**Business Insights Assessment**

### **Project Overview**

* **Objective:** Build a Data Engineering pipeline using provided orders related CSV files, calculate key metrics, and create business insights using the metrics as calculated.

### **Step-by-Step Plan**

#### **Step 1: Source Data Explanation**

1. **Task:**
   * Download the orders related data files from Google Drive.
   * Verify the integrity of the downloaded files (e.g., check for missing or corrupt values).

**All the files are present in the google drive link below**[**order\_items**](https://drive.google.com/file/d/1GXRZNgfngU6Yal6hzs5NClDgJoN3vEKZ/view?usp=drive_link)[**order\_item\_options**](https://drive.google.com/file/d/1l9anZqzpgTsQXe1ZTg-ihhn-9SBsa2H_/view?usp=drive_link) [**date\_dim**](https://drive.google.com/file/d/1v1rPl4nJp1B_nQmNm_Nrz2ZeBkppKRYh/view?usp=drive_link)

**1. order\_items *(203,519 records)***

Captures transactional-level data for each item in a customer's order.

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| app\_name | String | Name of the ordering platform or channel used |
| restaurant\_id | String / Int | Unique identifier for the restaurant |
| creation\_time\_utc | Timestamp | Timestamp of when the order was placed (UTC) |
| order\_id | String / Int | Unique identifier for the order |
| user\_id | String / Int | Unique identifier for the customer placing the order |
| printed\_card\_number | String | Tokenized or masked loyalty card number |
| is\_loyalty | Boolean | Flag indicating loyalty membership (True/False) |
| currency | String | Currency of the transaction (e.g., USD) |
| lineitem\_id | String / Int | Unique identifier for the specific item in the order |
| item\_category | String | Category of the item (e.g., Beverage, Entree, etc.) |
| item\_name | String | Name of the item |
| item\_price | Decimal | Unit price of the item |
| item\_quantity | Integer | Quantity of the item purchased |

**2. order\_item\_options *(193,017 records)***

Details add-ons, customizations, or modifiers associated with order items.

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| order\_id | String / Int | Identifier linking to the parent order |
| lineitem\_id | String / Int | Identifier linking to the specific order item |
| option\_group\_name | String | Group category for options (e.g., Size, Toppings) |
| option\_name | String | Selected option or customization (e.g., Extra Cheese) |
| option\_price | Decimal | Price of the option (negative if it's a discount) |
| option\_quantity | Integer | Number of times the option was added |

**3. date\_dim *(Date Dimension Table)***

Used for time-based joins and calendar aggregations.

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| date\_key | Date | Full calendar date |
| day\_of\_week | String | Day of the week (e.g., Monday) |
| week | Integer | Week number in the year |
| month | String | Month name (e.g., January) |
| year | Integer | Calendar year |
| is\_weekend | Boolean | Whether the date falls on a weekend |
| is\_holiday | Boolean | Flag indicating if the date is a recognized holiday |
| holiday\_name | String | Name of the Holiday |

#### **Step 2: Initial Data Analysis**

1. **Task:**
   * Analyze the CSV files to understand the schema, relationships, and data quality issues.(Manually)
   * Identify key columns for joins and insights.(Manually)

#### **Step 3: Define Data Engineering Pipeline/Architecture**

1. **Task:**
   * Design a pipeline architecture for ingesting, transforming, and storing data. Design a production level pipeline that includes scheduling, encryption, failure reload mechanism.
   * Design an architecture based on below requirements:

**Consider your source as SQL Server i.e you need to pull data from your database tables. A video step to load data in SQL Server can be found** [**here**](https://drive.google.com/file/d/10-YApwQA5rsO8s4dcSzI82BU_y4pLPDe/view?usp=drive_link)**. (WINDOWS OS)**

**The entire architecture should be designed using *AWS resources*. The company does not want to get new licenses. The company does not want to use Snowflake, DBT, etc external tools.**

**All logics should be done using *PYSPARK.***

***Please schedule the pipeline jobs once daily as a batch process.***

***Define a Data Model that satisfies below requirement:***

* **Customer Lifetime Value (LTV), and how would you structure it to show how LTV evolves for each customer daily? (Primary Goal)**
* **Look into Step 5 for additional business requirements while creating your data model. (Secondary Goal)**

***Present the architecture for Subject Matter Expert (SME) approval.***

***Move head once clearance from the SME***

**Deliverable:**

* + Pipeline architecture diagram and explanation.
  + Create a solution design document.
  + Written approval from the SME.
  + Use tools like draw.io, miro, etc for diagrams.

#### **Step 4: Build the Data Pipeline**

**Deliverable:**

* Working pipeline that can handle raw data ingestion, transformation in AWS.

#### **Step 5: Define and Calculate Metrics as per Business Requirement:**

***Primary Metrics:***

1. **Customer Lifetime Value (CLV):**

**Goal:** Estimate how much total revenue a customer will generate over their entire relationship with the business.

**Why it matters:** Helps prioritize high-value customers, plan marketing budgets wisely, and improve retention strategies.

**How to do it:**

* Use order\_items and order\_item\_options to compute revenue per order.
* Aggregate total spend per customer\_id**.**

**Group CLV values (for tagging):**

* High CLV: Top 20% customers
* Medium CLV: Mid 60%
* Low CLV: Bottom 20%

***Secondary Metrics***

1. **Customer Segmentation & Behavior:**

**Goal:** Group customers based on spending and activity to support campaign targeting.

**Why it matters:** Enables personalized offers and engagement.

**How to do it:**

* **Use RFM logic based on order\_items:**
  + Recency: Days since last purchase
  + Frequency: Number of purchases in last N months
  + Monetary: Total spend in last N months
* **Segment:**
  + VIPs: High R, F, M
  + New Customers: Low F, high R
  + Churn Risk: Low R, low F

1. **Churn Indicators:**

**Goal:** Build a customer activity profile to help marketing identify at-risk customers (no predictions).

**Why it matters:** Allows timely retention actions by analysts.

**How to do it:**

* For each customer\_id, compute:  
    
   • Days since last order  
   • Average gap between orders  
   • % change in spend over last N periods
* Tag customers based on inactivity thresholds (e.g., >45 days = “at risk”)

1. **Sales Trends Monitoring:**

**Goal:** Generate time-based summaries to analyze sales patterns.

**Why it matters:** Helps identify peak periods and plan resources.

**How to do it:**

* + Aggregate daily, weekly, and monthly revenue from order\_items
  + Break down by:  
     • Location  
     • Menu category (if available)  
     • Time of day (optional)

1. **Loyalty Program Impact**

**Goal:** Compare loyalty members vs non-members in terms of spend and engagement.

**Why it matters:** Evaluates ROI of the loyalty program.

**How to do it:**

* + Filter order\_items by is\_loyalty = true vs false
  + Compare per-customer:  
     • Average Spend  
     • Repeat Orders  
     • Lifetime Value

1. **Top-Performing Locations**

**Goal:** Identify best and worst-performing store locations.

**Why it matters:** Informs decisions about promotions, staffing, or expansion.

**How to do it:**

* Group order\_items by location\_id (or store\_id if available)
* Calculate:  
   • Total revenue  
   • Average order value  
   • Orders per day/week
* Rank locations based on revenue

1. **Pricing & Discount Effectiveness**

**Goal:** Measure how discounts affect revenue and profitability.

**Why it matters:** Helps optimize pricing strategies.

**How to do it:**

* Use order\_item\_options.option\_price to detect discounts (option\_price < 0)
* Compare:  
   • Revenue from discounted orders vs non-discounted  
   • Number of orders before/after applying discounts

#### **Step 6: Build the Dashboard using Streamlit:**

Please go through this link for working on Streamlit for Visualization:

[Streamlit Dashboard](https://www.youtube.com/watch?v=p2pXpcXPoGk)

1. **Task:**

Design an interactive dashboard showcasing the calculated metrics and insights.

**Deliverable:**

* **Customer Segmentation Dashboard:**
  + **Question:** What distinct customer segments emerge when grouping customers by purchase behavior (total spend, frequency, recency) and loyalty status?
  + **Focus:** Visualize segmentation (e.g., via RFM scores) to enable targeted marketing.
* **Churn Risk Indicators Dashboard:**
  + **Question:** Which metrics—such as days since last order, average order interval, and spend trends—correlate with a higher churn risk?
  + **Focus:** Highlight threshold-based alerts (e.g., customers at risk) to prompt re-engagement actions.
* **Sales Trends and Seasonality Dashboard:**  
  + **Question:** What are the monthly and seasonal trends in sales, and how do these vary by product category or location?
  + **Focus:** Track weekly/monthly sales aggregates and holiday spikes to support inventory planning and staffing decisions.
* **Loyalty Program Impact Dashboard:**  
  + **Question:** How does loyalty membership affect customer spending and repeat order rates?
  + **Focus:** Compare key metrics (CLV, average order value, repeat purchase rate) between loyalty and non-loyalty customers to assess program effectiveness and inform potential adjustments.
* **Location Performance Dashboard:**  
  + **Question:** Which restaurant locations generate the highest revenue, and what operational metrics (e.g., average order size, customer retention) distinguish top performers?
  + **Focus:** Rank locations and highlight actionable insights for expansion or targeted improvements.
* **Pricing and Discount Effectiveness Dashboard:**  
  + **Question:** How are discounts and promotions affecting overall sales volume and net revenue?
  + **Focus:** Compare revenue and profit (gross versus net after discount adjustments) for discounted versus full-price transactions to refine pricing strategies.

#### **Step 7: Submission and Review**

1. **Task:**
   * Submit all deliverables, including:
     + Pipeline documentation.
     + Code files (ETL scripts, Pyspark SQL queries).
     + Setup Configurations (if any)
     + Final dashboard.
     + Create a CI/CD pipeline using github.
2. **Instructions:**
   * Ensure all files are properly named and documented.
   * **Include a short video or presentation explaining the work**.
   * Explain **‘WHY**’ behind each tech stack used in your design.

*Thank You.*