Using SQL Built-in Functions

2.1 Create a table showing the number of Items and total price for Order 2000. Output should look something like this.

```
+-------------+--------------+
| NumberItems | Order2000Sum |
+-------------+--------------+
|           2 |       300.00 |
+-------------+--------------+
1 row in set (0.00 sec)
```

```
199.193.232.244 - PuTTY
 GNU nano 2.2.6                          File: sql_2-1.sql

SELECT COUNT(OrderNumber) AS NumberItems,
       SUM(ExtendedPrice) AS Order2000Sum
       FROM ORDER_ITEM WHERE OrderNumber = 2000;
```

```
mysql> source sql_2-1.sql
+-------------+--------------+
| NumberItems | Order2000Sum |
+-------------+--------------+
|           2 |       300.00 |
+-------------+--------------+
1 row in set (0.00 sec)
```

SQL Expressions in SQL SELECT Statements
2.2 Our company marks up all items 37% Write an SQL query that returns the SKU, Extended Price, and Profit for all items that make up order 3000

```
199.193.232.244 - PuTTY
 GNU nano 2.2.6                          File: sql_2-2.sql

SELECT SKU, ExtendedPrice,
       (ExtendedPrice * 0.37) AS Profit
       FROM ORDER_ITEM WHERE OrderNumber = 3000;
```

```
mysql> source sql_2-2.sql
+--------+---------------+----------+
| SKU    | ExtendedPrice | Profit   |
+--------+---------------+----------+
| 100200 |        300.00 | 111.0000 |
| 101100 |        100.00 |  37.0000 |
| 101200 |         50.00 |  18.5000 |
+--------+---------------+----------+
3 rows in set (0.00 sec)
```

Grouping Rows in Select Statement

2.3 List the Buyer and number of items sold (COUNT) from each department (GROUP) sorted by buyer name.
(NOTE: Grouping by department excludes one of the buyers, so I had to Group by buyer and department in order to preserve the buyer data).
ONE of the rows should look like:
| Jerry Martin | 2 |

```
199.193.232.244 - PuTTY
 GNU nano 2.2.6                              File: sql_2-3.sql

SELECT Buyer, COUNT(SKU) AS ItemsSold FROM SKU_DATA
       GROUP BY Buyer, Department
       ORDER BY Buyer;
```

```
mysql> source sql_2-3.sql
+--------------+-----------+
| Buyer        | ItemsSold |
+--------------+-----------+
| Cindy Lo     |         2 |
| Jerry Martin |         2 |
| Nancy Meyers |         2 |
| Pete Hansen  |         2 |
+--------------+-----------+
4 rows in set (0.00 sec)
```

Nested Queries

2.4 Give the total revenue (Extended Price) generated by items where Cindy Lo or Pete Hansen are the buyer; (Close to example in class)

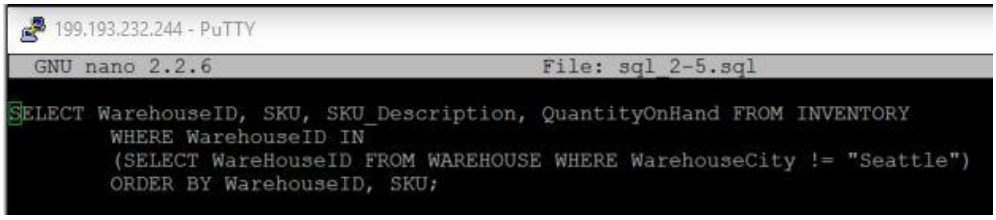
```
199.193.232.244 - PuTTY
 GNU nano 2.2.6                              File: sql_2-4.sql

SELECT SUM(ExtendedPrice) AS TotalRevenue FROM ORDER_ITEM
       WHERE SKU IN
       (SELECT SKU FROM SKU_DATA
       WHERE Buyer IN ("Cindy Lo", "Pete Hansen"));
```

```
mysql> source sql_2-4.sql
+--------------+
| TotalRevenue |
+--------------+
|       730.00 |
+--------------+
1 row in set (0.00 sec)
```

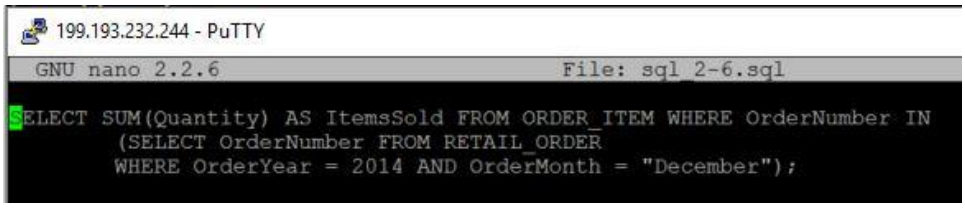
2.5 We have had a fire in the Seattle warehouse. Show the WarehouseID, SKU, SKU_Description, and QuantityOnHand for all items not in the Seattle warehouse. The final list should be ordered by WarehouseID then SKU.
The top few rows are

WareHouseID	SKU	SKU_Description	QuantityOnHand
100	100100	Std. Scuba Tank, Yellow	250
100	100200	Std. Scuba Tank, Magenta	200
100	101100	Dive Mask, Small Clear	0
100	101200	Dive Mask, Med Clear	100
100	201000	Half-dome Tent	2
100	202000	Half-dome Tent Vestibule	10



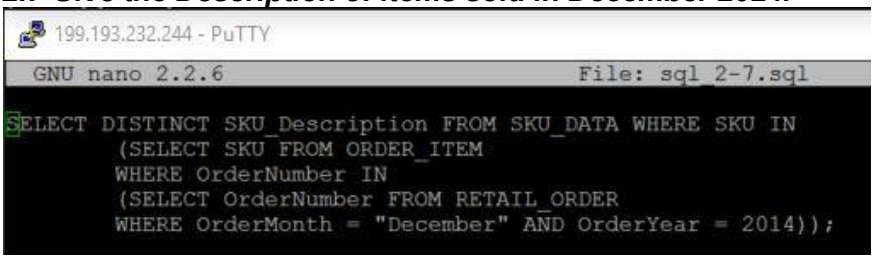

```
mysql> source sql_2-5.sql
+-------------+--------+---------------------------+----------------+
| WarehouseID | SKU    | SKU_Description           | QuantityOnHand |
+-------------+--------+---------------------------+----------------+
|         100 | 100100 | Std. Scuba Tank, Yellow   |            250 |
|         100 | 100200 | Std. Scuba Tank, Magenta  |            200 |
|         100 | 101100 | Dive Mask, Small Clear    |              0 |
|         100 | 101200 | Dive Mask, Med Clear      |            100 |
|         100 | 201000 | Half-dome Tent            |              2 |
|         100 | 202000 | Half-dome Tent Vestibule  |             10 |
|         100 | 301000 | Light Fly Climbing Harness|            300 |
|         100 | 302000 | Locking Carabiner, Oval   |           1000 |
|         200 | 100100 | Std. Scuba Tank, Yellow   |            100 |
|         200 | 100200 | Std. Scuba Tank, Magenta  |             75 |
|         200 | 101100 | Dive Mask, Small Clear    |              0 |
|         200 | 101200 | Dive Mask, Med Clear      |             50 |
|         200 | 201000 | Half-dome Tent            |             10 |
|         200 | 202000 | Half-dome Tent Vestibule  |              1 |
|         200 | 301000 | Light Fly Climbing Harness|            250 |
|         200 | 302000 | Locking Carabiner, Oval   |           1250 |
|         300 | 100100 | Std. Scuba Tank, Yellow   |            100 |
|         300 | 100200 | Std. Scuba Tank, Magenta  |            100 |
|         300 | 101100 | Dive Mask, Small Clear    |            300 |
|         300 | 101200 | Dive Mask, Med Clear      |            475 |
|         300 | 201000 | Half-dome Tent            |            250 |
|         300 | 202000 | Half-dome Tent Vestibule  |            100 |
|         300 | 301000 | Light Fly Climbing Harness|              0 |
|         300 | 302000 | Locking Carabiner, Oval   |            500 |
+-------------+--------+---------------------------+----------------+
24 rows in set (0.00 sec)
```

2.6 What total quantity of items were sold in December 2014?




```
mysql> source sql_2-6.sql
+------------+
| ItemsSold |
+------------+
|          8 |
+------------+
1 row in set (0.00 sec)
```

2.7 Give the Description of items sold in December 2014.




```
mysql> source sql_2-7.sql
+--------------------------+
| SKU_Description          |
+--------------------------+
| Dive Mask, Small Clear   |
| Dive Mask, Med Clear     |
| Half-dome Tent           |
| Half-dome Tent Vestibule |
+--------------------------+
4 rows in set (0.00 sec)
```

2.8 Launia Davis Nested Queries – extra credit challenge
Show the SKU Descriptions and the Department names from the SKU_DATA table where the buyer is either Pete or Cindy. Use a nested query to solve this problem
****(A nested query is not needed to solve this problem, it is redundant to nest a query from a table you are already in to that same table!)***

```
199.193.232.244 - PuTTY
 GNU nano 2.2.6                              File: sql_2-8.sql

SELECT SKU_Description, Department FROM SKU_DATA
       WHERE Buyer IN ("Pete Hansen", "Cindy Lo");
```

```
mysql> source sql_2-8.sql
+--------------------------+---------------+
| SKU_Description          | Department    |
+--------------------------+---------------+
| Std. Scuba Tank, Yellow  | Water Sports  |
| Std. Scuba Tank, Magenta | Water Sports  |
| Half-dome Tent           | Camping       |
| Half-dome Tent Vestibule | Camping       |
+--------------------------+---------------+
4 rows in set (0.00 sec)
```

2.9 Jake Hatfield CS364 Spring 2017 SQL Nested Query Challenge
How many tents are currently in inventory in all warehouses?
****(Nested Query is unnecessary, see submitted query.)***

```
199.193.232.244 - PuTTY
 GNU nano 2.2.6                              File: sql_2-9.sql

SELECT SUM(QuantityOnHand) AS NumberOfTents FROM INVENTORY
       WHERE SKU_Description LIKE "%Tent%";
```

```
mysql> source sql_2-9.sql
+----------------+
| NumberOfTents  |
+----------------+
|            373 |
+----------------+
1 row in set (0.00 sec)
```

2.10 Luis Loyh CS 364 SQL Nested Query Challenge

Show all of the items for camping and climbing currently on hand (from INVENTORY) in the warehouse in Bangor City. For this query, table SKU_DATA and WAREHOUSE must be nested in your query.

```
199.193.232.244 - PuTTY
 GNU nano 2.2.6                          File: sql_2-10.sql

SELECT * FROM INVENTORY WHERE SKU IN
       (SELECT SKU FROM SKU_DATA WHERE WarehouseID IN
              (SELECT WarehouseID FROM WAREHOUSE WHERE WarehouseCity = "Bangor"))
       AND QuantityOnHand > 0
       AND SKU IN (SELECT SKU FROM SKU_DATA
              WHERE Department IN ("Climbing", "Camping"));
```

```
mysql> source sql_2-10.sql
+-------------+--------+--------------------------+-----------------+------------------+
| WarehouseID | SKU    | SKU_Description          | QuantityOnHand | QuantityOnOrder |
+-------------+--------+--------------------------+-----------------+------------------+
|         300 | 201000 | Half-dome Tent           |             250 |                0 |
|         300 | 202000 | Half-dome Tent Vestibule |             100 |                0 |
|         300 | 302000 | Locking Carabiner, Oval  |             500 |              500 |
+-------------+--------+--------------------------+-----------------+------------------+
3 rows in set (0.00 sec)
```

2.11 John Mozingo

Create an SQL query that finds the CatalogID and the CatalogPage in CATALOG_SKU_2015 of the items in Order_Items table that are less than or equal to $50.00.

```
199.193.232.244 - PuTTY
 GNU nano 2.2.6                          File: sql_2-11.sql

SELECT CatalogID, CatalogPage FROM CATALOG_SKU_2015
       WHERE SKU IN (SELECT SKU FROM ORDER_ITEM WHERE Price <= 50);
```

```
mysql> source sql_2-11.sql
+-----------+-------------+
| CatalogID | CatalogPage |
+-----------+-------------+
|  20150003 |          27 |
|  20150004 |          27 |
+-----------+-------------+
2 rows in set (0.00 sec)
```