464 Part A Description

Description

As we described in the project specifications, we have taken that data provided in the Walmart Kaggle Data, and setup a database structure to help us answer question about strange weather patterns affect sold purchase. As mentioned in the case the main goal is to "predict the amount of each product sold around the time of major weather events," Walmart manager believe that there could be a trend here that affect purchasing behaviour, so we have a built a database suited to answering those inquiries.

Our database took existing Walmart data and merged it with market data to create tull data set to test inquiries. The structures were setup to remain simple to allow, fast query times & brief SQL queries. In their total there are eight entries, that maintain eight 1:M relationship. We have removed all M:M relationship within this model.

Figure 1: Data Manipulation

Given Data	Test Data (Randomized)
- Store id & station id - Entire weather entity data - Item number & units sold - Dates	- Product name, description, type, department, unit price - Store and station address, postal code, zip - Inventory quantity - Timestamp dates

Our Relationships

- Weather Event and Station: Each weather station has one or many weather events, and many weather events are part of exactly one station.
- Station and Store: Each store can have exactly one weather station, one weather station can be used for one or many stores.
- Store and Inventory: Each store can have multiple product inventories, but each inventory of product has only one store.
- Store and Sale: Each store can have multiple sales but each sale belongs to one store.
- Item and Inventory: Each item can be part of multiple set of inventory, but each inventory has only one occurrence of that item (items are separated by item brand)
- Item and Department: Each item belong to one department, but each department can have multiple items within that department
- Item and Line_item: Each item can be placed within multiple line_items, but each line_item can have only one occurrence of that item.
- Sale and Line_item: Each sale includes multiple line_items but each line_item can have only one sale.

Assumptions

We maintained some general assumptions in the design of our database. First, we have kept the the location where your derived price from in the both the sales entity and product entity. The product entity allows you to derive per unit price measurements, as well as the total sales by combining unit price with unit quantity. We believe these efficiency gains is worth any minor database redundancy that exists here. Secondly, we assumed that stores description will not require a unique name, since all the store are Walmart locations, we will refer to them via store address or store ID. Thirdly, we have extended the operational side of the database to include an inventory entity and a lineitem entity to allow users to gather further insights on changes in store inventory, as well as ordering and product details. Finally the largest assumption that we have made, is to not include a customers entity. Walmart has a very large operation,

to include a customers entity would be out-of-scope of this problem's scope, bring in huge data warehousing technology, increase the need security precautions around the data, substantially increase the size the database, and require Walmart to gather consumer data from it store via account creation (or possibly from credit card). Overall our group believes that our occam's razor approach reduces the complexity of the solution and provided the capability to address Walmart's concerns.

Potential Insights

This database is able to examine how different weather affects sales for a list of 111 items across Walmart stores. Each item is further defined with a description, department, price and inventory to better determine which variables affect sales data.We have also added the ability to constrain the query by date/time and time-stamps, as well adding a department entities & attributes, to allow the database to be used for more modern queries searches. The database can also aggregate sales to determine which dates generate the most revenue by combining the unit price and unit quantity from sales and line_item.