

## Unit 05 Instructions - Querying Multiple Tables

### Introduction to Enterprise Relational Databases

## Instructions

Read these instructions all the way through before you start.

Download the answer document. Be sure to put your name at the top of the answer document where that's indicated. Answer the questions below in that document and when you are done, upload it into Blackboard.

## Overview

In this unit we learn how join multiple tables in a query. This is the heart of query concepts in relational database. Since a well-designed database scatters facts among a number of tables, we use queries to “knit” together information from multiple tables. Queries allow us to present information in a more natural way.

### Creating the College Database

In this unit you'll use the College database for the first time in an assignment. To create the College database, download script **CollegeCreate.x.sql** and run it. It's in the **Databases used in the Class** folder inside **Units**. Once you run the script, be sure to click on the Refresh button in the upper right of the tree view to see the College database in the tree view. There's a second script named **CollegeDump.x.sql** that will display the records from all the tables in the College database.

In the folder with the create script is a PDF that will give you an overview of the College database. Reading though that before you start writing queries will speed up your work.

## Preparing your SQL

Be sure to follow these points in writing your queries.

- Make sure you include the comment and USE statement at the top of your SQL

```
-- Unit 5.1 - Your Name  
  
USE college ;  
  
-- List all departments and their faculty members.
```

- Beautify your SQL.
- Make your column headers look exactly like the examples shown.
- In your screen shots of the Result Grid you need to snip only first 6-10 rows. If you're not sure about this, see Unit 02 instruction for details on how to snip.

Note: Your data may be different than the data in the rows of the examples show below. The data itself may be different and you may get more or fewer rows that shown in the examples.

## Steps

In each of these queries you're shown the columns to display in your result.

1. [5] Write a query that lists the name of departments and the last and first names of faculty members that work in each department. Sort on all columns from left to right.

Department	Last	First
------------	------	-------

2. [5] Write a query that lists women students majoring in math or computer science. Sort on all columns from left to right.

Last	First	Major
Bennett	Betty	Computer Science

3. [5] Write a query that lists students' last and first names, the courses taken and the grades earned. Sort by Last, First and then Course. Tip: You'll need to join four tables to make this work. Section is one of the tables you'll need to join, even though no fields from Section appear in the result of the

query.

Last	First	Course	Grade
Alvarez	Kenneth	IT 1000 - Intro to Computer Systems	3.2
Alvarez	Kenneth	IT 1000 - Intro to Computer Systems	3.8

4. [5] Create a query that lists departments and the courses offered, including department(s) that don't have any courses.

Department	Course
------------	--------

5. [5] Write a query that lists only Majors that have no students in that Major. If a major has students it should not show in the results. (This may not be the major you get)

Major
Anthropology

6. [7] Write a Do-It-Yourself (DIY) query of your own design for the College database that inner joins at least four tables. One of the tables must be the Semester table. This must not be a query from elsewhere in the course and the query must have some meaningful purpose. Give the columns good names and sort the results as appropriate. Format the output neatly.

Write a one sentence explanation of the query. In your explanation tell me the purpose of the query, not how it works. i.e. don't tell me "I joined A to B and B to C..." I can see that by looking at your SQL.

2. [7] Write a DIY query of your own design for the College database that uses at least one **outer** join and shows at least one record that doesn't match on the join. This must not be a query from elsewhere in the course and the query must have some meaningful purpose. Give the columns good names and sort the results as appropriate. Format the output neatly. Write a one sentence explanation of the purpose of the query.