Lab – Introduction to SQL – Part 2 of 2

## Overview

In this lab, we will practice using DDL and DML SQL statements using our account on Microsoft SQL Server. Much of what we did last week using the GUI is now done this week via SQL.

### Learning Objectives

Upon completion of this learning unit you should be able to:

* Describe and use basic SQL commands on SQL Server
* Explain how and why SQL is used, and why it’s important
* Compare and contrast DML and DDL
* Use SQL commands to create metadata structures (tables and constraints) and perform **CRUD** operations to add data.

### What you will need

By now you should be proficient at logging on to your hosted SQL Server account. If you aren’t refer to the previous to Labs and Practice.

You’ll also need to complete part 1 of this lab before attempting these exercises. Part 1 of this lab is available in Blackboard.

All queries should be written in the Lab03\_Introduction\_to\_SQL\_Queries.sql File available in Blackboard. You should have entered all of the queries from part 1 into this file already.

### Lab Goals

This lab consists of 3 sections:

1. In section one we will explore a new aspect of the Fudgemart database schema for use in the lab. This section and the lab questions are available in Part 1 of this lab in Blackboard.
2. In section two; you will follow along entering and executing the SQL commands that are displayed in the screen shots. This lab will give you a brief explanation of what the commands are doing to help you through the muddy points. This section and the lab questions are available in Part 1 of this lab in Blackboard.
3. In section three, you’re on your own. You will be required to create and execute SQL to solve a variety of business problems.

This is a reminder from Part 1 of what our timesheets data looks like

**Table: fudgemart\_timesheets: columns**

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Default Value | Allow Null? |
| timesheet\_id | Int identity(1,1) |  | No |
| timesheet\_payrolldate | Datetime |  | No |
| timesheet\_employee\_id | Int |  | No |
| timesheet\_hours | decimal(3,1) | 40.0 | No |

**Table: fudgemart\_timesheets: constraints**

|  |  |  |
| --- | --- | --- |
| Constraint Name | Constraint Type | Data |
| pk\_timesheet\_id | Primary key | timesheet\_id |
| fk\_timesheet\_employee\_id | Foreign key | timesheet\_employee\_id REFERENCES fudgemart\_employees(employee\_id) |
| ck\_timesheet\_hours | Check | timesheet\_hours between 0 and 60 |

## Section 3: On your own: Fudgemart Employees and Timesheets

In this section we will create the **fudgemart\_timesheets** table object and add some required data. Make sure you’ve completed section 2 in part 1 (a separate document available in Blackboard) before working on this section. I will review your actual database to see the results of your efforts. In order to use these as helpful samples moving forward, write these queries in the sql file, Lab03\_Introduction\_to\_SQL\_Queries.sql which you should have used for part 1.

3.a) Write the SQL CREATE TABLE statement to create the **fudgemart\_timesheets** table columns only as outlined in section 1.b of the lab. Do not create any constraints at this time.

3.b) Write the SQL ALTER TABLE statement to add all the constraints to the **fudgemart\_timesheets** table constraints as outlined in section 1.b of the lab.

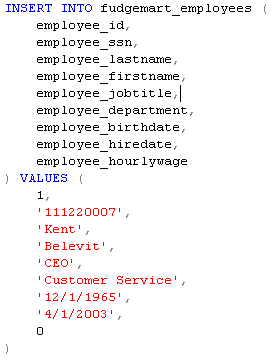
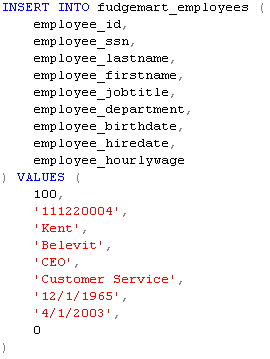
3.c) Write SQL INSERT statements to add one week of timesheets for each employee (6 inserts total) for the week of 9/10/2007. Each employee worked 45 hours total.

3.d) Correction. The two department managers worked 50 hours and not 45 hours. Write ONE SQL UPDATE statement to update both managers to 50 hours total.

3.e) Identity crisis! Lee Hvmeehom is tired of his name, and wants to change it to Mike Rofone. Write ONE SQL UPDATE statement to make this name change.

3.f) Write an SQL UPDATE statement to set the supervisor id of the ‘Hardware’ department employees to the employee id of the manager of that department.

## Section 4: Lab Questions: This is the Portion you will upload to Blackboard.

* 1. Write an SQL SELECT to show ALL of the rows in the table **fudgemart\_timesheets.** Copy the **results** here.
  2. Execute the following SQL. Explain in your own words why this command will not insert a row into the table.  
     
  3. Execute the following SQL. Explain in your own words why this command will not insert a row into the table.  
     
  4. Create a database diagram of the 4 tables created in this lab. Be sure we can see the physical constraints placed upon each column – just the column name won’t cut it.
  5. Explain in your own words why in  **fudgemart\_timesheets**, 61 hours would not be accepted if executed within the SQL code?
  6. Can you DROP a table if its primary key is used in a foreign key constraint? Why or why not? What error message will you get if you try to do so?
  7. Give one important advantage SQL DML has over building tables interactively via the GUI (graphical user interface). Give one disadvantage, too.