

Assignment #1: Oracle Database Mechanics**Due: February 7th, at the start of class****Part 1:**

The following is a partial description of a database of information typically recorded in the grade book of college professors. Create an Oracle database with contents that exactly match the tables described below.

CATALOG(CNO, CTITLE)

The CATALOG table contains information about course numbers and titles of courses taught by a particular professor.

<u>CNO</u>	<u>CTITLE</u>
AC200	Fixed-Asset Accounting
CS226	Introduction to Programming I
CS227	Introduction to Programming II
CS343	Introduction to Linux
MK173	Hi-Tech Marketing

STUDENTS(SID, FNAME, LNAME, MINIT)

The STUDENTS table contains information about the students taught by a particular professor.

<u>SID</u>	<u>FNAME</u>	<u>LNAME</u>	<u>MINIT</u>
1111	Susan	Jones	B
2222	Samuel	Corn	A
3333	Willie	Smith	
4444	Jerome	Franks	B
5555	Sally	Williams	G
6666	Willie	Brown	F
7777	Jonathan	Miller	
8888	Mary	Jones	C
9999	Ada	Jones	

COURSES(TERM, SECTNO, CNO, A, B, C, D)

The COURSES table contains various classes that have been taught by a particular professor. The TERM attribute corresponds to the term (such as Spring 2016 or Fall 2017) in which the course was taught; the SECTNO is a unique section number assigned by the registrar within a term. CNO is a foreign key to the CATALOG table. A, B, C, and D represent numeric values that correspond with each letter grade (e.g., A=90, B=80, C=70, D=60).

<u>TERM</u>	<u>SECTNO</u>	<u>CNO</u>	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>
F16	1031	CS226	90	80	65	50
F16	1032	CS226	90	80	65	50
S17	1031	CS227	90	80	70	55
F17	1032	CS343	92	82	75	60

COMPONENTS(TERM, SECTNO, COMPNAME, MAXPOINTS, WEIGHT)

The COMPONENTS table contains the various grading components (such as homework, quizzes, and exams) for a particular course taught by the professor. For each course taught (identified by TERM and SECTNO, so the combination of TERM and SECTNO is a foreign key to the COURSES table), each row holds the name of the grading component, the maximum points assigned to this component, and the weight of this component relative to the other components.

<u>TERM</u>	<u>SECTNO</u>	<u>COMPNAME</u>	<u>MAXPOINTS</u>	<u>WEIGHT</u>
F16	1031	Exam1	100	30
F16	1031	Quizzes	80	20
F16	1031	Final	100	50
F16	1032	Programs	400	40
F16	1032	Midterm	100	20
F16	1032	Final	100	40
S17	1031	Exam	100	50
S17	1031	Project	100	50
F17	1032	Paper	100	100

ENROLLS(SID, TERM, SECTNO)

The ENROLLS table shows which student was enrolled in which course taught by the professor. There are two foreign keys in the table: SID (which refers to the STUDENTS table) and the combination of TERM and SECTNO (which refers to COURSES).

<u>SID</u>	<u>TERM</u>	<u>SECTNO</u>
1111	F16	1031
2222	F16	1031
4444	F16	1031
5555	F16	1032
6666	F16	1032
3333	F16	1032
1111	S17	1031
3333	S17	1031
3333	F17	1032

SCORES(SID, TERM, SECTNO, COMPNAME, POINTS)

SCORES records the grading component scores (or points) for each student enrolled in a course. It has two foreign keys: the combination of SID, TERM and SECTNO (which refers to ENROLLS) and the combination of TERM, SECTNO and COMPNAME (which refers to COMPONENTS).

<u>SID</u>	<u>TERM</u>	<u>SECTNO</u>	<u>COMPNAME</u>	<u>POINTS</u>
1111	F16	1031	Exam1	90
1111	F16	1031	Quizzes	75
1111	F16	1031	Final	95
2222	F16	1031	Exam1	70
2222	F16	1031	Quizzes	40
2222	F16	1031	Final	82
4444	F16	1031	Exam1	83
4444	F16	1031	Quizzes	71
4444	F16	1031	Final	74
5555	F16	1032	Programs	400
5555	F16	1032	Midterm	95
5555	F16	1032	Final	99
6666	F16	1032	Programs	340
6666	F16	1032	Midterm	65
6666	F16	1032	Final	95
3333	F16	1032	Programs	380
3333	F16	1032	Midterm	75
3333	F16	1032	Final	88
1111	S17	1031	Exam	80
1111	S17	1031	Project	90
3333	S17	1031	Exam	80
3333	S17	1031	Project	85

Part 2:

- a. Formulate each of the queries below and run them on the database created above:
 1. Get the course number and title of courses in which Willie Smith has enrolled.
 2. Get the names of students who have been enrolled in the Intro to Programming II course.
 3. Get the SID value of students who did not enroll in any class during the Spring 2017 term (S17).
 4. Get the SID of students who have enrolled in all the courses in the catalog.
 5. Get the names of the student(s) who have not enrolled in any course.
 6. Get the names of the student(s) who have enrolled in the highest number of courses (sections).
 7. Get the term, section number, and course titles of all classes, along with their total enrollment (i.e., number of students enrolled).
 8. Get the term, section numbers, and course titles of courses that have enrollment greater than or equal to the average enrollment in all classes (sections) that have any students enrolled.
 9. Get the student IDs of students, the terms and section numbers of courses they have enrolled in, the component names of the courses, the student scores in the components of the courses, and the weighted score of each component.
 10. Given a term and section number of a course (for example, F16 and 1031. The query should prompt for the term and section number), get the student ID's and first and last names of all students enrolled in the class, along with each student's course average (rounded off to the nearest integer). The course average is the sum of the weighted averages of the individual component scores for that student for that class.
- b. For the grade book database, write SQL expressions to perform the following updates to the database:
 1. Delete the student named Jonathan Miller from the students table.
 2. Drop the student with sid = 4444 from the Fall 2016 course with sectno = 1031. Note: DON'T change the structure / definition of the table(s) themselves (i.e., keep primary and foreign keys in force).
 3. Give all students in the Fall 2016 course with SECTNO = 1031 a total of 10 extra points in the Exam1 component.
 4. Enroll all students in the Fall 2016 course with SECTNO = 1031 into the Spring 2017 course with SECTNO = 1031.
 5. Delete all the courses from the courses table that have enrollments of fewer than five students. Again, do NOT change the definition or structure of any tables.

Please note that you should provide queries that work for every possible instance of the database -- not just for the particular one you created above. For example, for the first query, it is NOT correct to write a query that explicitly asks for students enrolled in course CS227 -- though that would give the same answer for the sample data that is provided.

To Turn In:

For the CREATE TABLE and INSERT statements from Part 1, and each of the ten queries in Part 2a and the five UPDATE/INSERT/DELETES in Part 2b, turn in a printout with **both** the SQL statement(s) **and** the output from the query. **Also, post the SQL (only – the .sql file) for Part 2 (a & b) to Blackboard.**

Your full answer to the assignment will consist of printouts for each part of the assignment:

- * the SQL code for the CREATE TABLE and INSERT statements to build the tables (Part 1),
- * the SQL code and output of queries to display the (initial) contents of each table,
- * the SQL code and output of the queries asked for in Part 2a,
- * the SQL code and output necessary to demonstrate Part 2b (e.g., before and after listings of tables) .