OrderManager: A Database for Managing Products and Orders

This project is a portion of a an application for managing products, product inventory, and customer orders for an online store..

OrderManager is an e-commerce program that enables a business to manage information about products that can be sold to customers, to track current inventories of products, and to process orders for products from customers. The RDBMS maintains informaiton about products that can be ordered by costumers, tracks inventory levels of each product, and handles orders for product by customers. It enables a business to track sales of products, and allows customers to determine product availability and place orders.

The project provides a back-end that could be used for future projects in a web development class to create a front end to the database that allows both businesses and their customers access through a web or mobile interface. It could also be used as a starting point for a course in web-base software architecture to implement web services that can be accessed by web-based applications or integrated with a e-commerce system that contains product and order information.

Project details

The project is to develop and document a data model for representing the entities and relationships in an order management system, provide DDL for creating the tables, DML for adding entries in the tables, and DQL for making commonly used queries to retrieve product, inventory, and order information from the database.

The data model for the project is based on the concept of products that can be purchased, inventories of products available for purchase, customers who purchase products, and orders for products by costumers.

Product represents a product that can be purchased. It includes the name of the product, a product description, a vendor product SKU (Stock Keeping Unit) that identifies the product. For this exercise the SKU is a 12-character value of the form AA-NNNNNN-CC where A is an upper-case letter, N is a digit from 0-9, and C is either a digit or an uppper case letter. For example, "AB-123456-0N".

InventoryRecord is the number of units available for purchase and the price per unit for the current inventory (positive, with 2 digits after the decimal place).

Customer is information about the customer, including name, address, city, state, country, postal code. Make reasonable assumptions about the sizes of the fields, and whether state and country should be enumerated values or strings. The customer also has a customer id that is a numeric gensym. Payment information is not part of this database.

Order is an order for a set of products. It includes a customer ID, an order ID gensym, the order date, and shipment date indicating when the order was shipped. If shipment date is null, the order has not yet shipped. All items must be available in a single transaction to place an order.

OrderRecord is the record for an item in the order. it includes the order ID, the number of units, and the unit price. The item must be available and the inventory is automatically reduced when an order record is cretated for an order.