Fetch Data Analytics Manager Assessment -- Data Exploration

Abinav Bharadwaj

1) **Initial high-level data exploration**
    a) Products Table
        i) Manufacturer and Brand are provided but some values are missing
        ii) Products are identified by Barcode but there are Barcode values missing as well
        iii) Product Category is broken down into four levels for categorization, starting from the broad category name (CATEGORY_1) to the narrowest subcategory (CATEGORY_4)
        iv) Products and Transactions are linked by Barcode
    b) Transactions Table
        i) This table logs each transaction using fields Receipt ID, Purchase Date, Scan Date, User ID, etc.
        ii) Final Quantity and Final Sale seemingly show the number of products purchased and the total price but there are some unusual values in the Final Quantity column such as “zero” and decimal numbers
        iii) There is missing information in the Barcode and Final Sale columns
        iv) Transactions and Products are linked by Barcode
        v) Transactions and Users are linked by ID
    c) Users Table
        i) Contains information on the customers, identified by their ID, and with information on their Birth Date, State, Language and Gender
        ii) There is missing information in the Birth Date and Gender columns
        iii) Users and Transactions are linked by ID

2) **Are there any data quality issues present?**

I will focus on three types of potential data quality issues in my investigation: missing data, data consistency issues and incorrectly formatted data.

1) *Missing Data Products Table: Check for missing values in Barcode, Category 1, Manufacturer and Brand*

```
-- This query checks for missing or empty values in key fields of the
Products table
SELECT
    COUNT(*) AS total_rows,  -- Counts the total number of rows in the
Products table
    SUM(CASE WHEN BARCODE IS NULL OR BARCODE = '' THEN 1 ELSE 0 END) AS
missing_barcode,  -- Counts rows where BARCODE is either NULL or empty
    SUM(CASE WHEN CATEGORY_1 IS NULL OR CATEGORY_1 = '' THEN 1 ELSE 0 END)
AS missing_category_1,  -- Counts rows where CATEGORY_1 is NULL or empty
```

```
    SUM(CASE WHEN MANUFACTURER IS NULL OR MANUFACTURER = '' THEN 1 ELSE 0
END) AS missing_manufacturer,  -- Counts rows where MANUFACTURER is NULL or
empty
    SUM(CASE WHEN BRAND IS NULL OR BRAND = '' THEN 1 ELSE 0 END) AS
missing_brand  -- Counts rows where BRAND is NULL or empty
FROM products;  -- Querying the 'products' table
```

total_rows	missing_barcode	missing_category_1	missing_manufacturer	missing_brand
845552	4025	111	226474	226472

There are a significant number of missing values in the Manufacturer and Brand columns as well as ~4k missing Barcodes, which links Products and Transactions.

2) *Missing Data Transactions Table: Check for missing values in Barcode, User ID, Final Quantity, Final Sale and Purchase Date*

```
-- This query checks for missing or empty values in key fields of the
Transactions table
SELECT
    COUNT(*) AS total_rows,  -- Counts the total number of rows in the
Transactions table
    SUM(CASE WHEN BARCODE IS NULL OR BARCODE = '' THEN 1 ELSE 0 END) AS
missing_barcode,  -- Counts rows where BARCODE is either NULL or empty
    SUM(CASE WHEN USER_ID IS NULL OR USER_ID = '' THEN 1 ELSE 0 END) AS
missing_user_id,  -- Counts rows where USER_ID is NULL or empty
    SUM(CASE WHEN FINAL_QUANTITY IS NULL OR FINAL_QUANTITY = '' THEN 1 ELSE 0
END) AS missing_final_quantity,  -- Counts rows where FINAL_QUANTITY is NULL or
empty
    SUM(CASE WHEN FINAL_SALE IS NULL OR FINAL_SALE = '' THEN 1 ELSE 0 END) AS
missing_final_sale,  -- Counts rows where FINAL_SALE is NULL or empty
    SUM(CASE WHEN PURCHASE_DATE IS NULL OR PURCHASE_DATE = '' THEN 1 ELSE 0 END)
AS missing_purchase_date  -- Counts rows where PURCHASE_DATE is NULL or empty
FROM transactions;  -- Querying the 'transactions' table
```

total_rows	missing_barcode	missing_user_id	missing_final_quantity	missing_final_sale	missing_purchase_date
50000	5762	0	0	12500	0

Final_Quantity doesn't return "missing" values but I run into the issue of entries listed as "zero" and there being a corresponding value in the Final_Sale column, which seems bizarre. Additionally, there are 12500 Final_Sale values missing and they have a corresponding value in the Final_Quantity column. Finally, there are also nearly 6k missing Barcodes, making it impossible to link these transactions to products.

3) *Missing Data Users Table: Check for missing values in ID, Birth Date and Gender*

```
-- This query checks for missing or empty values in key fields of the Users
table
SELECT
    COUNT(*) AS total_rows,  -- Counts the total number of rows in the
Users table
    SUM(CASE WHEN ID IS NULL OR ID = '' THEN 1 ELSE 0 END) AS
missing_user_id,  -- Counts rows where USER_ID is NULL or empty
    SUM(CASE WHEN BIRTH_DATE IS NULL OR BIRTH_DATE = '' THEN 1 ELSE 0 END)
AS missing_birth_date,  -- Counts rows where BIRTH_DATE is NULL or empty
    SUM(CASE WHEN GENDER IS NULL OR GENDER = '' THEN 1 ELSE 0 END) AS
missing_gender  -- Counts rows where GENDER is NULL or empty
FROM users;
```

total_rows	missing_user_id	missing_birth_date	missing_gender
100000	0	3675	5892

There are several missing entries in Birth Date and Gender, but I don't think that the number of missing values is significant enough to strongly impact demographic analysis.

1) *Data Consistency Issues Products Table:*

This query checks for category hierarchy issues in the Products table. Specifically, it identifies cases where Category 2, Category 3, or Category 4 exist without the required preceding categories:

```
SELECT *
FROM products
-- Check if CATEGORY_2 is present but CATEGORY_1 is missing
WHERE (CATEGORY_2 IS NOT NULL AND CATEGORY_1 IS NULL)
-- Check if CATEGORY_3 is present but CATEGORY_1 or CATEGORY_2 are missing
   OR (CATEGORY_3 IS NOT NULL AND (CATEGORY_1 IS NULL OR CATEGORY_2 IS
NULL))
-- Check if CATEGORY_4 is present but CATEGORY_1, CATEGORY_2, or CATEGORY_3
```

```
are missing
   OR (CATEGORY_4 IS NOT NULL AND (CATEGORY_1 IS NULL OR CATEGORY_2 IS NULL
OR CATEGORY_3 IS NULL))
-- Limit the results to the first 10 rows for review
LIMIT 10;
```

We're off to a good start here, there are no products which violate the categorical hierarchy.

Now let's make sure that there aren't any invalid Barcodes:

```
-- This query checks for barcodes that contain non-numeric characters
SELECT *
FROM products
WHERE BARCODE NOT GLOB '[0-9]*'  -- Check for non-numeric characters
LIMIT 10;
```

Barcode length isn't consistent but every Barcode in the Products table is a string of numbers.

Finally, we'll check to make sure no products have the same Barcode

```
-- This query retrieves all product information for barcodes that are
duplicated
SELECT p.*
FROM products p
JOIN (
    -- Subquery to find duplicate barcodes
    SELECT BARCODE
    FROM products
    WHERE BARCODE IS NOT NULL
    GROUP BY BARCODE
    HAVING COUNT(*) > 1
) dupes
ON p.BARCODE = dupes.BARCODE
```

Here I find a consistency issue with two duplicated Barcodes, 52336919068 and 17000329260, as each of these Barcodes returns two products with all the same information except Brand -- one result lists Schwarzkopf and the other lists Göt2b.

2) *Data Consistency Issues Transaction Table:*

During the high-level data exploration, I noticed decimal values in the Final Quantity column so let's look into those

```
-- Check for decimal values in FINAL_QUANTITY that do not end in .00
```

```
SELECT *
FROM transactions
WHERE FINAL_QUANTITY LIKE '%.%'
  AND FINAL_QUANTITY NOT LIKE '%.00';
```

This returns a table with 110 rows of transactions for which Final Quantity is a decimal value. This would be expected for products that are sold by weight but would otherwise raise a flag. It looks like all of these transactions occurred at grocery stores, so it's possible that these are just purchases of fruits or vegetables measured by weight, but it's worth digging into further.

We can alter the query to link it to the Products table and bring in Product Categories:

```
-- Join transactions with decimal FINAL_QUANTITY values to the products
table through BARCODE
SELECT t.*, p.CATEGORY_1, p.CATEGORY_2, p.MANUFACTURER, p.BRAND
FROM transactions t
JOIN products p ON t.BARCODE = p.BARCODE
WHERE t.FINAL_QUANTITY LIKE '%.%'
  AND t.FINAL_QUANTITY NOT LIKE '%.00';
```

I'll also look at the data where there's a value in the Final Quantity column of the transaction but the Final Sale column returns 0:

```
-- Check for transactions where FINAL_QUANTITY is a number >= 1 and
FINAL_SALE is exactly 0
SELECT *
FROM transactions
WHERE CAST(FINAL_QUANTITY AS FLOAT) >= 1  -- Ensure the quantity is
numerically 1 or more
  AND CAST(FINAL_SALE AS FLOAT) = 0       -- Ensure FINAL_SALE is exactly 0
  AND FINAL_QUANTITY != 'zero';           -- Exclude rows where
FINAL_QUANTITY is the string "zero"
```

There are 321 rows for which there is a value >=1 for Final Quantity but a value of 0 for Final Sale. This is an inconsistency in the data, as there should be a Final Sale value for these transactions unless they are free products.

The next data consistency issue I want to check for are Scan Dates that occur before Purchase Date:

```
-- Check for transactions where SCAN_DATE occurs before PURCHASE_DATE
SELECT *
FROM transactions
WHERE SCAN_DATE < PURCHASE_DATE
```

Here I return 94 rows where Scan Date falls before Purchase Date. You can only scan a receipt after you complete your purchases so Scan Date should not be earlier than Purchase Date.

3) *Data Consistency Issues Users Table:*

I'll look into Birth Dates to make sure everything looks good there:

```
-- Check for invalid birth dates (future or very old)
SELECT *
FROM users
WHERE BIRTH_DATE > DATE('now')  -- Future dates
   OR BIRTH_DATE < '1924-01-01'  -- Very old dates (before 1924)
```

```
-- Check for birth dates that are repeated frequently
SELECT BIRTH_DATE, COUNT(*) AS count
FROM users
WHERE BIRTH_DATE IS NOT NULL
GROUP BY BIRTH_DATE
HAVING COUNT(*) > 1
ORDER BY count DESC
LIMIT 10;
```

These two queries output a few things to note. Fortunately, there aren't any future Birth Dates but there are 61 instances of Birth Dates over 100 years old, with some as far back as 1901. Additionally, there are 1272 instances of 01-01-1970 at 00:00:00 -- it seems like this might be the default birthdate when users create their account.

1) *Data Formatting Errors in Product Table:*

As mentioned earlier, Barcode lengths are different but the field is free from formatting errors. I'll check the Category, Manufacturer and Brand fields for formatting errors as well:

```
-- Check for category formatting issues (leading/trailing spaces)
SELECT *
FROM products
WHERE TRIM(CATEGORY_1) != CATEGORY_1
   OR TRIM(CATEGORY_2) != CATEGORY_2
   OR TRIM(CATEGORY_3) != CATEGORY_3
   OR TRIM(CATEGORY_4) != CATEGORY_4
```

```
 -- Check for manufacturer and brand formatting issues (leading/trailing
spaces or inconsistent capitalization)
```

```
SELECT *
FROM products
WHERE TRIM(MANUFACTURER) != MANUFACTURER
   OR TRIM(BRAND) != BRAND
```

There are no formatting issues in the Category fields, Manufacturer or Brand.

2) *Data Formatting Errors in the Transaction Table:*

Here, I want to check the Store Names column to see if the same store is listed by different names:

```
-- Identify inconsistent store names (e.g., different capitalizations or
formats)
SELECT STORE_NAME, COUNT(*)
FROM transactions
GROUP BY STORE_NAME
HAVING COUNT(*) > 1
ORDER BY STORE_NAME;
```

This query returns each Store Name along with the corresponding number of occurrences. For example, the output shows several instances of “Dick’s” and “Dick’s Sporting Goods,” which likely refer to the same entity. These variations should be consolidated under a single store name for consistency.

3) *Data Formatting Errors in the Users Table:*

Looking at the unstructured data, the Gender column sticks out as a field where there are data formatting errors:

```
-- Check for invalid or inconsistent gender values
SELECT DISTINCT GENDER
FROM USER_TAKEHOME
WHERE GENDER NOT IN ('male', 'female', 'non-binary', 'transgender',
'other', '')
```

From the output of this query I see that non-binary, not listed, and prefer not to say are formatted in various ways. Similar to the store names mentioned above, these variations should be consolidated for consistency.

3) Are there any fields that are challenging to understand?

1) *Products Table*
    a) I see that Category 1, Category 2, Category 3 and Category 4 represent a hierarchy but it's difficult to understand the rules behind the depth of the categorizations. Some products have fewer categories, others have more but it's not always clear as to why

```
-- Check how deeply products are categorized across the four category
levels
SELECT COUNT(*) AS total_products,
       SUM(CASE WHEN CATEGORY_4 IS NOT NULL THEN 1 ELSE 0 END) AS
fully_categorized,
       SUM(CASE WHEN CATEGORY_3 IS NOT NULL THEN 1 ELSE 0 END) AS
categorized_thru_3,
       SUM(CASE WHEN CATEGORY_2 IS NOT NULL THEN 1 ELSE 0 END) AS
categorized_thru_2,
       SUM(CASE WHEN CATEGORY_1 IS NOT NULL THEN 1 ELSE 0 END) AS
categorized_thru_1
FROM products;
```

There's a big dropoff from products that are categorized through Category 3 (784986) and products that are fully categorized (67459), so is Category 4 really necessary?

2) *Transactions Table*
    a) Mentioned this already but the decimal values in the Final Quantity column stick out. It's one thing if they are simply products measured and sold by weight but it's another thing if they're not.
    b) What does it mean when Final Sale is 0? Are these valid transactions, such as a free product or a return? More info on the business context would help clear this up.
    c) Again, there are instances where Scan Date falls before Purchase Date. Are there exceptions that allow for this to happen?
3) *Users Table*
    a) The Language field is confusing here

```
-- List all distinct values in the LANGUAGE field
SELECT DISTINCT LANGUAGE
FROM users
```

Language is either "en", "es-419" or NULL. I assume en is English and es is Spanish but why the "-419?" Does 419 represent a regional code for Spanish?

And if so, should there be a regional code for English for standardization purposes?