**CPSC 408: Database Development**

**Spring 2018**

**MySQL Lab**

The data and queries are based off of the Northwind database, which will be provided. Write the SQL queries for the following criteria.

1. Write a query to get Product name and quantity/unit

SELECT ProductName, QuantityPerUnit FROM Products;

1. Write a query to get current Product list (Product ID and name)

SELECT ProductID, ProductName FROM Products WHERE Discontinued=’0’;

1. Write a query to get discontinued Product list (Product ID and name)

SELECT ProductID, ProductName FROM Products WHERE Discontinued=’1’;

1. Write a query to get most expense and least expensive Product list (name and unit price)

**SELECT ProductName**,**UnitPrice FROM** Products **ORDER BY UnitPrice LIMIT** 1;

**SELECT ProductName**,**UnitPrice FROM** Products **ORDER BY UnitPrice DESC LIMIT** 1;

1. Write a query to get Product list (id, name, unit price) where current products cost less than $20

SELECT ProductId,ProductName,UnitPrice FROM Products WHERE Discontinued=’0’ AND UnitPrice<’20’;

1. Write a query to get Product list (id, name, unit price) where products cost between $15 and $25

SELECT ProductId,ProductName,UnitPrice FROM Products WHERE UnitPrice>’15’ AND UnitPrice<’25’;

1. Write a query to get Product list (name, unit price) of above average price.

SELECT ProductId, ProductName, UnitPrice FROM Products WHERE UnitPrice > (SELECT AVG(UnitPrice) from Products)

1. Write a query to get Product list (name, unit price) of ten most expensive products

Select ProductName,UnitPrice from Products ORDER BY Price DESC LIMIT 10

1. Write a query to count current and discontinued products. \*\*\*\*\*

SELECT count(\*) FROM Products;

1. Write a query to get Product list (name, units on order, units in stock) of stock is less than the quantity on order

SELECT ProductName, UnitsOnOrder, UnitsInStock FROM Products WHERE UnitsInStock<UnitsOnOrder;

Java Database Connectivity (java.sql)

Call-level interfaces allows external access to SQL database and provides methods to invoke queries and update commands

JDBC has a rich collection of classes and methods that make database programming simple and intuitive

Java Programming Languages

Establish a connection

* Make sure java jdbc and appropriate drivers are installed
  + Various drivers
    - Microsoft SQL server
    - Oracle
    - Mysql
    - Sqlite
* Load the vendor specific driver
  + ensures portability and code reuse
  + the API (java.sql) was designed to be as database vendor independent
  + DBMS’s have different behavior, we need to tell the driver manager which DBMS we wish to use, so that it can invoke the correct driver
* A MySQL driver is loaded using the following code snippet

Establishing a Connection cont.

* Once the driver is loaded we can establish the connection
  + Create an instance of a connection object using:
  + **Connection con = DriverManager.getConnection(“jdbc:mysql://hostname:3306/dbname”, username, pwd);**
* DriverManager.getCOnnection() parameters
  + the protocol: jdbc
  + the vendor: mysql
  + the server: hostname or ipaddress
  + port number: 3306
  + server database name: dbname
  + the ussername and password to access the database
* How do we invoke a SQL statement?
* JDBC statement
  + an object used to send sql statements to the dbms
  + a jdbc statement object requires an open/valid database connection (previous slides)
  + an active connection is needed to create a statement object
    - **Statement stmt = con.createStatement();**
  + If successful, a statement object exists
    - Now we can create sql statement to execute on to the dbms.
* Caching for prepared statements because the database already knows what to execute
* Preferred method for interfacing with a DBMS
* The primary difference between the statement base class, it’s initialized with a sql statement
* The SQL statement is then sent to the DBMS right away, where it is compiled and prepared for execution
* Prepared Statements have an advantage over statements
  + if you need to execute the same, or similar query with different parameters multiple times, the statement can be compiled and optimized by the DBMS just once
  + contrast this with a use of a normal statement where each use of th esame sql statement requires a compilation all over again