Retail Store Management

Introduction

This document describes the steps that were taken to deliver the project. The project consisted of designing and implementing a database that stores information for a retail store, Gap. This document was developed by eight students with the intent to create a database for Gap. The database reflects the paths the goods take to reach the customer from a vendor.

Overview

Gap is a big retail store that sells various items, not only clothing. This database is only a part of their vendor-to-customer system, however, the store has much more data that we won't be discussing in this document.

Assumptions and Special Considerations

It is assumed that an order can only have one product placed in it for easy accounting purposes and rules by this GAP store. For example, a T-shirt, an electronic item, a pair of pants, etc. all must be a part of their own separate order. However, a T-shirt, electronics, etc. can be part of multiple orders. Giving it that 1:M relationship between orders and products. Also, at this Gap distribution channel, a warehouse only has one designated vendor they work with to provide merchandise. However, a vendor could service multiple warehouses at the same time, giving it the 1:M relationship.

Requirements and Definition Document

The five entities that will be discussed are: Customer, Orders, Product, Warehouse and Vendor. Those entities and attributes are described below: Customer Entity The first entity is Customer where we will house Gap's customer data. The data that will be housed in this entity contains the 6 attributes: Customer_ID, First_Name, Last_Name , Address, Phone_no, and Email. The primary key in this entity is Customer_ID. There is no foreign key.

Orders Entity
The second entity is Orders where we will house Gap's orders for each transaction. The data that will be housed in this entity contains the 7 attributes: Order_ID, Product_ID, Customer_ID, Total_Price, Order_Date, Delivery_Address, and Payment_Method. The primary key in this entity is Order_ID. The foreign key is Customer_ID and Product_ID.
Payment_method could be cash, card, or check as a way of paying for your order

Product Entity

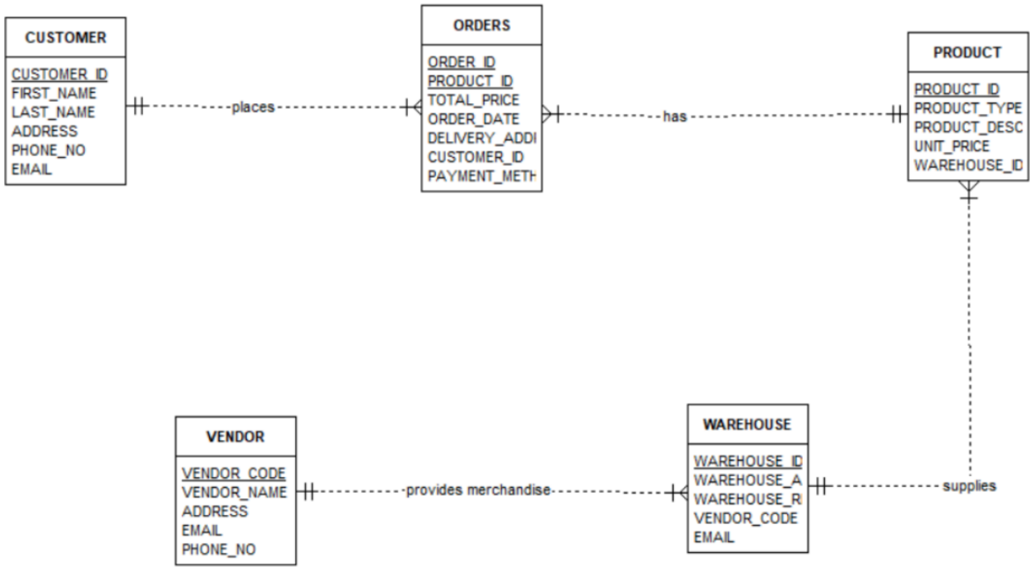
The third entity is the Product table where we will house Gap's products sold. The data that will be housed in this entity contains the 5 attributes: Product_ID, Product_Type, Product_Description,Unit_Price, and Warehouse_ID. The primary key in this entity is Product_ID. The foreign key in this entity comes from the Warehouse Table and Warehouse_ID. Product_type could be clothing, shoes, groceries, electronics, or other. Product_description is the description of the type of product, for example other could be chocolate or clothing could be white t-shirt.

Warehouse Entity

The fourth entity is Warehouse where we will house Gap's warehouse information for the store. The data that will be housed in this entity contains the 5 attributes: Warehouse_ID, Vendor_Code, Warehouse_Address, Warehouse_Region, and Email. The primary key in this entity is Warehouse_ID. The foreign key in this entity comes from the Vendor Table and it is Vendor_Code. Warehouse_Region is where it is located to service the Stores in that area (DFW, Plano, Etc).

Vendor Entity

The fifth and last entity is Vendor where we will house Gap's vendor's information. Its primary key is Vendor_code which provides a unique code for each vendor. The data that will be housed in this entity contains five attributes: Vendor_code, Vendor_name, Address, Email, and Phone_no. The primary key in this entity is Vendor_code. There is no foreign key.


CUSTOMER
CUSTOMER_ID
FIRST_NAME
LAST_NAME
ADDRESS
PHONE_NO
EMAIL
places
ORDERS
ORDER_ID
PRODUCT_ID
TOTAL_PRICE
ORDER_DATE
DELIVERY_ADDI
CUSTOMER_ID
PAYMENT_METH
has
PRODUCT
PRODUCT_ID
PRODUCT_TYPE
PRODUCT_DESC
UNIT_PRICE
WAREHOUSE_ID
VENDOR
VENDOR_CODE
VENDOR_NAME
ADDRESS
EMAIL
PHONE_NO
provides merchandise
WAREHOUSE
WAREHOUSE_ID
WAREHOUSE_A
WAREHOUSE_R
VENDOR_CODE
EMAIL
supplies