**MSIS 5663 - Data Warehousing**

CaseStudy2023 - Final ETL Project

Phase – 2

Team Pavuluri Pillai Vemula

Team Members:

Nagabhargav Pavuluri

Darshita Pillai

Sripriya Vemula

**High Level Project Plan:**

On analyzing the given project document and the requirements, our team understood that we could provide the executives of Lettuce Feed U the answer they require by performing a major chunk of operations in Visual Studio in an Integration Services Project Template (SSIS). A significant part of the project also seemed to require SSMS, especially for designing the data warehouse and for answering the executive questions. We divided our tasks into several distinct chunks and each team member finished their designated tasks.

**Assumptions:**

Over the course of our entire project, our team made the following assumptions:

* For products such as “Spicy Korean Chicken Sandwich” that had two ProducNbr values with all the same fields barring the ProductDesc, instead of removing it as a duplicate product, we owed the mistake to human error and simply renamed the product on what we thought would be the best fit based on the product description.
* For Products that had a value of 0 in their Product Desc, instead of giving our own description for each of those products, we simply gave a note that says, “Contact Us for Details!” This assumption has been made based on two factors: either the product must be something that is common and might not need any explanation (such as Bottled Water, Diet Coke etc. which don’t need any description) , or that the product is relatively new and might need to be explained by the company. (Even though such products might not be in the current dataset, they might pop up in the future!)
* Even though there are unique File1ID values for all the transactions, instead of treating each of that unique record as a transaction by a customer, we treated each ReceiptNbr value as a single customer.
* While answering the executive questions using SQL queries, while displaying the Top 10 Products in a time duration, we assumed that it means the highest number of products sold i.e., **We chose quantity as the measure of sales to display the Top 10 Products**
* The same assumption has been made for the executive question to display no or few sales i.e.,

**We chose a threshold quantity (taking total quantity as a measure of Total Sales) of 500 after observing the entire data and observing a sudden drop in the total number of each product sold.**

* Keeping in mind the possibility of future growth of the organization, which will obviously cause an increase in the amount of data, we assumed that the same executive questions might be required to view the data for different years. **Therefore, even though we have considered 2022 as the primary year of analysis in our project, instead of hardcoding the WHERE clause in the view, we have designed the view flexibly. However, we gave the flexibility to choose the desired time period (month name, week number, date, and year) in our SELECT statement. This decision has been taken solely to increase the usability of the view over time.**

**Project Implementation Steps and Task Assignments:**

* Getting an overall idea of the project (Sripriya, Darshita and Bhargav)
* Getting data from the CaseStudy2023 database into our database (Sripriya)
* Designing a data warehouse (Sripriya and Bhargav)
* Importing data into SSIS for ETL Process and removing duplicates (Darshita)
* Data cleanup (Bhargav and Sripriya)
* Populating the fact and dimension tables after cleanup in SSIS (Bhargav)
* Executing and testing the package repeatedly (Bhargav)
* Writing the SQL statements (Darshita and Sripriya)
* Making the data diagram (Darshita)
* Documentation (Bhargav and Darshita)

**Getting an overall idea of the project:**

All the team members proactively took out a significant amount of their time to go over the project document, under the requirements with no impending confusions and started to brainstorm ideas to fulfill the project objectives in the most optimal way.

**Getting data from the CaseStudy2023 database into our database:**

The original dataset was provided in a table CaseStudy2023AllRecs\_new present in the CaseStudy2023 database in SSMS. In one of our team member’s database **SP23\_ksbvemula**, we created a table **FinalCaseStudyTable** into which we imported the data from the CaseStudy2023 database, resulting in 96,577 records in our table.

**Designing a Data Warehouse:**

Once we imported the data into our database, we spent some time looking at all the attributes and then started designing our data warehouse. We decided to opt for a data warehouse that follows the **star schema** to keep the design uncomplicated and simple. We have spent a lot of time on this step because it will significantly affect both our data diagram and the ETL procedure in the next stages.

The following table depicts our fact and dimension tables respectively:

|  |  |  |
| --- | --- | --- |
| **Table Name** | **Fact/Dimension Table** | **Description** |
| dimStore | Dimension Table | Contains all Store related data |
| dimProduct | Dimension Table | Contains all Product related data |
| dimTransaction | Dimension Table | Contains all Transaction related data such as the date and time of transaction, ReceiptNbr etc. |
| dimDate | Dimension Table | Contains detailed information of all the dates for the years 2021 and 2022 (such as year, month,week, day etc.) |
| factHealthOptionsInc | Fact Table | Contains all the primary keys of all the dimension tables and some other attributes on whom aggregations can be performed. |

**NOTE:** For the dimDate dimension table, we populated the table with data in SSMS itself using a **Stored Procedure** called **dimDateProcedure**.

**Importing data into SSIS for ETL Process and removing duplicates:**

Once we finished the data warehouse design, we then proceeded to the ETL process in SSIS in Visual Studio. In a project titled **CaseStudy2023\_TeamPavuluriPillaiVemula**, we created a package with the same name.

We first created a Data Flow task in which we decided to perform all the cleanup operations including removing the duplicate records from the original dataset. After importing data using an **ADO.Net Source**, we used a Sort function to remove the duplicates. We used a combination of columns to achieve this: (ReceiptNbr,TransDate,TransTime,StoreNbr,ProductNbr,Quantity), which resulted in **74,111 unique rows**. We then proceeded to the cleanup operations.

**Data cleanup:**

To make sure that the final data is clean with the least number of inconsistencies, we performed the following cleanup operations:

* Trimmed the whitespaces in columns that had high possibilities of having whitespaces using the **TRIM** string function.
* Removed all instances of “?” in the ProductDesc column
* We then proceeded to perform some cleanup operations for some columns that were most prevalent on a first glance at the data:
  + Replaced the MenuCategory fields that had “Side” to “Sides” (since the data already had “Sides” as one of the categories, having both would be redundant)
  + Replaced all instances of 0 with “Contact Us for details!” in the ProductDesc column.
  + Replaced all instances of “Soup & Salad” with “Soups & Salads” and “Sandwiches” with “Sandwich” in the MenuType column to avoid having different types for similar items that basically fall under the same type.
  + Replaced all NULL values in the BuildingType column to “No Information” to remove NULLS in the data.
  + Replaced all instances of 0 with “NA” and “Soup & Salad” with “Soups & Salads” in MenuSubCategory (Because having Soup & Salad and Soups & Salads as two different sub categories will only cause confusion)
* Once we finished the basic cleanups, we then proceeded to perform a deeper cleanup with primary focus on the Menu elements as they were the ones that required the most corrections to be made. We implemented a **Conditional Split** to direct only records that satisfied a particular condition to a derived column rather than all the records.

**NOTE:** Even though a conditional split is not the best option when there are a lot of conditions, we had to go with it because in a derived column, each column can be modified only once. To overcome this issue, we had to take several derived columns, one for each condition in the conditional split.

* Replaced all instances of “Soup” with “Soups & Salads” in MenuSubCategory column.
* Replaced all instances of “Salad” with “Soups & Salads” in MenuSubCategory column.

**NOTE:** The above two replacements have been done because the “Soups& Salads” already exists as one of the sub categories and it would be pointless to have them both.

* Modified the MenuSubCategory of Toast Cup from “NA” to a new category called “Toast” since the product doesn’t fall under any existing sub category.
* Observed two products that had the same MenuName: Fried Fish Sandwich. This was because of an extra whitespace in one of the names, which has been removed.
* Modified the MenuSubCategory field of Club Sandwich to “Beef or Chicken” because initially there were two sub categories “Beef” and “Chicken” for the same item which caused some ambiguity.
* Modified the MenuSubCategory of the Product having ProductNbr 3300 from “Sandwich” to “Fish” based on its Menu Name (Spicy Fish Sandwich).
* Modified the MenuSubCategory of ProductNbr 3501 from “NA” to “Kids”, an already existing sub category based on the MenuName (Kid’s Grill Cheese Meal)
* Modified the MenuCategory of the product “Strawberry Banana Smoothie and Toast Cup” from Entrees to Drinks and the MenuType from “Soups & Salads” to “Beverage”.
* Modified MenuCategory of the Product “Veggie Sausage English Muffin” from Entrees to Breakfast based on the product description.
* Modified the MenuSubCategory of the Product “Fried Artichoke Sandwich” from “NA” to “Veggie” based on the product description.
* Corrected the name of the Product “Sweet Team-Large” to “Sweet Tea-Large”.
* Modified the MenuCategory of the Product “Chicken Biscuit” from “Entrees” to “Breakfast” and the MenuSubCategory from 0 to “Chicken” based on the product description.
* Modified the MenuType,MenuCategory and MenuSubCategory of the product “Vegetarian Breakfast Burrito with Salsa” from “Soups & Salads”,”Entrees”,0 to “Breakfast”,”Breakfast” and “Veggie” respectively, depending on the product description.
* There were two ProductNbr: 3400 and 5005, having the same MenuName “Korean Spicy Chicken Sandwich”. Based on the product description, modified the MenuName,MenuType,MenuCategory and MenuSubCategoryod ProductNbr 3400 to “Spicy Korean Salad”,”Soups & Salads”,”Salad” and “Soups & Salads”, thus creating a new menu item.

Once we finished the cleanup operations, we performed a union on all the records and sent them to a new table called **dummyTable** using which we would later populate our fact and dimension tables.

**Populating the fact and dimension tables after cleanup in SSIS:**

* Using the Control Flow tool Execute SQL Task, we populated dimension tables: dimProduct, dimStore and dimTransaction in SSIS, except the dimDate table (which we populated using a stored procedure in SSMS).
* Once we populated the dimension tables, we populated the factHealthOptionsInc (fact table) using the **Lookup** tool by using the surrogate key in each dimension table to lookup the data.
* We then proceeded towards execution of the package.

**Executing and testing the package repeatedly:**

* Once we finished populating all of our fact and dimension tables in SSIS, we executed our package successfully and could now view all the data in our dimension tables and the fact table.
* NOTE: Repeated testing of the package has been performed after dropping and creating the fact and dimension tables recursively to ensure error-free execution.
* The following table shows the number of records obtained in each of the tables after executing our package.

|  |  |  |
| --- | --- | --- |
| **Table Name** | **Table Type** | **Number of records after ETL process is finished** |
| dimStore | Dimension table | 5011 rows |
| dimProduct | Dimension table | 43 rows |
| dimTransaction | Dimension table | 74,111 rows |
| dimDate | Dimension table | 1095 rows |
| factHealthOptionsInc | Fact table | 74,111 rows |

**Writing the SQL statements:**

* Once the ETL process was successfully completed, we proceeded onto writing the SQL queries that were required to answer the executive questions as per the project requirement.
* Each of the queries were created inside a view without hardcoding any filters into the view considering there might be a need to use those views in future analyses.
* Another view has been written to display that all the fact table columns are being displayed without any errors when joined with all the dimension tables.

**Data Diagram:**

* In the final step of the project, we created the Data Diagram in the **Standard View**, which is as follows:

A screenshot of a computer

Description automatically generated with medium confidence