**Database Design Document**

**Team 7**

Sai Srilakshmi Kanumuri

Shrishti Khanna

Tyler McCarthy

Ben Reynolds

INFO6210 - Spring 2020

4/9/2020

**Database Purpose**

The purpose of our database is to track and monitor the data used by a grocery store to keep its inventory stocked and available for potential customers. This way they will not need to worry about running out of items. Some of the functionalities of the database include managing employees, products and suppliers.The primary users are administrators who are in charge of managing the store and keeping items and shelves stocked.

**Business Problems Addressed**

1. Allow store managers to keep a record of all employees including their personal information, the store they work at, their job title, and department.
2. Provide and track the amount of inventory at each store location to prevent running out of product or over-ordering.
3. Track orders and deliveries to hold suppliers accountable for making timely deliveries.
4. Keep a record of all customers to provide marketing with customer contact information and addresses.
5. Maintain sales records generating reports in the case of an audit and for predicting future revenues.

**Business Rules**

1. Each employee must work at exactly one store and in exactly one department
2. Each employee and customer (person) must have an address
3. Each sale must be to exactly one customer and performed by exactly one employee
4. Each sale must include at least one product
5. Each product must belong to exactly one category
6. Each store must have inventory of at least one product
7. Each store and supplier must have exactly one address
8. Each order must be delivered in one or more shipments, each of which is received by exactly one employee
9. Each order must be for exactly one store and contain one or more line items

**Design Decisions**

|  |  |  |
| --- | --- | --- |
| **Entity** | **Why Entity is Included** | **How Entity is Related to Other Entities** |
| Store | A store is the central entity in this database design. All other entities are derived based on their relationship to the store and in support of the store. The store does not have many attributes itself because the bulk of the information is stored in related tables such as Address and Employee. | The store has foreign key references to Employee (to record the store manager) and Address (to store the location of this store). It is also referenced by Sale and Inventory which can only exist in relation to a specific Store instance. |
| Address | Address contains data common to several other entities. It was separated into a separate entity in order to conform to the 3rd normal form. It provides a common format for storing information related to Store, Person, and Supplier. | Address has no foreign key references but it’s Id is referenced as a foreign key from Store, Person, and Supplier. This allows those three entities to act as the “owning” entity such that each of them “has an” Address. |
| Inventory | Inventory relates a given Store to the Products it currently has available for sale. In some ways, this table acts as a join table between Store and Product but provides additional information as well such as expiration date and unit price. For this reason, this entity has a composite natural key based on its relationships instead of a surrogate key. | Inventory references the Store for which it is tracking inventory and the product which it is counting. Each store has one Inventory per product. Inventory provides additional product information that is relevant to a single Store. |
| ProductCategory | ProductCategory only has a surrogate key and name attributes. It is a simple entity used to just provide a common set of categories which can be associated with products to correlate them. | The only relationship with ProductCategory is a foreign key reference from Product. This allows the product to know the category to which it belongs. |
| Product | Product is another entity which is central to the database design. Product contains information specific to a single type of sellable item, such as its brand and UPC code. This entity depends on its relationships to relate to a Store and its Sales. | Product has a foreign key reference to ProductCategory so each product can be related by a common set of categories. Product is also referenced in OrderLineItem, SaleLineItem, and Inventory. Those three entities act as join tables relating products to Stores, Sales, and Orders. |
| OrderLineItem | OrderLineItem is a join table between Order and Product. For this reason, it has no surrogate key but uses its relationship to Order and Product to form a natural composite key. | This entity has a many-to-one relationship with Order and a one-to-one relationship with product. This allows a single Order to include ordering many different types of products. |
| Order | Orders provide a coupling entity between Stores and Suppliers. It contains attributes like order date (to track delivery time) and order total (to accumulate the total price of all products included in the order). | Order references Store and Supplier to keep a record of who placed the order and who fulfilled it. It is also referenced by OrderLineItem in a one-to-many relationship such that each Order has many OrderLineItems. |
| Shipment | Shipments break Orders into smaller components based on individual deliveries to a Store. Shipment only tracks information which would be related to a specific package or truckload (such as tracking number and delivery date). | Shipment has a foreign key referencing Employee in order to keep track of which employee signed for the shipment upon its arrival. It also references the Order of which it is a part. |
| Supplier | Supplier is similar to Store in that it is a “standalone” entity, capable of existing on its own. In order to conform to the 3rd normal form, the address of a supplier is provided through a relationship to the shared Address entity. | Supplier has a foreign key reference to Address and is referenced by a foreign key on the Order entity. This provides tracking for which supplier is fulfilling Orders placed by the Store. |
| Department | Department is to Employee as ProductCategory is to Product. It serves only to provide a means of grouping Employees under a shared “tag” of sorts, in this case the Department they work in. | Department is related to and referenced only by Employee. Employees have a many-to-one relationship with Department. That is to say many Employees work in one Department but each employee is only in one Department. |
| Employee | A Store is nothing without its Employees. Employees execute sales and receive Shipments. They also “have an” address and “are a” Person. This prevents duplicating common information if an Employee also happens to be a Customer (a common use-case in real life). | Sales reference the Employee which performed them and Shipments reference the Employee which received / signed for them. Employees also reference Address (a common entity for address information) and Person (a common entity for information common to any human [Customer or Employee]). |
| Person | Person is an entity used to help the schema in conforming to the 3rd normal form. Without the Person entity, Customer and Employee would share many common attributes. Person provides a table which allows us to separate out those common attributes such as Name, Age, and Gender. This also allows us to have a Person who is both a Customer and an Employee without storing their personal information two places in the database. | Person is referenced by Employee and Customer through a foreign key. This provides an “is a” relationship such that Employee and Customer inherit the attributes provided by Person. Person has a foreign key reference to Address to similarly share information common to Person, Store, and Supplier. |
| Customer | Customer is a “sub-entity” of Person which provides additional attributes specific to a Person who has made a purchase at the Store. Attributes such as last sale date allow store managers to track how frequently a customer makes purchases. | Customer “is a” Person and reflects this relationship through a foreign key reference to the Person id. Each Sale executed by a store also contains a foreign key references to the Customer who made the purchase. |
| SaleLineItem | SaleLineItem is an entity which helps make up the Sale entity. SaleLineItems contain the price of each item being sold as well as the quantity sold in any particular sale. | SaleLineItem contains a foreign key reference to Sale as well as Product. This way, the products being sold as well as which sale they are a part of can be linked together. |
| Sale | Sale entities are a primary part of the database design. They track all the different sales that take place at a grocery store along with which employee made the sale and which customer purchased it. They contain other attributes such as the sale date, total, and mode of payment. | Sales have a foreign key reference to Store to indicate which store the sale took place in. Additionally, Sales also have foreign keys to Employee and Customer to show who sold and purchased the sale. |