

≡ [Item Navigation](#)

# Final Lab

## Overview

This lab challenges students to practice a variety of SQL queries utilizing all the SQL features we have learned in class.

This lab consists of TWO parts: Part One consists of SQL “SELECT” statements, and Part Two requires students to execute DDL/DML statements.

Part one queries will run against the tables contained in the “alanparadise/cm” repo. Part Two queries must run in your own bit.io demo\_repo.

For this lab, we will use the data in the Classic Models database. See Appendix “A” for an overview and introduction to the Classic Models database. You should become familiar with the Classic Models database prior to coding SQL queries for this lab.

## Results Submission

Copy and paste your SQL queries into a document. Number each query to match its problem. Save the document as a PDF and submit it for peer-review grading.

### Step-By-Step Instructions

## 1. Connect to the bit.io Lab Environment

NOTE: If you are not sure how to connect to the bit.io lab environment, please see the Lab 1 instructions.

## 2. Part One: Query Problems Using the “alanparadise/cm” repo

For this lab you must create and execute queries against the ClassicModels database to fulfill the requirements listed below. As a HINT, the expected number of rows in the answer set is shown in parentheses.

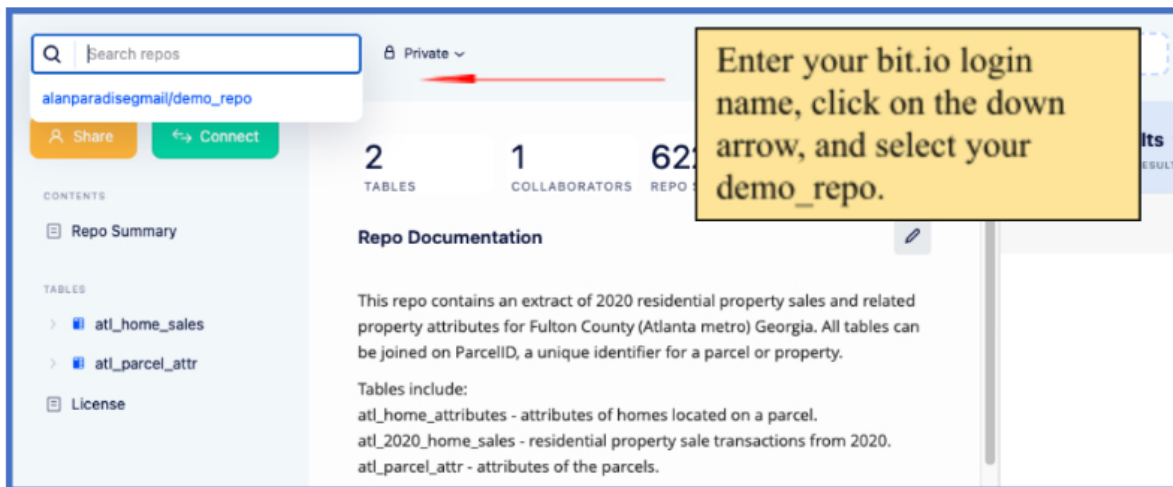
1. List the names of the cities in alphabetical order where Classic Models has offices. (7)
2. List the EmployeeNumber, LastName, FirstName, Extension for all employees working out of the Paris office. (5)
3. List the ProductCode, ProductName, ProductVendor, QuantityInStock and ProductLine for all products with a QuantityInStock between 200 and 1200. (11)
4. (Use a SUBQUERY) List the ProductCode, ProductName, ProductVendor, BuyPrice and MSRP for the least expensive (lowest MSRP) product sold by ClassicModels. (“MSRP” is the Manufacturer’s Suggested Retail Price.)

(1)

5. What is the ProductName and Profit of the product that has the highest profit (profit = MSRP minus BuyPrice). (1)
6. List the country and the number of customers from that country for all countries having just two customers. List the countries sorted in ascending alphabetical order. Title the column heading for the count of customers as "Customers". (7)
7. List the ProductCode, ProductName, and number of orders for the products with exactly 25 orders. Title the column heading for the count of orders as "OrderCount". (12)
8. List the EmployeeNumber, Firstname + Lastname (concatenated into one column in the answer set, separated by a blank and referred to as 'name') for all the employees reporting to Diane Murphy or Gerard Bondur. (8)
9. List the EmployeeNumber, LastName, FirstName of the president of the company (the one employee with no boss.) (1)
10. List the ProductName for all products in the "Classic Cars" product line from the 1950's. (6)
11. List the month name and the total number of orders for the month in 2004 in which ClassicModels customers placed the most orders. (1)
12. List the firstname, lastname of employees who are Sales Reps who have no assigned customers. (2)
13. List the customername of customers from Switzerland with no orders. (2)
14. List the customername and total quantity of products ordered for customers who have ordered more than 1650 products across all their orders. (8)

## 2. Part Two: Query Problems Using your demo\_repo

### 2.1. Navigate to **YOUR** demo\_repo



Students do not have permission to add database objects to "alanparadise/nw" repo. However, when you first launched bit.io, it created a "demo\_repo" for your login account. You can create database objects within your own demo\_repo.

In the screen shot above, you can see that I launched bit.io using an alternative bit.io userid. (I used my gmail account rather than my "alanparadise" account.)

You can use your own bit.io demo\_repo to create new database objects and update them.

## 2.2 Queries using your demo\_repo

1. Create a NEW table named “TopCustomers” with three columns: CustomerNumber (integer), ContactDate (DATE) and OrderTotal (a real number.) None of these columns can be NULL.
2. Populate the new table “TopCustomers” with the CustomerNumber, today’s date, and the total value of all their orders (PriceEach \* quantityOrdered) for those customers whose order total value is greater than \$140,000. (should insert 10 rows )
3. List the contents of the TopCustomers table in descending OrderTotal sequence. (10)
4. Add a new column to the TopCustomers table called OrderCount (integer).
5. Update the Top Customers table, setting the OrderCount to a random number between 1 and 10. Hint: use (RANDOM() \*10)
6. List the contents of the TopCustomers table in descending OrderCount sequence. (10 rows)
7. Drop the TopCustomers table. (no answer set)

### Appendix “A” – Overview of the Classic Models database

## Classic Models Company

Classic Models is a retailer of diecast miniature collectible model cars, motorcycles, airplanes, trains and ships.

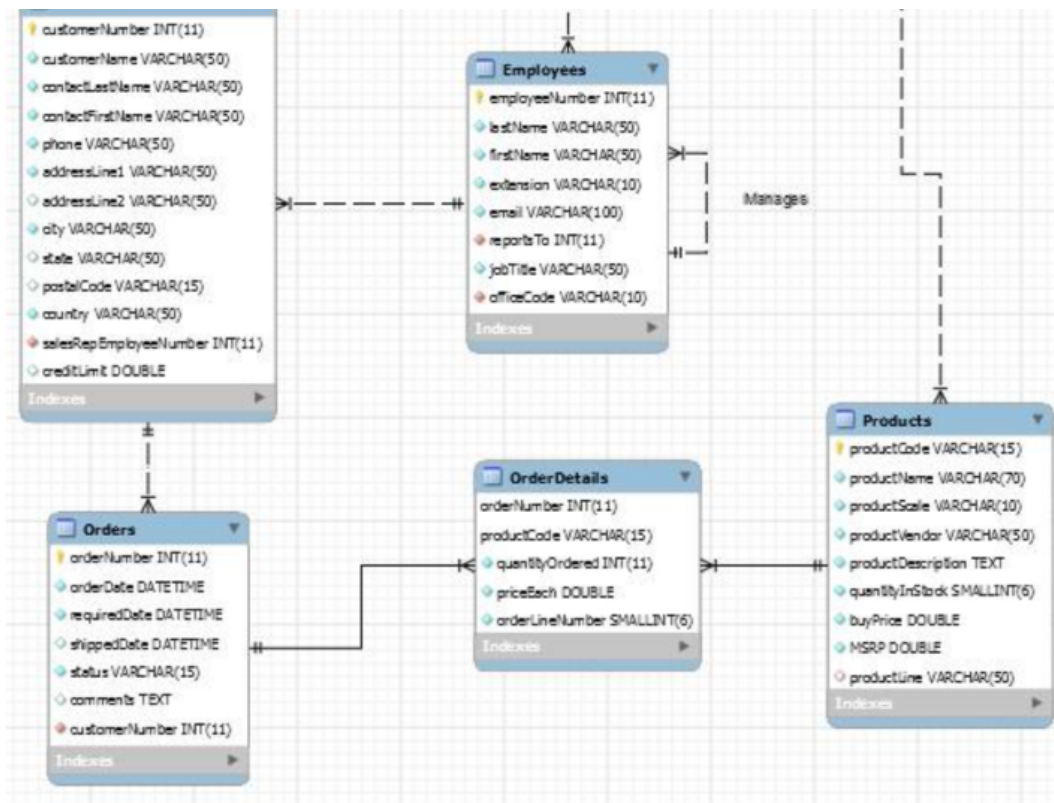
## Classic Models Database

The sample “Classic Models” database consists of the following eight tables:

<b>Customers</b>	<b>stores customer data.</b>
<b>Products</b>	stores information about the scale models.
<b>ProductLines</b>	stores a list of Classic Models’ various product lines.
<b>Orders</b>	stores sales orders placed by customers.
<b>OrderDetails</b>	stores sales order line items for each sales order.
<b>Payments</b>	stores payments made by customers for purchases
<b>Employees</b>	stores all Classic Models employee information
<b>Offices</b>	stores sales office data.

## Classic Models Data Model





✓ Completed

Go to next item

Like Dislike Report an issue