

```

1  /* Question 2.5 and Question 2.6 */
2
3  create or replace package course_registration as
4      /* Triggers will be explicitly written */
5
6      procedure showPreRequisitesforCourse(deptCode in Prerequisites.dept_code%type,
7      courseNo in Prerequisites.course#%type, isFirst Boolean, result out varchar2);
8
9      procedure enroll_student(param_B# in Enrollments.B#%type, param_classid in
10      Enrollments.Classid%type, messages out varchar2);
11
12      procedure drop_student(studentid_b# in students.b#%type, studentid_classid in
13      classes.classid%type, message out varchar );
14
15  end;
16  /
17  show errors
18
19  /* Body for the package course_registration */
20  create or replace package body course_registration as
21
22      procedure showPreRequisitesforCourse(deptCode in Prerequisites.dept_code%type,
23      courseNo in Prerequisites.course#%type, isFirst Boolean, result out varchar2) is
24          CURSOR courses_refcur is
25              select pre_dept_code, pre_course# FROM Prerequisites WHERE dept_code =
26              deptCode and course# = courseNo;
27              course courses_refcur%rowtype;
28              courseCount number;
29              begin
30                  select count(*) into courseCount from Courses where dept_code = deptCode and
31                  course# = courseNo;
32                  open courses_refcur;
33                  fetch courses_refcur into course;
34                  if(courses_refcur%found) then
35                      while(courses_refcur%found) loop
36                          showPreRequisitesforCourse(course.pre_dept_code, course.pre_course#, FALSE,
37                          result);
38                          if(result is NULL) then
39                              result := course.pre_dept_code || course.pre_course#;
40                          else
41                              result := result || ',' || course.pre_dept_code || course.pre_course#;
42                          end if;
43                          fetch courses_refcur into course;
44                      end loop;
45                  elsif(courseCount < 1 and isFirst) then
46                      result := deptCode || courseNo || ' does not exist';
47                  end if;
48              end
49      showPreRequisitesforCourse;
50
51      procedure enroll_student(param_B# in Enrollments.B#%type, param_classid in
52      Enrollments.Classid%type, messages out varchar2) is
53
54          enrolled_prerequisites number;
55          total_prerequisites number;
56          not_a_student exception;
57          not_a_class exception;
58          not_current_semester exception;
59          class_already_full exception;
60          already_enrolled exception;
61          max_classes exception;
62          relation_prerequisites exception;
63          relation_prerequisites_enroll exception;
64
65          student_count number;
66          class_count number;
67          sem_class_count number;
68          class_full_count number;
69          student_in_class_count number;
70          student_total_classes number;

```

```

62     student_prerequisites_count number;
63     prerequisiteClasses varchar2(200);
64
65     cursor c1 is
66     select dept_code, course# from classes where classid = param_classid;
67     class c1%rowtype;
68
69     begin
70         select count(*) into student_count from Students where B# = param_B#;
71         if(student_count < 1) then
72             raise not_a_student;
73         end if;
74
75         select count(*) into class_count from Classes where classid = param_classid;
76         if(class_count < 1) then
77             raise not_a_class;
78         end if;
79
80         select count(*) into sem_class_count from Classes where classid =
81         param_classid and year = 2018 and semester = 'Fall';
82         if(sem_class_count < 1) then
83             raise not_current_semester;
84         end if;
85
86         select count(*) into class_full_count from Classes where classid =
87         param_classid and year = 2018 and semester = 'Fall' and limit <> class_size;
88         if(class_full_count < 1) then
89             raise class_already_full;
90         end if;
91
92         select count(*) into student_in_class_count from Enrollments where classid =
93         param_classid and B# = param_B#;
94         if(student_in_class_count > 0) then
95             raise already_enrolled;
96         end if;
97
98         select count(*) into student_total_classes from Enrollments where B# =
99         param_B#;
100         if(student_total_classes >= 5) then
101             raise max_classes;
102         end if;
103
104         if(student_total_classes = 4) then
105             messages := 'The student will be overloaded with the new enrollment';
106         end if;
107
108         if(student_total_classes >= 5) then
109             raise max_classes;
110         end if;
111
112         open c1;
113         fetch c1 into class;
114
115         showPreRequisitesforCourse(class.dept_code, class.course#, TRUE,
116         prerequisiteClasses);
117
118         select count(*) into total_prerequisites from Classes where dept_code ||
119         course# in (
120         select regexp_substr(prerequisiteClasses, '[^,]+' , 1, level) from dual
121         connect by regexp_substr(prerequisiteClasses, '[^,]+' , 1, level) is not
122         null );
123
124         Select count(*) into enrolled_prerequisites from Enrollments e, Classes
125         c1 where e.classid = c1.classid and e.B# = param_B# and c1.dept_code ||
126         c1.course# in (
127         select regexp_substr(prerequisiteClasses, '[^,]+' , 1, level) from dual
128         connect by regexp_substr(prerequisiteClasses, '[^,]+' , 1, level) is not
129         null );
130

```

```

121         if(total_prerequisites != enrolled_prerequisites) then
122             raise relation_prerequisites_enroll;
123         end if;
124
125         showPreRequisitesforCourse(class.dept_code, class.course#, TRUE,
prerequisiteClasses);
126         select count(*) into student_prerequisites_count from Enrollments e,
Classes c where B# = param_B# and e.Classid = c.classid and
127         INSTR(prerequisiteClasses, c.dept_code || c.course#) > 0 and e.lgrade not
in ('A', 'A-', 'B+', 'B', 'B-', 'C+', 'C');
128         close c1;
129         if(student_prerequisites_count > 0) then
130             raise relation_prerequisites;
131         end if;
132
133     insert into Enrollments values (param_B#, param_classid, NULL);
134     messages := 'Student added successfully';
135
136 exception
137     when not_a_student then
138         messages := 'The B# is invalid';
139     when not_a_class then
140         messages := 'The Classid is invalid';
141     when not_current_semester then
142         messages := 'Cannot enroll into a class from a previous semester';
143     when class_already_full then
144         messages := 'The class is already full';
145     when already_enrolled then
146         messages := 'The student is already in the class';
147     when max_classes then
148         messages := 'Students cannot be enrolled in more than five classes in the same
semester';
149     when relation_prerequisites then
150         messages := 'Prerequisite not satisfied';
151     when relation_prerequisites_enroll then
152         messages := 'Prerequisite Issue : Student has not completed the Prerequisites';
153 end enroll_student;
154
155
156     procedure drop_student(studentid_b# in students.b#%type, studentid_classid in
classes.classid%type, message out varchar )
157 is
158     message1 varchar(50);
159     studentid_class_enroll_cnt int;
160     studentid_dept_code classes.dept_code%type;
161     message2 varchar(50);
162     studentid_prereq_cnt int;
163     studentid_class_size int;
164     studentid_course# classes.course#%type;
165     studentid_class_size_cnt int;
166     studentid_count int;
167     studentid_class_cnt int;
168
169     begin
170         select count(*) into studentid_count from students where b# = studentid_b#;
171         if studentid_count=0
172         then
173             message:='The B# is invalid';
174         else
175             select count(*) into studentid_class_cnt from classes where
classid=studentid_classid;
176             if studentid_class_cnt=0
177             then
178                 message:='The classid is invalid';
179             else
180                 select count(*) into studentid_class_enroll_cnt from enrollments
where classid=studentid_classid and b#=studentid_b#;
181                 if studentid_class_enroll_cnt=0
182                 then

```

```

183         message:= 'The student is not enrolled in the class';
184     else
185         select dept_code,course# into
186             studentid_dept_code,studentid_course# from classes where
187                 classid=studentid_classid;
188
189         select count(*) into studentid_prereq_cnt from classes
190             cl,prerequisites p where classid in (select classid from
191                 enrollments where classid<>studentid_classid and
192                 b#=studentid_b#) and
193             cl.dept_code=p.dept_code and cl.course# = p.course# and
194             p.pre_dept_code=studentid_dept_code and
195             p.pre_course#=studentid_course#;
196
197     if studentid_prereq_cnt>0
198     then
199         message:='The drop is not permitted because another class uses
200             it as a prerequisite';
201
202     else
203         delete from enrollments where b#=studentid_b# and
204             classid=studentid_classid;
205         message:='Student dropped successfully';
206
207         select class_size into studentid_class_size from classes where
208             classid=studentid_classid;
209
210         select count(*) into studentid_class_size_cnt from enrollments
211             where b#=studentid_b#;
212
213         if studentid_class_size=0
214         then
215             message1:='The class now has no students.';
216         end if;
217
218         if studentid_class_size_cnt=0
219         then
220             message2:='This student is not enrolled in any classes.';
221         end if;
222         message:=message||' '||message1||' '||message2;
223
224     end if;
225 end if;
226 end if;
227 end if;
228 end drop_student ;
229
230 end course_registration;
231 /
232 show errors

```