



# EBUS 3030

Assignment 1

Group: POE

Team members:

Sander Fabian Visvaseelan (c3418982)

Oh Chen Neen (c3415898)

Victor Chua Jia Zhi (c3418248)

Brent Lee Ting Zhen (c3415641)

Bui Ho Minh Tho (c3415439)

Wei Yang Cheong (c3415898)

# Table of Contents

1.0	Introduction .....	2
1.1	Problem Statement.....	2
1.2	Proposed Solution .....	2
2.0	Deriving an Answer .....	2
2.1	Assumptions .....	2
2.2	Identifying the Anomalies.....	3
2.3	Cleaning the Database .....	3
2.4	Creating a Datamart for Analysis .....	3
2.5	Findings.....	3
3.0	Conclusion .....	3
3.1	Suggestions .....	4
3.2	Benefits .....	4
3.3	Closing statement: .....	6
4.0	Appendices .....	6
4.1	Datamart Model .....	6
4.1.1	Datamart Diagram .....	6
4.1.2	DataMart Definitions .....	6
4.1.3	Design Rationale .....	10
4.1.4	DataMart Implementation.....	10
4.2	Anomaly Detection .....	14
	Solutions for cleaning the Data .....	16
4.3	ETL Processes.....	17
4.4	Base Analysis .....	19
4.5	References .....	24

## 1.0 Introduction

Hello Jenny Landers,

We are thrilled to present this business intelligence report to help “Bits&Bobs” identify their top salesperson and substantiate those claims with our findings and a relevant case study for their newly initiated incentive reward. This business report will further delve into detailed insights about the staff performance and have relevant recommendations to help improve the staff performance. Upon request, we have also developed an appropriate data mart specially catering to “Bits&Bobs” non-technical staff, ensuring simplicity and performance for future use.

### 1.1 Problem Statement

“Bits&Bobs” is currently struggling to identify their staff's sales productivity to provide an incentive program to motivate them. The organisation needs a team of impartial business intelligence specialists to help them produce a data mart and derive insights to identify the staff productivity and give feasible recommendations.

### 1.2 Proposed Solution

We are here to solve “Bits&Bobs” current struggles and have developed a data mart to store relevant information that will identify the top salesperson and is not limited to only identifying those criteria the organisation has requested. To address those issues, we developed a design for the data mart, identified a few anomalies in the Excel sheet the organisation provided and performed ETL to clear those anomalies through SQL Server Integration Services. The clean data would then be passed into the dimension tables and fact tables. We would then pluck those tables into a power BI and from there, extract meaningful insights to provide feasible recommendations.

## 2.0 Deriving an Answer

### 2.1 Assumptions

Based on the database presented, a few assumptions about the data were made:

1. Staff ID should be Unique.
2. All fields should have values.
3. Receipts should be associated to 1 customer.
4. Customer ID should be unique.
5. If a customer is loyal, with 5 or more unique items, all items will be discounted by 12.5%.
6. Receipts should not span over multiple dates.
7. Each receipt's transaction row can only be associated to a singular staff.

## 2.2 Identifying the Anomalies

The database was sent through a variety of checks to identify any anomalies in the data. This process is illustrated within Appendix 4.2.

## 2.3 Cleaning the Database

With the anomalies identified, the database was cleaned using SSIS. This process is illustrated within Appendix 4.3.

## 2.4 Creating a Datamart for Analysis

A Datamart was designed and created for use in analysis, the details are highlighted in Appendix 4.1. The implementation of the Datamart can be observed in 4.1.4.

## 2.5 Findings

From the Datamart, the team decided to weigh a few different KPIs relevant to appointing the best performance for sales. The KPIs are listed below, along with the results of the highest achieving within that category.

1. **Top salesperson by sales amount in dollars: Amber Hill**  
This KPI measures the highest earning salesperson in the year. Profit is an essential driver behind any organization.
2. **Top salesperson by inventory sold: Joseph Reed**  
This KPI shows an ability to move products quickly. Inventory space costs the organization money, moving products quickly can reduce inventory space used and the cost associated with it.
3. **Top salesperson by Average Transaction: Leah Harris**  
This KPI shows the ability to upsell, having customers purchase more than their initial needs.
4. **Top salesperson by Average Transaction per Month: Leah Harris**  
This KPI shows consistent monthly sales.
5. **Top salesperson by consistency: Megan James**  
This KPI also measures consistency with the standard deviation of the average monthly sales. The lowest deviation indicates the consistency of practice.

This is explored further in appendix 4.4.

## 3.0 Conclusion

The Organization should award Amber Hill. Amber Hill's achievement as the top salesperson in terms of highest profit is a crucial milestone for the company, driven by her exceptional ability to deliver outstanding financial results. The organisation's decision to reward her based on yearly bottom line and profit aligns with the following key considerations:

**Financial Impact:** Amber's track record of delivering the highest profit directly bolsters the company's financial strength. Her remarkable sales skills and consistent success ensure substantial revenue, positively impacting the company's overall financial stability.

**Strategic Focus:** Recognizing Amber's profit performance emphasises the strategic focus on financial results and profitability. It highlights the significance of bottom-line growth in achieving the

organisation's long-term financial objectives.

**Market Competitiveness:** Amber's ability to consistently drive the highest profit underlines the competitiveness and appeal of the company's products and services. Her achievement reflects market strength and customer demand.

In conclusion, rewarding Amber Hill based on her highest profit contribution directly serves the company's financial interests. Her impressive financial results validate the effectiveness of the company's strategies, emphasise financial stability, and help steer the organization towards higher profitability. This approach aligns with the organisation's emphasis on achieving a strong yearly bottom line and profit.

### 3.1 Suggestions

The following recommendations are aimed at enhancing sales staff morale and performance at "Bits&Bobs". These proposals are based on insights derived from our analysis of the provided data, as well as the business rules governing transactions at the store.

#### **Incentives for Sales Staff:**

We recommend implementing a structured incentive program to motivate the sales team. This program should include the following elements:

1. **Sales Bonuses or Commissions:** Offer monetary bonuses or commissions to top-performing sales staff based on their sales achievements. The rationale behind this is to directly reward and motivate individuals who consistently excel in sales.
2. **Recognition and Awards:** Recognize outstanding performance outside of pure profits could be a viable way to allow more staff to be recognised for efforts outside of dollars & cents. As noted prior, based on different KPIs, a different staff member was highlighted. This could be an opportunity to reward staff for, consistency or monthly performances and keep staff morale high.
3. **Monthly/Quarterly Accolades** Establish an "Employee of the Month" or "Employee of the Quarter" program. The selected employee should receive special recognition and rewards. This allows staff to feel their efforts validated on a regular basis.

### 3.2 Benefits

The benefits of, implementing systems to award staff can be observed in a case study of Laser clinics Australia.

#### **A Case Study of Laser Clinics Australia**

Laser Clinics Australia (LCA) is a well-known chain of cosmetic treatment clinics operating in Australia and New Zealand, with over 100 clinics across both countries. LCA's objective was to enhance their Key Performance Indicator (KPI) achievements across as many clinics as possible and incentivize their therapists to meet sales targets and complete training, with the prospect of becoming clinic owners. To achieve this, they implemented the "Diamond Rewards Program" in collaboration with

Power2Motivate. This case study highlights the successful outcomes of the program.  
(Power2Motivate, n.d.)

### **Introduction:**

Laser Clinics Australia (LCA) recognized the importance of achieving their KPIs and motivating therapists to excel in their roles. They aimed to foster an environment of performance and growth, allowing therapists to evolve into clinic owners while maintaining consistent KPI accomplishments. Their partnership with Power2Motivate allowed them to tailor a rewards program that addressed these objectives.

### **Custom Incentive Structure:**

Power2Motivate, in collaboration with LCA, thoroughly understood the business's needs. Multiple consultations were held to align the program with LCA's business objectives, goals, KPIs, and budget. After collecting the necessary information, a customised incentive structure was developed, tailored specifically to LCA's needs and objectives. The collaboration was a vital aspect of ensuring the program's success.

### **Achieving Diverse KPIs:**

The program aimed to boost KPI achievements in multiple clinics while motivating therapists. It had to cater to a broad spectrum of employees and encourage therapists to pursue clinic ownership through training. To address this, employees received informative materials such as FAQs, "How to Play" resources, and tutorial videos to familiarise themselves with the Power2Motivate (P2M) program. Also, the program consisted of a weekly reward structure recognizing the top 50 highest-performing therapists across the country. Therapists earned points based on their monthly focus, with each focus being associated with a pricing strategy determined by LCA. In addition to individual KPIs, each clinic has "Clinic Monthly Focus" goals that promote teamwork and collaboration.

### **Recognition and Motivation:**

The program recognized exceptional managers who led their teams effectively and contributed to the clinic's performance. Managers received points when their clinic won the monthly challenge, ranked among the top 5 clinics in the country, or experienced the most growth between quarters. The program's trophy case module further gamified the experience, motivating employees to meet sales targets and training goals. It recognized therapists for longer-term achievements, fostering diverse opportunities for celebrating employees' successes and milestones.

### **Results and Impact:**

LCA's implementation of the program led to remarkable outcomes. Within just two months of launching the program, the number of clinics reaching their KPIs increased threefold. The company witnessed an impressive 66% growth in revenue since the program's initiation. A high login rate of 75% was achieved, and over 7.5 million points were awarded across the organisation. Employees also collected nearly 1,600 "Sales Leader" trophies, celebrating their contributions to LCA's growth and success.

### **Program Management:**

Collaboration was key to the program's success. Regular progress meetings ensured the program ran smoothly, while monthly strategy meetings allowed for planning and identifying areas for improvement. The strong and ongoing partnership between Laser Clinics Australia and Power2Motivate underpinned the program's achievements.

Ultimately, the case study of Laser Clinics Australia's "Diamond Rewards Program" demonstrates the effectiveness of custom-tailored incentive programs in motivating employees, achieving KPIs,

and promoting growth. The results highlight the significant impact such programs can have on an organisation's performance and bottom line.

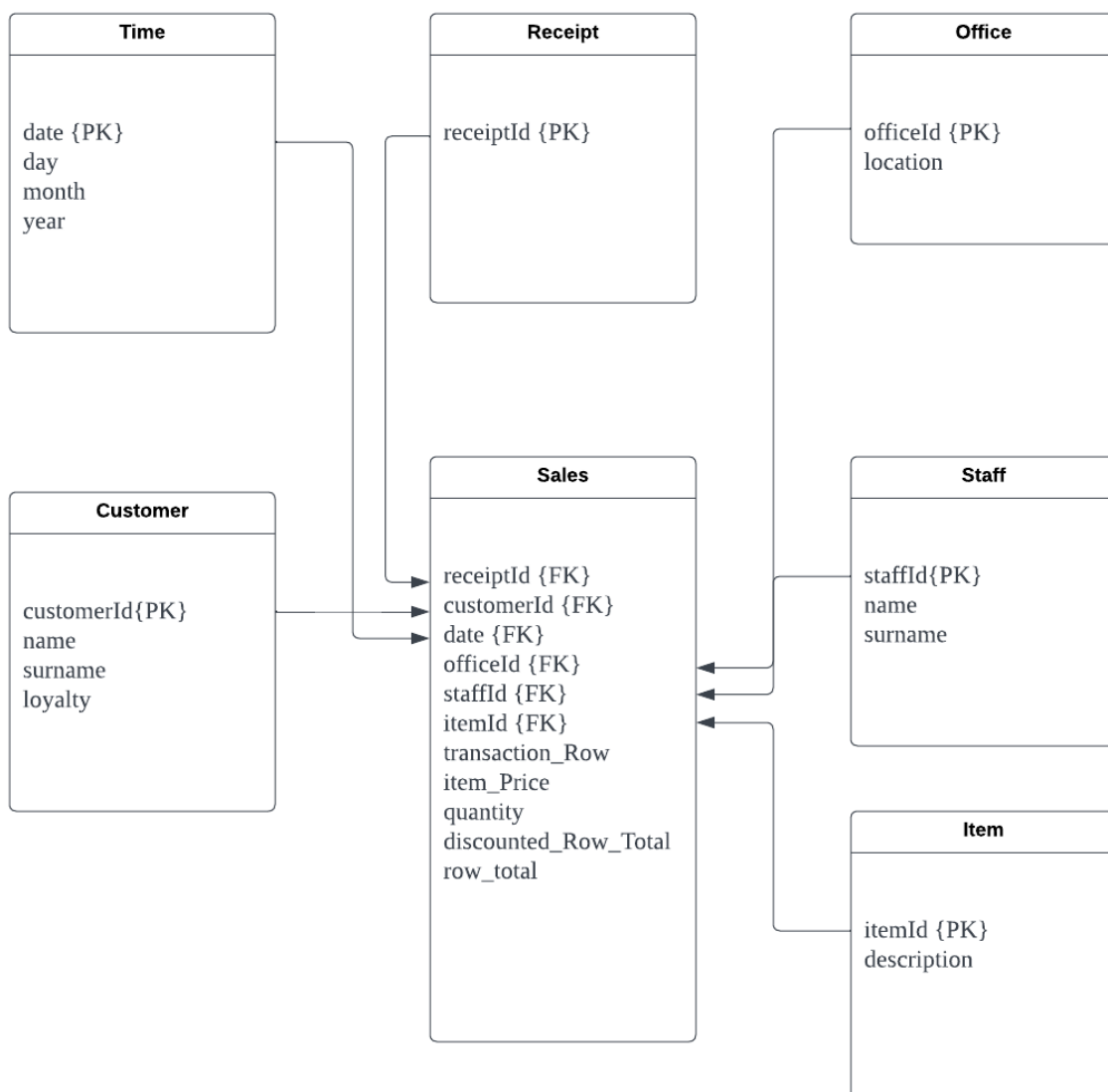
### 3.3 Closing statement:

We are confident that our design and findings would provide substantial evidence to reignite your staff passion and improve “Bits&Bots” overall sales performance. Please feel free to ask any questions or seek further assistance from our team.

## 4.0 Appendices

### 4.1 Datamart Model

#### 4.1.1 Datamart Diagram



#### 4.1.2 DataMart Definitions

##### Fact table (Sales)

We have developed this fact table (sales) to cover the areas of the sale transactions to derive meaningful insights for helping “Bits&Bobs” identify their best-performing staff. The fact table consists of multiple foreign keys to establish a relationship between the dimensions table to provide relevant additional information.

The fact table foreign keys consist of the following:

- **Date:** The time dimension table “date” primary key is referenced by the “date” column in the fact table as a foreign key to associate the sales transaction to the specific date. This would enable analysts to derive insights into customer behaviour patterns, staff sales patterns, sale patterns based on location, and popular items sold based on dates.
- **receiptId:** The receipt dimension table “receiptId” primary key is referenced by the “receiptId” column in the fact table as a foreign key to associate the sales transaction to the specific receipt. This would ensure that guarantee that every transaction can be traced back to the relevant receipt, making it more convenient for data analysts.
- **customerId:** The customer dimension table “customerId” primary key is referenced by the “customerId” column in the fact table as a foreign key to associate the sales transaction to the specific customer. This would enable analysts to analyse their purchase patterns based on dates, what are their frequently purchased items, and the average total amount spent per receipt.
- **officeId:** The office dimension table “officeId” primary key is referenced by the “officeId” column in the fact table as a foreign key to associate the sales transaction to the office. This would enable analysts to derive insights into which office has the highest number of items sold and what are the popular items in their area based on the dates.
- **staffId:** The staff dimension table “staffId” primary key is referenced by the “staffId” column in the fact table as a foreign key to associate the sales transaction to the specific staff. This would enable analysts to analyse the staff performance based on sales and analyse their sales performance behaviour. The information can be further used to offer feasible solutions to the company to improve its staff performance.
- **itemId:** The item dimension table “itemId” primary key is referenced by the “itemId” column in the fact table as a foreign key to associate the sales transaction to the specific item. This would enable analysts to derive meaningful insights further to determine the most popular and least popular products based on the dates and locations.

The fact table metrics consist of the following:

- **Transaction row:** Transaction row metrics as a field would allow the analysts to see the transaction row in the receipt belonging to which customer and staff.
- **Item price:** Putting the price of the item would allow the analyst to see the staff with the highest sales and potentially derive insights on product pricing based on the dates and whether it affects the sales. Furthermore, it stores the historical prices of the item, which can be used to derive insights into strategic plans for improving sales revenue.
- **Quantity:** The quantity of products metric would allow the analyst to see which staff have the most items sold under their staff ID based on the dates and potentially derive insights on the staff. working behaviour and customer purchase behaviour.
- **Discounted row total:** A discount metric would allow the analyst to see the customer behaviour on whether their current marketing tactic is working and which can be used to improve their current strategy. The discounted row total metric would automatically



discount the row total by 12.5 percent if the customer is a loyalty member, and if not, it would just put the row total amount. This way, the analyst can see the total amount of discounted items sold.

- Row total of each transaction row: A row total metric would allow the analyst to see the total amount of items sold in a transaction row to enable them to derive insights on customer behaviour patterns and staff sales patterns based on the dates.

### **Time dimension table**

The time dimension contains a collection of information about the date. The information in the data fields is important as it can provide more comprehensive information to be used for trend analysis.

Data fields:

- Date {Primary key}: Unique identifier for each date
- Day: Day of the date
- Month: Month of the date
- Year: Year of the date

### **Receipt dimension table**

receiptId: The receipt dimension table serves as a reference to associate the sales transaction with the receipt.

Data fields:

- receiptId {Primary key}: Unique identifier for each receipt

### **Customer dimension table**

The customer dimension table contains a collection of information about customers in the company, which enables the examination of sales transactions relating to the specific customer. It would enable the data analysts to extract meaningful insights into customer behaviour and item preferences to help with marketing tactics.

Data fields:

- customerId {Primary key}: Unique identifier for each individual customer
- name: Name of the customer
- surname: Surname of the customer
- loyalty: Whether the customer is a loyalty member or not.

### **Office dimension table**

The office dimension table contains a collection of information about the various locations within the company. It would enable data analysts to extract meaningful insights into which office has the highest sales and can investigate why certain office sales are lacking. From there, the company can perform an investigation into whether it is an internal or external problem.

Data fields:

- officeId {Primary key}: Unique identifier for each individual office
- location: Location of the office

### **Staff dimension table**

The staff dimension table contains a collection of information about the staffs in the company. It acts as a reference link to the sales transaction and the specific staff. This would enable data analysts to analyse each individual performance, item popularity, and customer behaviour based on the dates.

Data fields:

- staffId {Primary key}: Unique identifier for each individual staff
- name: Name of the staff
- surname: Surname of the staff

### **Item dimension table**

itemId: The item dimension table contains information about a variety of items offered by the company. It acts as a reference link to the sales transaction and specific item. This would enable data analysts to analyse the popularity of each product.

Data fields:

- itemId {Primary key}: Unique identifier for each item
- Description: Item name

Before designing the data mart schema, we investigated the “Bits&Bobs” business requirements and rules and have developed several assumptions, limitations, and questions appropriate for this design.

Assumptions before designing the data mart:

- Clean data
- Data consistency
- Types of user requirements

Limitations of designing the data mart:

- Redundancy of data: since our data mart is designed in a star schema, which is more denormalised than the snowflake schema, it might lead to inefficient data storage. To address this problem, we will implement only foreign keys and metrics in the fact table. This would avoid data repetition stored in the fact table.
- Data complexity: Star schema does not naturally fit hierarchical data. It might be a problem to display these structures effectively. To address this problem, we have structured them

into relevant dimension tables. This would maintain star schema simplicity while being able to handle hierarchical data.

#### 4.1.3 Design Rationale

We have decided to design the data mart model as a star schema to emphasise performance and simplicity to cater to “Bits&Bobs,” as from the assignment brief, they have stated that they are not tech-savvy people. Star schema has fewer joins and is more denormalised than snowflake schema, thus providing a more efficient query performance while maintaining simplicity, which makes this design the most appropriate for “Bits&Bobs” business intelligence future use.

#### 4.1.4 DataMart Implementation

With the cleaned database, the creation and population of the DataMart based on the data model can proceed.

##### 1. Creation & population of the time dimension

```
CREATE TABLE Dim_Time (  
    Sales_Date DATE PRIMARY KEY,  
    Day INT,  
    Month INT,  
    Year INT  
);  
  
-- Insert distinct date values into the Dim_Time table  
INSERT INTO Dim_Time (Sales_Date)  
SELECT DISTINCT Sale_Date  
FROM Clean_Table;  
  
-- Populate the Day, Month, and Year columns  
UPDATE Dim_Time  
SET Day = DAY(Sales_Date),  
    Month = MONTH(Sales_Date),  
    Year = YEAR(Sales_Date);
```

## 2. Creation & population of the customer dimension

```
-- Create the Dim_Customer dimension table
CREATE TABLE Dim_Customer (
    customerId varchar(50) PRIMARY KEY,
    name VARCHAR(255),
    surname VARCHAR(255),
    loyalty_ VARCHAR(50)
);

-- Insert distinct customer IDs into the Dim_Customer table
INSERT INTO Dim_Customer (customerId)
SELECT DISTINCT [Customer_ID]
FROM [Clean_Table];

-- Update the Dim_Customer table with customer details
UPDATE Dim_Customer
SET
    name = [Customer_First_Name],
    surname = [Customer_Surname],
    loyalty_ = [Loyalty]
FROM Dim_Customer
JOIN [Clean_Table] ON Dim_Customer.customerId = [Clean_Table].[Customer_ID];
```

## 3. Creation & population of the Receipt dimension

```
-- Create the Dim_Receipt dimension table
CREATE TABLE Dim_Receipt (
    receiptId varchar(50) PRIMARY KEY
);

-- Insert distinct receipt IDs into the Dim_Receipt table
INSERT INTO Dim_Receipt (receiptId)
SELECT DISTINCT [Reciept_Id]
FROM [Clean_Table];
```

## 4. Creation & population of the office dimension

```
-- Create the Dim_Office dimension table
CREATE TABLE Dim_Office (
    officeId INT PRIMARY KEY,
    location VARCHAR(255)
);

-- Insert distinct office data into the Dim_Office table
INSERT INTO Dim_Office (officeId, location)
SELECT DISTINCT [Staff_office], [Office_Location]
FROM [Clean_Table];
```

#### 5. Creation & population of the Staff dimension

```
-- Create the Dim_Staff dimension table
CREATE TABLE Dim_Staff (
    staffId VARCHAR(50) PRIMARY KEY,
    name VARCHAR(255),
    surname VARCHAR(255)
);

INSERT INTO Dim_Staff(staffId, name, surname)
SELECT DISTINCT [Staff_ID], [Staff_First Name], [Staff_Surname]
FROM [Clean_Table];
```

#### 6. Creation & population of the item dimension

```
-- Create the Dim_Item dimension table
CREATE TABLE Dim_Item (
    itemId INT PRIMARY KEY,
    description VARCHAR(255)
);

INSERT INTO Dim_Item(itemId, description)
SELECT DISTINCT [Item_ID], [Item_Description]
FROM [Clean_Table];
```

#### 7. Creation of the fact table

```
CREATE TABLE Fact_Sales (
    receiptId VARCHAR(50) FOREIGN KEY REFERENCES Dim_Receipt(receiptId),
    customerId VARCHAR(50) FOREIGN KEY REFERENCES Dim_Customer(customerId),
    date date FOREIGN KEY REFERENCES Dim_Time(Sales_Date),
    officeId INT FOREIGN KEY REFERENCES Dim_Office(officeId),
    staffId varchar(50) FOREIGN KEY REFERENCES Dim_Staff(staffId),
    itemId INT FOREIGN KEY REFERENCES Dim_Item(itemId),
    transaction_Row INT,
    item_Price DECIMAL(18, 2),
    quantity INT,
    discounted_Row_Total DECIMAL(18, 2),
    row_total DECIMAL(18, 2)
);
```

## 8. Insertion into the Fact\_sales Table and updating the discounted row column with the correct values

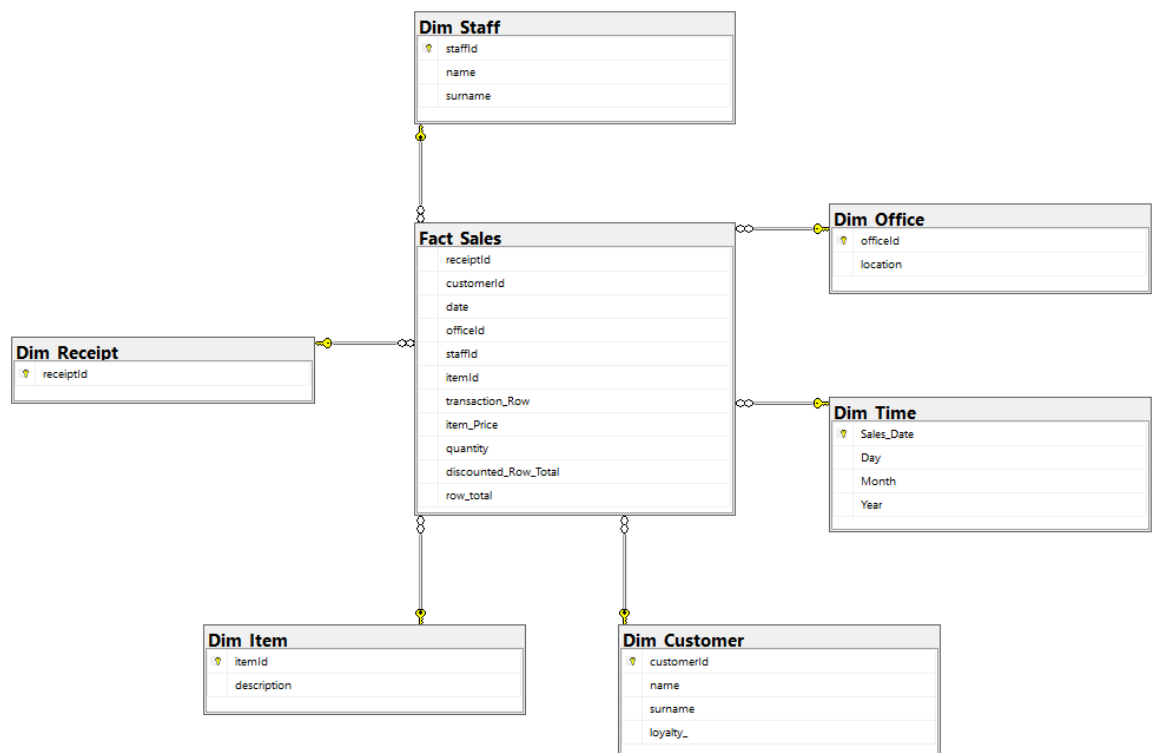
```

INSERT INTO Fact_Sales (receiptId, customerId, date, officeId, staffId, itemId, transaction_Row, item_Price, quantity, row_total)
SELECT
    Dim_Reciept.receiptId,
    Dim_Customer.customerId,
    Dim_Time.Sales Date,
    Dim_Office.officeId,
    Dim_Staff.staffId,
    Dim_Item.itemId,
    [Reciept Transaction Row ID],
    [Item Price],
    [Item Quantity],
    [Row Total]
FROM [Clean Table]
JOIN Dim_Reciept ON Dim_Reciept.receiptId = [Clean Table].[Reciept Id]
JOIN Dim_Customer ON Dim_Customer.customerId = [Clean Table].[Customer ID]
JOIN Dim_Time ON Dim_Time.Sales Date = [Clean Table].[Sale Date] -- Replace with the appropriate date key
JOIN Dim_Office ON Dim_Office.officeId = [Clean Table].[Staff office]
JOIN Dim_Staff ON Dim_Staff.staffId = [Clean Table].[Staff ID]
JOIN Dim_Item ON Dim_Item.itemId = [Clean Table].[Item ID];

UPDATE [A1 Dirty].[dbo].[Fact Sales]
SET discounted Row Total = CASE
    WHEN [receiptId] IN (
        SELECT [receiptId]
        FROM [A1 Dirty].[dbo].[Fact Sales]
        GROUP BY [receiptId]
        HAVING COUNT(DISTINCT [itemId]) > 4
    ) AND c.loyalty = 'Yes' -- Assuming IsLoyal is a column with values 'Yes' or 'No'
    THEN fs.Item Price * fs.Quantity * (1 - 0.125) -- 12.5% discount for loyal customers with more than 4 distinct items
    ELSE fs.Item Price * fs.Quantity -- No discount for others
END
FROM [A1 Dirty].[dbo].[Fact Sales] AS fs
JOIN [A1 Dirty].[dbo].[Dim Customer] AS c
ON fs.CustomerId = c.CustomerId;

```

## 9. Database diagram created by SMSS: Star schema



## 4.2 Anomaly Detection

1. Checked the number of NULL or empty strings on every column and summed the number of rows left blank or null.

```
SELECT
SUM(CASE WHEN [Sale_Date] IS NULL THEN 1 ELSE 0 END) AS Sale_Date_Null_Count,
SUM(CASE WHEN [Loyalty] IS NULL OR [Loyalty] = '' THEN 1 ELSE 0 END) AS Loyalty_Null_or_Empty_Count,
SUM(CASE WHEN [Receipt_Id] IS NULL THEN 1 ELSE 0 END) AS Receipt_Id_Null_Count,
SUM(CASE WHEN [Customer_ID] IS NULL OR [Customer_ID] = '' THEN 1 ELSE 0 END) AS Customer_ID_Null_or_Empty_Count,
SUM(CASE WHEN [Customer_First_Name] IS NULL OR [Customer_First_Name] = '' THEN 1 ELSE 0 END) AS Customer_First_Name_Null_or_Empty_Count,
SUM(CASE WHEN [Customer_Surname] IS NULL OR [Customer_Surname] = '' THEN 1 ELSE 0 END) AS Customer_Surname_Null_or_Empty_Count,
SUM(CASE WHEN [Staff_ID] IS NULL OR [Staff_ID] = '' THEN 1 ELSE 0 END) AS Staff_ID_Null_or_Empty_Count,
SUM(CASE WHEN [Staff_First_Name] IS NULL OR [Staff_First_Name] = '' THEN 1 ELSE 0 END) AS Staff_First_Name_Null_or_Empty_Count,
SUM(CASE WHEN [Staff_Surname] IS NULL OR [Staff_Surname] = '' THEN 1 ELSE 0 END) AS Staff_Surname_Null_or_Empty_Count,
SUM(CASE WHEN [Staff_office] IS NULL OR [Staff_office] = '' THEN 1 ELSE 0 END) AS Staff_Office_Null_or_Empty_Count,
SUM(CASE WHEN [Office_Location] IS NULL OR [Office_Location] = '' THEN 1 ELSE 0 END) AS Office_Location_Null_or_Empty_Count,
SUM(CASE WHEN [Receipt_Transaction_Row_ID] IS NULL OR [Receipt_Transaction_Row_ID] = '' THEN 1 ELSE 0 END) AS Receipt_Transaction_Row_ID_Null_or_Empty_Count,
SUM(CASE WHEN [Item_ID] IS NULL OR [Item_ID] = '' THEN 1 ELSE 0 END) AS Item_ID_Null_or_Empty_Count,
SUM(CASE WHEN [Item_Description] IS NULL OR [Item_Description] = '' THEN 1 ELSE 0 END) AS Item_Description_Null_or_Empty_Count,
SUM(CASE WHEN [Item_Quantity] IS NULL THEN 1 ELSE 0 END) AS Item_Quantity_Null_Count,
SUM(CASE WHEN [Item_Price] IS NULL THEN 1 ELSE 0 END) AS Item_Price_Null_Count,
SUM(CASE WHEN [Row_Total] IS NULL THEN 1 ELSE 0 END) AS Row_Total_Null_Count
FROM [A1_Dirty].[dbo].[A1_Dirty];
```

**Anomaly Detected:** 1 row for each column except transaction\_row\_id had a null value

Follow up:

Pull all rows where any column value is Null

```
SELECT *
FROM [A1_Dirty].[dbo].[A1_Dirty]
WHERE
[Sale_Date] IS NULL OR
[Loyalty] IS NULL OR
[Receipt_Id] IS NULL OR
[Customer_ID] IS NULL OR
[Customer_First_Name] IS NULL OR
[Customer_Surname] IS NULL OR
[Staff_ID] IS NULL OR
[Staff_First_Name] IS NULL OR
[Staff_Surname] IS NULL OR
[Staff_office] IS NULL OR
[Office_Location] IS NULL OR
[Item_ID] IS NULL OR
[Item_Description] IS NULL OR
[Item_Quantity] IS NULL OR
[Item_Price] IS NULL OR
[Row_Total] IS NULL;
```

**Anomaly confirmed:** 1 row had all columns NULL except transaction\_row\_id

2. Checking Whether offices are corresponding to the location

```
SELECT DISTINCT [Staff_Office]
FROM [A1_Dirty].[dbo].[A1_Dirty]
WHERE [Staff_Office] NOT IN (SELECT DISTINCT [Office_Location] FROM [A1_Dirty].[dbo].[A1_Dirty]);
```

3. Checking to see if there are any date/item id combos which yield more than 1 price to ensure consistent price for item on any given day

```
SELECT DISTINCT a.*
FROM [A1_Dirty].[dbo].[A1_Dirty] a
INNER JOIN [A1_Dirty].[dbo].[A1_Dirty] b
ON a.Sale_Date = b.Sale_Date
AND a.Item_Id = b.Item_Id
AND a.Item_Price <> b.Item_Price;
```

#### 4. Identifying receipts with more than 4 unique items (eligible for discount)

```
SELECT a.*
FROM [A1_Dirty].[dbo].[A1_Dirty] a
INNER JOIN (
    SELECT [Receipt_Id]
    FROM [A1_Dirty].[dbo].[A1_Dirty]
    GROUP BY [Receipt_Id]
    HAVING COUNT(DISTINCT [Item_Id]) > 4
) b ON a.[Receipt_Id] = b.[Receipt_Id]
WHERE a.[Loyalty] = 'yes';
```

#### 5. Ensuring the row\_totals correctly reflect the quantity \* price

```
SELECT *
FROM [A1_Dirty].[dbo].[A1_Dirty]
WHERE [Row_Total] != ([Item_Quantity] * [Item_Price]);
```

#### 6. Checking if there are any receipts that are associated with multiple customers

```
SELECT [Receipt_ID]
FROM [A1_Dirty].[dbo].[A1_Dirty]
GROUP BY [Receipt_ID]
HAVING COUNT(DISTINCT [Customer_ID]) > 1;
```

**Anomaly Confirmed:** Identified 4 erroneous receipt ids 21009, 21719, 22761, 22912

#### 7. Check that row\_transaction\_id is not repeated within a receipt

```
SELECT [Receipt_ID]
FROM [A1_Dirty].[dbo].[A1_Dirty]
GROUP BY [Receipt_ID]
HAVING COUNT(*) <> COUNT(DISTINCT [Receipt_Transaction_Row_ID]);
```

**Anomaly Confirmed:** Identified 4 erroneous receipt ids 21009, 21719, 22761, 22912

#### 8. Check if any receipts have the different dates across that receipt id

```
SELECT [Receipt_ID]
FROM [A1_Dirty].[dbo].[A1_Dirty]
GROUP BY [Receipt_ID]
HAVING COUNT(DISTINCT [Sale_Date]) > 1;
```

**Anomaly Confirmed:** Identified 2 erroneous members in particular 21009 and 21719

#### 9. Check if every receipt starts with a 1 in transaction row

```
SELECT [Receipt_ID]
FROM [A1_Dirty].[dbo].[A1_Dirty]
GROUP BY [Receipt_ID]
HAVING SUM(CASE WHEN LEFT([Receipt_Transaction_Row_ID], 1) = '1' THEN 1 ELSE 0 END) = 0;
```

#### 10. Check if customers have their own unique ID

```
SELECT [Customer_ID]
FROM [A1_Dirty].[dbo].[A1_Dirty]
GROUP BY [Customer_ID]
HAVING COUNT(DISTINCT [Customer_First_Name] + ' ' + [Customer_Surname]) > 1;
```



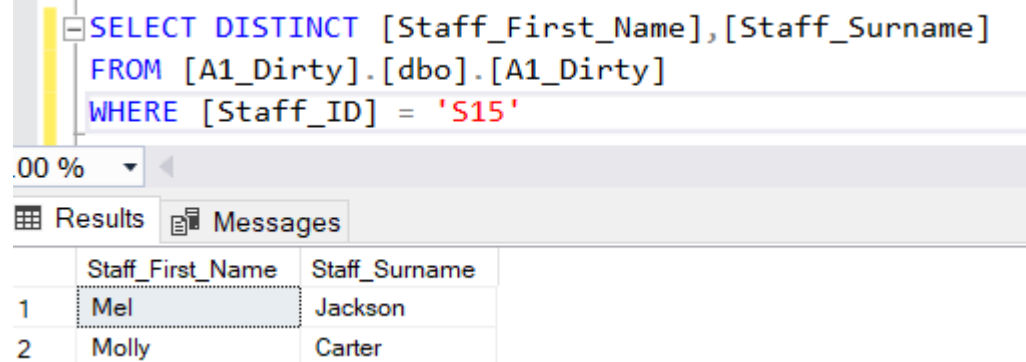
## 11. Check if Staff have their own unique ID

```
SELECT [Staff_ID]
FROM [A1_Dirty].[dbo].[A1_Dirty]
GROUP BY [Staff_ID]
HAVING COUNT(DISTINCT [Staff_First_Name]) > 1 OR COUNT(DISTINCT [Staff_Surname]) > 1;
```

**Anomaly Detected:** staff\_ID S15 is used twice

Follow up:

Identify staff with that ID



The screenshot shows a SQL query in the Enterprise Manager interface. The query is: `SELECT DISTINCT [Staff_First_Name],[Staff_Surname] FROM [A1_Dirty].[dbo].[A1_Dirty] WHERE [Staff_ID] = 'S15'`. Below the query editor, the 'Results' tab is active, displaying a table with two columns: 'Staff\_First\_Name' and 'Staff\_Surname'. There are two rows of data: Row 1 with 'Mel' and 'Jackson', and Row 2 with 'Molly' and 'Carter'.

	Staff_First_Name	Staff_Surname
1	Mel	Jackson
2	Molly	Carter

**Anomaly Confirmed:** S15 is used for MEL JACKSON and MOLLY CARTER

## 12. Checked if all item descriptions are unique and consistent to the item\_Id

```
SELECT [Item_ID] -- clean
FROM [A1_Dirty].[dbo].[A1_Dirty]
GROUP BY [Item_ID]
HAVING COUNT(DISTINCT [Item_Description]) > 1;
```

## Solutions for cleaning the Data

The data anomalies were not too pervasive, there were different options to handle them to create a clean set of data to be fed into the Data mart.

### Duplicate Identifiers (Unique Keys):

1. Omitting rows that contain non-unique identifier keys
2. Supplementing those Keys with a version alphabet/ or duplicate affix

Supplementing an affix was chosen for greater coverage in analysis and reduced loss of integral data. By omitting erroneous rows that may potentially represent a large percentage of total rows, results of an analysis may be greatly skewed.

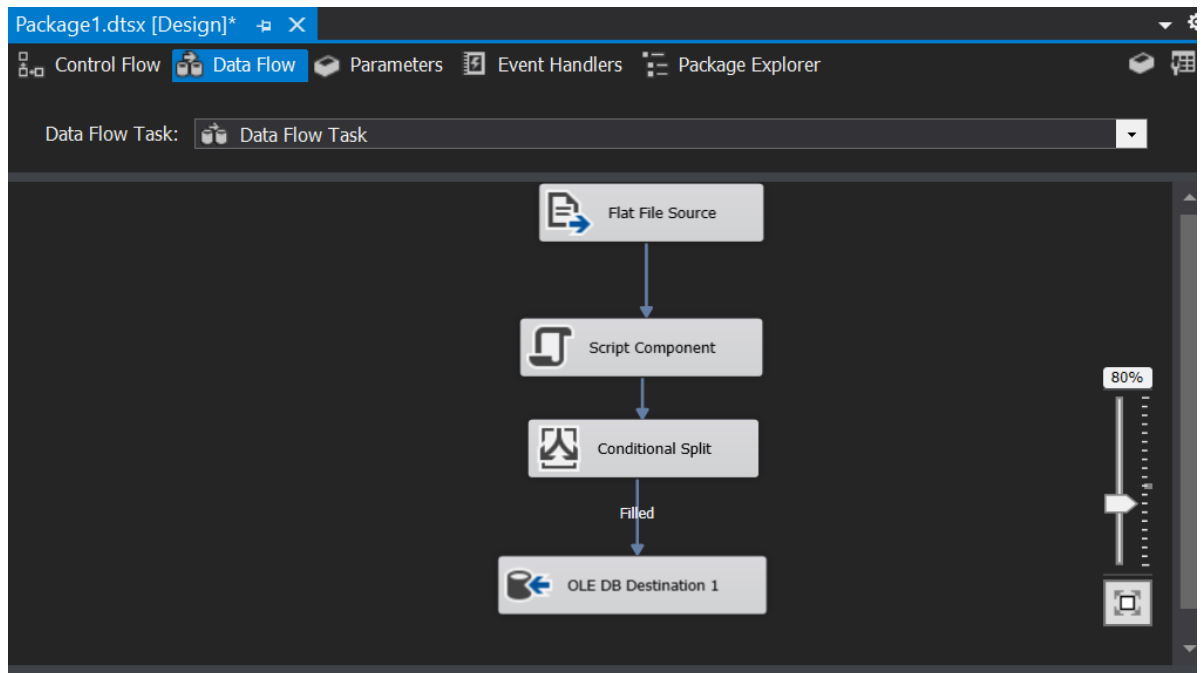
### NULL Values:

Since the detection process determined that there was only a singular instance of Null values within the database table, the choice to omit that row was made as it had virtually no valuable data captured and was likely a product of a glitch or accidental entry. In this unique case, this removal has no bearings on the analysis.

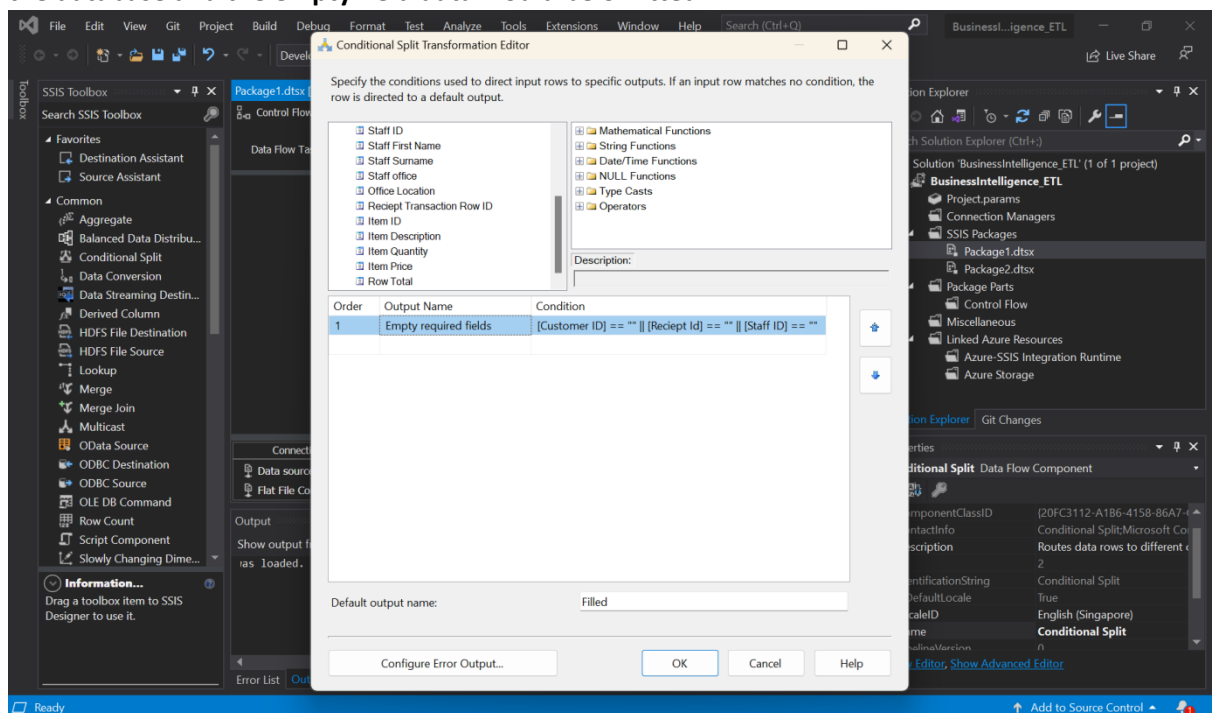
## 4.3 ETL Processes

The ETL process was handled with SSIS. The cleaning mostly consisted of supplementing duplicate members with a d to the ID to retain essential data meant for analysis. An example of this was the duplication of the Staff\_ID, Receipt\_ID and the row of NULL values. The key processes are highlighted below:

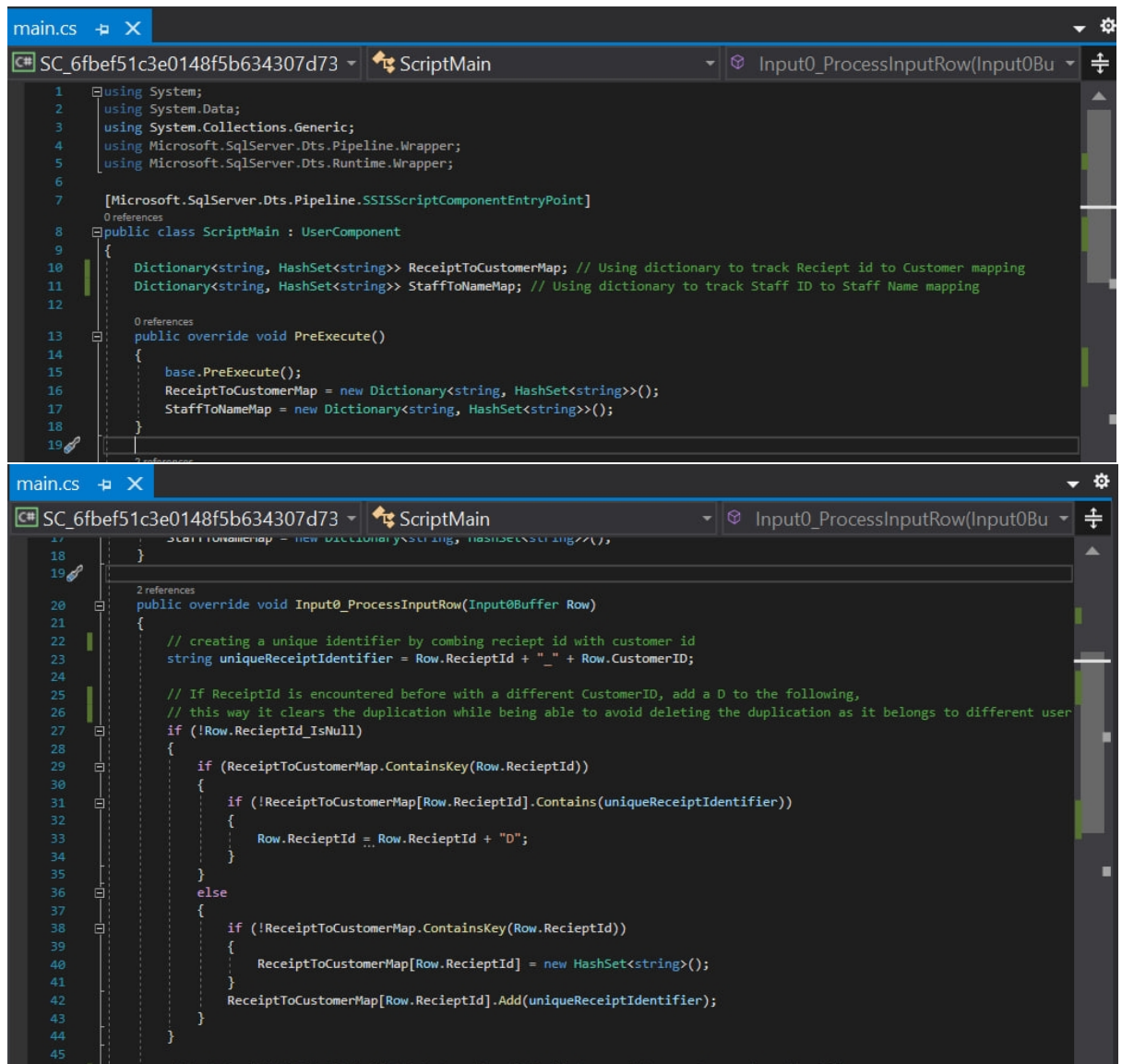
### 1. Data flow of the SSIS



### 2. Conditional split transformation editor, this way, it will split the data into two parts which is data with empty fields and data that is filled. The data that is filled will be directed to the database and the empty field data would be omitted.



3. The SSIS script responsible for supplementing a alphabet "D" for duplicated receipt id and staff id that shares different users.



```
main.cs  SC_6fbef51c3e0148f5b634307d73  ScriptMain  Input0_ProcessInputRow(Input0Bu
1  using System;
2  using System.Data;
3  using System.Collections.Generic;
4  using Microsoft.SqlServer.Dts.Pipeline.Wrapper;
5  using Microsoft.SqlServer.Dts.Runtime.Wrapper;
6
7  [Microsoft.SqlServer.Dts.Pipeline.SSISScriptComponentEntryPoint]
8  public class ScriptMain : UserComponent
9  {
10     Dictionary<string, HashSet<string>> ReceiptToCustomerMap; // Using dictionary to track Receipt id to Customer mapping
11     Dictionary<string, HashSet<string>> StaffToNameMap; // Using dictionary to track Staff ID to Staff Name mapping
12
13     public override void PreExecute()
14     {
15         base.PreExecute();
16         ReceiptToCustomerMap = new Dictionary<string, HashSet<string>>();
17         StaffToNameMap = new Dictionary<string, HashSet<string>>();
18     }
19
20     public override void Input0_ProcessInputRow(Input0Buffer Row)
21     {
22         // creating a unique identifier by combining receipt id with customer id
23         string uniqueReceiptIdentifier = Row.ReceiptId + "_" + Row.CustomerID;
24
25         // If ReceiptId is encountered before with a different CustomerID, add a D to the following,
26         // this way it clears the duplication while being able to avoid deleting the duplication as it belongs to different user
27         if (!Row.ReceiptId_IsNull)
28         {
29             if (ReceiptToCustomerMap.ContainsKey(Row.ReceiptId))
30             {
31                 if (!ReceiptToCustomerMap[Row.ReceiptId].Contains(uniqueReceiptIdentifier))
32                 {
33                     Row.ReceiptId = Row.ReceiptId + "D";
34                 }
35             }
36             else
37             {
38                 if (!ReceiptToCustomerMap.ContainsKey(Row.ReceiptId))
39                 {
40                     ReceiptToCustomerMap[Row.ReceiptId] = new HashSet<string>();
41                 }
42                 ReceiptToCustomerMap[Row.ReceiptId].Add(uniqueReceiptIdentifier);
43             }
44         }
45     }
46 }
```

```

main.cs
SC_6fbef51c3e0148f5b634307d73 ScriptMain Input0_ProcessInputRow(Input0Bu
44 }
45
46 // Combine "Staff_ID," "Staff_First_Name," and "Staff_Surname" to create a unique identifier
47 // creating a unique identifier by combining staff id with staff first name and surname
48 string uniqueStaffIdentifier = Row.StaffID + "_" + Row.StaffFirstName + "_" + Row.StaffSurname;
49
50 // If StaffID is encountered before with a different staff name, add a with "D" to the following,
51 // this way it clears the duplication while being able to avoid deleting the duplication as it belongs to different user
52 if (!Row.StaffID_IsNull)
53 {
54     if (StaffToNameMap.ContainsKey(Row.StaffID))
55     {
56         if (!StaffToNameMap[Row.StaffID].Contains(uniqueStaffIdentifier))
57         {
58             Row.StaffID = Row.StaffID + "D";
59         }
60     }
61     else
62     {
63         if (!StaffToNameMap.ContainsKey(Row.StaffID))
64         {
65             StaffToNameMap[Row.StaffID] = new HashSet<string>();
66         }
67         StaffToNameMap[Row.StaffID].Add(uniqueStaffIdentifier);
68     }
69 }
70 }
71
72 }

```

65 % No issues found Ln: 19 Ch: 5 SPC LF

## 4.4 Base Analysis

With the DataMart in place, we can create a few queries to check the top salesperson based on different criteria as to what should be taken into consideration.

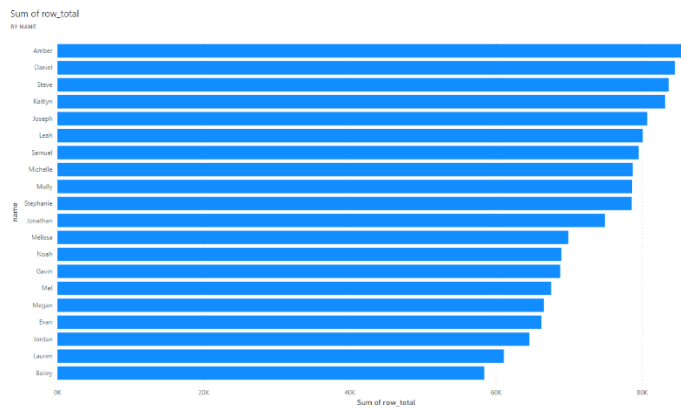
### 1. Top salesperson by sales amount in dollars: Amber Hill

```
--KPI: Top salesperson by sales amount in dollars
```

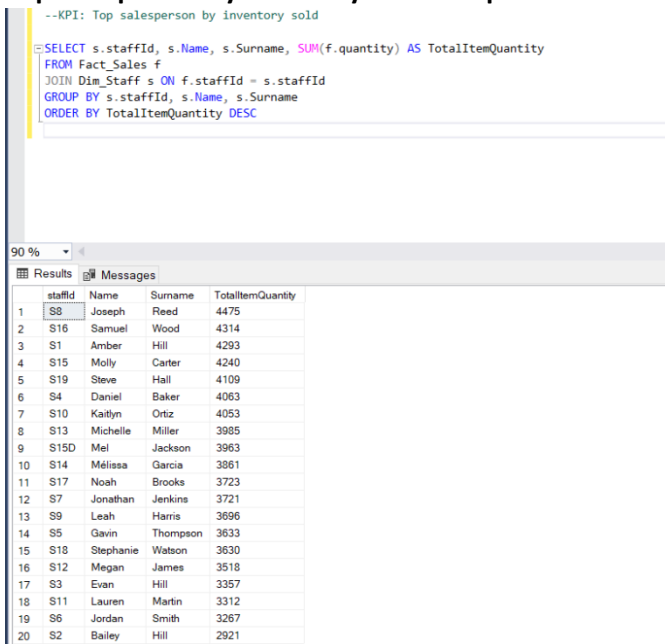
```
SELECT s.staffId, s.Name, s.Surname, SUM(f.row_total) AS TotalSalesAmount
FROM Fact_Sales f
JOIN Dim_Staff s ON f.staffId = s.staffId
GROUP BY s.staffId, s.Name, s.Surname
ORDER BY TotalSalesAmount DESC
```

	staffId	Name	Surname	TotalSalesAmount
1	S1	Amber	Hill	86324.63
2	S4	Daniel	Baker	84505.94
3	S19	Steve	Hall	83659.40
4	S10	Kaitlyn	Ortiz	83163.23
5	S8	Joseph	Reed	80736.75
6	S9	Leah	Harris	80112.79
7	S16	Samuel	Wood	79557.56
8	S13	Michelle	Miller	78727.88
9	S15	Molly	Carter	78662.15
10	S18	Stephanie	Watson	78601.42
11	S7	Jonathan	Jenkins	74932.67
12	S14	Mélissa	Garcia	69927.34
13	S17	Noah	Brooks	68984.36
14	S5	Gavin	Thompson	68828.33
15	S15D	Mel	Jackson	67575.31
16	S12	Megan	James	66582.81
17	S3	Evan	Hill	66241.92
18	S6	Jordan	Smith	64607.41
19	S11	Lauren	Martin	61117.99

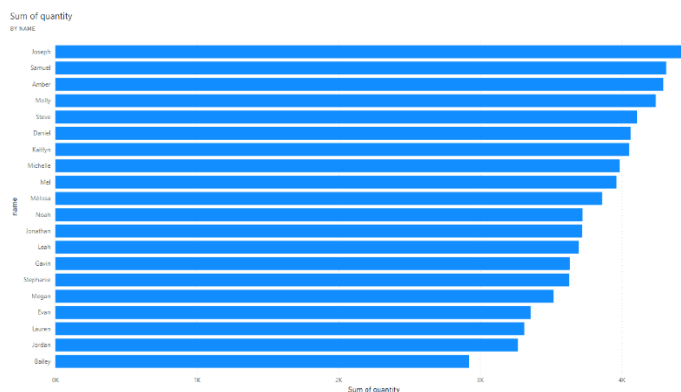
## Visualised in Power BI:



## 2. Top salesperson by inventory sold: Joseph Reed



## Visualised in Power BI:



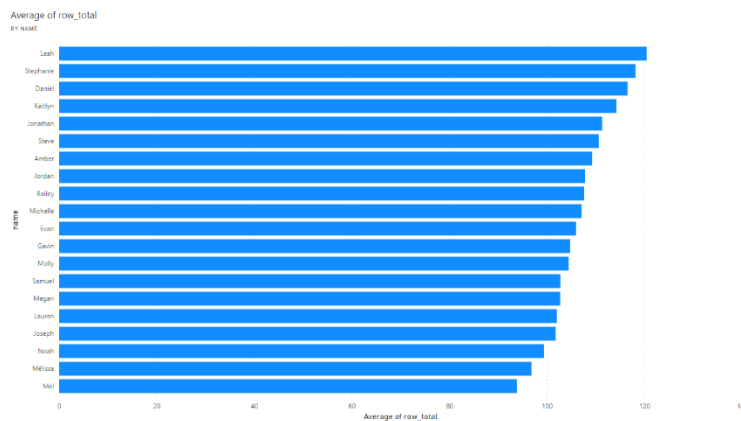
### 3. Top salesperson by Average Transaction: Leah Harris

```
--KPI: Top salesperson by average transaction

SELECT s.staffId, s.Name, s.Surname, AVG(f.row_total) AS AvgTransactionValue
FROM Fact_Sales f
JOIN Dim_Staff s ON f.staffId = s.staffId
GROUP BY s.staffId, s.Name, s.Surname
ORDER BY AvgTransactionValue DESC
```

staffId	Name	Surname	AvgTransactionValue
S9	Leah	Harris	120.470360
S18	Stephanie	Watson	118.197624
S4	Daniel	Baker	116.559917
S10	Kaitlyn	Ortiz	114.235206
S7	Jonathan	Jenkins	111.341263
S19	Steve	Hall	110.660582
S1	Amber	Hill	109.271683
S6	Jordan	Smith	107.858781
S2	Bailey	Hill	107.620202
S13	Michelle	Miller	107.112761
S3	Evan	Hill	105.987072
S5	Gavin	Thompson	104.761537
S15	Molly	Carter	104.465006
S16	Samuel	Wood	102.787545
S12	Megan	James	102.751250
S11	Lauren	Martin	102.033372
S8	Joseph	Reed	101.811790
S17	Noah	Brooks	99.401095
S14	Mélissa	Garcia	96.852271
S15D	Mel	Jackson	93.854597

### Visualised in Power Bi:



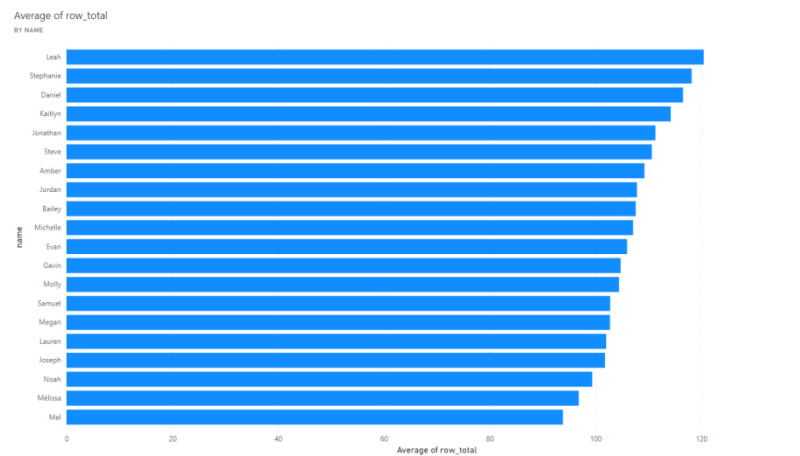
#### 4. Top salesperson by Average Transaction per Month: Leah Harris

```
--KPI: Top salesperson by monthly average sales

SELECT
    s.staffId,
    s.Name,
    s.Surname,
    AVG(f.row_total) AS AvgMonthlySales
FROM Fact_Sales f
JOIN Dim_Staff s ON f.staffId = s.staffId
JOIN Dim_Time t ON f.date = t.Sales_Date
WHERE t.Sales_Date >= DATEADD(MONTH, -12, GETDATE()) -- Consider the past 12 months since it was from july 2022 to june 2023
GROUP BY s.staffId, s.Name, s.Surname
ORDER BY AvgMonthlySales DESC -- Sort in descending order of average monthly sales
```

staffId	Name	Surname	AvgMonthlySales
S9	Leah	Harris	129.570426
S10	Kaitlyn	Ortiz	124.440692
S4	Daniel	Baker	119.847829
S6	Jordan	Smith	116.823611
S18	Stephanie	Watson	116.187541
S2	Bailey	Hill	115.681005
S1	Amber	Hill	113.685325
S3	Evan	Hill	112.034479
S11	Lauren	Martin	110.578965
S15	Molly	Carter	110.262826
S16	Samuel	Wood	107.118601
S19	Steve	Hall	105.827771
S13	Michelle	Miller	104.675252
S7	Jonathan	Jenkins	102.247955
S8	Joseph	Reed	100.379891
S5	Gavin	Thompson	99.277408
S15D	Mel	Jackson	99.068658
S17	Noah	Brooks	98.777415
S14	Mélissa	Garcia	96.520782

#### Visualised in Power Bi:

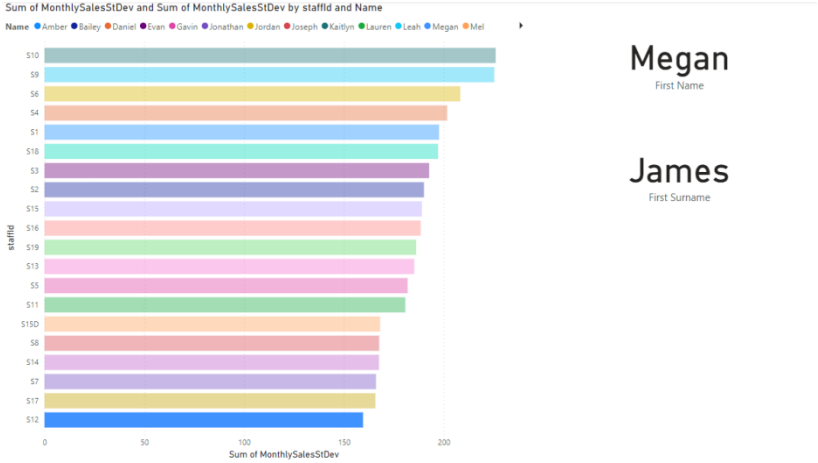


#### 5. Top salesperson by consistency: Megan James

```
SELECT
    s.staffId,
    s.Name,
    s.Surname,
    AVG(f.row_total) AS AvgMonthlySales,
    STDEV(f.row_total) AS MonthlySalesStDev,
    RANK() OVER (ORDER BY STDEV(f.row_total)) AS ConsistencyRank
FROM Fact_Sales f
JOIN Dim_Staff s ON f.staffId = s.staffId
JOIN Dim_Time t ON f.date = t.Sales_Date
WHERE t.Sales_Date >= DATEADD(MONTH, -12, GETDATE())
GROUP BY s.staffId, s.Name, s.Surname
ORDER BY ConsistencyRank
```

staffId	Name	Surname	AvgMonthlySales	MonthlySalesStDev	ConsistencyRank
S12	Megan	James	94.532434	175.900991565996	1
S17	Noah	Brooks	98.777415	179.850381059963	2
S7	Jonathan	Jenkins	102.247955	183.197455171105	3
S8	Joseph	Reed	100.379891	185.78631812523	4
S15D	Mel	Jackson	99.068658	187.962359656863	5
S14	Mélissa	Garcia	96.520782	188.561325707486	6
S11	Lauren	Martin	110.578965	197.400229915232	7
S5	Gavin	Thompson	99.277408	201.284754247181	8
S19	Steve	Hall	105.827771	205.198366525497	9
S13	Michelle	Miller	104.675252	206.563345588645	10
S16	Samuel	Wood	107.118601	210.053820319922	11
S3	Evan	Hill	112.034479	210.256239986902	12
S15	Molly	Carter	110.262826	213.331264713373	13
S2	Bailey	Hill	115.681005	213.76993032434	14
S4	Daniel	Baker	119.847829	220.006165605434	15
S18	Stephanie	Watson	116.187541	220.448282026406	16

Visualised in Power Bi:



Megan

First Name

James

First Surname



## 4.5References

Power2Motivate. (n.d.). *Laser Clinics Australia*. Retrieved from Power2Motivate:  
<https://www.power2motivate.com/PM/files/92/92347698-a734-4b88-83e0-37e217e25382.pdf>