BUS 316
Professor Ballenger
March 10, 2020

Question 1

Output

Caffeine Type	Product Type	Product Name
Regular	Coffee	Amaretto
Regular	Coffee	Colombian
Regular	Espresso	Caffe Latte
Regular	Espresso	Caffe Mocha
Regular	Espresso	Regular Espresso
Regular	Tea	Darjeeling
Regular	Tea	Earl Grey
Regular	Tea	Green Tea
Decaf	Coffee	Decaf Irish Cream
Decaf	Espresso	Decaf Espresso
Decaf	Herbal Tea	Chamomile
Decaf	Herbal Tea	Lemon
Decaf	Herbal Tea	Mint

Written Explanation

This problem requires little explanation, only requiring the data scientist to call from one table, products. There were only three columns needed to be called on; type, product_type, and product. Lastly the type column needed to be sort in descending order, the other columns in ascending order.

Code

```
SELECT products.type "Caffeine Type",
products.Product_Type "Product Type",
Products.Product "Product Name"
FROM products
ORDER BY products.type DESC,
products.Product_Type ASC,
Products.Product ASC
;
```

Question 2

Output

Market	State
Central	Colorado
Central	Illinois
Central	Iowa
Central	Missouri
Central	Ohio
Central	Wisconsin
East	Connecticut
East	Florida
East	Massachusetts
East	New Hampshire
East	New York
South	Louisiana
South	New Mexico
South	Oklahoma
South	Texas
West	California
West	Nevada
West	Oregon
West	Utah
West	Washington

Written Explanation

The question asks to create a list of states grouped by their respective, unique, markets. Given that, I first called in the market column from the location stable along with the state column from the same table. After, I ordered by market first and then state.

Code

```
SELECT DISTINCT locations.market "Market",
locations.State
FROM locations
ORDER BY locations.Market,
locations.State
;
```

Question 3

Output

```
+---------------+
| State         |
+---------------+
| Connecticut   |
| Iowa          |
| Louisiana     |
| Missouri      |
| Nevada        |
| New Hampshire |
| New Mexico    |
| Oklahoma      |
| Oregon        |
| Utah          |
| Washington    |
| Wisconsin     |
+---------------+
```

Written Explanation

This question asks to create a distinct list of states that operate in a small market, listed in alphabetical order. Given that, I used a select distinct function to create the distinct list of states. After, I created a where function to isolate state operating in a small market and ordered the rows alphabetically, using an order by function to put the data in ascending, or alphabetical, order.

Code

```
SELECT DISTINCT locations.State
FROM locations
WHERE locations.market_size = "Small Market"
ORDER BY locations.state ASC
;
```

Question 4

Output

```
+--------------+---------------------+
| Market Size  | Number of Locations |
+--------------+---------------------+
| Major Market |                 114 |
| Small Market |                  42 |
+--------------+---------------------+
```

Written Explanation
This question called to create a count of locations operating in Small and Major markets. In order to create this output, I selected the Locations.Market_size column, Locations.Area_code column from the locations table. After a created a count for the number of area codes, generating a number of locations. After my from statement, I finished with a group by function, grouping by market size.

Code

```
SELECT locations.market_size "Market Size",
COUNT(locations.area_code) "Number of Locations"
FROM locations
GROUP BY locations.market_size
;
```

Question 5

Output

```
+-----------+----------------+--------------+-------+
| Area Code | Product        | Product Type | Sales |
+-----------+----------------+--------------+-------+
|       415 | Caffe Latte    | Espresso     |   509 |
|       415 | Caffe Mocha    | Espresso     |   318 |
|       415 | Decaf Espresso | Espresso     |   626 |
|       415 | Chamomile      | Herbal Tea   |   261 |
|       415 | Lemon          | Herbal Tea   |   486 |
+-----------+----------------+--------------+-------+
```

Written Explanation

The question called the scientist to list the locations.Area_code column, products.product column, products.Product_Type column, and financials.sales column in the output, all of which I called on in my solution. The sales and area code columns came from the financials table, while the product and product type columns came from the product table, meaning a join was necessary. I chose to join on the productID column. I restricted the data using where functions, requiring the data to be from the month of April and collected from the location(s) within the area code 415. Lastly, I limited the selection to the top five selling problems using the limit function.

Code

```
SELECT financials.Area_Code "Area Code",
products.Product,
products.Product_Type "Product Type",
financials.Sales
FROM products JOIN financials
ON products.ProductId=financials.ProductId
WHERE Area_Code = 415
AND financials.date = '2013-04-01'
LIMIT 5
;
```

Question 6

Outputs

Date	State	Area Code	Product ID	Product Name	Sales	Margin	Profit	Gross Profit Margin %	Net Profit Margin %
December-2012	California	951	1	Amaretto	102	-25	-88	-24.51	-86.27
December-2012	California	916	11	Darjeeling	302	179	134	59.27	44.37
December-2012	California	916	4	Caffe Latte	501	251	157	50.10	31.34
December-2012	California	805	10	Mint	99	56	29	56.57	29.29
December-2012	California	714	3	Decaf Irish Cream	192	-32	-149	-16.67	-77.60
December-2012	California	707	2	Colombian	699	420	271	60.09	38.77
December-2012	California	626	8	Chamomile	210	124	89	59.05	42.38
December-2012	California	619	12	Earl Grey	133	79	53	59.40	39.85
December-2012	California	530	5	Caffe Mocha	322	187	31	58.07	9.63
December-2012	California	415	9	Lemon	525	284	188	54.10	35.81
December-2012	California	213	13	Green Tea	168	101	47	60.12	27.98
December-2012	California	209	6	Decaf Espresso	576	329	216	57.12	37.50

Written Explanation

Starting the question, I immediately began to use the select statement to call on the columns required; financial.date column, the locations.state column, the financials.ProductID column, the Products.product column, the financial.sales column, the financials.margin column, and the financials.profit column. After, I calculated the Gross Profit Margin % and Net Profit Margin % columns, all in that order. Next, I joined the financials and locations table using the area code columns in each table. After doing this, I rounded all the necessary columns and formatted the date. Lastly, I restricted all the data to only include information from California during December of 2012 and ordered the data in descending order.

Code

```
SELECT DATE_FORMAT(financials.date,'%M-%Y') "Date",
 locations.state "State",
 locations.Area_code "Area Code",
 financials.ProductId "Product ID",
 products.Product "Product Name",
 financials.Sales,
 financials.Margin,
 financials.Profit,
 ROUND(((financials.sales-financials.cogs)/financials.sales)*100, 2) "Gross Profit Margin %",
 ROUND(((financials.Margin-financials.Total_Expenses)/financials.sales)*100, 2) "Net Profit Margin %"
FROM financials JOIN locations
ON financials.Area_Code=locations.Area_Code
JOIN products
ON financials.ProductID=products.productid
WHERE financials.date LIKE '2012-12%'
AND locations.state = 'California'
ORDER BY financials.Area_Code DESC
;
```

Question 7

Output

Date	Product ID	Product	Profit	Budget Profit	Profit Variance
December	5	Caffe Mocha	-224	-210	-14
May	7	Regular Espresso	187	210	-23
September	10	Mint	-238	-180	-58
July	11	Darjeeling	138	160	-22
April	13	Green Tea	83	100	-17

Written Explanation

Beginning the problem, I called on all the columns necessary to complete the problem; the financials.date column, the products.productID column, the products.product column, the financials.profit column, the financials.budget_profit column, and the created a new column called the profit variance column by calculating the variance. Next, I joined the financials and products tables on the productID column. I restricted the data, limiting the data to include information from the year 2012 and used a subquery to limit the data to information collected from locations in New York City. Lastly, I grouped the table by products.product and order them by products.productID to get information for each product and Profit Variance.

Code

```
SELECT DATE_FORMAT(financials.date,'%M') "Date",
financials.ProductId "Product ID",
products.Product,
financials.Profit,
financials.Budget_Profit "Budget Profit",
(financials.profit)-(financials.Budget_Profit) "Profit Variance"
FROM financials JOIN products
USING (productID)
WHERE financials.date LIKE '2012%'
AND area_code=(SELECT Area_Code
                FROM locations
                WHERE locations.city = 'New York City'
                GROUP BY locations.city)
GROUP BY Products.product
ORDER BY ProductId,
(financials.profit)-(financials.Budget_Profit)
;
```

Question 8

Output

Year	Product Type	Product Name	Total Sales	Total Cost of Goods Sold	Total Gross Margin %	Total Expenses	Total Net Profit
2013	Coffee	Amaretto	13,428	6,366	52.59	4,405	2,070
2013	Espresso	Caffe Latte	18,340	7,647	58.30	5,173	4,739
2013	Espresso	Caffe Mocha	43,367	18,995	56.20	15,026	7,516
2013	Espresso	Decaf Espresso	39,922	16,659	58.27	9,362	12,219
2013	Tea	Earl Grey	34,102	14,277	58.13	8,374	10,019
2013	Herbal Tea	Chamomile	38,609	15,693	59.35	9,981	11,295
2013	Herbal Tea	Mint	18,230	9,965	45.34	4,900	2,615
2013	Coffee	Decaf Irish Cream	31,779	14,784	53.48	9,778	5,907
2013	Herbal Tea	Lemon	48,970	20,747	57.63	13,732	12,477
2013	Tea	Darjeeling	37,352	15,267	59.13	8,509	12,023
2013	Coffee	Colombian	65,487	24,198	63.05	15,524	23,102
2013	Tea	Green Tea	16,785	9,539	43.17	6,491	-269
2013	Espresso	Regular Espresso	12,281	5,199	57.67	2,410	4,141

Written Explanation

Starting this problem, I created a select statement that would call on all the required columns; the financials.date column which only displays the year, the products.product_type column, the products.product columns. The other columns needed to be calculated. In order to generate the Total sales and total expense columns seen above, financials.sales and financials.expenses needed to have sum, format, and lpad functions appleid to them, so that they included columns, were aligned correctly on the left, and were totals. Total Gross Margin % and Total Net Profit Margin also required that all the variables were totaled and, as the last step, rounded correctly, including a format function in the latter column's case. After, the financials and products table are joined on productID. Next, the data are restricted between the dates of 2013-01-01' and '2013-12-01 to only account for data during the appropriate time span. The data is then grouped by products.product to group according to the product name and ordered by the financials.date column, products.Product_Type column, and the products.product column to order based on the date, product type, and name.

Code

```
SELECT DATE_FORMAT(financials.date,'%Y') "Year",
products.product_Type "Product Type",
products.Product "Product Name",
LPAD(FORMAT(SUM(financials.Sales), 0), 20, ' ') "Total Sales",
LPAD(FORMAT(SUM(financials.COGS), 0), 20, ' ') "Total Cost of Goods Sold",
ROUND((SUM(financials.sales)-SUM(financials.cogs))/(SUM(financials.sales))*100,2) "Total Gross Margin %",
LPAD(FORMAT(SUM(financials.total_expenses), 0), 20, ' ') "Total Expenses",
FORMAT((SUM(financials.margin)-SUM(financials.total_expenses)),0) "Total Net Profit"
FROM financials JOIN products
USING (Productid)
WHERE financials.date BETWEEN '2013-01-01' AND '2013-12-01'
GROUP BY products.product
ORDER BY financials.date,
products.Product_Type,
products.product
;
```

Question 9

Output

Year	Product Name	Total Sales	Total Profit	Net Profit Margin %
2013	Amaretto	13,428	2,907	-3.09
2013	Caffe Mocha	43,367	10,477	1.00
2013	Decaf Irish Cream	31,779	8,281	6.61
2013	Lemon	48,970	17,674	17.89
2013	Decaf Espresso	39,922	17,477	19.31
2013	Mint	18,230	3,643	26.14
2013	Earl Grey	34,102	14,313	27.38
2013	Caffe Latte	18,340	6,739	27.64
2013	Green Tea	16,785	-141	27.66
2013	Chamomile	38,609	16,138	34.97
2013	Colombian	65,487	33,027	35.43
2013	Darjeeling	37,352	17,209	38.46
2013	Regular Espresso	12,281	5,973	53.37

Written Explanation

Beginning this problem, I used a select statement to call on the required columns; the financials.date column, the products.product column, the financials.sales column, and the financials.profit column. Additionally, I applied to date format function to the financials.date column, lpad, format, and sum function to the financials.sales and financial.profit columns in order to create totals with commas, that are correctly formatted. Next, I joined the financials and products table on the productID. Later, I restricted the data to the dates between 2013-01-01 and 2013-12-01, as required. I grouped the output by the products.product column and order by the Net Profit Margin.

Code

```
SELECT DATE_FORMAT(financials.date,'%Y') "Year",
products.Product "Product Name",
LPAD(FORMAT(SUM(financials.Sales), 0), 20, ' ') "Total Sales",
LPAD(FORMAT(SUM(financials.Profit),0), 20, ' ') "Total Profit",
ROUND(((financials.Margin-financials.Total_Expenses)/financials.sales)*100, 2) "Net Profit Margin %"
FROM financials JOIN products
USING (Productid)
WHERE financials.date BETWEEN '2013-01-01' AND '2013-12-01'
GROUP BY Products.product
ORDER BY  ROUND(((financials.Margin-financials.Total_Expenses)/financials.sales)*100, 2) ASC
;
```

Question 10

Output

Month	Year	Area Code	Product Type	Product Name	Beginning Inventory	Ending Inventory	Change in Inventory
November	2013	603	Coffee	Amaretto	316	327	11
November	2013	603	Coffee	Colombian	452	462	10
November	2013	603	Espresso	Caffe Mocha	325	335	10
November	2013	603	Espresso	Regular Espresso	211	218	7
November	2013	603	Herbal Tea	Lemon	243	250	7
November	2013	603	Tea	Darjeeling	848	851	3
November	2013	603	Tea	Green Tea	387	344	-43

Written Explanation

Starting this problem, I created a select statement to call in all the necessary columns and used the appropriate functions. I called in date column twice, using the date format function to display the months and year in separate columns. After, I called in the area code column, products.product_type column, the products.product column, inventory column from the ending table created later in the subquery, and last created a column calculating the difference between the ending inventory and beginning inventory, calling it the change in inventory column. In the from statement, I referenced the initial table. Then I join to locations on the area code column, and join to products on the productID column. Immediately after, I join to a subquery to generate the ending inventory, using a select statement that calls on the identically formatted date functions from the select statement in the main query, the area code column, the product_type column, the product column, and the inventory column. In the same subquery, I join financials to locations using the area code column and to products using the productId column, restricting the data in a where clause to information from the city of New Hampshire and during 2013-12, ending the subquery. Afterward, I referenced ending to create a table alias using the product column. Back in the main query, I used a where statement to restrict the data to information from the city of New Hampshire during 2013-11, to get the correct beginning inventory and products. Afterward, I used an order by statement to order the data based on the products.product_type column and the products.product column.

Code

```
SELECT DATE_FORMAT(date,'%M') "Month",
DATE_FORMAT(date,'%Y') "Year",
initial.Area_Code "Area Code",
products.Product_Type "Product Type",
products.product "Product Name",
initial.inventory "Beginning Inventory",
ending.inventory "Ending Inventory",
(ending.inventory - initial.inventory) "Change in Inventory"
from financials initial
Join locations
USING(Area_Code)
JOIN products
USING(productID)
JOIN    (Select DATE_FORMAT(date,'%M'),
        DATE_FORMAT(date,'%Y'),
        area_code,
        Product_Type,
        Product,
        inventory
        FROM financials JOIN locations
        USING(Area_Code)
        JOIN products
        USING(productID)
        WHERE date = '2013-12-01'
        AND City like '%New Hampshire') ending USING(Product)
WHERE date = '2013-11-01'
AND city like '%New Hampshire'
ORDER BY products.Product_Type
AND Products.product
;
```

Error Log

March 5th at 10:00am- How to use subqueries in mysql?-tutorial on mysql subqueries-https://www.w3schools.com/sql/-A

The Pledge

“On my honor, I have neither given nor received any unacknowledged aid on this assignment.”

John Schmitt

Completed on Tuesday, March 10th