

```

1  /* Question 2.5 and Question 2.6 */
2  /* This package has been written for any student to enroll or the drop from course */
3  /* showPreRequisitesforCourse this procedure has been used in enroll_student for
4  prerequisite looping query */
5
6  create or replace package course_registration as
7      /* Triggers will be explicitly written */
8      /* deptCode and Course Number has been taken as IN parameter for which we need to
9      check whether prerequisites are present */
10     /* Boolean flag is checking whether condition for the combination of input */
11     /* result as OUT to display appropriate results in JAVA*/
12     procedure showPreRequisitesforCourse(deptCode in Prerequisites.dept_code%type,
13     courseNo in Prerequisites.course#%type, isFirst Boolean, result out varchar2);
14
15     /* It takes Student B# and Class ID as IN parameter and */
16     /* Message has been used for displaying results in java */
17     /* This query results in enrolling any student for given Class ID if conditions are
18     met */
19     procedure enroll_student(param_B# in Enrollments.B#%type, param_classid in
20     Enrollments.Classid%type, messages out varchar2);
21
22     /* It takes Student B# and Class ID as IN parameter and */
23     /* Message has been used for displaying results in java */
24     /* This query results in dropping any student for given Class ID*/
25     procedure drop_student(studentid_b# in students.b#%type, studentid_classid in
26     classes.classid%type, message out varchar );
27
28 end;
29 /
30 show errors
31
32 /* Body for the package course_registration */
33 create or replace package body course_registration as
34
35     procedure showPreRequisitesforCourse(deptCode in Prerequisites.dept_code%type,
36     courseNo in Prerequisites.course#%type, isFirst Boolean, result out varchar2) is
37     CURSOR courses_refcur is
38     select pre_dept_code, pre_course# from Prerequisites where dept_code = deptCode and
39     course# = courseNo;
40     course courses_refcur%rowtype;
41     courseCount number;
42     begin
43         select count(*) into courseCount from Courses where dept_code = deptCode and
44         course# = courseNo;
45         open courses_refcur;
46         fetch courses_refcur into course;
47         if(courses_refcur%found) then
48             while(courses_refcur%found) loop
49                 showPreRequisitesforCourse(course.pre_dept_code, course.pre_course#, FALSE,
50                 result);
51                 result := result || ' ' || course.pre_dept_code || course.pre_course#;
52                 fetch courses_refcur into course;
53             end loop;
54         elsif(courseCount < 1 and isFirst) then
55             result := 'The combination you entered
56             '||course.pre_dept_code||course.pre_course#||' does not exists';
57         end if;
58     end showPreRequisitesforCourse;
59
60     procedure enroll_student(param_B# in Enrollments.B#%type, param_classid in
61     Enrollments.Classid%type, messages out varchar2) is
62
63         student_count number;
64         class_count number;
65         sem_class_count number;
66         class_full_count number;
67         student_in_class_count number;

```

```

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

```

120         end if;
121
122         insert into Enrollments values (param_B#, param_classid, NULL);
123         messages := 'Student added successfully';
124
125     exception
126     when not_a_student then
127         messages := 'The B# is invalid';
128     when not_a_class then
129         messages := 'The Classid is invalid';
130     when not_current_semester then
131         messages := 'Cannot enroll into a class from a previous semester';
132     when class_already_full then
133         messages := 'The class is already full';
134     when already_enrolled then
135         messages := 'The student is already in the class';
136     when max_classes then
137         messages := 'Students cannot be enrolled in more than five classes in the same
138         semester';
139     when relation_prerequisites then
140         messages := 'Prerequisite not satisfied';
141 end enroll_student;
142
143 procedure drop_student(studentid_b# in students.b#%type, studentid_classid in
144 classes.classid%type, message out varchar )
145 is
146     message1 varchar(50);
147     studentid_class_enroll_cnt int;
148     studentid_dept_code classes.dept_code%type;
149     message2 varchar(50);
150     studentid_prereq_cnt int;
151     studentid_class_size int;
152     studentid_course# classes.course#%type;
153     studentid_class_size_cnt int;
154     studentid_count int;
155     studentid_class_cnt int;
156
157 begin
158     select count(*) into studentid_count from students where b# = studentid_b#;
159     if studentid_count=0
160     then
161         message:='The B# is invalid';
162     else
163         select count(*) into studentid_class_cnt from classes where
164         classid=studentid_classid;
165         if studentid_class_cnt=0
166         then
167             message:='The classid is invalid';
168         else
169             select count(*) into studentid_class_enroll_cnt from enrollments
170             where classid=studentid_classid and b#=studentid_b#;
171             if studentid_class_enroll_cnt=0
172             then
173                 message:= 'The student is not enrolled in the class';
174             else
175                 select dept_code,course# into
176                 studentid_dept_code,studentid_course# from classes where
177                 classid=studentid_classid;
178
179                 select count(*) into studentid_prereq_cnt from classes
180                 cl,prerequisites p where classid in (select classid from
181                 enrollments where classid<>studentid_classid and
182                 b#=studentid_b#) and
183                 cl.dept_code=p.dept_code and cl.course# = p.course# and
184                 p.pre_dept_code=studentid_dept_code and
185                 p.pre_course#=studentid_course#;
186
187                 if studentid_prereq_cnt>0

```

```

178         then
179             message:='The drop is not permitted because another class uses
180             it as a prerequisite';
181
182         else
183             delete from enrollments where b#=studentid_b# and
184             classid=studentid_classid;
185             message:='Student dropped successfully';
186
187             select class_size into studentid_class_size from classes where
188             classid=studentid_classid;
189
190             select count(*) into studentid_class_size_cnt from enrollments
191             where b#=studentid_b#;
192
193             if studentid_class_size=0
194             then
195                 message1:='The class now has no students.';
196             end if;
197
198             if studentid_class_size_cnt=0
199             then
200                 message2:='This student is not enrolled in any classes.';
201             end if;
202             message:=message||' '||message1||' '||message2;
203
204         end if;
205     end if;
206 end drop_student ;
207
208 end course_registration;
209 /
210 show errors

```