rectangle

|  |
| --- |
|  |
|  |



rectangle

**Step 1: Business Understanding**

**Industry**

Alliance Sales and Marketing is a large food broker. They are a marketing and consulting company that represents brands, products, and large companies. We used data provided by Alliance as a basis for this project.

The database and data visualization project are to provide the best insights for Sales and Alliance’s higher up managers to see what products are performing the best on a unit and dollar basis. With Alliance being the middleman between manufacturers and retailers, the higher ups would wish to know the metrics of each product to know the best products to put their name behind. They will view the overall market pictures in regard to sales, divisions, locations, and products. In this particular instance, our goal is to provide visualizations from Kroger’s sales data in relation to the chain’s yogurt sales during 2019 and 2020. Not only can Alliance demonstrate large-scale, company wide data, but it also can provide specific visualizations to items as simple as yogurt. With these visualizations, Alliance will better understand the situation at hand and become further equipped when consulting clients.

**Scenario**

Kroger has hired alliance sales representatives to create a process to produce sales reports in a timely manner. They ask for a transformation process to take weekly sales data for the Dairy and Natural Food Departments. Alliance Sales representatives Nathan, Kayla, and Aswini are the team that will produce this process. Kroger will be able to upload their weekly sales data into a designated ETL process that will create fact and dimension tables for further analysis.

**Managerial Topics**

* What are the peak selling days, months, quarters, and years for the items contained in the datasets?
* Depending on the week and year, managers need to pull reports in a timely and accurate manner.
* Visualization of the data will be essential.
* Each Dimension needs investigating on the best and lowest performing outputs.

Our project uses the CRISP-DM as a standard process model. CRISP-DM breaks the process of data mining (or other data projects) into six major phases:

* Step 1: Business Understanding
* Step 2: Data Understanding
* Step 3: Data Preparation
* Step 4: Model Building
* Step 5: Testing and Evaluation
* Step 6: Deployment

**Step 2: Data Understanding**

There are 4 files that contain both 2019 and 2020 weekly data for two departments and their Commodity. They are Dairy: Yogurt and Natural Foods: NF Yogurt. The files contain Kroger weeks which are expressed in the form “YYYY PD ## WK # (##)”. In the parentheses, is the actual week number of that year. This essentially acts as the primary key to uniquely identify each row.

Csv files were transferred to excel. We decided to create more Sale Transaction like data with an actual date assigned to each record. This will help to create visualizations later on in the project. Each record randomly assigned a date in the range of 1/1/2019 and 12/31/2020. We then created a Calendar Table using excel manipulations and calculations to each record. The original Kroger Week was kept in this table for reporting services for Top Management. Captures a concept of domain knowledge that might be needed for an organization. This way as new data is uploaded, new Kroger weeks can be generated if needed.

All columns were analyzed in R programing using summary functions to understand the values and the suggested datatypes of each. Text

Description automatically generated with medium confidence

The most significant attributes are in the following table.

Table

Description automatically generated

**Prototype ERD**

Our team also used Visio to create a sample ERD to help inform data decisions in the model creation process. This helped the team to really understand the relations between the entities. Without a clear objective, the team would have had a hard time creating a database. Timeline

Description automatically generated

**Step 3: Data Preparation**

**Database Creation**

Using SQL Server 2019, a database was created and named Alliance Sales database. The team then worked to create a SQL script to produce each table. A snapshot of script executed to create the actual database is shown on the right.

**Integration Services Setup**Graphical user interface, text, application

Description automatically generated with medium confidence

To begin the ETL process, we created an integrated service project using Visual Studios 2019, on one of the team’s personal PC. This is what we will refer to as the server part of the project. This is where the business would dump new data files to the process. Within the connection manager, a link was made to the excel file to look up different tables contained within.

First, we started loading the Calendar table. Within the dataflow, the data moved from the source connection to the data conversion. This step is to ensure that the proper datatypes are used to be loaded into the database previously created. A lookup was created to then match the appropriate columns to the columns contained in the database. The naming schema was different and so it took some time to correctly match the correct attributes. Data was then connected to the OLE DB destination.

Data type conversion proved to be an issue. It would be simple to load tables from ETL to the database using the new table functionality. But we wanted to transfer data to existing tables in the database. The issue revolves around varchar datatypes. We also noticed that we incorrectly mapped attributes in the destination SSIS step, we needed to map the copy of our attributes to the columns found in our database. This corrected many issues.

Diagram

Description automatically generated

We then loaded smaller dimension tables that were manipulated in excel such as Manufacture, Division, Commodity, Sub Commodity, and Department. If a new file were to be uploaded with new records, then we would pull the unique field from the sales transaction data file when needed. For simplicity, directly adding the dimensions saves time. Using multiple file connections to excel sheets, we were able to create data flows to transfer all records to their respective database tables. More issues with matching the correct data types occurred. Using the tooltip feature, we were able to solve numerous data type conversion issues. It is also important to note that having the source connections open will not allow a connection to be established within SISS. 

The Item and Fact tables needed special attention. We had to merge 4 separate files that contain the actual transaction data. We used the flat file connection to source the data. One could access our ETL structure to change the connection to a new file when needed.

The Item table’s main challenge was about the performance of our package. Due to the size of the files, some contain 400,000 records, transformation steps within Visual Studio caused the program to slow. We opted to use a sort to remove duplicates based on a certain column, in this case the UPC code. We removed any rows that are not used in subsequent transformation steps. Some truncation errors were also encountered. We had to edit the character length of description fields and ensure all data types were matching throughout the dataflow. Lookup steps still took a significant amount of time. The team opted to lookup keys in excel and map them manually to save project time.

Data was loaded with source steps, data conversions, surrogate key lookups, and destination steps. For the fact table, data files were loaded separately and then were validated on the rows transferred to the database.

| **ETL Package Steps (In order)** | |
| --- | --- |
| **Step** | **Photo** |
| **1 – Calendar Table** |  |
| **2 – Manufacturer Table** |  |
| **3 – Division Table** |  |
| **4 – Department Table** |  |
| **5 – Commodity Table** |  |
| **6 – Sub Commodity Table** |  |
| **7 – Item Table** |  |
| **8 – Input File Extraction** |  |
| **9 – Extract Fact Transactions p1** |  |
| **10 – Check contents of factSales** |  |
| **11 – Extract Fact Transactions p2** |  |
| **12 – Check contents of factSales** |  |
| **13 – Extract Fact Transactions p3** |  |
| **14 – Check contents of factSales** |  |
| **15 – Extract Fact Transactions p4** |  |
| **16 – Check contents of factSales** |  |
| **End of Lab** | |

**Step 4: Model Building (Cubes, Visuals, and Labs)**

This section is dedicated to applying the database to other applications. We first began with a basic cube formation. Next, we wanted to experiment with access to see if you could update records as needed. Next, PowerBI with its data model, transformation steps, and visualization were utilized to help great powerful insights. Lastly, we used SSRS to build similar reports. Four labs are provided below to help guide future users.

| **Lab 1: Cube Creation for Easy Data Browsing** | |
| --- | --- |
| 1 – Change Server Name | Graphical user interface, text, application, email  Description automatically generated |
| 2 – Create Data Source, place Server Name and Database | `Graphical user interface, application  Description automatically generated |
| 3 – Data Source View has been created. We will need a new Item Table that combines PK to FK relationships. This will help make the STAR schema. | Graphical user interface  Description automatically generated |
| 4 – Diagram View, we will be using the 5 tables at the top of the page. | Diagram, schematic  Description automatically generated |
| 5 – Select Measure Groups (Fact Table). | Graphical user interface, text, application, email  Description automatically generated |
| 6 – Select Dimension Tables (Everything else). | Graphical user interface, text, application, email  Description automatically generated |
| 7 – New Cube View | Graphical user interface  Description automatically generated with low confidence |
| 8 – Solution Explorer |  |
| 9 – Building Cube |  |
| 10 – Realized Dimensions were missing key columns, needed to add columns to each DIM one by one. |  |
| 11 – Example: Management needs a data pull on Division 6 on the total sales for a Kroger Week. |  |
| 12 – Example: Management needs another data pull this time with Total Sales filtered by the Month December in 2020 in the each department available. |  |
| **End of Lab** | |

| **Lab 2: How to Update Records directly in Access** | |
| --- | --- |
| 1 – Use External Data Sources in Access in the upper ribbon. Select Link to Data Source, this option is what lets one update and pass the updates directly to the Server. |  |
| 2 – Click Machine Data Source, then New. |  |
| 3 – Click OK to bypass. |  |
| 4 – Select User Data Source |  |
| 5 – We will be using SQL Server Native Client 11.0 since it is the most recent driver. |  |
| 6 – Name your Connect and place your Server and Instance name (Machine Name/Instance). |  |
| 7 – Select the database in the Server you wish to connect tables. Place the READWRITE option in the application intent section. Keep all defaults. |  |
| 8 – Test Connection |  |
| 9 – Select all Tables that you would like to use. |  |
| 10 – Tables should appear in the left-hand pane. |  |
| 11 – Example: SQL Server View of all Commodities |  |
| 12– Example: Adding new commodities. Always remember to hit save. |  |
| 13 – Example: Updates the actual SQL Server database that Access is linked to. This provides a great source to update records manually. |  |
| **End of Lab** | |

| **Lab 3: PowerBI** | |
| --- | --- |
| 1 – Connected to SQL Server database to populate queries | Graphical user interface  Description automatically generated |
| 2 – Schema | Graphical user interface, application  Description automatically generated |
| 3 – Total Sales Page    Slicers on the left-hand side so you can filter by Division Name, Department Name, Commodity Name, and Manufacturer Name  Each graph displays the total sales by year in columns as well as the average of total sales with the line. | Graphical user interface, application, Teams  Description automatically generated |
| 4 – Unit Sales Page    Slicers on the left-hand side so you can filter by Division Name, Department Name, Commodity Name, and Manufacturer Name  Each graph displays the total unit sales by year in columns as well as the average of unit sales with the line. | Graphical user interface, application  Description automatically generated |
| 5 – Total Sales and Unit Sales by Division  This graph displays each divisions Total Sales (columns) and Unit Sales (line) for all the data in the database. | Chart, histogram  Description automatically generated |
| 6 – Top 15 Items by Total Sales and Unit Sales  This graphs shows the top 15 items that Kroger sells in terms of Total Sales (columns) and Unit Sales (line). | Chart, histogram  Description automatically generated |
| 7 – Total Sales and Unit Sales by Kroger Week  This graphs shows Total Sales (columns) and Unit Sales (line) by Kroger Week. A Kroger Week is how the company tracks their sales, it is listed as the year – week #. | Chart, bar chart, histogram  Description automatically generated |
| **End of Lab** | |

| **Lab 4: Reports in SSRS** | |
| --- | --- |
| 1- Change deployment properties |  |
| 2 - Annual Sales by Department - Query | SELECT dimDept.ITM\_SCN\_DEPT\_DESC as Department, dimCalendar.yr as Year, format(SUM(factSales.total\_sales),'c') AS Total\_Sales  FROM dimCalendar INNER JOIN  factSales ON dimCalendar.date\_id = factSales.date\_id INNER JOIN  dimItem ON factSales.ITEM\_ID = dimItem.ITEM\_ID INNER JOIN  dimDept ON dimItem.DEPT\_ID = dimDept.DEPT\_ID  GROUP BY dimDept.ITM\_SCN\_DEPT\_DESC, dimCalendar.yr  ORDER BY dimDept.ITM\_SCN\_DEPT\_DESC, dimCalendar.yr |
| 2 - Annual Sales by Department |  |
| 3 - Average Sales by Department - Query | SELECT dimDept.ITM\_SCN\_DEPT\_DESC as Department, dimCalendar.mnth\_name as Month, format(SUM(factSales.total\_sales),'c') AS Average\_Monthly\_Sales  FROM dimCalendar INNER JOIN  factSales ON dimCalendar.date\_id = factSales.date\_id INNER JOIN  dimItem ON factSales.ITEM\_ID = dimItem.ITEM\_ID INNER JOIN  dimDept ON dimItem.DEPT\_ID = dimDept.DEPT\_ID  GROUP BY dimDept.ITM\_SCN\_DEPT\_DESC, dimCalendar.mnth\_name  ORDER BY dimDept.ITM\_SCN\_DEPT\_DESC |
| 3 a - Average Sales by Department - 05 Dairy |  |
| 3 b - Average Sales by Department - 17 NATURAL FOODS |  |
| 4 - Average Sales based on Day of week - Query | SELECT dimDept.ITM\_SCN\_DEPT\_DESC as Department, dimCalendar.day\_of\_week, format(AVG(factSales.total\_sales),'c') AS Average\_Sales  FROM dimCalendar INNER JOIN  factSales ON dimCalendar.date\_id = factSales.date\_id INNER JOIN  dimItem ON factSales.ITEM\_ID = dimItem.ITEM\_ID INNER JOIN  dimDept ON dimItem.DEPT\_ID = dimDept.DEPT\_ID  GROUP BY dimDept.ITM\_SCN\_DEPT\_DESC, dimCalendar.day\_of\_week  ORDER BY dimDept.ITM\_SCN\_DEPT\_DESC, dimCalendar.day\_of\_week |
| 4 a - Average Sales based on Day of week - 05 Dairy |  |
| 4 b - Average Sales based on Day of week - |  |
| 5 - Top Items based on Quantity Sold - Query | SELECT top 40 dimItem.ITEM\_DESCRIPTION, dimCalendar.yr as year, SUM(factSales.unit\_sales) AS [Quantity Sold]  FROM dimCalendar INNER JOIN  factSales ON dimCalendar.date\_id = factSales.date\_id INNER JOIN  dimItem ON factSales.ITEM\_ID = dimItem.ITEM\_ID  GROUP BY dimItem.ITEM\_DESCRIPTION, dimCalendar.yr  order by SUM(factSales.unit\_sales) desc,dimCalendar.yr |
| 5 a - Top 20 Items based on Quantity Sold - 2019 |  |
| 5 b - Top 20 Items based on Quantity Sold - 2020 |  |
| 6 - Top Items based on Total Sales - Query | SELECT top 40 dimItem.ITEM\_DESCRIPTION, dimCalendar.yr as year, format(SUM(factSales.total\_sales),'c') AS [Total Sales]  FROM dimCalendar INNER JOIN  factSales ON dimCalendar.date\_id = factSales.date\_id INNER JOIN  dimItem ON factSales.ITEM\_ID = dimItem.ITEM\_ID  GROUP BY dimItem.ITEM\_DESCRIPTION, dimCalendar.yr  order by SUM(factSales.total\_sales) desc,dimCalendar.yr |
| 6 a - Top 20 Items based on Total Sales - 2019 |  |
| 6 b - Top 20 Items based on Total Sales - 2020 |  |
| **End of Lab** | |

**Step 5 and 6: Testing and Deployment**

**Lessons Learned**

* Access provides some functionality to easy update records.
* Working as a team on a database project was very helpful to see different ideas of how to go about its creation. This does model the real world.
* PowerBI is a great resource to produce visualizations.
* Visual Studio has some issues when it comes to loading and processing large data files. With the proper resources, performance could be improved.
* Our project was limited to only two Kroger Departments, more significant insights could be discovered if the team had more datasets.
* Data preparation takes a significant amount of time compared to the actual model building.
* Reports can be replicated in various applications easily, choosing the best application depends on business context and capabilities.