# LAB 05 – SQL SELECT STATEMENT PART 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:

* Use of SQL aggregate functions and the GROUP BY and HAVING clauses
* Complex SQL Joins
* Sub queries (Queries within queries)

### 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 parts:

1. The first part will “set the stage” for the lab, giving you the background you need to understand the data model.
2. In the second part, you will mainly respond by typing in the provided SQL statements, and then try explaining what each statement does in plain old English.
3. In part 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. 

You will have to hand in pieces from parts 2 and 3 on this week’s learning assessment.

### 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: This lab uses the Fudgemart tables. If you did the previous lab than you are all set. If you did not you can get the script in the previous unit.
2. Connect to your SQL Server instance.

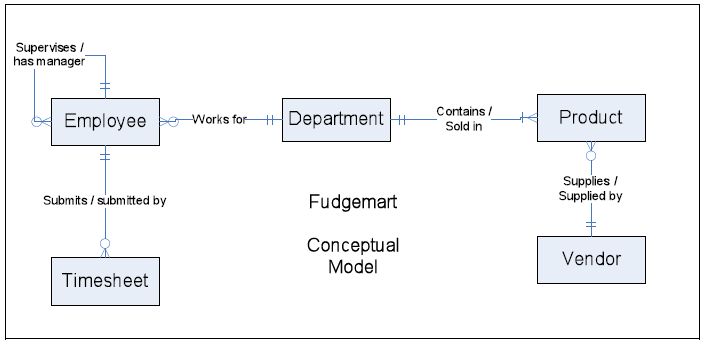
## PART 1: THE FUDGEMART DATABASE SCHEMA (REDUX)

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

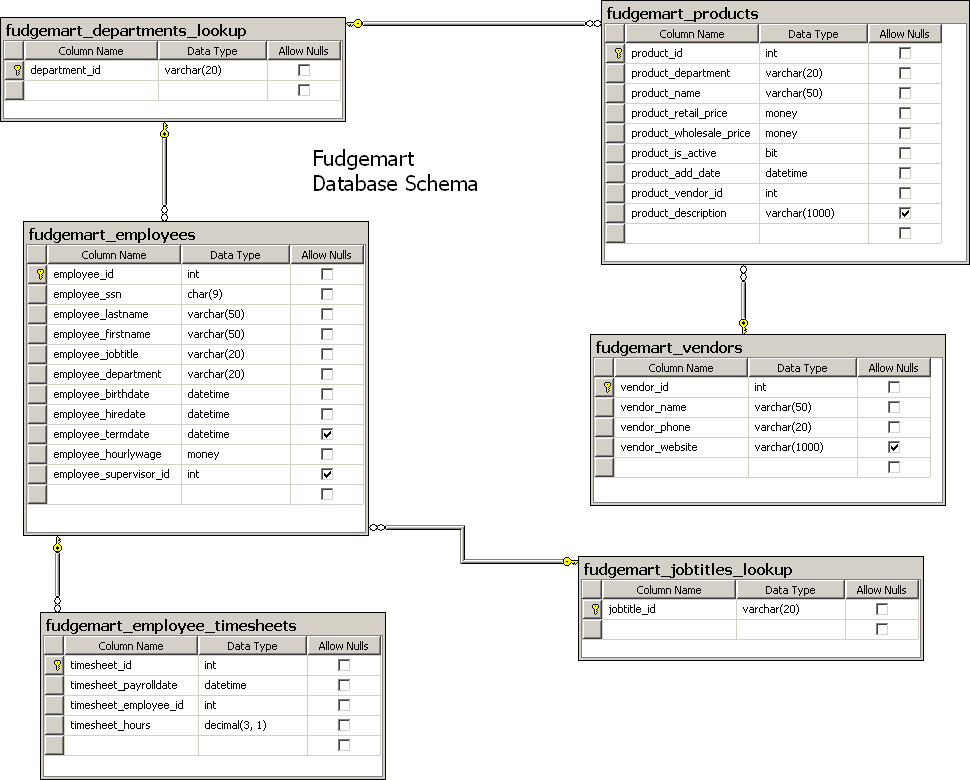
Let’s combine the conceptual models from the first two labs into a new mega Conceptual model. The Conceptual model is represented by a Chen Entity-­‐Relationship Diagram.

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. 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.



## PART 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 execute and observe the output of each SQL statement.



2.a)

Answer: counts the number of active fudgemart products

Example:

|  |
| --- |
| 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) |