

Lab 04 – SQL (DML)

Objectives:

The purpose of this lab is to introduce you to the DDL set of statements in SQL. By writing SQL to create tables, constraints, and views, you will have the tools needed to implement database designs that you will create later in the course. By finishing this lab, the student will be able to:

- create, modify, and drop tables based on design specifications provided,
- inserting new data into tables, update data in tables, and delete data from tables while considering referential integrity,
- enforce constraints on tables to ensure data integrity and consistency,
- create a table using the structure and data from an existing table,
- Import data into a table from other tables.

Submission:

Your submission will be a single text-based **.sql** file with the solutions provided.

DBS211_L04_Group_Number.sql

Your submission needs to include a comment header block and be commented to include the questions and the solutions. Make sure every SQL statement terminates with a semicolon.

Example Submission

```
-- *****
-- Name: Your Name
-- ID: #####
-- Date: The current date
-- Purpose: Lab 04 DBS211
-- *****

-- Q1 SOLUTION --
SELECT * FROM TABLE;

-- Q2 SOLUTION --
SELECT * FROM TABLE;
```

Locate, select, and submit the file to the Lab 04 link.

Setup

Create a new worksheet in SQL developer and add an appropriate comment header that includes your name, student id, the date and the purpose of the file (i.e. DBS211 – Lab 04).

Style Guide

Your SQL should be written using the standard coding style:

- all keywords are to be upper case,
- all user-defined names are to be lower case, (example: table and field names)
- there should be a carriage return before each major part of the SQL statements (i.e. before SELECT, FROM, WHERE and ORDER BY)

See the following sample:

```
SELECT columns
FROM tables
WHERE conditions
ORDER BY column1, column2;
```

Marking Scheme

Question	Points	Question	Points
1	0.5	9	1
2	1	10	1
3	0.5	11	0.5
4	1	12	0.5
5	0.5	13	1
6	0.5	14	0.5
7	0.5	15	0.5
8	0.5		

Total: 10

Grade Policy

- Solutions with errors do not get any marks. (They get zero.)
 - Execute your *.sql* file using the “Run Script” button to make sure there is no errors in your file.
- If your result in a question does not match the sample output results, you do not get any marks.
- You do not receive marks for the missing or incomplete solutions.
- The name of the tables you will create in this lab have to match exactly with the names given this lab document.

Tasks:

Add

```
SET AUTOCOMMIT ON;
```

under the comment header and execute it.

Consider the following table specifications:

Remember to comment the question number for each solution.

DML

1. Create a new **empty** table **employee2** the same as table **employees**. Use a single statement to create the table and insert the data at the same time.
2. Modify table **employee2** and add a new column **username** to this table. The value of this column is not required and does not have to be unique.
3. Delete all the data in the **employee2** table
4. Re-insert all data from the **employees** table into your new table **employee2** using a single statement. Add constraints to employee2 table same as in the employees table.
5. Create a statement that will insert yourself as an employee into **employee2**.
 - a. Use a unique employee number of your choice (**Hint: Find the highest value of the employee number in the db211_employees table, increase the value by one and use it as your employee number.**)

To find the highest value of the employee number you can sort the rows in the descending order. The first row will then contain the highest value.

Or, you can run the following statement (Do not include this statement in your submission.)

```
SELECT max(employee_number)
FROM db211_employees;
```

This statement returns the maximum value of the employee number in table db211_employees .
 - b. Use your school email address
 - c. *Your extension is 'x2222'*
 - d. Your job title will be "Cashier"
 - e. Office code will be 4
 - f. You will report to employee 1088
 - g. You do not have any username
6. Create a query that displays your, and only your, employee data.
7. Create a statement to update your job title to "Head Cashier".

Hint: Be careful. You may update other rows or all rows in the employee table. You only need to update one row which belongs to you and update your job title. Make sure that your query updates only one employee using a WHERE clause.

8. Create a statement to insert another fictional employee into **employee2**. This employee will be a “Cashier” and will report to you. Make up fake data for the other fields. The fake employee does not have any username.
9. Create a statement to delete yourself from **employee2**. Did it work? If not, **why**?
10. Create a statement to delete the fake employee from **employee2** and then rerun the statement to delete yourself. Did it work? Explain why?
11. Create a **single** statement that will insert both yourself and the fake employee at the same time. This time you and the fake employee will report to 1088.
12. Create a **single** statement to delete both yourself and the fake employee from **employee2**.
13. In table **employee2**, generate the email address for column **username** for each student by concatenating the first character of employee’s first name and the employee’s last name. For instance, the username of employee Peter Stone will be **pstone**. NOTE: the username is in all lower case letters.
14. In table **employee2**, remove all employees with office code 4.
15. Drop table **employee2**.