Final Project:

Ridex Database

Spencer Fairbairn, Chidi Nzerem, Jackson Yee,

Whitney Goit, Rohit Jain

MSBX 5405 | Group 9

Keith Sparling

Project Scenario

Description:

Ridex is a new ride-sharing app that is poised to compete with Uber and Lyft. With Ridex, drivers will have to register in order to offer rides through the app. The CEO of Ridex wants to be able to better track key performance indicators about how Ridex is fairing in the competitive market of ride-share apps. Ridex is looking towards expanding into new US cities, to do so they want to make sure they have all the relevant information about their consumers.

Once a driver is registered with Ridex, they will be able to start taking rides from our consumers. Ridex performs drug tests as well as background checks on all their drivers, to ensure the safety of the riders. Ridex will also be keeping track of location data for its drivers in order to better optimize future trips. We want to use this data in order to be more prepared for peak ride-sharing hours, by strategically having riders in closer proximity to hot-spots in an attempt to decrease wait time.

Ridex is a relatively new company, with little infrastructure set up. They currently have the majority of their data stored in an Excel spreadsheet to track their consumers. Due to this, Ridex is wanting to manage their data more efficiently in order to make their information more accessible for their employees. Organizing consumer and driver's information will allow Ridex to show only the necessary information to different departments within the company. This will lead to increased efficiency as Ridex employees will no longer be required to comb through a large, unruly spreadsheet.

Relationships:

Our database is one that focuses on the inner workings of the relationship between a ridex driver, its customers and location. That being said our database can be broken down into sections. The logic in working with this type of relationship was that we would be able to see how one side of this transaction affects the other which would be useful for the executive time to make or change company protocols.

The first half of the database we would be focusing on is the customer side of the database. On this side of the database we wanted to focus on the user information. Mainly the customer bookmarks which gave us specific locations the customer frequently visited.
The other table that was key in the database was the location table. This table documented where drivers picked up and dropped off customers. This was a key relationship because it allowed us

to make a connection between where drivers were and where needed to be based on the bookmark locations customers wanted.

Another important table relationship we made was the relationship between the promotions table and the date table, which showed the days customers travelled the most and the list. Based on this we could put out promotions on days customers travelled least to insensitive them to use our drivers. These are just some examples of the relationships/importance between the tables in our database.

Relevant Tables:

Since our dataset was self created we did not need to transform any of the data. Because the set was formatted in a way we wanted it. Some relevant tables in our database were the Dim_Cust_Bookmark, Dim_location, Dim_account and the Dim_promo.

Dim_cust_bookmark and Dim_location are too very important columns because of the relationships between them. From these two tables we can tell locations frequently visited by customers as well as locations frequently driven by the drivers. This could be very important for final analysis in order to get more rides from customers.

The other two important tables are Dim_promo, dim_account and Dim_date. This relationship between these tables could be beneficial for figuring out what specific promotions customers are using. As well as what days customers ride less or more in order to increase or decrease promotion time during those periods.

Normalization:

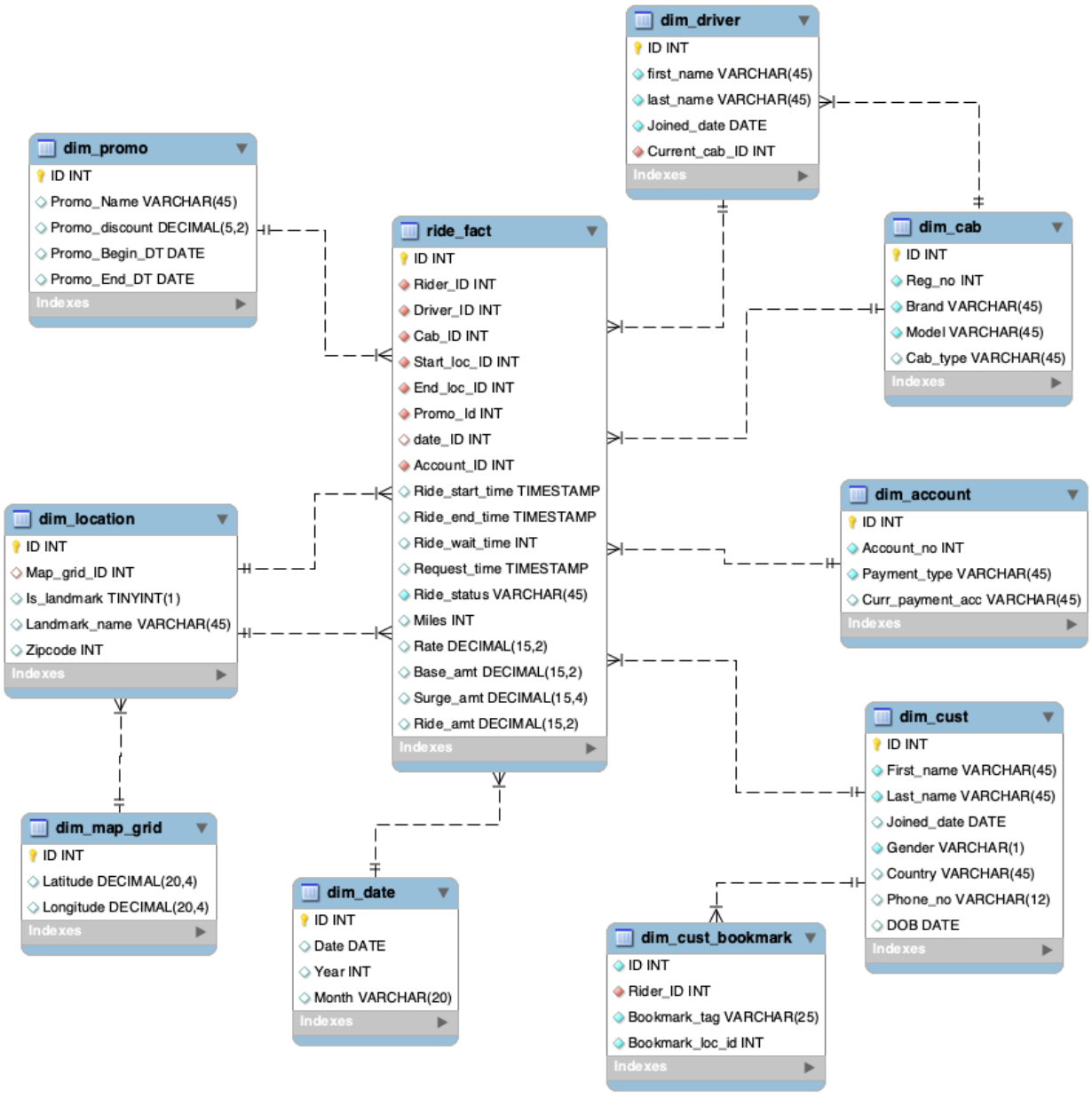
To normalize the database we initially began by separating tables based on referenceable IDs. We wanted to use a star schema with our main trip fact table pulling data in from other smaller tables, as to make the entire database more legible and efficient. To do this we separated our tables into dimension tables and a fact table that pulls information from the dimension tables using IDs.

We normalized the Dim_Cust table by taking out the customer bookmarks such as “house” or “gym”. We decided to use another table to hold this information as some riders may not have bookmarks saved to their profile. This made it easier to display only the necessary bookmark’s by assigning the customer a bookmark_ID that is referenced within the Dim_Cust table. We normalized the Dim_Location table in a similar way by putting the latitude and

longitude data in a separate table, Dim_Map_Grid. Since latitude and longitude data is not always the most important, we decided to make it a referenceable table.

To help normalize our dataset we broke the drivers table into Dim_Driver and Dim_Cab tables to separate the drivers and their vehicle types. We can use specific IDs within our Trip_Fact table to find out the driver of a ride, and what kind of car they were driving. This is useful if a driver ever changes cars, it will be much easier to update all of the information within the Trip_Fact table.

ER Model:


dim_driver
ID INT
first_name VARCHAR(45)
last_name VARCHAR(45)
Joined_date DATE
Current_cab_ID INT
Indexes
dim_promo
ID INT
Promo_Name VARCHAR(45)
Promo_discount DECIMAL(5,2)
Promo_Begin_DT DATE
Promo_End_DT DATE
Indexes
ride_fact
ID INT
Rider_ID INT
Driver_ID INT
Cab_ID INT
Start_loc_ID INT
End_loc_ID INT
Promo_Id INT
date_ID INT
Account_ID INT
Ride_start_time TIMESTAMP
Ride_end_time TIMESTAMP
Ride_wait_time INT
Request_time TIMESTAMP
Ride_status VARCHAR(45)
Miles INT
Rate DECIMAL(15,2)
Base_amt DECIMAL(15,2)
Surge_amt DECIMAL(15,4)
Ride_amt DECIMAL(15,2)
Indexes
dim_cab
ID INT
Reg_no INT
Brand VARCHAR(45)
Model VARCHAR(45)
Cab_type VARCHAR(45)
Indexes
dim_location
ID INT
Map_grid_ID INT
Is_landmark TINYINT(1)
Landmark_name VARCHAR(45)
Zipcode INT
Indexes
dim_account
ID INT
Account_no INT
Payment_type VARCHAR(45)
Curr_payment_acc VARCHAR(45)
Indexes
dim_map_grid
ID INT
Latitude DECIMAL(20,4)
Longitude DECIMAL(20,4)
Indexes
dim_date
ID INT
Date DATE
Year INT
Month VARCHAR(20)
Indexes
dim_cust
ID INT
First_name VARCHAR(45)
Last_name VARCHAR(45)
Joined_date DATE
Gender VARCHAR(1)
Country VARCHAR(45)
Phone_no VARCHAR(12)
DOB DATE
Indexes
dim_cust_bookmark
ID INT
Rider_ID INT
Bookmark_tag VARCHAR(25)
Bookmark_loc_id INT
Indexes