TUNIS BUSINESS SCHOOL
UNIVERSITY OF TUNIS

Business intelligence and Database Management System

Business Intelligence Research on Starbucks

Prepared by:

Rayen Rouaissi

Mohamed Aziz Belhadj

Achref Khairi

Maha Jdidi

Contents

1 Introduction

This business intelligence project is focused on analyzing the characteristics of Starbucks customers and their ratings based on quality, service, and promotions. This study aims to give insights into how different factors such as age, income, and gender can influence customer ratings.

After understanding the relationships and interactions between the facts and dimensions we succeeded in having a full view of preferences, which can help Starbucks improve their offerings and consequently drive more business.

Throughout this project, we aim to improve overall customer satisfaction and build better implementation strategies for Starbucks in the United States.

2 Implementation

2.1 Data Gathering

We extracted the Starbucks stores' and customers' ratings basically from GitHub and Kaggle datasets. All the raw data is stored in the Raw Data folder.

2.2 Data Preparation

For the data preparation, we used Python to manipulate data and configure it for the data warehouse. First, we started by converting all the data into one format (JSON) from different formats (CSV, Excel, JSON).

We employed the use of pandas' library in Python and Talend to make necessary transformations and apply the ETL process. All the code employed is stored in the Transformations folder.

Then we moved to change. Firstly, we created a script that allows us to clean and manipulate the customer file because it contains some undesired columns, and we fixed their IDs, so they match with the payment method. Additionally, we created a transfrom_income function that helped us to transform the format of average spent per visit to the dollar sign ($) since we are dealing with Starbucks stores based in the United States. Moreover, we created a new dataset called working hours where we calculated the total number of working hours per store in each day. For the map and sort, we used Talend. All the transformations are stored in the result folder.

```
from google.colab import files
temp_file_path = "Category.json"
with open(temp_file_path, 'w') as temp_file:
    temp_file.write(valid_json)
files.download(temp_file_path)
```

```
Data.insert(0, 'user_id', range(1, len(Data) + 1))
```

```
print(Data)
```

```
     user_id Visit_Frequency Membership Avg_amount_spent_pervisit  \
0          1           Rarely        Yes             Less than RM20
1          2           Rarely        Yes             Less than RM20
2          3          Monthly        Yes             Less than RM20
3          4           Rarely         No             Less than RM20
4          5          Monthly         No        Around RM20 - RM40
..       ...              ...        ...                       ...
117      118          Monthly        Yes        Around RM20 - RM40
118      119          Monthly        Yes            More than RM40
119      120           Rarely         No             Less than RM20
120      121           Rarely         No             Less than RM20
121      122           Rarely         No             Less than RM20
```

```
data.head()
```

	name	location	Date	Rating	Review	Image_Links
0	Helen	Wichita Falls, TX	Reviewed Sept. 13, 2023	5.0	Amber and LaDonna at the Starbucks on Southwes...	['No Images']
1	Courtney	Apopka, FL	Reviewed July 16, 2023	5.0	** at the Starbucks by the fire station on 436...	['No Images']
2	Daynelle	Cranberry Twp, PA	Reviewed July 5, 2023	5.0	I just wanted to go out of my way to recognize...	['https://media.consumeraffairs.com/files/cach...
3	Taylor	Seattle, WA	Reviewed May 26, 2023	5.0	Me and my friend were at Starbucks and my card...	['No Images']
4	Tenessa	Gresham, OR	Reviewed Jan. 22, 2023	5.0	I'm on this kick of drinking 5 cups of warm wa...	['https://media.consumeraffairs.com/files/cach...

```
data.drop(['Review',  'Image_Links'], axis=1, inplace=True)
```

```
data.head()
```

```python
def transform_income(income_str):
    income_str = income_str.replace('RM', '').replace(',', '')

    if 'Less than' in income_str:
        return income_str.replace('Less than', '$')
    elif 'More than' in income_str:
        return income_str.replace('More than', '$')

    if 'Around' in income_str:
        return income_str.replace('Around', '').strip()

    if '-' in income_str:
        lower, upper = map(int, income_str.replace('$', '').split('-'))
        return f"${(lower + upper) / 2:.0f}"

    return income_str
```

```python
Data['Avg_amount_spent_pervisit'] = Data['Avg_amount_spent_pervisit'].apply(transform_income)
```

```python
def calculate_working_hours(time_range):
    if pd.notna(time_range) and ' to ' in time_range:
        start, end = time_range.split(' to ')
        start_time = pd.to_datetime(start.replace(' AM', '').replace(' PM', ''), format='%I:%M')
        end_time = pd.to_datetime(end.replace(' AM', '').replace(' PM', ''), format='%I:%M')
        end_time += pd.Timedelta(hours=12)
        return (end_time - start_time).seconds / 3600
    else:
        return 0
```

```python
for day in df_result.columns[1:]:
    df_result[day] = df_result[day].apply(calculate_working_hours)
```

```python
df_result.head()
```

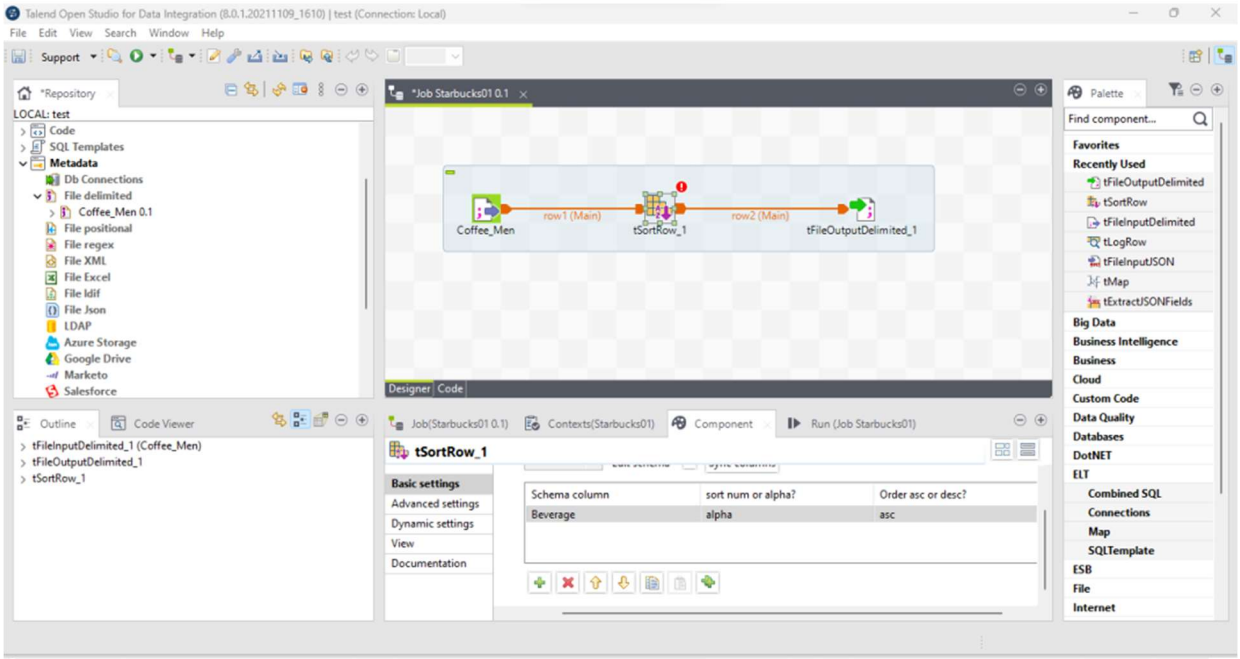
	storeId	sunday	monday	tuesday	wednesday	thursday	friday	saturday
0	S1000	13.0	13.0	13.0	13.0	13.0	13.0	13.0
1	S1001	14.0	14.5	14.5	14.5	14.5	14.5	14.0
2	S1002	12.0	14.0	14.0	14.0	14.0	14.0	12.0
5	S1005	14.0	16.0	16.0	16.0	16.0	16.0	14.0
8	S1008	14.0	14.0	14.0	14.0	14.0	14.0	14.0

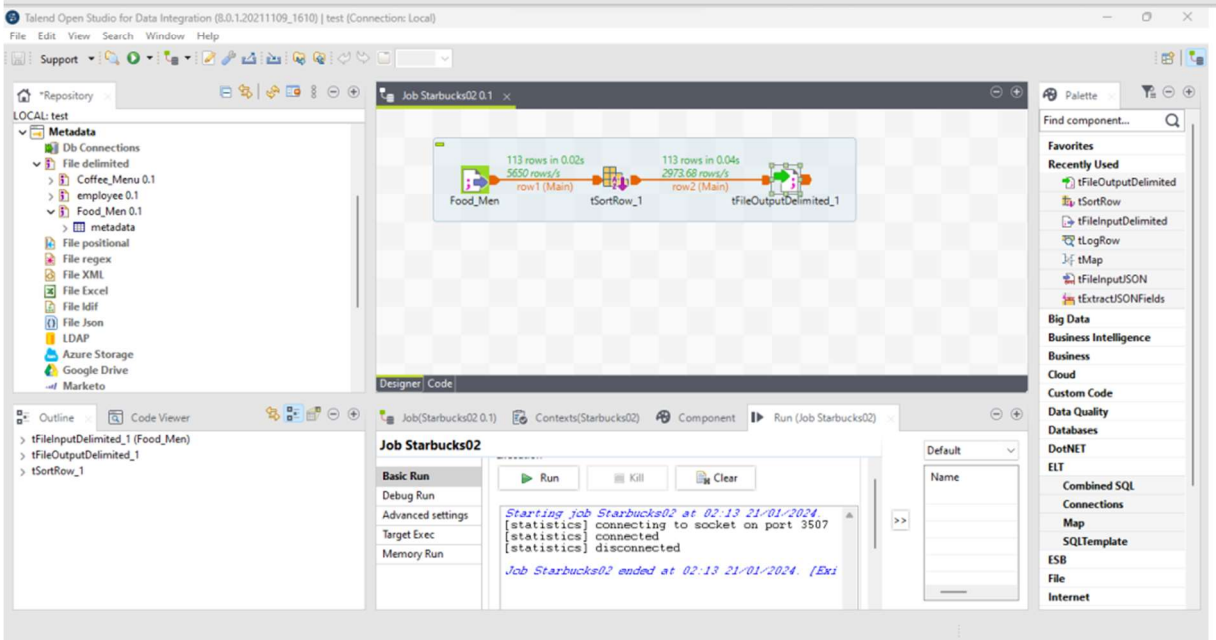
```
data["became_member_on"] = data["became_member_on"].apply(lambda x: str(x))
```

```
data["became_member_on"] = data["became_member_on"].apply(lambda x: x[0:4] + '-' + x[4:5] + '-' + x[5:6])
```

```
data.head()
```

	gender	age	id	became_member_on	income
0	None	118	68be06ca386d4c31939f3a4f0e3dd783	2017-0-2	NaN
1	F	55	0610b486422d4921ae7d2bf64640c50b	2017-0-7	112000.0
2	None	118	38fe809add3b4fcf9315a9694bb96ff5	2018-0-7	NaN
3	F	75	78afa995795e4d85b5d9ceeca43f5fef	2017-0-5	100000.0
4	None	118	a03223e636434f42ac4c3df47e8bac43	2017-0-8	NaN





2.3 Data Storage
2.3.1 Storage
For data storage, we employed the use of SQL to insert and map tables in our code and PostgreSQL as the database management system.
We have 2 main tables in the database, Customers, and Stores. the other tables are Ratings, Menu Category, Employees, Working Hours, Customer Payment. Attached here is the code for the SQL.

```
CREATE TABLE "Coffee_Men" (
    "Product_ID" TEXT PRIMARY KEY,
    "Beverage_category" TEXT,
    "Beverage" TEXT,
    "Beverage_prep" TEXT,
    "Calories" INT,
    "Total_Fat_g" TEXT,
    "Trans_Fat_g" NUMERIC(2, 1),
    "Saturated_Fat_g" NUMERIC(2, 1),
    "Sodium_mg" INT,
    "Total_Carbohydrates_g" INT,
    "Cholesterol_mg" INT,
    "Dietary_Fibre_g" INT,
    "Sugars_g" INT,
    "Protein_g" NUMERIC(3, 1),
    "Vitamin_A_DV" NUMERIC(2, 0),
    "Vitamin_C_DV" NUMERIC(3, 0),
    "Calcium_DV" NUMERIC(2, 0),
    "Iron_DV" NUMERIC(5, 3),
    "Caffeine_mg" TEXT,
    "Type_Menu1" TEXT
);
```

```
CREATE TABLE  "Customers" (
    "User_ID" INT PRIMARY KEY,
    "name" TEXT,
    "location" TEXT,
    "Gender" TEXT,
    "Age" INT,
    "income" NUMERIC(7, 1)
);
CREATE TABLE  "Payment" (
    "User_ID" INT PRIMARY KEY,
    "PAY_method" TEXT,
    FOREIGN KEY ("User_ID") REFERENCES "Customers" ("User_ID")
    );
CREATE TABLE "Store" (
    "StoreId" TEXT PRIMARY KEY,
    "name" TEXT,
    "city" TEXT,
    "country" TEXT
);
```

```sql
CREATE TABLE "working_Hours" (
    "StoreId" TEXT PRIMARY KEY,
    "sunday" NUMERIC(3, 1),
    "monday" NUMERIC(4, 2),
    "tuesday" NUMERIC(4, 2),
    "wednesday" NUMERIC(4, 2),
    "thursday" NUMERIC(4, 2),
    "friday" NUMERIC(4, 2),
    "saturday" NUMERIC(3, 1),
    FOREIGN KEY ("StoreId") REFERENCES "Store" ("StoreId")
);
CREATE TABLE  "Category" (
    "StoreId" TEXT PRIMARY KEY,
    "Type_Menu1" TEXT,
    "Type_Menu2" TEXT,
    FOREIGN KEY ("StoreId") REFERENCES "Store" ("StoreId")
);

CREATE TABLE "Food_Men" (
    "Food_ID" TEXT PRIMARY KEY,
    "Menu_Item" TEXT,
    "Calories" INT,
    "Fat_g" NUMERIC(3, 1),
    "Carb_g" INT,
    "Fiber_g" INT,
    "Protein_g" INT,
    "Type_Menu2" TEXT
);
```

```
CREATE TABLE "employees" (
    "EMPLOYEE_ID" INT PRIMARY KEY,
    "FIRST_NAME" TEXT,
    "LAST_NAME" TEXT,
    "EMAIL" TEXT,
    "PHONE_NUMBER" TEXT,
    "HIRE_DATE" TIMESTAMP,
    "SALARY" INT,
    "StoreId" TEXT,
    FOREIGN KEY ("StoreId") REFERENCES "Store" ("StoreId")
);
CREATE TABLE "Ratings" (
    "Store_Id" TEXT,
    "User_ID" INT PRIMARY KEY,
    "Visit_Frequency" TEXT,
    "Membership" TEXT,
    "Avg_amount_spent_pervisit" TEXT,
    "Brand_satisfaction" INT,
    "price_satisfaction" INT,
    "promotions_satisfaction" INT,
    "Ambiance" INT,
    "wifi_quality" INT,
    "Service_satisfaction" INT,
    "Overall_satisfaction" INT,
    "Channels" TEXT,
    "Loyalty" TEXT,
    FOREIGN KEY ("User_ID") REFERENCES "Customers" ("User_ID"),
    FOREIGN KEY ("Store_Id") REFERENCES "Store" ("StoreId")
);
```

Query Query History

```
1 copy public."working_Hours" ("StoreId","sunday", "monday", "tuesday", "wednesday", "thursday", "friday", "saturday")
2 FROM 'C:/Users/medaz/DOWNLO~1/WORKIN~1.CSV'
3 WITH CSV HEADER;
```

Query Query History

```
1 copy public."Ratings" ("Store_Id", "User_ID", "Visit_Frequency", "Membership",
2 "Avg_amount_spent_pervisit", "Brand_satisfaction", "price_satisfaction", "promotions_satisfaction",
3 "Ambiance","wifi_quality", "Service_satisfaction",
4 "Overall_satisfaction", "Channels", "Loyalty") FROM 'C:/Users/medaz/DOWNLO~1/Ratings.csv' WITH CSV HEADER;
```

Query Query History

```
1 copy public."Food_Men" ("Food_ID", "Menu_Item", "Calories", "Fat_g",
2 "Carb_g", "Fiber_g", "Protein_g", "Type_Menu2") FROM 'C:/Users/medaz/DOWNLO~1/FOOD_M~2.CSV'
3 WITH CSV HEADER;
```

```
copy public."Customers" ("User_ID", "name", "location", "Gender", "Age", "income")
FROM 'C:/Users/medaz/DOWNLO~1/CUSTOM~1.CSV'
WITH CSV HEADER;
```

Query | Query History

```
copy public."Food_Men" ("Food_ID", "Menu_Item", "Calories", "Fat_g",
"Carb_g", "Fiber_g", "Protein_g", "Type_Menu2") FROM 'C:/Users/medaz/DOWNLO~1/FOOD_M~2.CSV'
WITH CSV HEADER;
```

Query | Query History

```
copy public."Payment" ("User_ID", "PAY_method") FROM 'C:/Users/medaz/DOWNLO~1/Payment.csv'
WITH CSV HEADER;
```

Query | Query History

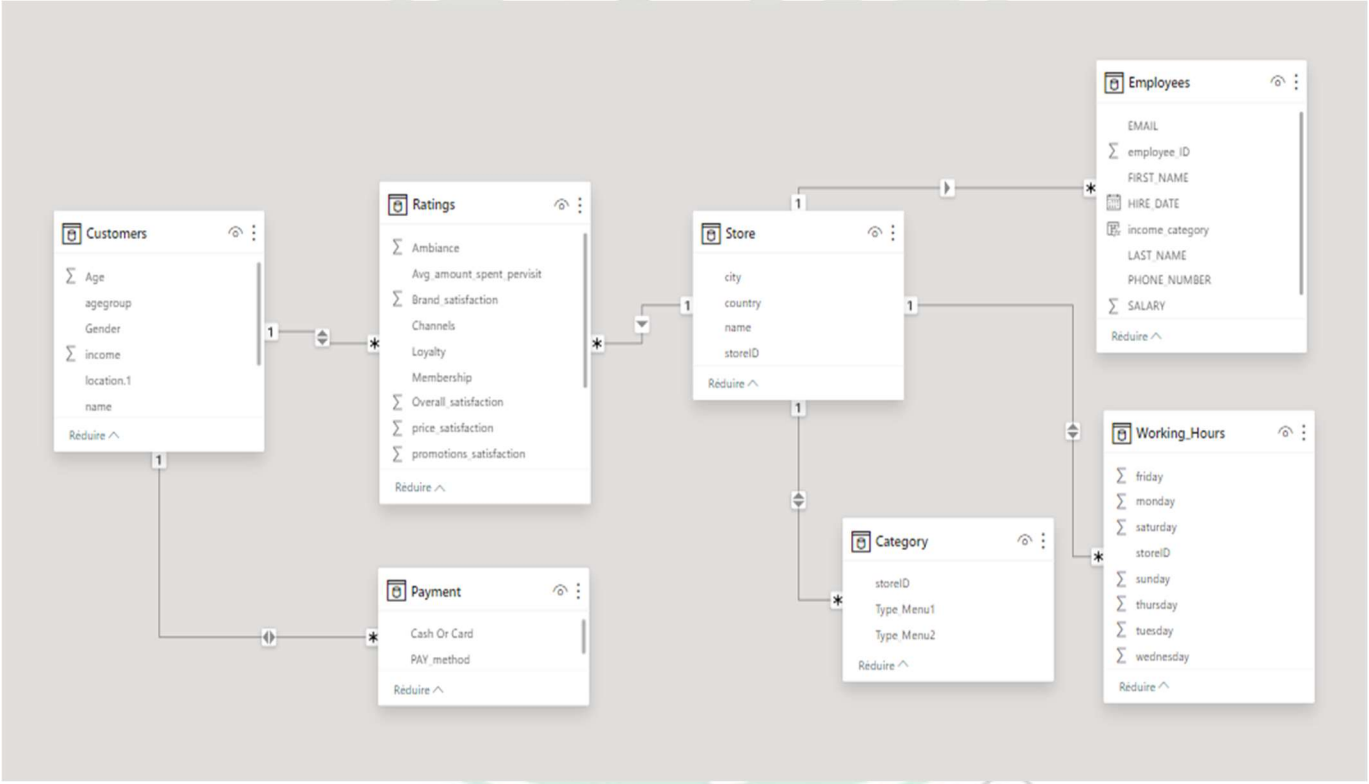
```
copy public."employees" ("EMPLOYEE_ID", "FIRST_NAME", "LAST_NAME", "EMAIL", "PHONE_NUMBER", "HIRE_DATE", "SALARY", "S
FROM 'C:/Users/medaz/DOWNLO~1/EMPLOY~1.CSV'
WITH CSV HEADER;
```

Query | Query History

```
copy public."Coffee_Men" ("Product_ID", "Beverage_category", "Beverage", "Beverage_prep",
"Calories", "Total_Fat_g", "Trans_Fat_g", "Saturated_Fat_g", "Sodium_mg", "Total_Carbohydrates_g",
"Cholesterol_mg", "Dietary_Fibre_g", "Sugars_g", "Protein_g", "Vitamin_A_DV", "Vitamin_C_DV", "Calcium_DV", "Iron_DV"
FROM 'C:/Users/medaz/DOWNLO~1/COFFEE~1.CSV'
WITH CSV HEADER;
```

- **Customers:** represents the customer information.

- **Store:** represents the store information (store ID, name, city, and country).
- **Ratings:** represents the service, promotions, Wi-Fi, and overall satisfaction. Additionally, the loyalty and membership of each customer to the store.
- **Category:** represents the type of menu of each store.
- **Coffee Menu:** represents the drinks available on the menu and their characteristics.
- **Food Menu:** represents the foods available on the menu and their characteristics.
- **Working Hours:** represents the number of working hours for each store in each day.
- **Employees:** represents the information of employees including their salary working in a given store.
- **Payment:** represents the payment method desired for each customer.



2.3.2 Fact

The fact in this data is the ratings given by the customer to each store. The types of

ratings include:
- **Ambiance Rating**
- **Brand Rating**
- **price Rating**
- **Wifi Rating**
- **Promotions Rating**
- **Overall Rating**

2.3.3 Dimensions

Dimensions included in this data are:
- **Store**
- **Customer**

Dimensions that are derived from the customer dimensions include:
- **Payment Method**

Dimensions that are derived from the restaurant's dimensions include:
- **Working Days and hours**
- **Menu type.**
- **Employees**

Finaly, we made the Snowflake Schema as Data Warehouse Schema because of one-to-many relationships between customers' and stores' values in the customer.

2.4 Data Visualization

For the Data visualization, we used power bi to understand customer behaviours and their satisfaction about the offerings of Starbucks. It also helped us to get insights on some points and channels that Starbucks can work on to attract more customers and serve them better.

• Most customers in the dataset live in California.
• The average income of customers is 66.800$ and it ranges from 30 to 120K.
• Most of the customers are males.
• Most of the customers belong to Generation X and Z.
• Most of the customers use cash as a payment method.
• Most customers are satisfied with service, ambiance, and Brand.
• Customers are fairly satisfied with the price and Wi-Fi.
• Each customer spends 20$ on average for each visit.
• 77% of current customers are satisfied with Starbucks and consider staying loyal in the future.
• 49.18% of current customers have a membership card.
• The average monthly salary for employees is 6180$.
• Most of the employees are in the low- and medium-income category.

3 Conclusion

In conclusion, the research proved that most of the customers in the dataset are living in California. They are mostly males who have a medium level of income and belong to generation X and Z. Most of them are satisfied with what Starbucks is offering and consider staying loyal to the brand in the future. Wi-Fi and price could be improved to meet the customers' needs.