## unjChapter Nine Assignment

To perform the following assignments, refer to the tables created in the “*New Build Script Chapter 9.sql*” script in Black Board Learn/Course Content/ Chapter 9 Joining Data from Multiple Tables. Generate and test two SQL, queries for each of the following asks:

1. The SQL statement needed to perform the stated task with the traditional approach.
2. The SOL statement needed to perform the stated task with the JOIN keyword.
3. Create a list of all students (first name and last name separated by a comma) listed as “Student”, and their professors name (first name and last name separated by a comma) listed as “Professor”.

*select distinct student.stu\_fname ||',' || stu\_lname as "student", professor.PROFESSOR\_FIRST\_NAME || ',' || professor.PROFESSOR\_LAST\_NAME as professor from student, professor where professor.prof\_num=student.prof\_num;*

*B*

*Select distinct student.stu\_fname ||',' || stu\_lname as "student", professor.PROFESSOR\_FIRST\_NAME || ',' || professor.PROFESSOR\_LAST\_NAME as professor from student natural join professor;*

*Screen Shot goes here.*



1. Create a distinct list of all professors (first name and last name separated by a space) listed as “Professor”, and their courses listed as “Courses”.

*Select distinct PROFESSOR\_FIRST\_NAME ||' '|| PROFESSOR\_LAST\_NAME as professor, course.crs\_code as course from professor, course , class*

*where course.crs\_code=class.crs\_code and class.prof\_num=professor.prof\_num;*

*select distinct PROFESSOR\_FIRST\_NAME ||' '|| PROFESSOR\_LAST\_NAME as professor, course.crs\_code as course*

*from professor join class on class.prof\_num=professor.prof\_num*

*join course on course.crs\_code=class.crs\_code;*

*Screen Shot goes here.*



1. Create a distinct list of all professors (first name and last name separated by a space) listed as “Professor”, and the description of the courses they teach listed as “Courses”.

*Select distinct PROFESSOR\_FIRST\_NAME ||' '|| PROFESSOR\_LAST\_NAME as professor, course.CRS\_DESCRIPTION as course from professor, course , class*

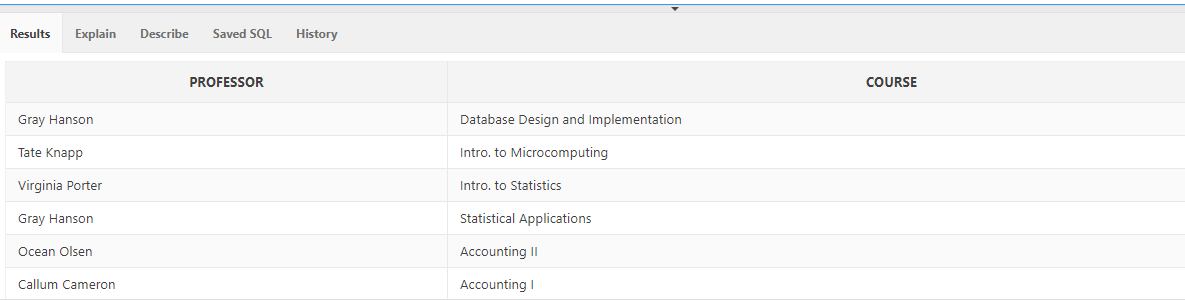
*where course.crs\_code=class.crs\_code and class.prof\_num=professor.prof\_num;*

*select distinct PROFESSOR\_FIRST\_NAME ||' '|| PROFESSOR\_LAST\_NAME as professor, course.CRS\_DESCRIPTION as course*

*from professor join class on class.prof\_num=professor.prof\_num*

*join course on course.crs\_code=class.crs\_code;*

*Screen Shot goes here.*



1. Create a list of students (first name and last name separated by a comma) listed as “Student” enrolled in course, also list the professors name (first name and last name separated by a comma) listed as “Professor” of the courses, the course description listed as “Course Description” and the student’s grade for the course listed as “Grade”.

*select student.stu\_fname ||',' || student.stu\_lname as "student", professor.PROFESSOR\_FIRST\_NAME || ',' || professor.PROFESSOR\_LAST\_NAME as professor, course.CRS\_DESCRIPTION as CRS\_description, course.crs\_code as course, enroll.ENROLL\_GRADE as grade*

*from student, professor, Enroll, course, class*

*where student.stu\_num=enroll.stu\_num*

*and student.prof\_num=professor.prof\_num*

*and enroll.class\_code=class.class\_code*

*and class.crs\_code=course.crs\_code*

*order by student.stu\_lname desc;*

*b.*

*select student.stu\_fname ||',' || stu\_lname as "student", professor.PROFESSOR\_FIRST\_NAME || ',' || professor.PROFESSOR\_LAST\_NAME as professor, course.CRS\_DESCRIPTION as CRS\_description, course.crs\_code as course, enroll.ENROLL\_GRADE as grade*

*from student join enroll on student.stu\_num=enroll.stu\_num*

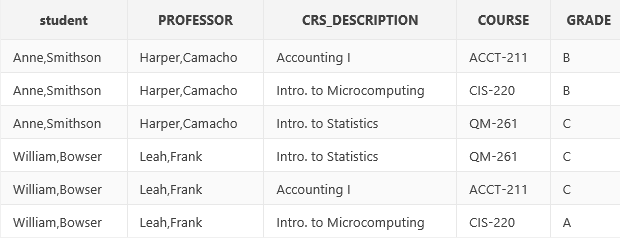
*join professor on professor.prof\_num=student.prof\_num*

*join class on class.class\_code=enroll.class\_code*

*join course on course.crs\_code=class.crs\_code*

*order by student.stu\_lname desc;*

*Screen Shot goes here.*



1. Create a list of all the stores and their managers names (first and last separated by a space including there title) list the stores even if there is no manager. Hint: There is only one way to do this problem and it in involves outer joins, but **DO NOT** use the **JOIN** keyword.
2. *select employee\_x.EMP\_fname||' '||employee\_x.EMP\_lname as manager, store.store\_name as store*

*from employee\_x, store*

*where employee\_x.store\_code (+)= store.store\_code*

*and job\_code(+)='MGR';*

*Screen Shot goes here.*

