# ITAD 138 SQL

# Final Exam

Midterm Exam is a “hands on”, “open book” test. You can use any resources you wish during the test.

**Restrictions**

* Use of e-mail is not allowed during the test.
* Only workstations in the classroom can be used to take the test. You cannot use your own laptops.
* No flash drives are allowed in the computer.

Please log in into Canvas.

Find the MidTerm Exam under Assignments -> Tests -> MidTerm Exam

**After you completed your work:**

* Submit your result

**Database**

You will be working with database called *classicmodels*. The database keeps info for a business that sells scale models of cars, motorcycles, ships etc. You can find an ER diagram of the database in the FinalExam folder in the file **ClassicModelsDBSchema.jpg**

The database consists of eight tables:

* Offices: Sales offices
* Employees: All employees, including sales reps who work with customers.
* Customers
* Orders: Orders placed by customers
* Order Details: Additional info about an order.
* Payments: Payments made by customers against their account
* Products: The list of scale model cars
* Product Lines: The list of product line classification

**To Create Database**

Run the script called **classicmodels.sql** that can be found in FinalExam folder. This script creates all the tables and populates them with data.

**Test Questions**

**Total - 100 pts.**

*First 6 questions have corresponding screenshots of resulting tables. All the screenshots are located in the FinalExam folder in subfolder ‘QueryScreenshots’*

1. (10 pts.) Write a query that finds all the customers that are located in the same city as ‘Vitachrome Inc.’. Please use only given information (customer name) when querying the database. Order by customer number. Returns 5 rows, see screenshot Q1.
2. (10 pts.) Write a query that retrieves all the customers that made payments in June 2004. Include customer number, date of payment (in format ‘Jun 15 2004’), customer name, and the phone number. Order by order date. Returns 6 rows, see screenshot Q2.
3. (10 pts.) Write a query that finds all sales representatives who have more than 6 customers. Fetch the employee number, first and last names, and the number of customers. Order by number of customers in descending order. Returns 6 rows, see screenshot Q3.
4. (10 pts.) Create a view called train\_orders that shows all the orders that were placed in 2003 and included models of trains (from ‘Trains’ product line). Include orderNumber, orderDate, shippedDate, and customerNumber fields into the view.

Run *SELECT \* FROM train\_orders;* query on the view. It returns 17 rows, see screenshot Q4.

**HINT:**

The next 4 questions are about writing stored procedures. For each question, I would suggest you to start with writing a query (or a set of queries) that perform the required actions. Then create the procedure, copy the query into procedure code, and modify the query (if necessary) to incorporate the procedure variables. When you run procedures that change the database you might want to re-run **classicmodels.sql** in order to restore the database to the original version.

1. (15 pts.) Create a procedure named *direct\_subordinates* that takes first and last name of an employee and returns all employees directly supervised by the given person. The procedure should take two parameters of type VARCHAR(50).

Call the procedure for the employee Mary Patterson. See the results in Q5 screenshot.

1. (15 pts.) Write a function named *get\_orders\_total* that calculates and returns a total amount of all the orders placed by a customer. The function takes customer number (INT) as a parameter and returns a value of type DOUBLE.

Test your function: call the function to find the total amount of all orders placed by the customer 103. Include customer’s name into the output. See screenshot Q6 for results.

Hint: first write the query that finds the total amount of all orders placed by customer 103, and then incorporate the query into a function.

1. Create procedure called delete\_order that takes order number as parameter and deletes order from the orders table. In addition to the order itself, all the line items of that order must be deleted from the orderdetails table.

Call the procedure to make sure it runs properly. Run queries that allow you to see the changes in the employees and customer tables.

1. Write a procedure named add\_rep that adds a new sales rep to the employees table, and then assigns him all Canadian customers. Following is the new employee’s info:

ID: 1324

First name: Sam

Last name: Smith

Extension: x105

Email: samsmith@classicmodelcars.com

Office Code: 3

Reports to: 1143

Job Title: Sales Rep

Call the procedure to make sure it runs properly. Run queries that allow you to see the changes in the employees and customer tables.