**UMUC Data 620 Assignment 6.1**

**Your Team:**

**Individual names:**

**Date:**

---

This is a team assignment. Recall from our syllabus:

> *For team assessments: Each person on a team will typically receive the same grade for an assessment. The highest possible grade on the team project will be based on what is submitted as the team end-product. However, a team member's grade may be adjusted downward for a specific assessment if the faculty member determines the quality of that person's participation to be substandard. To allow your faculty member to review team member contributions to all team assessments, each team member is required to post all contributions in the team's discussion area.*

Your company wants to merge its old product order data into a new data mart to facilitate analysis. Your team has been tasked with writing an ETL (extract, transform, and load) code sequence, and executing it on three years' worth of order data. Your team will produce:

- A .csv data file suitable for direct upload to the data mart, to match the data mart format given in the assignment
- A Microsoft word memo to the executive team, outlining what you did and what your recommendations are for moving forward. In the Appendix of the memo you will put the SQL code you wrote.

Additionally, you will submit a peer evaluation of your own and your teammates' performance after completion of this project.

Of course it is possible to perform ETL using a variety of software packages, even Excel, but for this project, please do *all* of your programming in My SQL Workbench in our Virtual Lab. A correct answer obtained by using something other than MySQL in our VDA will not receive credit.

Rubric:

| Element: | Possible Points | Notes |
|---|---|---|
| **.csv file deliverable** | **20** | Graded according to correctness of .csv file data over all years, product lines, and other summary fields. File should have headers describing the columns. Columns should be sorted per instructions. .csv data supports answers to Executive Memo below. |
| **SQL code** | **50** | SQL Code is submitted as a separate attachment and labeled as required.<br><br>Graded according to SQL rubric. Include comments and easy-to-follow queries. If we cannot generate your .csv file from the input files using your SQL code and following your directions, you will receive a 0 on this part.<br><br>You are welcome to include screenshots, but please also include SQL code such that we can run it. If your only SQL is a screenshot, you will receive very little credit on this part. |
| **Executive Memo** | | |
| ERD, ETL Documentation and Metadata | **20** | ERD is clearly documented and contains sentences denoting cardinality of relationships.<br>Process explanation is clear and in business English, not "technology-speak." Diagrams are encouraged. There is no use of SQL in this part, but instead references are made to SQL code by caption number in Appendix where needed.<br>Metadata is clear and comprehensive, and would be sufficient for a new programmer to come up to speed quickly. |
| Question 1 – Granularity | **30** | Complete answer to question, with examples where needed to support points. Demonstrates understanding of granularity in data marts. |
| Question 2 - Ramon | **30** | Complete answer to question, with examples where needed. Demonstrates understanding of what Ramon would need to answer the query; can run an example with one or two pieces of final data to illustrate. |
| Question 3 – different format for the data | **30** | Complete answer to question. Demonstrates understanding of advantages and disadvantages of two different types of two data layouts. Identifies any missing ideas and defends answer. |
| APA Formatting | **20** | Memo conforms to desired formatting: APA formatting for everything except feel free to put diagrams and charts in body of main paper as well as in Appendix. Few grammatical or spelling errors. Passes a Turnitin plagiarism check. |
| **TOTAL** | **200** | |

Getting Started:

This assignment starts with the script, “Week6_business_units.sql” . This script should create a table called “business_unit” and a table called “Product_BU.” Unfortunately, the metadata descriptions have been lost, so you will need to figure out what you can from the SQL script. The only thing you know about the metadata is that the company runs several individual strategic business units, such as “On The Go” and “Snack.” Each of these business units is run under an umbrella designation, such as “Growth” or “Decline.” The company will run marketing for growth products differently than it would run marketing for products on the decline.

You also have product order files from 2012, 2013, and 2014. They are attached as .csv files titled

- “2012_product_data_students.csv”
- “2013_product_data_students.csv”
- “2014_product_data_students.csv”

Your job is to use SQL to perform an ETL which will accomplish the following:

1. Extract data from the 2012, 2013, and 2014 order files
2. Transform the data according to the given rules
3. Load it into one final table
4. Export your final output table under the name “GX_output_final.csv” . (You may create as many or as few data objects as you like in your work, but the data in the .csv file named “GX_output_final.csv” will be the data evaluated.

Please name the computer files you submit for this assignment with a “GX” prefix, where “X” is your group number. For example, if you are in Group 3, you might create an SQL script named “G3_extract_2012.sql” (Ensure your group number and group member names are commented in any script you turn in as well.)

This is so when we grade the work, it’s clear which bit came from which group. You should get credit for your good work!

You may write one large SQL script to accomplish the entire process. You may also break your SQL commands into smaller groups, interspersed with MySQL GUI commands. If you do this, your notes should reflect what you did (for example, in the Appendix you could say “We created database YYY, and then used the “import” button on the MySQL GUI interface to upload the .csv file into Table Z. Then we ran the script shown in Figure X …”)

Please only use MySQL in our VDA in this assignment. The only exceptions here are minor edits made using Notepad or Excel, such as putting headers on column names. Document these carefully in your Appendix; if your SQL script doesn't write column headers, but your output file magically has them, we want to know how they got there. You can just say something like "After we did << XXX >> to export the data, we used Notepad to insert Row 1, which are the header names.")

Remember, you have learned how to download and run a .sql script in the Virtual Lab. And in Week 4, we learned how to use FileZilla to retrieve the results of an outfile. You will need both of these skills this week.

A note about outfile names: we know the SQL server and FileZilla don't let you easily overwrite an output file, so you may find yourself going through several iterations of output file names, such as "outfile_01" and "outfile_02." It's okay if you need to manually rename your final output file from "outfile_99" to the name requested above just before you turn it in. Just make a note if you did this. (You don't need to hold your breath and hope you get the code to run perfectly the very first time.)

Detailed Instructions:

**Extraction**: Your extracted data should meet the following criteria for each of the 2012, 2013, and 2014 data sets.

1. Business Unit Designations are for "Growth" and "Mature" only; do not choose any orders which are associated with a "Decline" designation
2. You will need to make a business decision about whether you want to extract records with a quantity of 0 or an order total of 0. Please note your decision and the logic behind it in your management memo.

**Transformation**: Your output file should follow this format, for loading into the data mart. A sample of some output is given below; note that your data may or may not match these numbers.

1. BU Designation – this is Growth and Mature; please roll up by this field
2. BU Name – no transformations; roll up by this field within BU Designation
3. Product – no transformations; roll up by this field within BU Name
4. Region – no transformations; roll up by this field within Product
5. Year – no transformations; roll up by this field within Region
6. Month – no transformations; roll up by this field within Year

7. Sum of Quantity – this reflects the sum of the "Quantity" field in the relevant data. For example, in the data below, the first line indicates that for Growth/Energy/Purple Pain/Eastern/2012/April, there was a total of 20 Purple Pain packets sold. This could reflect twenty 1-packet sales, four 5-packet sales, or one sale of 20 packets.
8. Sum of Order Total – this reflects the sum of the "Order total" field in the relevant data. For example, in the data below, the first line indicates that for Growth/Energy/Purple Pain/Eastern/2012/April, there was a total of 6960 cents in revenue from the 20 Purple Pain packets sold. (This implies a price of 6960/20 = 348 cents, or $3.48 per Purple Pain Packet in 2012.) You can assume pricing is stable throughout a calendar year, and any price changes happen instantaneously at midnight on December 31 and apply to the entire next year.

| BU Designation | BU Name | Product | Region | Year | Month | Sum of Quantity | Sum of Order Total |
|---|---|---|---|---|---|---|---|
| **Growth** | Energy | Purple Pain | Eastern | 2012 | 4 | 20 | 6960 |
| **Growth** | Energy | Purple Pain | Eastern | 2012 | 8 | 19 | 6612 |
| **Growth** | Energy | Purple Pain | Western | 2012 | 6 | 0 | 0 |
| **Growth** | Energy | Red Hot Chili Peppers | Eastern | 2012 | 1 | 33 | 14190 |
| **Growth** | Energy | Red Hot Chili Peppers | Eastern | 2012 | 8 | 30 | 12900 |
| **Growth** | Energy | Red Hot Chili Peppers | Midwest | 2012 | 6 | 37 | 15910 |
| **Growth** | Energy | Red Hot Chili Peppers | Western | 2012 | 2 | 12 | 5160 |
| **Growth** | Energy | Red Hot Chili Peppers | Western | 2012 | 3 | 33 | 14190 |
| **Growth** | Snack | Crocodile Tears | Eastern | 2012 | 2 | 26 | 7332 |
| **Growth** | Snack | Crocodile Tears | Southeast | 2012 | 4 | 4 | 1128 |
| **Growth** | Snack | Crocodile Tears | Western | 2012 | 3 | 18 | 5076 |
| **Mature** | Health | Panda Gummies | Eastern | 2012 | 4 | 69 | 10074 |
| **Mature** | Health | Panda Gummies | Midwest | 2012 | 7 | 16 | 2336 |

**Load**: Your deliverable is a single .csv file with the applicable data in it. It should contain only the fields listed above, and should be sorted alphabetically (or numerically) ascending in each field, with the leftmost fields having precedence. For example, you should first sort on BU Designation, and within that, sort on BU_Name. Your one data file should contain the data from all three years (2012, 2013, and 2014). Make sure to use your .csv editor (such as Notepad or Excel) to insert the field names on your .csv file after you have exported from SQL.

**Management Memo**

Your team writes a memo to management outlining your answers to the following questions:

1. Create and explain an ERD to go with this data. Your ERD should describe the business situation in existence as best as you can infer it. Since your input files are not necessarily in the best shape, your ERD should not simply map the input files. Your output file is by definition a flat file with no major database schema, so your ERD shouldn't map that either. As a hint, consider this: based on the data here, what relationship can you infer exists between BU Designation and Product? One to one? One to many? Must-have or may-have? Use ER Assistant to do your ERD, and incorporate a screenshot of your ERD in the management memo. (You do not need to attach the ER Assistant file.)
2. Document your ETL process. Which functions did you use, and what logic did you follow? This should be at the level that your boss, who has an MBA but not an IT/database background, can follow it. Do not use "computer-ese" here; use regular business English.
3. Give metadata for your final deliverable file. The analysts who follow you will thank you.
4. Your boss has a question for you. "We think this is about the right level of granularity for our data mart. What do you think? Should we extract more detailed information, and if so, what? Or would you recommend going to a coarser level of granularity, and if so, what fields would you recommend we drop?" Give your rationale. Think critically, and demonstrate a good understanding of data management.
5. Your boss wants to know the answer to this business question: "We believe our Growth segment should show at least 10% year over year growth in either quantity sold or order total. We also believe our Mature segment should remain pretty much the same in terms of quantity and order totals. If I give the final data file you produced to Ramon (an expert analyst), can he run queries to answer this?" (You may wish to run a query or two as proof of concept.) Tell the boss if you believe the data as laid out like it is will easily support Ramon in that sort of analysis. If it will, what about it makes it easy? If it won't, how could it change to support this analysis?

6. Your boss has another question: "Our database folks have suggested we use a different format for the ETL if I'm so interested in growth. It's copied below. It's the exact same data, just a little differently arranged. What do you think of it? Bobby, one of my IT people, thinks a data mart with this layout is a brilliant answer to the growth question. But Susie, another one of my IT people, has concerns that this data layout will make it hard to query on any other dimension, such as whether a particular product is doing well or poorly in a given region, regardless of year, or monthly seasonal trends. Am I missing anything here? What do you recommend? If we had to go with just one layout of our data mart, which layout should it be?"

Existing layout:

| << Other fields such as Designation here >> | Year | Sum of Quantity | Sum of Order Total |
|---|---|---|---|
| XXX | 2012 | 15 | 150 |
| XXX | 2013 | 16 | 160 |
| XXX | 2014 | 17 | 170 |

Proposed layout:

| << Other fields such as Designation here >> | Sum of Quantity for 2012 | Sum of Order Total for 2012 | Sum of Quantity for 2013 | Sum of Order Total for 2013 | Sum of Quantity for 2014 | Sum of Order Total for 2014 |
|---|---|---|---|---|---|---|
| XXX | 15 | 150 | 16 | 160 | 17 | 170 |

A successful memo will meet the following criteria:

- Times New Roman, double spaced, 12-point font, with 1-inch margins
- Contain a cover page with your group's number and all group members' names on it
- Contain a bibliography in APA format citing appropriate references (you may need to only cite this classroom and the Reference Manual; if you look up other sources cite them too.)
- Pass a Turnitin check for plagiarism
- Be in memo form, addressed to your boss, in business English (not computer-ese). Technical talk goes in the Appendix.
- Be of reasonable length. There are no page minimums or maximums, but please be reasonable. Something on the order of 10 pages or less for the written memo should probably suffice; the Appendix may run longer.

- An Appendix with any technical information you want to include. This could be technical explanations of how you used the GUI, or other nerdspeak.
- Classic APA formatting calls for all figures, exhibits, and tables to be in the Appendix. I'm relaxing this requirement here. If a diagram (for example, a flowchart of something) would make more sense in the body of your paper, put it in the body. If it would make more sense in the Appendix, leave it in the Appendix.

Submit:

- Your memo, labeled "GX_memo.docx" (or .pdf), where "X" is your group number. (If you are Group 3, this will be called "G3_memo.docx"
- Your final output file, labeled "GX_output_final.csv", where "X" is your group number.
- The SQL code file(s) you used to make this happen. If there is more than one file, label them to make it easy to find and assemble them. If you have 3 files, you can call them "GX_1_extract.sql", "GX_2_transform.sql", "GX_3_load.sql", etc. The GX is your group designation. The 1, 2, 3 are the orders in which we should run the scripts. And the word is a summary of what it might do – the words you use are entirely up to you.

**2012 Data Notes:**

Your order 2012 data is contained in the attached file, "2012_product_data_students.csv." A sample of this file's type of data is contained below in Table 1 Sample of order data from 2012. (Note your file may or may not have the same data in it.)

Your field definitions follow:

- Month: integer, corresponds to the month of the sale. For example, 5 = May.
- Country: character, should all be USA. (All data in this exercise should be USA.)
- Region: character, represents the regions within the country.
- State: two characters, USPS state abbreviations. Each state is within one region.
- Product: character. This is the name of a packaged food product.
- Per-unit price: integer. This represents the per-unit price in cents; for example, 466 indicates that Blue Rock Candy sells for $4.66 per package. (For the purposes of this exercise, you may disregard all currency formatting and just use 466 to represent $4.66. If you choose to do this, make sure you note it in your final product.)
- Quantity: integer. This represents how many items were in that particular order. The first order here was for 3 packages of Blue Rock Candy.
- Order Total: integer. This is the per-unit price x the quantity. The first line here indicates that 466 x 3 = 1398 (or $13.98) was the price of the first order.

**Table 1 Sample of order data from 2012**

| Month | Country | Region | State | Product | Per-Unit Price | Quantity | Order Total |
|---|---|---|---|---|---|---|---|
| 5 | USA | Midwest | MN | Blue Rock Candy | 466 | 3 | 1398 |
| 5 | USA | Eastern | RI | Pink Bubble Gum | 318 | 15 | 4770 |
| 4 | USA | Southeast | MO | Crocodile Tears | 282 | 4 | 1128 |
| 1 | USA | Eastern | MD | Yellow Zonkers | 258 | 27 | 6966 |

**2013 Data Notes:**

Your order 2013 data is contained in the attached file, "2013_product_data_students.csv"

A sample of this file's data is contained below as Table 2 Sample of order data from 2013. (Note your file may or may not have the same data in it.)

Your field definitions follow:

- Month: integer, corresponds to the month of the sale. For example, 5 = May.
- Region: character, represents the regions within the country.
- Customer_ID: numeric, represents the customer's unique Customer ID number.
- Product: character. This is the name of a packaged food product. The product name is consistent between 2012, 2013, and 2014; for example, if something is called "Orange Creepies" in 2012, those characters refer to the same product in 2013 and 2014.
- Per-unit price: integer. This represents the per-unit price in cents; for example, 293 indicates that Crocodile Tears sells for $2.93 per package. (For the purposes of this exercise, you may disregard all currency formatting and just use 293 to represent $2.93. If you choose to do this, make sure you note it in your final product.)
- Quantity_1: integer. This represents how many items were in the first shipment of that particular order. This year we had shipping problems, and could often not ship the entire order all at once. Orders were split into two shipments where necessary, and Quantity_1 reflects how many units were shipped first. (Assume all shipments were completed in the month listed, and that no shipments had the first shipment in one month and the second shipment in the subsequent month.)
- Quantity_2: integer. This represents how many items were in the second shipment of that particular order. A 0 indicates a second shipment was not necessary. To get the total number of items shipped, you need to add Quantity_1 and Quantity_2.
- The first line here reflects that Crocodile Tears has a first shipment of 13 units, and a second shipment of 1 unit, all within the month of May, for a total of 13 + 1 = 14 units.

**Table 2 Sample of order data from 2013**

| Month | Region | Customer_ID | Product | Per-Unit Price | Quantity_1 | Quantity_2 |
|---|---|---|---|---|---|---|
| 5 | Southeast | 857 | Crocodile Tears | 293 | 13 | 1 |
| 9 | Midwest | 785 | Blue Rock Candy | 489 | 16 | 10 |
| 5 | Eastern | 906 | Nap Be Gone | 427 | 24 | 4 |
| 2 | Western | 939 | Yellow Zonkers | 253 | 8 | 5 |
| 7 | Western | 558 | Pink Bubble Gum | 318 | 26 | 7 |

**2014 Data Notes:**

Your order 2014 data is contained in the attached file, "2014_product_data_students.csv."

A sample of this file's data is contained below as Table 3 Sample of order data from 2014. (Note your file may or may not have the same data in it.)

Your field definitions follow:

- Month: integer, corresponds to the month of the sale. For example, 5 = May.
- Country: character, represents the country of the customer. Should all be USA.
- Region: character, represents the regions within the country.
- State: USPS code for the 50 United States.
- Product: character. Same as in 2012 and 2013 data.
- Per-unit price: integer. This represents the per-unit price in cents; for example, 425 indicates that Red Hot Chili Peppers sells for $4.25 per package. (For the purposes of this exercise, you may disregard all currency formatting and just use 425 to represent $4.25. If you choose to do this, make sure you note it in your final product.)

- Quantity: This represents how many items were in that particular order. The first order here was for 32 packages of Red Hot Chili Peppers.
- Order Subtotal: This represents the order subtotal, calculated as per-unit price x quantity. For example, the first order here reflects a per-unit price of 425 cents x 32 units, for a subtotal of 13,600 (or $136.00).
- Quantity Discount: This represents the new policy (effective January 1, 2014) that all orders over 3 dozen (36 units) will automatically earn a 10% discount. An order of exactly 36 units does not earn the discount. All order discounts have been rounded to the nearest penny, so you can assume this field has no decimals in it. In the data below,
    - The first order, of 32 Red Hot Chili Peppers to New Jersey, did not qualify for the Quantity Discount. Therefore, the Order total would simply be the Order subtotal.
    - The fourth order, of 60 Green Lightning to Rhode Island, did qualify for the Quantity Discount, which has already been computed as 2340, or 10% of 23400. In this case, the final order total would be 23,400 – 2,340 = 21,060 (or $210.60).
- When you roll up the discounts, the Order Subtotal, Quantity Discount, and Order Total should just add up. It should work like this (below is a very abbreviated data set, designed to show only treatment of order totals):

| Product Name | Order Subtotal | Quantity Discount | Order Total |
|---|---|---|---|
| **Product A** | 100 | 0 | 100 |
| **Product A** | 500 | 50 | 450 |
| **Product A** | 200 | 0 | 200 |

Rolled up, this data should give:

| Product Name | Sum of Order Subtotal | Sum of Quantity Discount | Sum of Order Total |
|---|---|---|---|
| **Product A** | 800 | 50 | 750 |

**Table 3 Sample of order data from 2014**

| Month | Country | Region | State | Product | Per-Unit Price | Quantity | Order Subtotal | Quantity Discount |
|---|---|---|---|---|---|---|---|---|
| 6 | USA | Eastern | NJ | Red Hot Chili Peppers | 425 | 32 | 13600 | 0 |
| 5 | USA | Eastern | DE | Giant Gummies | 428 | 34 | 14552 | 0 |
| 5 | USA | Southeast | LA | Orange Creepies | 466 | 25 | 11650 | 0 |
| 6 | USA | Eastern | RI | Green Lightning | 390 | 60 | 23400 | 2340 |
| 10 | USA | Eastern | GA | Giant Gummies | 428 | 44 | 18832 | 1883 |