**StoreRunner Tech Database**

Team: Jake Christy, Alec DeVries, Justin Sorensen, Brian Turnbo, Linhao Yuan

CIS353 - 01

Fall 2021

**Database requirements:**

This database will be for an imaginary store. There are several things that a business may wish to keep track of. Customer information, orders, shipments, and inventory are just some of the things that might need to be tracked. To help keep track of all of these, the database shall have several tables to store and organize the various types of information.

Below is a list of requirements that the database must meet. An ER diagram, basic relational schema, and integrity constraints are included as well.

**Customer**

The database shall keep track of customers. For each customer, we need to track their name, address, email address, and password. Customers can have multiple addresses (e.g. home address, work address). Customers will be tracked by their email address.

**Orders**

The database shall track store orders. Each order will have a unique order id, the customer who ordered it, the order date, the total cost, and the address to ship it to.

**Products**

The database shall track products available for sale. Each product shall have a unique product id. It shall also have a name, a price, and the quantity in stock.

**Order lines**

The database shall keep a list of all of the lines of the order. Each line shall contain the product, the quantity purchased of the product, the price of the product, and the order that it belongs to.

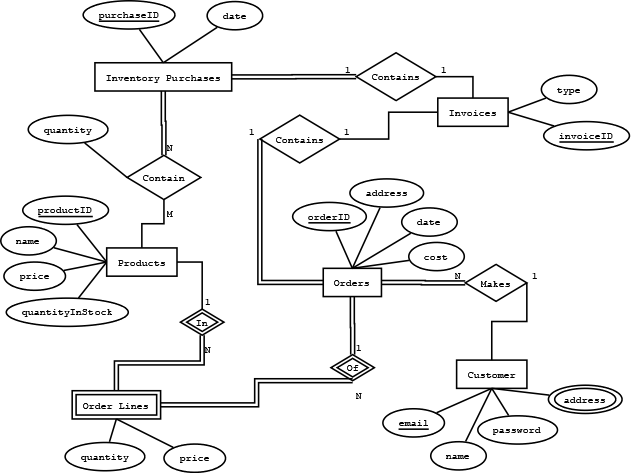
**Inventory Purchases**

The database shall keep track of orders for new inventory. Inventory purchases will have a unique id, can contain multiple products and must record the date, and the quantity purchased of each product.

**Invoices**

The database shall keep a list of invoices. There shall be invoices for both orders and inventory purchases. Invoices shall have a unique id and have a record of the order or purchase and whether the invoice is for an order or a purchase.

**ER Diagram:**

****

**Basic Relational Schema:**

Customer(email, name, password)

Orders(orderID, address, date, cost, invoiceID, email)

Products(productID, name, price, quantityInStock)

InventoryPurchases(purchaseID, date, invoiceID)

Invoices(invoiceID, type)

OrderLines(orderID, productID, quantity, price)

Address(email, address)

ProductPurchases(purchaseID, productID, quantity)

**-------------------------------------------------------------------------------------------------------------------**

**Integrity constraints:**

(state in plain English one type of ICs in the table)

| IC Name | IC Type | English Statement |
| --- | --- | --- |
| custC1 | Key | Email must be in the format of [account]@[domain] and must be unique |
| ordIC5 | Foreign Key | Order email references customer table |
| ordIC3 | 1-Attribute | Order cost should be greater than $0 |
| lineIC6 | 2-Attribute, 1-row | Orderline quantity must be less than or equal to product quantity in stock |

**Customer ICs:**

custIC1: Customer email must be in the format [account]@[domain] and must be unique (primary key)

custIC2: Customer name must have a name

custIC3: Customer password should be at least 8 characters long

**Orders ICs:**

ordIC1: Order ID is primary key

ordIC2: Order address must be not null

ordIC3: Order cost should be greater than $0

ordIC4: Order invoiceID must be foreign key referencing Invoice Table

ordIC5: Order email must be foreign key referencing Customer Table

**Products ICs:**

prodIC1: Product ID is primary key

prodIC2: Product cost should be greater than $0

prodIC3: Product quantity in stock should not be negative

prodIC4: Product name should include only alphanumeric characters and spaces and not null

**InventoryPurchases ICs:**

invpurIC1: Inventory purchase ID is primary key

invpurIC2: Inventory invoiceID is foreign key referencing Invoice Table

**Invoices ICs:**

invoIC1: Invoice ID is primary key

invoIC2: Invoice type must be “ORDER” or “INVPURCH”

**OrderLines ICs:**

lineIC1: Orderline quantity must be greater than 0

lineIC2: Orderline price must be greater than $0

lineIC3: Orderline productID is foreign key referencing Product Table

lineIC4: Orderline orderID is foreign key referencing Order Table

lineIC5: orderID and productID form composite key

lineIC6: Orderline quantity must be less than or equal to Product[productID].quantityInStock

**Address ICs:**

addIC1: Addresses shall be not null

addIC2: Address email is foreign key referencing Customers Table

**ProductPurchases ICs:**

prodpurIC1: Product Purchase purchaseID is foreign key referencing InventoryPurchase Table

prodpurIC2: Product Purchase productID is foreign key referencing Product Table

prodpurIC3: purchaseID and productID are composite key