

# Assignment 1

## Business Analysis of Maven Market

### Instructions:

Maven Market is a multi-national grocery chain with locations in Canada, Mexico and the United States. You have to perform the task of connecting and shaping the source data, building a relational model, adding calculated columns and measures, and designing a report using different visuals.

Refer to the following files in the MavenMarket\_dataset folder on Teams:

1. MavenMarket\_Calendar.csv
2. MavenMarket\_Customers.csv
3. MavenMarket\_Products.csv
4. MavenMarket\_Regions.csv
5. MavenMarket>Returns\_1997-1998.csv
6. MavenMarket\_Stores.csv
7. MavenMarket\_Transactions\_1997.csv
8. MavenMarket\_Transactions\_1998.csv

Open a new PowerBI Desktop file and perform the tasks mentioned below. Save this file as yourPRN\_MavenMarket.pbix file which should be submitted on Microsoft Teams:

1. Check that Locale for import is set to "English (United States)" in the Regional Settings tab
2. Connect to the MavenMarket\_Customers csv file
  - a. Name the table "Customers", and make sure that headers have been promoted
  - b. Confirm that data types are accurate (Note: "customer\_id" should be whole numbers, and both "customer\_acct\_num" and "customer\_postal\_code" should be text)
  - c. Add a new column named "full\_name" to merge the the "first\_name" and "last\_name" columns, separated by a space
  - d. Create a new column named "birth\_year" to extract the year from the "birthdate" column, and format as text
  - e. Create a conditional column named "has\_children" which equals "N" if "total\_children" = 0, otherwise "Y"
3. Connect to the MavenMarket\_Products csv file
  - a. Name the table "Products" and make sure that headers have been promoted

- b. Confirm that data types are accurate (Note: "product\_id" should be whole numbers, "product\_sku" should be text), "product\_retail\_price" and "product\_cost" should be decimal numbers)
  - c. Use the statistics tools to return the number of distinct product brands, followed by distinct product names. You should see 111 brands and 1,560 product names
  - d. Add a calculated column named "discount\_price", equal to 90% of the original retail price
  - e. Format as a fixed decimal number, and then use the rounding tool to round to 2 digits
  - f. Select "product\_brand" and use the Group By option to calculate the average retail price by brand, and name the new column "Avg Retail Price". You should see an average retail price of \$2.18 for Washington products, and \$2.21 for Green Ribbon
  - g. Delete the last applied step to return the table to its pre-grouped state
  - h. Replace "null" values with zeros in both the "recyclable" and "low-fat" columns
4. Connect to the MavenMarket\_Stores csv file
  - a. Name the table "Stores" and make sure that headers have been promoted
  - b. Confirm that data types are accurate (Note: "store\_id" and "region\_id" should be whole numbers)
  - c. Add a calculated column named "full\_address", by merging "store\_city", "store\_state", and "store\_country", separated by a comma and space
  - d. Add a calculated column named "area\_code", by extracting the characters before the dash ("-") in the "store\_phone" field
5. Connect to the MavenMarket\_Regions csv file
  - a. Name the table "Regions" and make sure that headers have been promoted
  - b. Confirm that data types are accurate (Note: "region\_id" should be whole numbers)
6. Connect to the MavenMarket\_Calendar csv file
  - a. Name the table "Calendar" and make sure that headers have been promoted
  - b. Use the date tools in the query editor to add the following columns:
    - Start of Week (starting Sunday)
    - Name of Day
    - Start of Month
    - Name of Month
    - Quarter of Year
    - Year
7. Connect to the MavenMarket>Returns csv file
  - a. Name the table "Return\_Data" and make sure that headers have been promoted

- b. Confirm that data types are accurate (all ID columns and quantity should be whole numbers)
8. Connect to MavenMarket\_Transactions\_1997 and MavenMarket\_Transactions\_1998 csv files and combine the data using append query. Name the new table as Transaction data
9. Click Close & Apply and Exit query editor
10. In relationship view ensure that tables are connected properly
  - a. Connect Transaction\_Data to Customers, Products, and Stores using valid primary/foreign keys
  - b. Connect Transaction\_Data to Calendar using both date fields, with an inactive "stock\_date" relationship
  - c. Connect Return\_Data to Products, Calendar, and Stores using valid primary/foreign keys
  - d. Connect Stores to Regions as a "snowflake" schema
11. In the data view of Desktop
  - a. Update all date fields (across all tables) to the "M/d/yyyy" format using the formatting tools in the Modeling tab
  - b. Update "product\_retail\_price", "product\_cost", and "discount\_price" to Currency (\$ English) format
  - c. In the Customers table, categorize "customer\_city" as City, "customer\_postal\_code" as Postal Code, and "customer\_country" as Country/Region
  - d. In the Stores table, categorize "store\_city" as City, "store\_state" as State or Province, "store\_country" as Country/Region, and "full\_address" as Address
12. In the Data View, add following Calculated Columns
  - a. In the Calendar table, add a column named "Weekend"
    - Equals "Y" for Saturdays or Sundays (otherwise "N")
  - b. In the Calendar table, add a column named "End of Month"
    - Returns the last date of the current month for each row
  - c. In the Customers table, add a column named "Current Age"
    - Calculates current customer ages using the "birthdate" column and the TODAY() function
  - d. In the Customers table, add a column named "Priority"
    - Equals "High" for customers who own homes and have Golden membership cards (otherwise "Standard")
  - e. In the Customers table, add a column named "Short\_Country"
    - Returns the first three characters of the customer country, and converts to all uppercase
  - f. In the Customers table, add a column named "House Number"

- Extracts all characters/numbers before the first space in the "customer\_address" column (hint: use SEARCH)
- g. In the Products table, add a column named "Price\_Tier"
  - Equals "High" if the retail price is >\$3, "Mid" if the retail price is >\$1, and "Low" otherwise
- h. In the Stores table, add a column named "Years\_Since\_Remodel"
  - Calculates the number of years between the current date (TODAY()) and the last remodel date
- 13. In Report view, add the following Measures (Use matrix visual to match the spot-check values)
  - a. Create new measures named "Quantity Sold" and "Quantity Returned" to calculate the sum of quantity from each data table
    - Spot check: You should see total Quantity Sold = 833,489 and total Quantity Returned = 8,289
  - b. Create new measures named "Total Transactions" and "Total Returns" to calculate the count of rows from each data table
    - Spot check: You should see 269,720 transactions and 7,087 returns
  - c. Create a new measure named "Return Rate" to calculate the ratio of quantity returned to quantity sold (format as %)
    - Spot check: You should see an overall return rate of 0.99%
  - d. Create a new measure named "Weekend Transactions" to calculate transactions on weekends
    - Spot check: You should see 76,608 total weekend transactions
  - e. Create a new measure named "% Weekend Transactions" to calculate weekend transactions as a percentage of total transactions (format as %)
    - Spot check: You should see 28.4% weekend transactions
  - f. Create new measures named "All Transactions" and "All Returns" to calculate grand total transactions and returns (regardless of filter context)
    - Spot check: You should see 269,720 transactions and 7,087 returns across all rows (test with product\_brand on rows)
  - g. Create a new measure to calculate "Total Revenue" based on transaction quantity and product retail price, and format as \$
    - Spot check: You should see a total revenue of \$1,764,546
  - h. Create a new measure to calculate "Total Cost" based on transaction quantity and product cost, and format as \$
    - Spot check: You should see a total cost of \$711,728
  - i. Create a new measure named "Total Profit" to calculate total revenue minus total cost, and format as \$
    - Spot check: You should see a total profit of \$1,052,819

- j. Create a new measure named "Unique Products" to calculate the number of unique product names in the Products table
- Spot check: You should see 1,560 unique products

#### 14. VISUALS – Try to fit all visuals on the same page

a. Matrix Visual

- Insert a Matrix Visual to show Total Transactions, Total Profit, Profit Margin, and Return Rate by Product\_Brand.
- Add conditional formatting to show data bars on the Total Transactions column, and color scales on Profit Margin (White to Green) and Return Rate (White to Red).
- Add a visual level Top N filter to only show the top 30 product brands, then sort descending by Total Transactions

b. Map Visual

- Add a Map visual to show Total Transactions by store city
- Add a slicer for store country, Under the "selection controls" menu in the formatting pane, activate the "Show Select All" option
- Change the orientation in the "General" formatting menu to horizontal and resize to create a vertical stack (rather than a list)

c. Treemap Visual

- Add a Treemap visual to break down Total Transactions by store country
- Pull in store\_state and store\_city beneath store\_country in the "Group" field to enable drill-up and drill-down functionality
- 

- d. Add any 2 more visuals of your choice. The visuals may cover some of the measures / dimensions not already covered above.

*Just to repeat - Save your file as yourPRN\_MavenMarket.pbix file which should be submitted on Microsoft Teams*

Along with uploading the .pbix file, publish your project on PowerBI and share the link as well.

#### **End of Assignment**