**Unit 5 Graded Exercise 1**

The following questions come from the “Check your understanding” examples in Chapter 15 in your textbook.

After you are finished, please submit a Microsoft Word file that contains screenshots of the SQL queries and the output, along with a comment in each query containing your name. Your document should be named **U5\_GradedExercise1\_Lastname.docx**.

(15-1) Question 1:

1. Here is some information about the size of two tables:

Table 1: 10 columns 100,000 rows

Table 2: 10 columns 50,000 rows

What is the maximum size of the union of these two tables? What is the maximum size of the inner join of the tables? By “maximum size” I mean the maximum number of columns and rows.

2. Write a *select* statement to form the *union* of the *twos* table and the threes table.

(15-2) Question 2:

Write a *select* statement to form a union of the *twos* table and the *threes* table. Use *union all*. How does this differ from using a regular *union*?

(15-3) Question 3:

1. What is wrong with this *select* statement?

*select number\_2*

*from twos*

*union*

*select number\_3,*

*word\_3*

*from threes;*

2. Goal 1: Show that a *union* is similar to an *insert* statement in that it can add new data to the result table.

3. Goal 2: Show a *union* that uses more than two *select* statements. The following select statement shows the number of lunches that each employee will attend, but it does not account for Carol Rose or Paula Jacobs because they are not attending any lunches. Modify this statement to show that these two people will not attend any lunches.

*select a.first\_name,*

*a.last\_name,*

*count(b.lunch\_id) as number\_of\_lunches*

*from l\_employees a*

*inner join l\_lunches b*

*on a.employee\_id = b.employee\_id*

*group by a.first\_name,*

*a.last\_name;*

(15-4) Question 4:

Modify the following *union*. Add an *order* by clause to it to sort the rows by the last name. Try all four methods. Which ones work?

*select a.first\_name,*

*a.last\_name,*

*count(b.lunch\_id) as number\_of\_lunches*

*from l\_employees a*

*inner join l\_lunches b*

*on a.employee\_id = b.employee\_id*

*group by a.first\_name,*

*a.last\_name*

*union all*

*select 'Carol',*

*'Rose',*

*0*

*from dual*

*union all*

*select 'Paula',*

*'Jacobs',*

*0*

*from dual;*

(15-5) Question 5:

Goal: Show that a *union* can add new rows of data to a table. This is similar to what an *insert* statement does.

First, create a *select* statement that lists all the columns and rows of the *l\_employees* table and uses a *union all* to add the following new employee. Then save the result table as a new table called *sec1505\_employees*

Employee\_id: 301

First\_Name: Gail

Last\_Name: Jones

Dept\_code: Sal

Hire\_date: Feb 15, 2011

Credit\_limit: $25.00

Phone\_number: (null)

Manager\_id: 202

(15-6) Question 6:

Run the code from this section. Use the methods of **section 7-12** to examine the datatypes of the columns of the beginning tables and of the new view created by the *union*. Have any of the datatypes changed in the process of forming the union?

(15-7) Question 7:

Modify the following select statement. Convert the datatypes of all the columns to text. (Actually, sometimes this code will work as it is and the conversion of the datatypes is done automatically for you behind the scenes.)

*select date\_1,*

*date\_1,*

*date\_1*

*from sec1507\_first*

*union*

*select number\_2,*

*word\_2,*

*date\_2*

*from sec1507\_second;*

(15-8) Question 8:

Modify the following *select* statement to make it work. Add one more columns to the second *select* statement. You can use either a null or a literal value.

*select number\_1,*

*word\_1,*

*date\_1*

*from sec1508\_more\_columns*

*union*

*select number\_2,*

*word\_2*

*from twos;*

(15-10) Question 9:

The following *select* statement creates a *union* of the *twos* table with the *threes* table. Add a new column to show the table from which each row comes.

*select number\_2,*

*word\_2*

*from twos*

*union*

*select number\_3,*

*word\_3*

*from threes*;