Introduction to Databases
Assignment 2

Instructions: Design your database as outlined in the scenario below. You must specify the field name, data type, description, validation rules as needed, and a primary key information for each table. You must use good naming practices. State any assumptions that you have made in addition to the ones given below. Your assumptions must not contradict the information included here.

Please use A2_ as a prefix in all your tables in this assignment, so in keeping with good naming standards the first table should be named A2_STUDENT

Scenario: The dean has asked you to create a new database that will store information about students enrolled in School of Information and Library Science. The initial database that you create will be just a prototype, and as such will not capture all of the assumptions within the real world constraints. Your database will capture important information about the courses offered during the current term, and students' registration for courses. You have looked at the ER diagram (that someone else generated) and you have ascertained that three tables will be required:

(1) The student table has information about currently enrolled students. You should capture the following information:

- SID - the student identification number
- lastNm - the student's last name
- firstNm - the student's first name
- street - the student's local street address
- city - the city in which the student lives
- state - the state in which the student lives
- zip - the student's zip code
- credits - the number of credits the student has earned to date

The primary key for this table is the SID.

(2) The course table has information about the courses being offered this term. The table has 4 fields:

- courseNo - the number of the course, including department prefix and section number (e.g., LIS490DB)
- courseNm - the name of the course (e.g., Database)
- location - the building and room number in which the course is taught (e.g., GSLIS 131)
- instructor - the last name of the course instructor. Default value is "Staff", indicating an instructor has not yet been found for the course.

The primary key for this table is courseNo.

(3) The registration table has information about which students are registered for which courses. It has 2 fields:

- SID - which refers to student.SID as a foreign key
- courseNo - which refers to course. courseNo as a foreign key
- You need a compound primary key consisting of both SID and courseNo for this table.

Assignment details:
Part A – Create the tables listed above. In addition to the primary keys, you should apply validation rules to ensure that the values entered are consistent with the domain of each attribute.

Part B – Define the relationships between each of these tables. You can find information from help, or the constraint option on right-click menu of each table. Relationships in Oracle SQL Developer reflect foreign key constraints. You have again looked at the ER diagram and observed that there is a one-to-many relationship between student and registration, and a one-to-many relationship between course and registration. Capture those relationships in your database. Choose to enforce referential integrity, cascade delete. That means that if a student withdraws from school, and the instructor removes his or her record from the database, then the database should require the removal of any additional registrations for that student. Similarly, if an instructor drops a course (e.g., not enough students registered for it), all registrations for that course will be deleted.

Part C – Add records to each table. Enter data for at least 8 students and 5 courses. Be sure that you have data that will allow you to get answers for each query in part D. Enter records that will test that your constraints are in place correctly. In addition you should enter records that enable you to test the queries in part D. For example, be sure that you have "Staff" listed as the instructor for at least one course so that you can complete Query B.

Part D – Now that you have created your database, you need to test that the relationships are working. The best way to do this is to run some sample queries. You look back at your interview notes and find several questions that the dean would like the database to answer. Where X appears in the question, you should hard-code a value from the test data that you entered in Part C.

a) How many credits does the student with SID X have?

b) List the course number and name of all courses for which we still need to find instructors (instructor = "Staff"). Order the records in your answer by course number.

c) Which courses meet in room X, and who are the instructors for these courses?

d) List the names and SIDs of all students registered for course X. Order by last name. What are the names and numbers of courses for which student X is registered?

e) How many students registered for course X?

f) Give the name and SID of the student who has the most credits to date. If more than one student has earned the same maximum number, your answer should include records for each of them.

Grant the instructors access to your tables.

Submission instructions
Upload a single file that contains your table creation statements and referential integrity constraints (parts A and B) along with your sql queries (part D) to moodle. Don't forget to grant the instructors access to your tables.

Grading:

Table and relationship definitions	35 points
Good testing records	20 points
Queries	45 points