# Project 2

Relational Database Coding

152-125

BD05370_

Due: April 18, 2015

11:59 p.m.

1. Each script must begin with **comment** line(s) that list: (1) the author’s name, (2) project number, (3) request number, and (4) date accomplished. (-1/script if not done)

1. The submitted solution file will consist of input query statements within a single script file labeled Proj2Lastname.sql where Lastname is the student’s last name. Submit the file using the Black Board Submit Projects tab.
2. Projects must by submitted by the due date. (-1 point per day late)
3. All project work must be accomplished individually.

REQUESTs:

**Request 1:** (5 pts) Write/execute SQL statement(s) that create a new Employees table that has the following characteristics:

1. The table name is *EMPS*.
2. *EMPS* will contain all the rows and columns currently in the *employees* table, with the exception that employee #178 (Kimberly Grant)’s row must no longer be in the new table, since that employee has left the company. (Note that some constraints may be copied from the original EMPLOYEES table when EMPS is created. Verify for yourself which constraints exist on the new *EMPS* table by executing a query against an appropriate data dictionary view that returns the constraint name, constraint type, table name and search condition for *EMPS*.)
3. There is a primary key constraint on employee\_id.
4. There is an additional column that contains the employee first name concatenated with the employee last name with a blank space in-between the two names, a column which is named FULL\_NAME.
5. There is a constraint in place that will prevent anyone from adding a new row to the table that contains a department\_id value that is not present in the *departments* table.
6. There are constraints in place that will prevent anyone from adding a row to the table that contains nulls in the first\_name, last\_name, hire\_date, FULL\_NAME and job\_id columns.
7. There is a constraint in place that will prevent anyone from adding a row to the table that contains a duplicate email address, although null values are allowed in the email address column.

**Request 2:** (5 pts) Write/execute a SQL statement that lists only the following four columns from the *EMPS* table: employee\_id, FULL\_NAME, salary and department\_id.

The order of the rows in this listing must be in order by employee **last name**, even though the employee last name field is not to be a part of the listing.

**Request 3:** (5 pts) Create a view, named as you choose, that will use the *EMPS* table as the base underlying table. This view must group the rows of the *EMPS* table by job\_id, and for each group must provide the job\_id, the average salary, the maximum salary, the minimum salary and the difference between the maximum and minimum salaries. In addition to turning in the statement(s) which created the view, prove that the view functions correctly by executing a SELECT \* query against the view to list out the data “in” the view.

**Request 4:** (5 pts) Write/execute a single SQL statement that runs against the *EMPS* table that lists all employees’ ids, their FULL\_NAME and the FULL\_NAME of their manager. If an employee has no manager, then the FULL\_NAME of the manager on the row for that employee may be left blank, but that employee must still be in the listing. The column names on the listing must clearly indicate what the column represents.

**Request 5**: (5 pts) Create an exact copy of the data in the *departments* table, with the copy named ***DEPS.*** The only constraint present on the *DEPS* table must be that the department\_id column is a primary key. Verify for yourself which constraints exist on the *DEPS* table by executing a query against an appropriate data dictionary view that returns the constraints for *DEPS*. Then do the following things to this *DEPS* table in the order listed:

1. Delete all the row(s) for any department that is in location 1400.
2. Add a new row to the table for a department with an id of 200, a name of “Human Resources,” a null value for manger\_id, and a location\_id of 2700.
3. Change the name of the Administration department to the “Operations” department.
4. Change the case of all of the values in the Department\_Name column so that the letters are all upper case.
5. Add a new column to the *DEPS* table, with the new column named SECURITY\_ID. The new column must be numeric, and able to hold a 6 digit number. The contents of the new column must be 101 in every row of existing data after adding the column.

3/2015