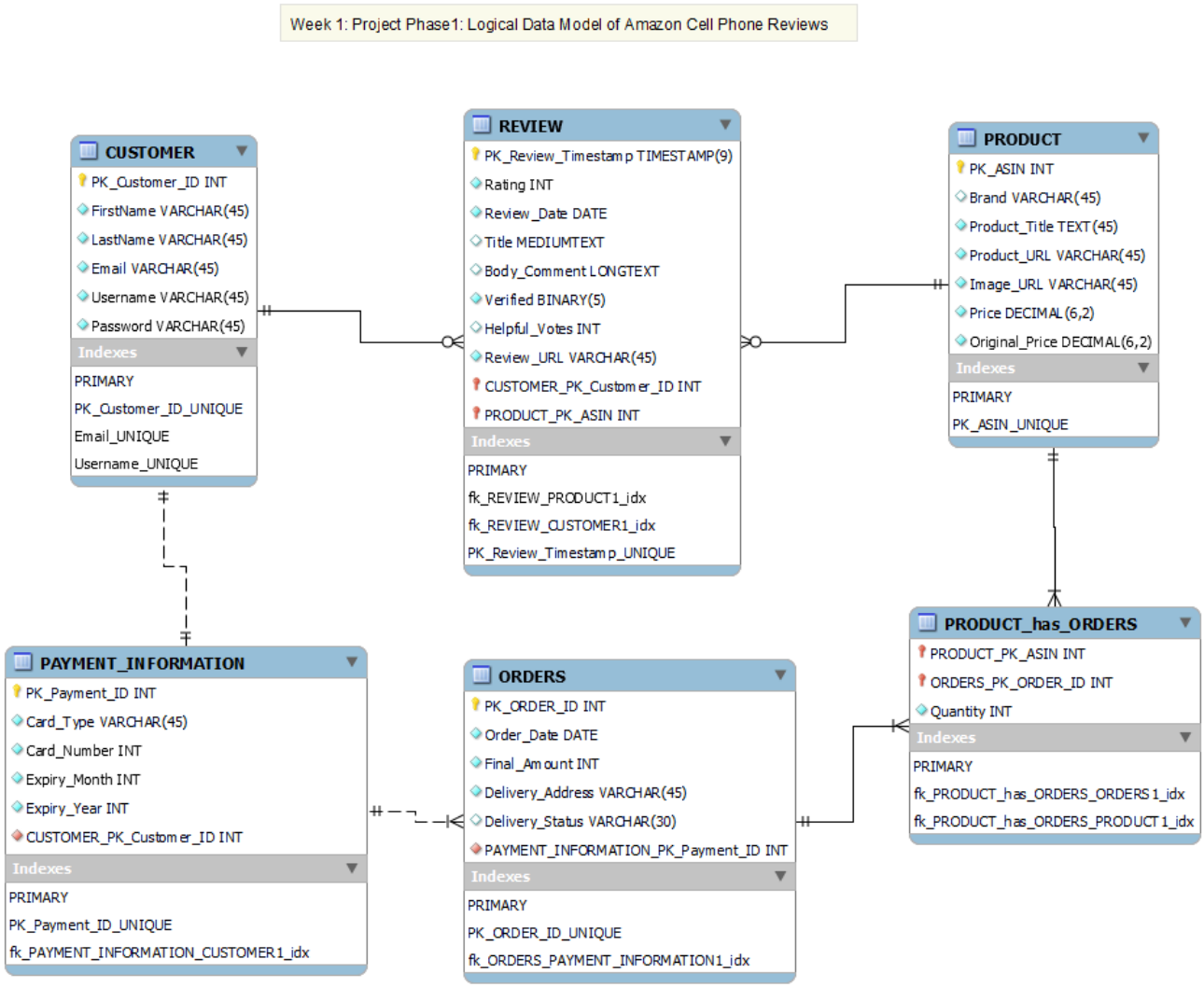
Datawarehouse Project : Logical Data Model of Amazon Cell Phone Reviews



1. Tool Used: MySQL Workbench

2. The intersection/ associate table review, has a composite key made of PK_Customer_ID , PK_ASIN , PK_Review_Timestamp and is used as primary key. PK_Customer_ID, PK_ASIN are also the foreign keys in the review table.

3. The intersection/ associate table product_has_orders, has a composite key made of PK_ASIN and PK_Order_ID and is used as primary key . PK_ASIN and PK_Order_ID are also the foreign keys in product_has_orders table.

4. Columns rating and total reviews (from items.csv file) are not included in this E- R Diagram as it doesnot go with the grain of the review table which is one review per product per customer per timestamp.Column rating in items.csv file is an average of all the ratings given in reviews.csv file.
So, Rating and Total Reviews can be found by simple SQL query and these columns can be added to the dimensional model, as degenerate dimension if required, in the review fact table by adding SQL query in my ETL logic.

5. Tables: Payment_Information ,Orders, Product_has_Orders(associate table) - were not a part of the csv files. I have included them in my E-R diagram to give a complete picture of logical data model.

6.In my customer table, I have added columns PK_Customer_ID , Email, Username, Password. These columns were not a part of my csv files.
I have split the column name (in items.csv file) into FirstName and LastName.

- Dataset used: Amazon Cell Phone Reviews
  https://www.kaggle.com/grikomsn/amazon-cell-phones-reviews?select=20191226-items.csv

- Tool Used: MySQL Workbench

- Entities:
  i. Customer
  ii. Product
  iii. Reviews (intersection / associate table between customer and product)
  iv. Payment
  v. Orders
  vi. Product_has_orders (intersection / associate table between product and order)

- Cardinalities:
  i. Each customer may or may not give reviews and each product may or may not have customer reviews (Many- to- Many Relationships i.e., Optional Many).
  ii. Each product can have one or many orders and each order can have one or many products (Many- to-Many Relationship) (Mandatory Many)
  iii. Each payment information can be used for one or many orders (Mandatory One to Many Relationship).
  iv. Each order will have only one payment information (Mandatory One Relationship).
  v. Each payment belongs to one and only customer and each customer has only one payment information (Mandatory One to One Relationship).