DOCUMENTATION

The following documentation provides detailed explanation about every procedure that is created.

The PL/SQL package created for the application is named: **student_registration_system.**

Following is the description about various procedures, triggers used in all the questions.

1) Sequence used: **logid_seq**.

The sequence generates auto incrementing logid in logs table when new log records are inserted into the logs table. The sequence starts with 1000.

2) Procedures used are:

i. **sp_show_students**: This procedure is used to display all the tuples in the *students* table.
ii. **sp_show_courses**: This procedure is used to display all the tuples in the *courses* table.
iii. **sp_show_course_credit**: This procedure is used to display all the tuples in the *course_credit* table.
iv. **sp_show_prerequisites**: This procedure is used to display all the tuples in the *prerequisites* table.
v. **sp_show_classes**: This procedure is used to display all the tuples in the *classes* table.
vi. **sp_show_enrollments**: This procedure is used to display all the tuples in the *enrollments* table.
vii. **sp_show_grades**: This procedure is used to display all the tuples in the *grades* table.
viii. **sp_show_logs**: This procedure is used to display all the tuples in the *logs* table.

3) Procedures used are:

i. **sp_classes_taken_by_student:** When B# is given as an input parameter, the procedure lists every class the student has taken or is taking.
All attributes of the class table are listed except the limit and class_size along with the letter grade and number grade the student received for that class (including null grades)
If the classid is not present in the students table, the procedure reports: “The B# is invalid.”
If the student has not taken any course, the procedure reports: “The student has not taken any course.”

4) Procedures used are:

i. **sp_find_prerequisites**: When *dept_code* and *course#* are given as input parameters, this procedure returns all courses that need this course as a prerequisite, including the courses that need this course as prerequisite directly or indirectly.
The prerequisites with more than two levels away are also returned.

5) Procedures used are:

i. **sp_get_class_students:** When *classid* is given as input parameter, this procedure can list the classid and course title of the class as well as all the students (along with their B# and firstname) who took or are taking the class.
If the classid is not present in the classes table, the procedure reports: "The classid is invalid".
If no student took or is taking the class, the procedure reports: "No student has enrolled in the class.

6) Procedures used are:

i. **sp_enroll:** When *B#* of a student and *classid* of the class are provided as parameters, this procedure enrolls the student into the class.
If the student with the entered B# is not in the students table, the procedure reports: "The B# is invalid."
If the class is not in the classes table, the procedure reports: "The classid is invalid."
If the enrollment of a student into a class causes the class_size to exceed the limit, for example, say there are 40 students enrolled in a class with limit equals to 40, then the enrollment of a new student is not possible. Hence the enrollment is rejected and the procedure outputs: "The class is full."
If a student is already in the class, the procedure reports: "The student is already in the class."
If the student is already enrolled in three other classes and in the same semester and same year, for example, say a student with B# B002 is already enrolled to classes c0001, c0003 and c0006, all three in Fall 2015, then the procedure reports: "You are overloaded." However, the enrollment of the student is still accepted in this case.
If the student is already enrolled in four other classes and in the same semester and same year, the procedure reports: "Students cannot be enrolled in more than four classes in the same semester."
If the student has not completed the required prerequisite courses with grade at least C, the enrollment is rejected and the procedure reports: "Prerequisite not satisfied."

When a successful enrollment of a student into a specific class occurs, a trigger is used to implement the corresponding update of the values in the table.

7) Procedures used are:

i. **sp_drop_enrollment**: When *B#* of the student and *classid* of the class are provided as parameters, this procedure drops a student from a class.
If the student is not in the students table, the procedure reports: “The B# is invalid.”
If the classid is not in the classes table, the procedure reports: “The student is not enrolled in the class.”
If a class with classid say c0002 of course say CS432 is a prerequisite of a course say CS532 with classid c0004, then dropping a student from class c0002 will cause the violation of prerequisite requirement. Therefore, the drop attempt is rejected and the procedure will report: “The drop is not permitted because another class uses it as a prerequisite.”
If the class is last class of the student, the procedure reports: “This student is not enrolled in any class.”
If the dropped student was the last student of the class, then the procedure reports: “The class now has no students.”

8) Procedures used are:

i. **sp_delete_student:** When *B#* of a student is given as a parameter, this procedure deletes the corresponding student from the student table.
Suppose a student with B# B005 is not in the students table, the procedure reports: “The B# is invalid.”
When a student is deleted from the students table, all tuples in the enrollments table involving that student are also dropped.” This is implemented using a trigger.

9) Triggers used: **trigger_students** and **trigger_enroll.**

The trigger updates data on successful Insert or Delete on students table as well as when a student is successfully enrolled into or dropped from a class, i.e. in the enrollments table.