

## Project

### part1

1.

```
declare
v_stu_id enrollments.stu_id%type:=:student_id;
v_class_id enrollments.class_id%type:=:class_id;
v_date enrollments.enrollment_date%type:=sysdate;
begin
insert into enrollments (enrollment_date, class_id,stu_id,status)
values(v_date,v_class_id,v_stu_id,'Enrolled');
end;
```

2.

```
declare v_stu_id enrollments.stu_id%type:=:student_id;
v_class_id enrollments.class_id%type:=:class_id;v_count number;
begin
delete from enrollments where stu_id=v_stu_id and class_id=v_class_id;
v_count:=sql%rowcount;
if v_count=0 then
for v_record in (select * from enrollments)loop
dbms_output.put_line(v_record.enrollment_date ||
v_record.class_id||v_record.stu_id||v_record.final_numeric_grade||
v_record.final_letter_grade||v_record.status);
end loop;
end if;
end;
```

3.

```
declare
v_stud_id enrollments.stu_id%type:=:Student_Id;
cursor c_cursor(p_stu_id in enrollments.stu_id%type)
is
select enrollment_date, class_id, status from enrollments
where stu_id=p_stu_id and enrollment_date > add_months(sysdate, -12*10);
begin
for r_record in c_cursor(v_stud_id)
loop
dbms_output.put_line('Enrollment_Date:'|| r_record.enrollment_date ||', Class_Id:
'||r_record.class_id||', Status:'||r_record.status);
end loop;
end;
```

4.

```
declare
v_number_of_classes number := 1;
v_class_id number;
```

```

v_next_id v_class_id%type;
v_curr_date date:=sysdate;
v_course_id number := 1002;
v_period varchar2(20):='First';
v_frequency varchar2(20) := 'Weekly'
;v_instructor_id number := 3002;
begin
for i in 1..v_number_of_classes
loop
select max(class_id) into v_class_id from classes;
v_next_id:=v_class_id + 1;
insert into classes values
(v_next_id,v_curr_date,v_period, v_frequency, v_instructor_id, v_course_id);
end loop;
end;

```

## part 2

1.

```

declare
v_course_id courses.course_id%type:=:Course_Id;
v_instructor_id instructors.instructor_id%type:=:Instructor_Id;
cursor c_students(p_c_course_id courses.course_id%type,
p_c_instructor_id instructors.instructor_id%type)
is
select e.class_id, e.status, s.first_name, s.last_name
FROM students s
left join enrollments e on s.stu_id = e.stu_id
left join classes c on e.class_id = c.class_id
where c.course_id = p_c_course_id AND c.instr_id=p_c_instructor_id;
begin
for r_record in c_students(v_course_id,v_instructor_id)
loop
dbms_output.put_line ('Class_Id: '||r_record.class_id || ',
Status: '|| r_record.status || ', First_Name: ' || r_record.first_name || ', Last_Name: '|| r_record.last_name);
end loop;
end;

```

2.

```

declare
e_invalid_grade EXCEPTION;
v_grade_number number:=:Grade;
v_grade_char char(1);
begin
if v_grade_number >= 90 AND v_grade_number <=100 then
v_grade_char:='A';
elsif v_grade_number >=80 AND v_grade_number < 90 then
v_grade_char:='B';
elsif v_grade_number >=70 AND v_grade_number < 80 then
v_grade_char:='C';

```

```

elsif v_grade_number >=60 AND v_grade_number <70 then
v_grade_char:='D';
elsif v_grade_number < 60 then
v_grade_char:='F';
else
RAISE e_invalid_grade;
end if;
dbms_output.put_line('Grade: ' || v_grade_char);
exception
when e_invalid_grade then
dbms_output.put_line('Invalid');
end;

```

3.

```

declare
v_class_id enrollments.class_id%type:=:class_ID;
v_count number;
begin
select count(stu_id) into v_count from enrollments where class_id=v_class_id;
dbms_output.put_line('Grade is: ' || v_count);
end;

```

4.

```

declare
v_description assessments.description%type:=:Opis;
v_grading_criteria assessments.grading_criteria%type:=:Numeric;
begin
insert into assessments values (assessment_id_seq.nextval,v_description, v_grading_criteria);
end;

```

5.

```

declare
v_numeric_grade class_assessments.numeric_grade%type:=:Ocena(Numeric);
v_class_assessment_id class_assessments.class_assessment_id%type:=:Class_Assessments_Id;
v_class_id class_assessments.class_id%type:=:Class_Id;
v_stu_id class_assessments.stu_id%type:=:Stu_Id;
v_date class_assessments.date_turned_id%type:=:sysdate;
begin
insert into class_assessments(class_assessment_id, date_turned_in, numeric_grade, class_id,
stu_id, assessment_id) values (v_class_assessment_id, v_date v_numeric_grade, v_class_id, v_stu_id,
v_class_assessment_id);
end;

```

part 3

1.

```

declare
v_start_date enrollments.enrollment_date%type:=add_months(sysdate,-12);
v_end_date enrollments.enrollment_date%type:=sysdate;
cursor c_enrollments(p_c_start_date enrollments.enrollment_date%type,
p_c_end_date enrollments.enrollment_date%type)

```

```

is
select * from enrollments where final_numeric_grade is null and final_letter_grade is null
and enrollment_date between p_c_start_date and p_c_end_date
order by enrollment_date desc;
begin
for r_record in c_enrollments(v_start_date, v_end_date)
loop
dbms_output.put_line(|| v_start_date || ' do: ' || v_end_date || ' Class_Id: ' ||r_record.class_id || ',
Student_Id: ' || r_record.stu_id||', Status: ' || r_record.status);
end loop;
end;

```

```

2.
declare
v_class_id class_assessments.class_id%type:=:Id_Class;
v_average_grade number;
begin
select avg(numeric_grade) into v_average_gradefrom class_assessments
where class_id=v_class_id;
dbms_output.put_line('Average: ' || v_average_grade);
end;

```

```

3.
declare
v_course_id classes.course_id%type:=:Course_Id;
v_number_of_classes number;
begin
select count (*) into v_number_of_classes
from classes
where course_id=v_course_id
;dbms_output.put_line ('Number of classes: ' || v_number_of_classes);
end;

```

```

4.
declare
v_start_date date :=:Datum_od;
v_end_date date :=:Datum_do
;cursor c_classes (p_c_start_date date, p_c_end_date date)
is
select c.class_id, c.start_date, i.first_name, i.last_name, co.title, co.section_code
from classes c
left join instructors i on i.instructor_id = c.instr_i
dleft join courses co on c.course_id = co.course_id
where c.start_date between p_c_start_date and p_c_end_date;
begin
for r_record in c_classes(v_start_date, v_end_date)
loop
dbms_output.put_line(v_start_date|| ' - ' || v_end_date|| 'Class_ID: '||r_record.class_id||', Start_Date: '||
r_record.start_date||', Instructor name: '||r_record.first_name||' '||r_record.last_name||', Title: '||

```

```
r_record.title||', Selection_Code: '||r_record.section_code);  
end loop;  
end;
```