Lab – Introduction to SQL Select – Part 1 of 2

## Overview

In this lab, we will practice using DML (mainly the SQL SELECT statement) and the myriad of options available to it that we learned in class, mainly:

* SQL Functions for manipulating dates, text, and converting data types.
* Column Aliasing.
* Use of the TOP and DISTINCT keywords
* Use of SQL JOINS to combine output from multiple tables
* Use of SQL aggregate functions and the GROUP BY and HAVING clauses

### Learning Objectives

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

* Compare, contrast scalar and aggregate functions.
* Describe and use various ways to join tables.
* Solve problems using aggregate functions and joins.
* Compare join and sub queries.
* Understand how to use special constructs on the SELECT statement.

### Lab Goals

This lab consists of 3 sections. Part I is made up of Sections 1 and 2. Part 2 is the final section:

1. The first section will “set the stage” for the lab, giving you the background you need to understand the data model.
2. In the second section, you will mainly respond by typing in the provided SQL statements, and then try explaining what each statement does in plain old English. That concludes Part 1.
3. In section three, you will have to craft your own SQL statements from my descriptions using our account on Microsoft SQL Server. Style points will be rewarded.

**For Part 1 you will have to hand in pieces from parts 1.b and all of section 2. An Answer Template is provided at the end. Please insert your answers and upload into BB.**

### What you will need to begin

By now you should be proficient at logging on to your hosted SQL Server account. If you aren’t I suggest practice, practice, practice!

1. IMPORTANT: Finish the previous lab before starting this one.
2. Connect to your SQL Server instance.
3. The SQL file fudgemart\_lab3.sql this file will re-create the Fudgemart schema and populate it with a heaping helping of data. (This file can be downloaded from the same place where you got the lab.)
4. Open the SQL file in  and click  to create the tables, constraints, and populate the data. Once the data and tables are there you’re ready to rock.

## Section 1: The Fudgemart Database Schema (Reloaded)

Throughout the semester we will use several different case studies to help enforce the concepts we learn in class. One of the recurring case-studies we will use in class and the labs is the Fudgemart database. This database supports the business operations of a fictitious mega-store retailer and e-tailer called Fudgemart. The Fudgemart database supports all aspects of the business from human resources, to payroll, to sales transactions, and e-commerce. In each lab we will add new database objects and data to the Fudgemart schema.

### 1a: The enhanced Conceptual Model

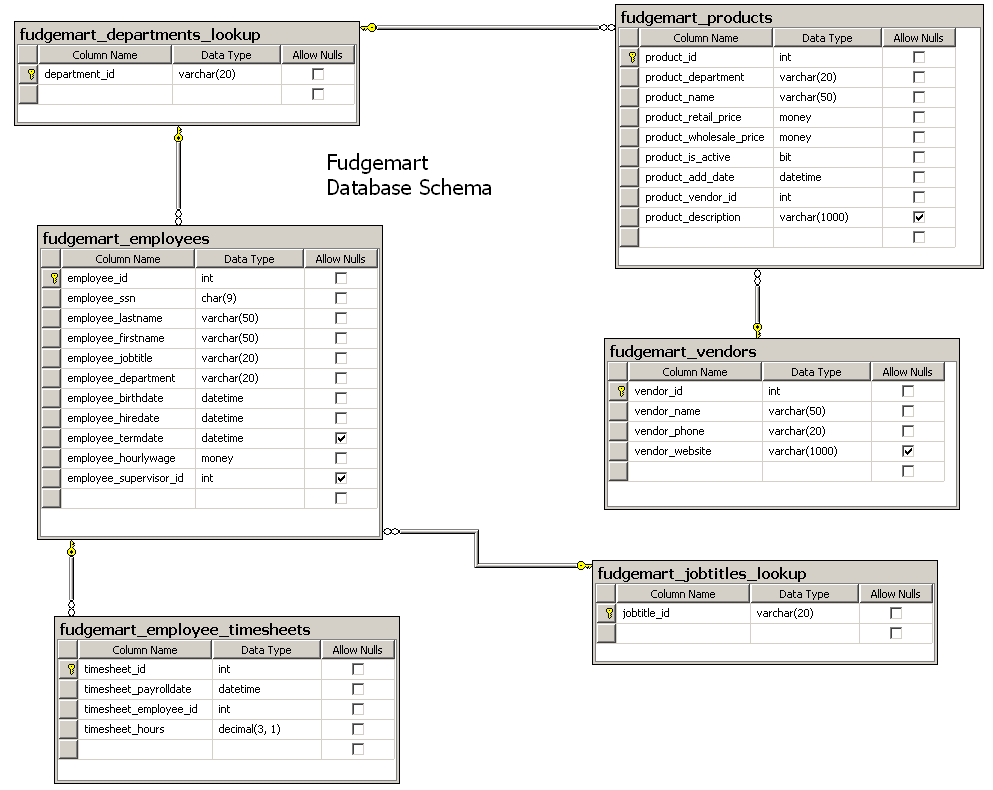
Let’s combine the conceptual models from the first two labs into a new mega Conceptual model. The Conceptual model is represented by a Chen Entity-Relationship Diagram.

As you’ve probably already figured out, the boxes will be tables in our Fudgemart database, and the lines connecting the boxes become foreign key constraints.



### 1b: Fudgemart Internal Model (Schema)

Here is the internal model. Remember there is a 1-1 correspondence between the logical model and the internal model. Also notice how much more “complicated” the internal model is when compared to the conceptual model. Remember, the conceptual model exists to define the *requirements* of the database, the logical model is a *blueprint* of that database, while the internal model *\*is\** the database. The only difference between the logical model and the internal model is the latter is actually implemented in a DBMS. Notice, that although not represented in the diagram, the check constraints, unique constraints, and default values all have all been implemented in the database schema just like they were in the previous labs.



**NOTE:** create this database diagram in your SQL database. It is good practice and will also help you verify your sql file was executed correctly. Before asking for help have your Gone to the HELP feature in Management Studio and looked for directions about creating a Database Diagram. If not, start there.

**Create the database Diagram (be sure to show data types and FK>>PK.)**

**Paste it into the Template Answer Sheet that will hold all of your answers. It is the last page of this document.**

## Section 2: Describe what each SQL statement does

In this part you will attempt to describe what each of the following SQL statements represent in plain English. The best way to accomplish this, of course, is to type in and execute the SQL and observe the output of each SQL statement. Change up the command, experiment, you can’t break ANYTHING. These are only SELECT statements.

|  |
| --- |
| Example:  Answer: Lists all departments sorted from last to first (Z-A) |
| 2.a)  What does this statement do? |
| 2.b)  What does this statement do? |
| 2.c)   Etc… |
| 2.d) |
| 2.e) |
| 2.f) |
| 2.g) |
| 2.h) |
| 2.i) (hint) leave distinct out, execute, then add it back in and execute to see what it’s doing. |
| 2.j) |

# Lab: SQL SELECT Statement Part 1 Answer Sheet Place Your Name Here:\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Place your Answers to Each Section in the Corresponding Area Below. Add as much additional space as you need so your answers are easy to read. Only Upload these Pages, not the Entire LaB.

**Answer 1.B Your Database Diagram Should Be Placed Below. DO Not COPY the one provided in your lab! Create your own!**

**Answer 2.a-J For each SQL statement, Describe what the statement does**

2.a

2.b

2.c

2.d

2.e

2.f

2.g

2.h

2.i

2.j