel varchar2

)

is

begin

update tblTeacher set name = pname, ssn = pssn, tel = ptel where seqTeacher = pseqTeacher;

commit;

exception

when others then

rollback;

end procUpdateTeacher;

-- 4.1.1. 교사 정보 수정 예

begin

procUpdateTeacher(1, '김찬승', '770203-1511312', '010-5112-3219');

end;

-- 4.2. 강의 가능 과목 수정

-- 4.2.1. 전체삭제 후 전체추가

create or replace procedure procDeleteAvailableSubject(

pseqTeacher number

)

is

begin

delete from tblAvailableSubject where seqTeacher = pseqTeacher;

commit;

exception

when others then

rollback;

end procDeleteAvailableSubject;

-- 4.2.2. 강의 가능 과목 수정 예 (삭제 후 추가)

begin

procDeleteAvailableSubject(11);

procAddAvailableSubject(11, 1);

procAddAvailableSubject(11, 2);

procAddAvailableSubject(11, 3);

procAddAvailableSubject(11, 0);

end;

-- 4.3.1. 특정 과목 선택하여 수정

create or replace procedure procUpdateAvailableSubject(

pseqTeacher number,

ppreseqBasicSubject number,

pnewseqBasicSubject number

)

is

begin

update tblAvailableSubject set seqBasicSubject = pnewseqBasicSubject where seqTeacher = pseqTeacher and seqBasicSubject = ppreseqBasicSubject;

commit;

exception

when others then

rollback;

end procUpdateAvailableSubject;

-- 4.3.2. 특정 과목 수정 예

declare

vseqTeacher number;

vpreseqBasicSubject number;

vnewseqBasicSubject number;

begin

vseqTeacher := 1;

vpreseqBasicSubject := 15;

vnewseqBasicSubject := 14;

procUpdateAvailableSubject(vseqTeacher, vpreseqBasicSubject, vnewseqBasicSubject);

end;

-- 5.1. 교사 정보 삭제

-- 5.1.1. 강의중인지 확인하기

create or replace procedure procLectureState(

pseqTeacher number,

presult out number

)

is

-- 강의중인 과정인지?

cursor vcursor is select at.seqTeacher as seqTeacher

from tblTeacher t

inner join tblAssignTeacher at

on t.seqTeacher = at.seqTeacher

inner join tblOpenCourse oc

on at.seqOpenCourse = oc.seqOpenCourse

where sysdate between oc.startDate and oc.endDate;

vseqTeacher number;

begin

open vcursor;

loop

fetch vcursor into vseqTeacher;

if vseqTeacher is not null then

presult := 1; -- 강의중

else

presult := 0; -- 강의중이지 않음

end if;

exit when vcursor%notfound;

end loop;

close vcursor;

end procLectureState;

-- 5.1.2. 교사 삭제

create or replace procedure procDeleteTeacher(

pseqTeacher number

)

is

vresult number;

begin

-- 강의중이지 않은 교사만 삭제 가능

procLectureState(pseqTeacher, vresult); -- vresult = 0 -> 강의중이지 않은 개설과정

if vresult = 0 then

delete from tblAvailableSubject where seqTeacher = pseqTeacher;

delete from tblAssignTeacher where seqTeacher = pseqTeacher;

--delete from tblTeacher where seqTeacher = pseqTeacher; -- 완전 삭제하기

update tblTeacher set id = null, ssn = null, tel = null where seqTeacher = pseqTeacher; -- 교사번호, 이름 남기고 나머지 항목 null로 변경

else

-- 강의중이면 삭제X

dbms\_output.put\_line('교사는 강의중입니다.');

end if;

commit;

exception

when others then

rollback;

end procDeleteTeacher;

-- 5.2. 교사 정보 삭제 예

declare

vseqTeacher number;

begin

vseqTeacher := 2;

procDeleteTeacher(vseqTeacher);

end;

--A006, A007 시험관리, 성적 관리

create or replace View vwStudentTestScore

as

select

ts.seqTestPercent as testSeq,

s.seqStudent as stuSeq,

s.name as studentName,

s.ssn as studentSsn,

bci.name as courseName,

oc.startDate as courseStart,

oc.EndDate as courseEnd,

os.seqOpenSubject,

bs.name as subjectName,

os.startDate as subjectStart,

os.endDate as subjectEnd,

t.name as teacherName,

ts.writtenScore,

ts.practicalScore,

ts.attendanceScore,

tp.writtenPercent,

tp.practicalPercent,

tp.attendancePercent,

rc.studentState,

r.name as roomName,

t.seqTeacher as tSeq

from tblOpenSubject os

right outer join tblAvailableSubject asj

on asj.seqAvailableSubject = os.seqAvailableSubject

right outer join tblTeacher t

on t.seqTeacher = asj.seqTeacher

left outer join tblBasicSubject bs

on bs.seqBasicSubject = asj.seqBasicSubject

left outer join tblBasicTest bt

on bt.seqOpenSubject = os.seqOpenSubject

left outer join tblTestPercent tp

on tp.seqBasicTest = bt.seqBasicTest

left outer join tblTestScore ts

on ts.seqTestPercent = tp.seqTestPercent

left outer join tblRegCourse rc

on rc.seqRegCourse = tp.seqRegCourse

right outer join tblStudent s

on s.seqStudent = rc.seqStudent

left outer join tblopencourse oc

on oc.seqopencourse = os.seqopencourse

left outer join tblBasicCourseInfo bci

on bci.seqBasicCourseInfo = oc.seqBasicCourseInfo

inner join tblroom r

on r.seqRoom = oc.seqRoom;

-- A008 출결 관리 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- A008-1 과정별 출결 조회

-- 모든 개설 과정을 조회한다. (과정번호, 과정명, 시작날짜, 종료날짜, 강의실명, 교사명, 수강인원, 개설과목등록여부)

create or replace procedure procCourseTotList(

pcursor out sys\_refcursor

)

is

begin

open pcursor for

select

oc.seqopencourse, -- 과정번호

bci.name, -- 과정명

to\_char(oc.startdate, 'yyyy-mm-dd') startDate, -- 과정 시작날짜

to\_char(oc.enddate, 'yyyy-mm-dd') endDate, -- 과정 종료날짜

r.name roomName, -- 강의실명

t.name teacherName, -- 교사명

oc.membercount || '명' capacity, -- 수강인원

case

when oc.courseRegistrationState = 't' then '개강'

when oc.courseRegistrationState = 'f' then '종료'

end courseRegState -- 개설과목등록여부

from tblOpenCourse oc

inner join tblBasicCourseInfo bci

on oc.seqBasicCourseInfo = bci.seqBasicCourseInfo

inner join tblRoom r

on oc.seqroom = r.seqRoom

inner join tblAssignTeacher at

on oc.seqOpenCourse = at.seqOpenCourse

inner join tblTeacher t

on t.seqTeacher = at.seqTeacher;

end;

-- A008-1-2 특정 개설 과정의 전체 교육생 리스트 조회

-- 특정 개설과정의 번호를 입력 받았을 때 그 개설과정을 수강하는 교육생 전체 목록을 출력한다.

-- 교육생번호, 교육생이름, 교육생ID, 교육생PW(주민번호뒷자리), 교육생등록일, 교육생수강상태

create or replace procedure procStudentlist(

pcursor out sys\_refcursor,

pseqOpenCourse number -- 개설과정번호

)

is

begin

open pcursor for

select

rc.seqstudent,

s.name,

s.id,

substr(s.ssn, 8, 7) pw,

to\_char(s.firstRegistrationDate, 'yyyy-mm-dd') regDate,

case

when rc.studentState in ('수료', '중도탈락') then rc.studentState || '(' || to\_char(rc.failOrDoneDate,'yyyy-mm-dd') || ')'

else rc.studentState

end studentState

from tblRegCourse rc

inner join tblStudent s

on s.seqStudent = rc.seqStudent

inner join tblOpenCourse oc

on oc.seqopencourse = rc.seqopencourse

where oc.seqOpenCourse = pseqOpenCourse; -- 특정 과정번호 입력

-- where oc.seqOpenCourse = 1; -- 1번 과정번호 입력

end;

-- A008-1-4 특정 개설 과정의 특정 교육생 특정 기간 출결 리스트 조회

-- 교육생번호, 교육생이름, 입실시간, 퇴실시간, 근태상태

create or replace procedure procAttPeriodList(

pcursor out sys\_refcursor,

pseqOpenCourse number, -- 입력한 개설 과정 번호

pseqRegCourse number, -- 입력한 수강번호

pstartDate varchar2, -- 입력한 특정기간(시작일)

pendDate varchar2 -- 입력한 특정기간(종료일)

)

is

begin

open pcursor for

select

to\_char(vw.CourseDate, 'yyyy-mm-dd') attendDate, -- 교육생 수강날짜

to\_char(a.inTime, 'hh24:mi:ss') inTime, -- 입실시간

to\_char(a.outTime, 'hh24:mi:ss') outTime, -- 퇴실시간

case

when a.attendState is null and h.name is not null then h.name -- 공휴일

when to\_char(vw.CourseDate, 'd') in (1, 7) then '휴일'

when a.attendState is null then '결석'

else a.attendState

end attendState -- 출결상태

from tblRegCourse rc

inner join tblOpenCourse oc

on rc.seqOpenCourse = oc.seqOpenCourse

-- and rc.seqRegCourse = 21 -- 수강번호 21번만 보기 (특정 교육생)

and rc.seqRegCourse = pseqRegCourse -- 특정 수강번호 (특정 교육생)

inner join tblBasicCourseInfo bci

on bci.seqBasicCourseInfo = oc.seqBasicCourseInfo

inner join tblStudent s

on rc.seqStudent = s.seqStudent

inner join tblAttendance a

on a.seqRegCourse = rc.seqRegCourse

right outer join (select -- 개설과정테이블에서 개설과정번호 2번인 과정의 (수업시작일 - 종료일)을 가져와 모든 날짜를 출력하는 서브쿼리

(select startDate

from tblOpenCourse where seqOpenCourse = /\*2\*/pseqOpenCourse) + level - 1 as courseDate

from dual

connect by level < (select

to\_date((case

when endDate < sysdate then endDate -- 수강종료일이 현재날짜보다 이후라면 현재날짜까지만 출력한다.

else sysdate

end), 'yyyy-mm-dd')

from tblOpenCourse

where seqOpenCourse = /\*2\*/pseqOpenCourse) - (select

to\_date(startDate, 'yyyy-mm-dd')

from tblOpenCourse

where seqOpenCourse = /\*2\*/pseqOpenCourse) + 2) vw

on to\_date(vw.CourseDate, 'yy-mm-dd') = to\_date(a.attendanceDate, 'yy-mm-dd')

left outer join tblHoliday h

on vw.courseDate = h.regDate

-- where vw.courseDate between '2020-11-01' and '2020-12-01' -- 특정기간 (2020년 11월 1일~ 12월 1일)

where vw.courseDate between pstartDate and pendDate

order by vw.courseDate;

end;

-- A008-1-5 특정 개설 과정의 특정 교육생 출결 상태 수정 (ex) 결석 -> 병가)

-- 교육생번호, 교육생이름, 입실시간, 퇴실시간, 출결상태

create or replace procedure procUpdateAttState(

pcursor out sys\_refcursor,

pseqRegCourse number, -- 입력한 수강번호

pattendanceDate varchar2, -- 입력한 출결 날짜

pattendState varchar2 -- 수정할 출결 상태

)

is

begin

update tblAttendance

set attendState = pattendState

where seqRegCourse = pseqRegCourse and attendanceDate = pattendanceDate;

end;

/\*A-009 관리자 연계기업 채용공고관리\*/

----------------------A-009 등록된 전체 채용공고 조회

create view vwCompanyInfo

as

select

seqcompanyinfo, --번호

name, --기업명

to\_char(startdate,'yyyy-mm-dd') as startdate, --채용시작일

to\_char(enddate,'yyyy-mm-dd') as enddate, --채용종료일

comfield, --채용분야

salary, --연봉

employmenttype, --채용형태

comsize, --기업규모

address, --주소

case

when startdate > sysdate then '채용예정'

when startdate <= sysdate and enddate > sysdate then '채용중'

when enddate < sysdate then '채용마감'

end as state --채용상태

from tblCompanyInfo

order by startdate desc,enddate desc;

--------------------------A005 소재지별 취업공고 조회, //교육생과같음

create or replace procedure procCityCompany(

paddress vwCompanyInfo.address%type,

pcursor out sys\_refcursor

)

is

begin

open pcursor

for select \* from vwcompanyinfo where address like '%' || paddress || '%';

end;

-----------------------------------A005 업무별 채용공고 조회, //교육생과같음

create or replace procedure procFieldCompany(

pcomField vwCompanyInfo.comField%type,

pcursor out sys\_refcursor

)

is

begin

open pcursor

for select \* from vwCompanyInfo where comField like '%' ||pcomField||'%';

end;

--------------------------------A005 연봉별 채용공고 조회, //교육생과같음

create or replace procedure procSalaryCompany(

psalary in vwCompanyInfo.salary%type, ---첫번째 입력값 ex)20000000이상

psalary2 in vwCompanyInfo.salary%type, -- 두번째 입력값 ex)30000000미만

pcursor out sys\_refcursor

)

is

begin

open pcursor

for select \* from vwcompanyinfo where salary >=psalary and salary <psalary2;

end;

-------------------------------A-009 1.채용공고 등록

create or replace procedure procAddCompany(

pname tblcompanyinfo.name%type, --회사이름

pstartDate tblcompanyinfo.startdate%type, --채용시작일

pendDate tblcompanyinfo.enddate%type, --채용종료일

pcomField tblcompanyinfo.comfield%type, --취업분야

psalary tblcompanyinfo.salary%type, -- 연봉

pemploymentType tblcompanyinfo.employmenttype%type, --고용형태

pcomSize tblcompanyinfo.comsize%type, --회사규모

paddress tblcompanyinfo.address%type --주소

)

is

begin

-- 채용종료일이 현재날짜보다 이전이면 등록되지 않는다.

if(pendDate < sysdate)

then

dbms\_output.put\_line('마감된 공고는 등록할 수 없습니다.');

rollback;

---- 채용시작일이 채용마감일보다 늦을 수 없다

elsif (pstartdate>penddate)

then

dbms\_output.put\_line('채용시작일이 종료일보다 늦을 수 없습니다.');

rollback;

else

insert into tblCompanyInfo(seqCompanyInfo,name,startDate,endDate,comField,salary,employmentType,comSize,address)

values (seqCompanyInfo.nextVal,pname,pstartDate,pendDate,pcomField,psalary,pemploymentType,pcomSize,paddress);

commit;

end if;

exception

when others then

rollback;

end;

--------------------------------------------A-009 2.채용공고 수정

create or replace procedure procUpdateCompany(

pseq in number, --수정할 기업번호

pname tblcompanyinfo.name%type, --기업명

pstartDate tblcompanyinfo.startdate%type, --채용시작일

pendDate tblcompanyinfo.enddate%type, --채용종료일

pcomField tblcompanyinfo.comfield%type, --취업분야

psalary tblcompanyinfo.salary%type, -- 연봉

pemploymentType tblcompanyinfo.employmenttype%type, --고용형태

pcomSize tblcompanyinfo.comsize%type, --회사규모

paddress tblcompanyinfo.address%type --주소

)

is

begin

-- 채용종료일이 현재날짜보다 이전이면 수정되지 않는다.

if(pendDate < sysdate)

then

dbms\_output.put\_line('마감된 공고는 등록할 수 없습니다.');

rollback;

-- 채용시작일이 채용마감일보다 늦을 수 없다.

elsif (pstartdate>penddate)

then

dbms\_output.put\_line('채용시작일이 종료일보다 늦을 수 없습니다.');

rollback;

else

update tblCompanyInfo

set name=pname,

startDate=pstartdate,

endDate=penddate,

comField=pcomfield,

salary=psalary,

employmentType=pemploymentType,

comSize=pcomsize,

address=paddress

where seqCompanyInfo = pseq;

commit;

end if;

exception

when others then

rollback;

end;

-------------------------------------------A-009 3. 연계기업채용공고 삭제

create or replace procedure procDeleteCompany(

pseq in number --삭제할 기업번호

)

is

begin

update tblCompanyInfo

set name=null, -----null값으로 변경

startDate=null,

endDate=null,

comField=null,

salary=null,

employmentType=null,

comSize=null,

address=null

where seqCompanyInfo = pseq;

commit;

exception

when others then

rollback;

end;

/\* A-0010관리자 수료생취업현황관리\*/

------------------------------------------A-0010 최신 취업일별 수료생 전체취업현황조회

create view vwGetJobInfoA

as

select

gj.seqgetjobinfo as gjseq,

s.name as name, --이름

s.id as id, --학생id

gj.name as companyname, --회사명

gj.duty as duty, --업무

gj.form as form, --채용형태

gj.salary as salary, --연봉

to\_char(gj.getjobdate,'yyyy-mm-dd') as getjobdate, --취업일

gj.location as location, --회사주소

rc.seqregcourse as rcseq, --과정번호

bc.name as course --수료한과정명

from tblGetJobInfo gj

inner join tblRegCourse rc

on gj.seqregcourse = rc.seqregcourse

inner join tblStudent s

on rc.seqstudent =s.seqstudent

inner join tblOpenCourse oc

on rc.seqopencourse = oc.seqopencourse

inner join tblBasicCourseinfo bc

on oc.seqbasiccourseinfo = bc.seqbasiccourseinfo

order by gj.getjobdate desc;

------------------------------------A-0010 소재지별 수료자취업정보조회

create or replace procedure procCityGetJobA(

plocation vwgetjobinfoa.location%type,

pcursor out sys\_refcursor

)

is

begin

open pcursor

for select \* from vwgetjobinfoa where location like '%' || plocation || '%';

end;

---------------------------------A-0010 업무별 수료자 취업정보 조회

create or replace procedure procDutyGetJobA( --에러뜨지만.. 실행됨

pduty vwgetjobinfoa.duty%type,

pcursor out sys\_refcursor

)

is

begin

open pcursor

for select \* from vwgetjobinfoa where duty like '%' ||pduty||'%';

end;

--------------------------------------------A-0010 연봉별 수료자 취업정보 조회

create or replace procedure procSalaryGetJobA(

psalary vwgetjobinfoa.salary%type,

psalary2 vwgetjobinfoa.salary%type,

pcursor out sys\_refcursor

)

is

begin

open pcursor

for select \* from vwgetjobinfoa where salary >= psalary and salary < psalary2;

end;

------------------------------------A-0010 1.수료생 취업현황 등록

create view vwCompleteStudent --수료한 사람들 전체

as

select rc.seqregcourse as rcseq, s.name as sname, s.id as id , bc.name as course from tblregcourse rc

inner join tblStudent s

on rc.seqstudent = s.seqstudent

inner join tblopencourse oc

on rc.seqopencourse = oc.seqopencourse

inner join tblbasiccourseinfo bc

on oc.seqbasiccourseinfo = bc.seqbasiccourseinfo

where rc.studentstate ='수료'

order by s.name;

create view vwGetJobStudent --수료한 사람들 중 취업등록이되어있는사람

as

select rc.seqregcourse as rcseq, s.name as sname, s.id as id ,bc.name as course

from tblgetjobinfo gj

inner join tblregcourse rc

on gj.seqregcourse = rc.seqregcourse

inner join tblstudent s

on rc.seqstudent = s.seqstudent

inner join tblopencourse oc

on rc.seqopencourse = oc.seqopencourse

inner join tblbasiccourseinfo bc

on oc.seqbasiccourseinfo = bc.seqbasiccourseinfo

where rc.studentstate ='수료'

order by s.name;

--수료한 사람들 중 취업등록이 안되어있는 학생 조회

select \* from vwCompleteStudent minus select \* from vwGetJobStudent;

create or replace procedure procAddGetJobInfo(

pseqRegCourse in number, --수강정보번호

plocation tblgetjobinfo.location%type, --회사주소

pname tblgetjobinfo.name%type, --회사명

pduty tblgetjobinfo.duty%type, --직무

psalary tblgetjobinfo.salary%type, --연봉

pform tblgetjobinfo.form%type, --고용형태

pgetJobDate tblgetjobinfo.getjobdate%type --취업일

)

is

vstudentstate tblregcourse.studentstate%type;

cursor vcursor is select seqregcourse from tblregCourse where studentstate = '수료' ;

begin

open vcursor;

loop

fetch vcursor into vstudentstate;

exit when vcursor%notfound;

end loop;

close vcursor;

if pseqregcourse = vstudentstate then-- 수료한학생의 수강정보번호와 입력한 번호가 일치해야 등록가능.

insert into tblGetJobInfo (seqGetJobInfo,seqRegCourse,location,name,duty,salary,form,getJobDate)

values (seqGetJobInfo.nextVal,pseqRegCourse,plocation,pname,pduty,psalary,pform,pgetJobDate);

commit;

else

dbms\_output.put\_line('아직 과정을 수료하지 않은 학생입니다');

rollback;

end if;

exception

when others then

rollback;

end;

------------------------------------A-0010 2. 수료생 취업현황 수정

create or replace procedure procUpdateGetJobInfo(

pseq in number, --수정할번호

plocation tblgetjobinfo.location%type, --회사주소

pname tblgetjobinfo.name%type, --회사명

pduty tblgetjobinfo.duty%type, --직무

psalary tblgetjobinfo.salary%type, --연봉

pform tblgetjobinfo.form%type, --고용형태

pgetJobDate tblgetjobinfo.getjobdate%type --취업일

)

is

begin

update tblGetJobInfo

set location = plocation,

name = pname,

duty = pduty,

salary = psalary,

form = pform,

getJobDate = pgetJobDate

where seqGetJobInfo = pseq;

commit;

exception

when others then

rollback;

end;

---------------------------------------------A-0010 3.수료생 취업현황 삭제

create or replace procedure procDeleteGetJobInfo(

pseq in number --삭제할번호

)

is

begin

update tblGetJobInfo -----null값으로 변경

set seqRegCourse = null ,

location = null ,

name = null,

duty = null,

salary =null,

form = null,

getJobDate =null

where seqGetJobInfo = pseq;

commit;

exception

when others then

rollback;

end;

-------------------------------A-0010 4.관리자 수료생 연계기업취업현황 조회

create view vwAllEmpStatus

as

select

es.seqempstatus as seq, ----번호

s.name as name, --이름

s.id as id, --수료생아이디

gj.name as companyname, --회사명

gj.duty as duty, --업무

gj.form as form, --채용형태

gj.salary as salary, --연봉

to\_char(gj.getjobdate,'yyyy-mm-dd') as getjobdate, --취업일

gj.location as location, --회사주소

bc.name as course --수료한과정명

from tblEmpStatus es

inner join tblcompanyinfo ci

on es.seqcompanyinfo = ci.seqcompanyinfo

inner join tblgetjobinfo gj

on es.seqgetjobinfo = gj.seqgetjobinfo

inner join tblregcourse rc

on es.seqregcourse = rc.seqregcourse

inner join tblstudent s

on rc.seqstudent = s.seqstudent

inner join tblopencourse oc

on rc.seqopencourse = oc.seqopencourse

inner join tblbasiccourseinfo bc

on oc.seqbasiccourseinfo = bc.seqbasiccourseinfo

order by gj.getjobdate desc;

-------------------------- A-0010 4.1 관리자 수료생 연계기업취업현황 등록

select \* from tblcompanyinfo; --연계기업 조회,(원하는 연계기업의 번호입력할 수있게해야한다.)

select gj.seqgetjobinfo as gjseq, rc.seqregcourse as rcseq, s.name as sname, gj.name as cname, gj.getjobdate as getjobdate, bc.name as course

from tblgetjobinfo gj

inner join tblregcourse rc

on gj.seqregcourse = rc.seqregcourse

inner join tblstudent s

on rc.seqstudent = s.seqstudent

inner join tblopencourse oc

on rc.seqopencourse = oc.seqopencourse

inner join tblbasiccourseinfo bc

on oc.seqbasiccourseinfo = bc.seqbasiccourseinfo

order by gj.getjobdate desc;--취업현황 조회, (연계기업번호와 같은 이름의 기업의 취업현황번호와 수강정보번호 선택입력할 수 있게 해야한다.)

create view vwEmpStatus -- 연계기업취업등록할때 이미 등록이 되어있는 사람은 빼고 등록할수 있도록 출력해주는 것..

as

select gjseq, rcseq, name, id, companyname, getjobdate, location from vwgetjobinfoa

minus

select

gj.seqgetjobinfo as gjseq,

rc.seqregcourse as rcseq,

s.name as name,

s.id as id,

gj.name as companyname,

to\_char(gj.getjobdate,'yyyy-mm-dd') as getjobdate,

gj.location as location

from tblEmpStatus es

inner join tblRegCourse rc

on es.seqregcourse = rc.seqregcourse

inner join tblgetjobinfo gj

on es.seqgetjobinfo = gj.seqgetjobinfo

inner join tblStudent s

on rc.seqstudent = s.seqstudent;

-------------------------등록 프로시저

create or replace procedure procAddEmpStatus(

pseqcompanyinfo in number,

pseqgetjobinfo in number,

pseqregcourse in number

)

is

begin

insert into tblempstatus (seqempstatus, seqcompanyinfo, seqgetjobinfo, seqregcourse)

values(seqempstatus.nextVal,pseqcompanyinfo,pseqgetjobinfo,pseqregcourse);

commit;

exception

when others then

rollback;

end;

-------------------------------A-0010 4.2 관리자 수료생 연계기업 취업현황 수정

create or replace procedure procUpdateEmpStatus(

pseqcompanyinfo in number,

pseqempstatus in number

)

is

begin

update tblempstatus set

seqcompanyinfo = pseqcompanyinfo

where seqempstatus = pseqempstatus;

end;

-------------------------------A-0010 4.2 관리자 수료생 연계기업 취업현황 삭제

create or replace procedure procDeleteEmpStatus(

pseqempstatus in number

)

is

begin

update tblempstatus -----null값으로 변경

set seqcompanyinfo = null,

seqgetjobinfo=null,

seqregcourse=null

where seqempstatus = pseqempstatus;

commit;

exception

when others then

rollback;

end;

commit;

-- A011. 강의 및 시설 평가 조회

-- 종료된 과정명, 시작일, 종료일, 수강인원, 교사명 출력

create or replace view vwRegCourse

as

select

distinct

rc.seqOpenCourse as seqOpenCourse,

bc.name as course,

oc.startDate as startDate,

oc.endDate as endDate,

(select count(\*) from tblRegCourse rc inner join tblOpenCourse oc on rc.seqOpenCourse = oc.seqOpenCourse where oc.endDate < sysdate) as studentCount,

at.seqTeacher as seqTeacher,

t.name as teacher

from tblBasicCourseInfo bc

inner join tblOpenCourse oc

on bc.seqBasicCourseInfo = oc.seqBasicCourseInfo

inner join tblRegCourse rc

on oc.seqOpenCourse = rc.seqOpenCourse

inner join tblAssignTeacher at

on oc.seqOpenCourse = at.seqOpenCourse

inner join tblTeacher t

on at.seqTeacher = t.seqTeacher

where oc.endDate < sysdate;

select \* from vwRegCourse;

-- 특정 과정 선택

create or replace view vwSpecificRegCourse

as

select

\*

from vwRegCourse

where seqOpenCourse = 1;

-- 특정 과정 평가 조회 (교육생 번호, 교육생 이름, 평가항목...)

create or replace view vwEvaluation

as

select

rc.seqStudent,

s.name,

e.processScore,

e.understandScore,

e.communicationScore,

e.usefulScore,

e.satisfactionScore,

e.facilityScore,

e.managementScore

from tblRegCourse rc

left outer join tblEvaluation e

on rc.seqRegCourse = e.seqRegCourse

inner join tblStudent s

on rc.seqStudent = s.seqStudent

where seqOpenCourse = (select seqOpenCourse from vwSpecificRegCourse)

order by rc.seqStudent;

select \* from vwEvaluation;

-- 특정 교사 평가 조회

-- 종료된 과정 중 특정 교사가 강의한 과정 출력

create or replace view vwSpecificTeacher

as

select \* from vwRegCourse where seqTeacher = 1;

-- 선택한 과정의 교사 평가 조회

select

rc.seqStudent,

s.name,

e.processScore,

e.understandScore,

e.communicationScore,

e.usefulScore,

e.satisfactionScore

from tblEvaluation e

right outer join tblRegCourse rc

on e.seqRegCourse = rc.seqRegCourse

inner join tblStudent s

on rc.seqStudent = s.seqStudent

inner join tblOpenCourse oc

on rc.seqOpenCourse = oc.seqOpenCourse

inner join tblAssignTeacher at

on oc.seqOpenCourse = at.seqOpenCourse

where at.seqOpenCourse = (select seqOpenCourse from vwSpecificTeacher);

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

--A-003 개설과정 관리

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

--A-003-1 전체개설과정 정보조회

--전체개설과정 조회 뷰

create or replace view vTotalOpenCourse

as

select \* from (

select rownum num, a.\* from

(select b.seqbasiccourseinfo as seqBasicCourse,

o.seqOpenCourse as seqOpenCourse,

b.name as courseName,

to\_char(o.startdate, 'yyyy-mm-dd') as startDate,

to\_char(o.enddate, 'yyyy-mm-dd') as endDate,

r.name as room,

o.memberCount as countMember,

case

when endDate > sysdate and startDate < sysdate then '강의중'

when endDate < sysdate then '강의종료'

when startDate > sysdate then '강의예정'

end as courseState

from tblOpenCourse o

inner join tblBasicCourseInfo b

on o.seqBasicCourseInfo = b.seqBasicCourseInfo

inner join tblRoom r

on r.seqroom = o.seqroom

order by o.startDate desc)a);

--A-003-1-1 특정개설과정의 과목조회(오라클용)

create or replace procedure procViewSubject(

pseqOpenCourse in number --과정번호

)

is

cursor vcursor is select oc.seqopencourse as ocSeq,

bs.name as bsName, --과목명

os.startdate as osSd, --시작일

os.endDate as osEd, --종료일

t.name asTn, --교사명

b.name as bName, --교재명

case --과목진행여부

when os.endDate > sysdate and os.startDate < sysdate

then '진행중'

when os.endDate < sysdate

then '수료'

when os.startDate > sysdate

then '진행예정'

end as state

from tblAvailableSubject availSub

inner join tblBasicSubject bs

on availsub.seqbasicsubject = bs.seqBasicSubject

inner join tblOpenSubject os

on os.seqAvailableSubject = availsub.seqavailablesubject

inner join tblOpenCourse oc

on oc.seqopencourse = os.seqopencourse

inner join tblBook b

on bs.seqBook = b.seqBook

inner join tblTeacher t

on t.seqteacher = availsub.seqteacher

where oc.seqOpenCourse = PseqOpenCourse

order by os.startdate asc;

num number; --과목 앞에 번호 붙일 변수

begin

num := 1;

for vrow in vcursor loop

dbms\_output.put\_line('과목명' || num || ': ' || vrow.bsName);

dbms\_output.put\_line('시작일: ' || vrow.osSd);

dbms\_output.put\_line('종료일: ' || vrow.osEd);

dbms\_output.put\_line('교사명: ' || vrow.asTn);

dbms\_output.put\_line('과목진행여부: ' || vrow.state);

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

num := num + 1;

end loop;

end;

--A-003-1-1 특정개설과정의 과목조회 프로시저 호출

begin

procViewSubject(1);

end;

--A-003-1-1 특정개설과정의 과목조회(JDBD용)

create or replace procedure procViewSubject2(

pseqOpenCourse in number, --과정번호

pcursor out sys\_refcursor --커서

)

is

begin

open pcursor for select seqOpenCourse, name, startDate, endDate, teacherName, bookName, state

from vOpenSubject --뷰

where seqOpenCourse = PseqOpenCourse

order by startdate asc;

end;

--특정개설과정의 과목조회 뷰

create or replace view vOpenSubject

as

select

os.seqOpenCourse as seqOpenCourse, --과정번호

bs.name as name, --과목명

os.startdate as startDate, --시작일

os.endDate as endDate, --종료일

t.name teacherName, --교사명

b.name as bookName, --교재명

case --과목진행여부

when os.endDate > sysdate and os.startDate < sysdate

then '진행중'

when os.endDate < sysdate

then '수료'

when os.startDate > sysdate

then '진행예정'

end as state

from tblAvailableSubject availSub

inner join tblBasicSubject bs

on availsub.seqbasicsubject = bs.seqBasicSubject

inner join tblOpenSubject os

on os.seqAvailableSubject = availsub.seqavailablesubject

inner join tblOpenCourse oc

on oc.seqopencourse = os.seqopencourse

inner join tblBook b

on bs.seqBook = b.seqBook

inner join tblTeacher t

on t.seqteacher = availsub.seqteacher

order by os.startdate asc;

--A-003-1-2 특정개설과정의 교육생조회(오라클용)

create or replace procedure procViewStudent(

pseqOpenCourse number --과정번호

)

is

cursor vcursor is select s.seqStudent as seqS,

s.name as name,

s.ssn as ssn,

s.phone as tel,

s.firstRegistrationDate as registDate,

r.studentState as state

from tblStudent s

inner join tblRegCourse r

on s.seqstudent = r.seqstudent

inner join tblOpenCourse oc

on oc.seqOpenCourse = r.seqopencourse

where pseqOpenCourse = oc.seqOpenCourse;

begin

for vrow in vcursor loop

dbms\_output.put\_line('번호: ' || vrow.seqS);

dbms\_output.put\_line('이름: ' ||vrow.name);

dbms\_output.put\_line('주민번호: ' || vrow.ssn);

dbms\_output.put\_line('전화번호: ' || vrow.tel );

dbms\_output.put\_line('등록일: ' || to\_char(vrow.registDate, 'yyyy-mm-dd'));

dbms\_output.put\_line('수강상태: ' || vrow.state);

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

end loop;

end;

--A-003-1-2 특정개설과정의 교육생조회 프로시저 호출

begin

procViewStudent(1);

end;

--A-003-1-2 특정개설과정의 교육생조회(JDBD용)

create or replace procedure procViewStudent2(

pseqOpenCourse in number,

pstart in number,

pend in number,

pcursor out sys\_refcursor

)

is

begin

open pcursor for select num, seqOpenCourse, seqStudent, name, ssn, tel, registDate, state

from vViewStudent --뷰

where seqOpenCourse = pseqOpenCourse and num between pstart and pend;

end;

--특정개설과정 교육생 조회

declare

vcursor sys\_refcursor;

vrow vViewStudent%rowtype;

begin

procViewStudent2(2, 3, 8, vcursor);

loop

fetch vcursor into vrow;

exit when vcursor%notfound;

dbms\_output.put\_line(vrow.name);

end loop;

end;

--특정개설과정의 교육생조회 뷰

create or replace view vViewStudent

as

select \* from (select a.\*, rownum num from (select oc.seqOpenCourse as seqOpenCourse,

s.seqStudent as seqStudent,

s.name as name,

s.ssn as ssn,

s.phone as tel,

to\_char(s.firstRegistrationDate, 'yyyy-mm-dd') as registDate,

r.studentState as state

from tblStudent s

inner join tblRegCourse r

on s.seqstudent = r.seqstudent

right outer join tblOpenCourse oc

on oc.seqOpenCourse = r.seqopencourse

order by s.name asc)a)b;

--A-003-2 개설과정등록(오라클용)

create or replace procedure procRegistCourse(

proom in number, --강의실번호

pbasicCourse in number, --기초강좌번호

pstartDate in varchar2, --시작일

pendDate in varchar2, --종료일

pmemberCount in number --인원

)

is

vresult number; --강의실 상태 결과

vnum number; --방금 들어간 과정번호 복사할 변수

vresultCourse number; --과정 > 과목 할당 결과

begin

checkRoomState(proom, vresult); --강의실 중복검사

if --강의중인 강의실 -> 1

vresult = 1 then

dbms\_output.put\_line('해당 강의실은 이용중입니다.');

rollback;

elsif

to\_date(pstartDate, 'yyyy-mm-dd') > to\_date(pendDate, 'yyyy-mm-dd') then

dbms\_output.put\_line('날짜를 확인해주세요.');

rollback;

elsif --과정기간의 제약

(to\_date(pendDate, 'yyyy-mm-dd') - to\_date(pstartDate, 'yyyy-mm-dd')) / 30 > 9

or (to\_date(pendDate, 'yyyy-mm-dd') - to\_date(pstartDate, 'yyyy-mm-dd')) / 30 < 5 then

dbms\_output.put\_line('과정기간은 5.5개월, 6개월, 7개월만 등록할 수 있습니다.');

rollback;

else

insert into tblOpenCourse(seqOpenCourse, seqRoom, seqBasicCourseInfo, startDate, endDate, memberCount)

values(seqOpenCourse.nextVal, proom, pbasicCourse, to\_date(pstartDate, 'yyyy-mm-dd'), to\_date(pendDate, 'yyyy-dd-mm'), pmemberCount);

commit;

end if;

select max(seqOpenCourse) into vnum from tblOpenCourse; --방금 들어간 seqOpenCourse.nextVal 변수에 복사

checkRegistOpenSubject(vnum, vresultCourse); --개설과정이 개설과목에 할당 되었는가?

if

vresultCourse = 1 then --할당 되어있으면 t

update tblOpenCourse set courseRegistrationState = 't' where vnum = seqOpenCourse;

commit;

elsif

vresultCourse = 0 then --아니면 f

update tblOpenCourse set courseRegistrationState = 'f' where vnum = seqOpenCourse;

commit;

else

null;

end if;

exception when others then

dbms\_output.put\_line('등록할 수 없습니다. 다시 입력해주세요.');

rollback;

end;

--A-003-2 개설과정등록 프로시저 호출

begin

procRegistCourse(2, 3, '2021-11-14', '2022-05-12', 30);

end;

--\*\* 개설과목등록여부 검사

create or replace procedure checkRegistOpenSubject(

pseqOpenCourse number, --개설과정번호

presult out number --결과값

)

is

vnum number; --개설과정등록여부 복사할 변수

begin

select

case

when os.seqOpenCourse is null

then 0 --개설과목 > 개설과정 등록완료

else 1 --등록 x

end as 개설과정등록여부 into vnum

from tblOpenSubject os

right outer join tblOpenCourse oc

on os.seqOpenCourse = oc.seqOpenCourse

where pseqOpenCourse = oc.seqOpenCourse;

if

vnum = 1 then

presult := 1; --등록됨

else

presult := 0; --등록 x

end if;

exception when others then

dbms\_output.put\_line('등록할 수 없습니다.');

rollback;

end;

--\*\* 강의실 중복 검사

create or replace procedure checkRoomState(

proom in number, --강의실번호

presult out number --결과값

)

is

vresult number; --결과 넣을 변수

vroom number; --강의실 번호 담을 변수

begin

select

case

when sysdate between oc.startDate and oc.endDate

then 1 --강의중인 강의실

else 0 --사용가능한 강의실

end as 강의실 into vroom

from tblOpenCourse oc

right outer join tblRoom r

on oc.seqRoom = r.seqRoom

where proom = r.seqRoom;

if vroom = 1 then --강의중인 강의실

vresult := 1;

else

vresult := 0;

end if;

exception when others then

dbms\_output.put\_line('강의실 에러');

rollback;

end;

--\*\* 과정 기간 안에 교사 중복 검사

--1. 현재 강의중인 교사 조회

select t.seqTeacher as 교사번호,

t.name as 교사명,

bc.name as 강좌명,

to\_char(oc.startdate, 'yyyy-mm-dd') as 시작일,

to\_char(oc.enddate, 'yyyy-mm-dd') as 종료일

from tblOpenCourse oc

inner join tblBasicCourseinfo bc

on oc.seqBasicCourseInfo = bc.seqBasicCourseInfo

inner join tblAssignTeacher at

on at.seqOpenCourse = oc.seqOpenCourse

inner join tblTeacher t

on t.seqteacher = at.seqteacher

where sysdate between oc.startDate and oc.endDate;

--2. 교사중복여부

create or replace procedure checkAssignTeacher(

pseqTeacher in number, --교사번호

presult out number --결과값

)

is

vteacher number; --중복 교사번호 담을 변수

vresult number; --강의여부 상태를 담을 변수

begin

select

case

when sysdate between oc.startDate and oc.endDATE

then 1 else 0

end as tState into vteacher

from tblTeacher t

inner join tblAssignTeacher at

on t.seqTeacher = at.seqTeacher

inner join tblOpenCourse oc

on oc.seqOpenCourse = at.seqOpenCourse

where pseqTeacher = at.seqTeacher;

if

vteacher = 1 then

vresult := 1; --강의중이면 1

dbms\_output.put\_line('현재 강의중인 교사입니다.');

else

vteacher := 0; --아니면 0

end if;

exception when others then

dbms\_output.put\_line('존재하지 않는 강좌 또는 교사 번호입니다.');

rollback;

end;

--3. 개설강좌 배정

create or replace procedure procRegistOC(

popenCourse number, --강좌번호

pteacher number --교사번호

)

is

vresult number; --결과값 넣을 변수

vteacher number; --교사 프로시저 out값 담을 변수

begin

select oc.seqOpenCourse into vresult --개설강좌번호 담기

from tblOpenCourse oc

where popenCourse = oc.seqOpenCourse;

checkAssignTeacher(pteacher, vteacher); --강의중인 교사 체크

if vteacher = 1 or vresult > 0 then --강의중인 교사 또는 이미 추가된 강좌인 경우

rollback;

dbms\_output.put\_line('강좌가 중복되었습니다.');

else

insert into tblAssignTeacher(seqAssignTeacher, seqOpenCourse, seqTeacher)

values (seqAssignTeacher.nextVal, popenCourse, pteacher);

commit;

end if;

exception when others then

dbms\_output.put\_line('존재하지 않는 강좌 또는 교사 번호입니다.');

rollback;

end;

--개설강좌 & 교사배정 프로시저 호출

begin

procRegistOC(6, 4);

end;

--A-003-2 개설과정등록(JDBC용)

create or replace procedure procRegistCourse2(

proom in number, --강의실번호

pbasicCourse in number, --기초강좌번호

pstartDate in varchar2, --시작일

pendDate in varchar2, --종료일

pmemberCount in number --인원

)

is

begin

insert into tblOpenCourse(seqOpenCourse, seqRoom, seqBasicCourseInfo, startDate, endDate, memberCount)

values(seqOpenCourse.nextVal, proom, pbasicCourse, to\_date(pstartDate, 'yyyy-mm-dd'), to\_date(pendDate, 'yyyy-dd-mm'), pmemberCount);

commit;

exception when others then

raise\_application\_error(-20002, '과정을 등록할 수 없습니다.');

end;

--날짜제약(JDBC용)

create or replace procedure checkDate(

pstartDate varchar2,

pendDate varchar2,

presult out number

)

is

begin

if

to\_date(pstartDate, 'yyyy-mm-dd') > to\_date(pendDate, 'yyyy-mm-dd')

or to\_char(to\_date(pendDate), 'mm') - to\_char(to\_date(pstartDate), 'mm') not between 5 and 8

then presult := 1;

else presult := 0;

end if;

Exception when others then

raise\_application\_error(-20001, '날짜를 확인해주세요.');

end;

select to\_char(sysdate, 'mm') - to\_char(to\_date('2020-09-20'), 'mm') from dual;

--A-003-3 개설과정수정

--1-1. 개설과정 강의실 수정

create or replace procedure procEditRoom(

pseqOc in number, --과정번호

proom in number --바꿀 강의실 번호

)

is

vresult number; --결과값 받을 변수

begin

checkRoomState(proom, vresult); --강의중인 강의실 체크

if

vresult = 1 then

rollback;

else

--강의실 수정

update tblOpenCourse set seqRoom = proom where seqOpenCourse = pseqOc;

commit;

end if;

exception when others then

dbms\_output.put\_line('수정할 수 없습니다. 다시 입력해주세요.');

rollback;

end;

--1-1. 강의실 수정 프로시저 호출

begin

procEditRoom(8,6);

end;

--2-1. 개설과정 날짜 수정

create or replace procedure procEditCourse(

pseqOpenCourse in number, --수정할 과정 번호

pstartDate in varchar2, --시작일

pendDate in varchar2 --종료일

)

is

begin

if --시작일이 종료일보다 크거나 같은 경우

to\_date(pstartDate, 'yyyy-mm-dd') >= to\_date(pendDate, 'yyyy-mm-dd') then

dbms\_output.put\_line('날짜를 확인해주세요.');

rollback;

elsif --과정기간의 제약

(to\_date(pendDate, 'yyyy-mm-dd') - to\_date(pstartDate, 'yyyy-mm-dd')) / 30 > 9

or (to\_date(pendDate, 'yyyy-mm-dd') - to\_date(pstartDate, 'yyyy-mm-dd')) / 30 < 5 then

dbms\_output.put\_line('과정기간은 5.5개월, 6개월, 7개월만 등록할 수 있습니다.');

rollback;

else

--개설과정수정

update tblOpenCourse set startDate = to\_date(pstartDate, 'yyyy-mm-dd'), endDate = to\_date(pendDate, 'yyyy-mm-dd')

where seqOpenCourse = pseqOpenCourse;

commit;

end if;

exception when others then

dbms\_output.put\_line('수정할 수 없습니다. 다시 입력해주세요.');

rollback;

end;

--2-2.개설과정 날짜수정 프로시저 호출

begin

procEditCourse(1,'2020-06-10', '2020-08-20');

end;

--3-1. 개설과정정보의 기초강좌번호 수정

create or replace procedure procBasicCourseSeq(

pseqOpenCourse in number, --개설과정번호

pbasicCourseInfo in number --수정할 기초강좌번호

)

is

begin

update tblOpenCourse set seqBasicCourseInfo = pbasicCourseInfo where pseqOpenCourse = seqOpenCourse;

commit;

end;

--A-003-4 개설과정삭제

create or replace procedure procDeleteCourse(

pseqOpenCourse in number --삭제 할 과정 번호

)

is

vstartDate date; --startDate를 복사할 변수

vendDate date; --endDate를 복사할 변수

begin

select oc.startDate, oc.endDate into vstartDate, vendDate

from tblOpenCourse oc

where oc.seqOpenCourse = pseqOpenCourse; --startDate를 vstartDate에 복사, endDate를 vendDate에 복사

if

vendDate > sysdate and vstartDate < sysdate then --이미 진행중인 과정인 경우

dbms\_output.put\_line('진행중인 과정은 삭제할 수 없습니다.');

else --개설과정seq를 제외한 나머지 컬럼 null

update tblOpenCourse set seqRoom = null,

seqBasicCourseInfo = null,

endDate = null,

startDate = null,

memberCount = null,

courseRegistrationState = null

where seqOpenCourse = pseqOpenCourse;

commit;

--개설과정seq 참조하는 교사배정 개설과정번호 = null

update tblAssignTeacher at set at.seqOpenCourse = null where pseqOpenCourse = at.seqOpenCourse;

commit;

--개설과정seq 참조하는 수강정보 개설과정번호 = null

update tblRegCourse rg set rg.seqOpenCourse = null where pseqOpenCourse = rg.seqOpenCourse;

commit;

--개설과정seq 참조하는 개설과목 개설과정번호 = null

update tblOpenSubject os set os.seqOpenCourse = null where pseqOpenCourse = os.seqOpenCourse;

commit;

--개설과정 삭제

delete from tblOpenCourse where seqOpenCourse = pseqOpenCourse;

commit;

dbms\_output.put\_line('삭제완료');

end if;

exception when others then

dbms\_output.put\_line('과정을 삭제할 수 없습니다.');

rollback;

end;

--A-003-4 개설과정삭제 프로시저 호출

begin

procDeleteCourse(65);

end;

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

--A-004 개설과목관리

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

--전체개설과목 조회 뷰

create or replace view openSubjectView

as

select \* from (select a.\*, rownum num2 from (select os.seqOpenSubject as seqOpenSubject,

asub.seqAvailableSubject as seqAvailableSubject,

bs.name as subjectName,

b.name as bookName,

t.name as teacherName,

bc.name as courseName,

to\_char(os.startdate, 'yyyy-mm-dd') as startDate,

to\_char(os.enddate, 'yyyy-mm-dd') as endDate,

oc.seqOpenCourse as seqOpenCourse

from tblBasicSubject bs

inner join tblAvailableSubject asub

on bs.seqBasicSubject = asub.seqBasicSubject

inner join tblOpenSubject os

on asub.seqAvailableSubject = os.seqAvailableSubject

inner join tblOpenCourse oc

on oc.seqOpenCourse = os.seqOpenCourse

inner join tblBasicCourseInfo bc

on bc.seqbasiccourseinfo = oc.seqbasiccourseinfo

inner join tblBook b

on b.seqbook = bs.seqbook

inner join tblTeacher t

on t.seqTeacher = asub.seqTeacher

order by oc.startdate asc)a);

--A-004-1 특정과정의 개설과목조회

--A-003-1-1과 동일

--개설과목조회 프로시저 호출

begin

procViewSubject(2);

end;

--A-004-1-1 개설과목등록

create or replace procedure procRegistSubject(

pseqAvailSub number, --강의가능과목 번호

pseqOpenCourse number, --개설강좌번호

pstartDate varchar2, --시작일

pendDate varchar2 --종료일

)

is

vresult number; --강의중인 과목 체크 결과값 받을 변수

presult number; --과목날짜의 합 < 과정날짜 체크 결과값 받을 변수

begin

CheckAvailSub(pseqAvailSub, vresult); --강의중인 과목 체크

checkOpenSubjectDate(pseqOpenCourse, presult); --과목날짜의 합 < 과정날짜 체크

if

vresult = 1 or presult = 1 then

rollback;

elsif

--과목 시작일이 종료일보다 크거나 같을 경우

to\_date(pendDate, 'yyyy-mm-dd') <= to\_date(pstartDate, 'yyyy-mm-dd') then

dbms\_output.put\_line('날짜를 확인해주세요.');

rollback;

else

insert into tblOpenSubject(seqOpenSubject, seqAvailableSubject, seqOpenCourse, startDate, endDate)

values (seqOpenSubject.nextVal, pseqAvailSub, pseqOpenCourse, to\_date(pstartDate, 'yyyy-mm-dd'), to\_date(pendDate, 'yyyy-mm-dd'));

commit;

end if;

exception when others then

dbms\_output.put\_line('등록할 수 없습니다. 다시 입력해주세요.');

rollback;

end;

--A-004-1-1 개설과목등록 프로시저 호출

begin

procRegistSubject(33, 1, '1993-02-18', '1998-03-19');

end;

--A-004-1-2 개설과목수정

--1-1. 개설과목 -> 과정 수정

create or replace procedure procEditSubject(

pseqOpenSubject in number, --수정할 개설과목 번호

pseqOpenCourse in number --수정할 과정번호

)

is

vresult number; --과목날짜의 합 < 과정날짜 체크 결과값 받을 변수

begin

checkOpenSubjectDate(pseqOpenCourse, vresult); --과목날짜의 합 < 과정날짜 체크 프로시저

if

vresult = 1 then

rollback;

else

update tblOpenSubject set seqOpenCourse = pseqOpenCourse where pseqOpenSubject = seqOpenSubject;

commit;

end if;

exception when others then

dbms\_output.put\_line('수정할 수 없습니다. 다시 입력해주세요.');

rollback;

end;

--1-2. 개설과목 -> 과정 수정 프로시저 호출

begin

procEditSubject(1, 3);

end;

--2-1. 개설과목 -> 강의가능과목번호 수정

create or replace procedure procEditAvailSub(

pseqOpenSub in number, --개설과목번호

pseqAvailSub in number --강의가능과목번호

)

is

vresult number; --강의가능과목 담을 변수

begin

CheckAvailSub(pseqAvailSub, vresult); --강의가능과목 중복 확인

if --강의가능한 과목번호가 진행중인 과정에 있을 경우

vresult = 1 then

rollback;

else

update tblOpenSubject set seqAvailableSubject = pseqAvailSub;

commit;

end if;

end;

--2-2. 개설과목 -> 강의가능과목 번호 수정 프로시저 호출

begin

procEditAvailSub(3, 10);

end;

--\*\*강의가능과목 중복 검사

create or replace procedure CheckAvailSub(

pseqavailSub in number,

presult out number

)

is

vnum number; --결과 복사할 변수

begin

select

case

when sysdate between oc.startDate and oc.endDate

then 1 --이미 강의중인 강좌

else 0 --강의중 x

end as avss into vnum

from tblOpenCourse oc

inner join tblOpenSubject os

on oc.seqOpenCourse = os.seqOpenCourse

right outer join tblAvailableSubject avs

on avs.seqAvailableSubject = os.seqAvailableSubject

where pseqAvailSub = avs.seqAvailableSubject;

if

vnum = 1 then

dbms\_output.put\_line('현재 진행중인 과목입니다.');

presult := 1; --진행중

else

presult := 0; --진행중 x

end if;

end;

--\*\*전체 개설과목 날짜의 합 < 개설과정 날짜 체크

create or replace procedure checkOpenSubjectDate(

popencourse in number, --과정번호

presult out number --결과값

)

is

cursor vcursor is select subjectDate, courseDate, seqOpenCourse from vcheckSubjectDate where popenCourse = seqOpenCourse;

vrow vchecksubjectdate%rowtype;

vdate number;

begin

vdate := 0;

open vcursor;

loop

fetch vcursor into vrow;

exit when vcursor%notfound;

vdate := vdate + vrow.subjectDate; --과목날짜 합 누적

end loop;

close vcursor;

if

vdate > vrow.courseDate --과목날짜 합이 과정 날짜를 넘겼을 경우 1

then

dbms\_output.put\_line('전체 과목기간은 과정기간을 넘길 수 없습니다.');

presult := 1;

else

presult := 0;

end if;

end;

--과목날짜 체크 뷰

create or replace view vcheckSubjectDate

as

select os.endDate - os.startDate as subjectDate,

oc.endDate - oc.startDate as courseDate,

os.seqOpenCourse as seqOpenCourse

from tblOpenSubject os

left outer join tblOpenCourse oc

on os.seqopencourse = oc.seqopencourse;

--3-1. 개설과목 -> 날짜 수정

create or replace procedure procEditSubjectDate(

pseqOpenSubject in number, --개설과목번호

pstartDate in varchar2, --시작일

pendDate in varchar2 --종료일

)

is

begin

if

pendDate > pstartDate then

dbms\_output.put\_line('날짜를 확인해주세요.');

rollback;

else

update tblOpenSubject set StartDate = to\_date(pstartDate, 'yyyy-mm-dd'),

EndDate = to\_date(pendDate, 'yyyy-mm-dd')

where seqOpenSubject = pseqOpenSubject;

commit;

end if;

end;

--3-2. 날짜 수정 프로시저 호출

begin

procEditSubjectDate(1, '2020-10-09', '1999-03-28');

end;

--A-004-1-3 개설과목삭제

create or replace procedure procRegistSubject(

pseqOpenSubject in number --개설과목 번호

)

is

cursor vcursor is select seqOpenCourse from tblopensubject;

vrow number;

begin

open vcursor;

loop

fetch vcursor into vrow;

exit when vcursor%notfound;

if

vrow > 0 then

dbms\_output.put\_line('개설 된 강좌에 있는 과목은 삭제할 수 없습니다.');

rollback;

exit;

else

--개설과목번호 삭제 전 다 null값으로 만들기

update tblOpenSubject set seqAvailableSubject = null,

seqOpenCourse = null,

startDate = null,

endDate = null

where seqOpenSubject = pseqOpenSubject;

--개설과목번호 참조하는 시험정보 테이블에서 개설과목번호 null로 바꾸기

update tblBasicTest set seqOpenSubject = null where seqOpenSubject = pseqOpenSubject;

--개설과목번호 삭제

delete from tblOpenSubject where seqOpenSubject = pseqOpenSubject;

dbms\_output.put\_line('삭제되었습니다.');

commit;

end if;

end loop;

close vcursor;

exception when others then

dbms\_output.put\_line('삭제할 수 없습니다. 다시 시도해주세요.');

rollback;

end;

--A-004-1-3 개설과목삭제 프로시저 호출

begin

procRegistSubject(3);

end;

--A-005 교육생관리-추가

create or replace procedure addStudent(

pseqStudent tblStudent.seqStudent%type,

pname tblStudent.name%type,

pid tblStudent.id%type,

pssn tblStudent.ssn%type,

pphone tblStudent.phone%type,

pemail tblStudent.email%type,

pfRegDate tblStudent.firstregistrationdate%type,

pemployField tblstudent.employmentfield%type

)

is

begin

insert into tblStudent (seqStudent,name,id,ssn,phone,email,firstregistrationdate,employmentfield)

values (pseqStudent,pname,pid,pssn,pphone,pemail,pfRegDate,pemployField);

dbms\_output.put\_line('추가 성공.');

commit;

exception

when DUP\_VAL\_ON\_INDEX then

dbms\_output.put\_line('ID 중복');

end addStudent;

--A-005 교육생관리-추가-유효성검사

create or replace trigger trgAddStudent

before

insert on tblStudent

for each row

begin

if :new.ssn not like '\_\_\_\_\_\_-\_\_\_\_\_\_\_' then

raise\_application\_error(-20000,'올바른 주민등록번호를 입력하세요.');

elsif :new.phone not like '01\_-%-\_\_\_\_' then

raise\_application\_error(-20001,'올바른 번호를 입력하세요.');

elsif :new.email not like '%@%.%' then

raise\_application\_error(-20002,'올바른 이메일 주소를 입력하세요.');

end if;

end trgAddStudent;

--A-005 교육생관리-전체조회

create or replace procedure chkStudentAll

is

result number:=0;

cursor c1

is

select

s.seqStudent as seqStudent\_s,s.name as name\_s,id,ssn,phone,email,firstRegistrationDate,employmentField,b.name as name\_b,r.seqRegCourse as seqRegCourse

from tblstudent s

left outer join tblRegCourse r

on s.seqStudent=r.seqStudent

left outer join tblOpenCourse o

on o.seqOpenCourse=r.seqOpenCourse

left outer join tblBasicCourseInfo b

on b.seqBasicCourseInfo=o.seqBasicCourseInfo

order by 1;

vrow c1%rowtype;

begin

open c1;

loop

fetch c1

into vrow;

exit

when c1%notfound;

if vrow.id is not null then

dbms\_output.put(

vrow.seqStudent\_s||','||vrow.name\_s||','||vrow.id||','||substr(vrow.ssn,8,14)||','||vrow.phone||','||vrow.email||','||vrow.seqRegCourse||','||vrow.employmentField||','||to\_char(vrow.firstRegistrationDate,'yyyy-mm-dd')

);

if vrow.name\_b is not null then

dbms\_output.put(','||vrow.name\_b);

end if;

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

result:=result+1;

end if;

end loop;

close c1;

dbms\_output.put\_line(result||'건 조회 완료.');

end chkStudentAll;

--A-005 교육생관리-수강번호로조회

create or replace procedure chkStudent(pnum number)

is

result number:=0;

cursor c1

is

select

s.seqStudent as seqStudent\_s,s.name as name\_s,id,ssn,phone,email,firstRegistrationDate,employmentField,b.name as name\_b,r.seqRegCourse as seqRegCourse

from tblstudent s

left outer join tblRegCourse r

on s.seqStudent=r.seqStudent

left outer join tblOpenCourse o

on o.seqOpenCourse=r.seqOpenCourse

left outer join tblBasicCourseInfo b

on b.seqBasicCourseInfo=o.seqBasicCourseInfo

where r.seqRegCourse=pnum

order by 1;

vrow c1%rowtype;

begin

open c1;

loop

fetch c1

into vrow;

exit

when c1%notfound;

dbms\_output.put(

vrow.seqStudent\_s||','||vrow.name\_s||','||vrow.id||','||substr(vrow.ssn,8,14)||','||vrow.phone||','||vrow.email||','||vrow.seqRegCourse||','||vrow.employmentField||','||to\_char(vrow.firstRegistrationDate,'yyyy-mm-dd')

);

if vrow.name\_b is not null then

dbms\_output.put(','||vrow.name\_b);

end if;

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

result:=result+1;

end loop;

close c1;

dbms\_output.put\_line(result||'건 조회 완료.');

end chkStudent;

--A-005 교육생관리-아이디로 삭제

create or replace procedure delStudent(pid varchar2)

is

vid tblStudent.id%type;

begin

select id into vid from tblStudent where id=pid;

update tblStudent set id='', ssn='',phone='',email='',firstRegistrationDate='',Employmentfield=''

where id=pid;

dbms\_output.put\_line('삭제 성공.');

commit;

--exception

-- when others then

-- dbms\_output.put\_line('일치하는 데이터 없음.');

end delStudent;

--A-009 성적정보(우수훈련생) 출력기능 시험배점정보→시험성적정보

drop table tbltest1;

create table tbltest1(

seq number primary key,

seqRegCourse number,

writtenScore number,

attendanceScore number,

practicalScore number

);

select \* from tbltest1;

drop sequence seqtest1;

create sequence seqtest1;

create or replace procedure chkTopStudent

is

cursor c1

is

select

p.seqTestPercent as seqTestPercent,

r.seqRegCourse as seqRegCourse,

max(s.name) as name,

round(avg(sc.writtenScore)) as score1,

round(avg(sc.attendanceScore)) as score2,

count(case

when a.attendState='출석' then 1

end) as score3

from tblStudent s --교육생

right join tblRegCourse r --수강정보

on r.seqStudent=s.seqStudent

left outer join tblAttendance a --출결정보

on r.seqRegCourse=a.seqRegCourse

left outer join tblTestPercent p --시험배점정보

on r.seqRegCourse=p.seqRegCourse

left outer join tblTestScore sc --시험성적정보

on p.seqTestPercent=sc.seqTestPercent

left outer join tblOpenCourse o --개설과정정보

on o.seqOpenCourse=r.seqOpenCourse

group by r.seqRegCourse,p.seqTestPercent

having count(a.attendState) != 0 and count(sc.writtenScore) != 0 and count(sc.attendanceScore) != 0

order by 2 desc;

vrow c1%rowtype;

begin

open c1;

loop

fetch c1 into vrow;

exit when c1%notfound;

--dbms\_output.put\_line(vrow.seqRegCourse||','||vrow.seqTestPercent||','||vrow.name||','||vrow.score1||','||vrow.score2||','||vrow.score3);

--insert into tbltest1 (seq,seqRegCourse,writtenScore,attendanceScore,practicalScore) values (seqtest1.nextVal,vrow.seqTestPercent,vrow.score1,vrow.score2,vrow.score3);

insert into tblTestScore (seqTestScore,seqTestPercent,writtenScore,attendanceScore,practicalScore) values (seqTestScore.nextVal,vrow.seqTestPercent,vrow.score1,vrow.score2,vrow.score3);

end loop;

close c1;

end chkTopStudent;

--A-012 우수훈련생(출결/성적) top3 출력 (시험성적정보테이블→우수훈련생테이블)

create or replace procedure chkTopStudent

is

cursor c1

is

select

t.\*

from(select

\*

from tbltest1

where WRITTENSCORE is not null

order by (WRITTENSCORE+ATTENDANCESCORE)/2 desc

) t

where rownum<=3;

vrow c1%rowtype;

cursor c2

is

select

t.\*

from(select

\*

from tbltest1

where WRITTENSCORE is not null

order by (WRITTENSCORE+ATTENDANCESCORE)/2 desc

) t

where rownum<=3;

vrow2 c2%rowtype;

begin

open c1;

loop

fetch c1 into vrow;

exit when c1%notfound;

--dbms\_output.put\_line(vrow.seq||','||vrow.SEQREGCOURSE||','||vrow.WRITTENSCORE||','||vrow.ATTENDANCESCORE||','||vrow.PRACTICALSCORE||',성적');

insert into tblTopStudent (seqTopStudent,seqTestScore,seqScholarship) values (seqTopStudent.nextVal,vrow.seq,1);

end loop;

close c1;

open c2;

loop

fetch c2 into vrow2;

exit when c2%notfound;

--dbms\_output.put\_line(vrow2.seq||','||vrow2.SEQREGCOURSE||','||vrow2.WRITTENSCORE||','||vrow2.ATTENDANCESCORE||','||vrow2.PRACTICALSCORE||',출결');

insert into tblTopStudent (seqTopStudent,seqTestScore,seqScholarship) values (seqTopStudent.nextVal,vrow.seq,2);

end loop;

close c2;

end chkTopStudent;