URL: https://web.engr.oregonstate.edu/~angusjo/index
Team Members in Group 117:

Joel Angus

Daniel Nikolov

Feedback by the peer reviewer:
Match names on the schema and SQL document
Inconsistent capitalization of attributes
Add more update and delete queries to SQL

Actions based on the feedback:
Fixed inconsistencies in names on the SQL and schema to match each other
Fixed ERD to make customers to order header a mandatory relationship
Added many update, select, and delete queries to the DML sql

Upgrades to the Draft version:
Fixed capitalization of attributes on this document
Fixed ERD to make customers to order header a mandatory relationship
Made the titles more descriptive and better relate to the ERD

Overview

Title: Retail and Online Store Transaction Management System and Inventory Tracker

A retail company with $20 million in annual sales requires a database-driven website to manage 20,000 transactions yearly. They are in need of a system which tracks their customers, employees, products, orders, and the allocation of products for their orders from a variety of inventory locations. The system will record orders (sales and returns), connecting products to customers for order processing. With a focus on tracking inventory, processing orders, and generating sales reports, the website ensures real-time insights into product availability and customer preferences.

Database Outline, In Words

We will have several tables to track sales data, such as customers, products, and transactions (returns, sales, etc.). However, in the same database, we will also retain other information concerning the store, such as tables regarding employees, expenses, etc.

The database also tracks online transactions and allows for the simultaneous purchase of several products, with varying quantities, and allows for product quantity to be fulfilled by multiple locations (warehouses.)

When a user places an order, their customer information, the aiding employees information, and general order information are stored in the Order Header.

- An Order Header has a one to many relationship with Order Items
    - Order items represent a single product within the user's full order and stores how many of the item they requested, along with the product_id of the good.

        - An Order Item is also a foreign key within the Order Allocation, which stores how many of an Order Item can be fulfilled by an inventory location (warehouse)

        - Ultimately, the product inventory foreign key in Order Allocation denotes the specific inventory location where the allocated quantity is found
            - The inventory table has 2 primary keys
                - One key represents the inventory location itself
                - The other represents the desired product which is being requested by the customer
            - In addition, the inventory table stores the the onhand quantity of the product at the specific location and the name of the location

Customers: records the details of Customers we do business with

- cust_id: int, auto_increment, unique, NOT NULL, PK
- cust_name: char, NOTNULL
- cust_email: char, NOT NULL
- cust_phone_num: char, NOTNULL
- cust_date_joined: date, NOTNULL
- Relationship: M:N relationship between Customers and Order Header as customers can place many transactions, and a transaction can be placed by many different customers.

Employees: Records the employees that make the transaction to sell product, employees do not appear in online orders

- emp_id: int, auto_increment, unique, NOT NULL, PK
- hourly_wage: int, not NULL, FK
- emp_name: char, not NULL
- emp_email: char, not NULL
- emp_phone_num: char, not NULL
- date_hired: date, not NULL
- Relationship: M:N relationship between Employees and Order Header as employees can confirm many transactions, a transaction can be conformed by many transactions

Products: Keeps track of every item which can be sold to the customers

- product_id: int, auto_increment, unique, NOT NULL, PK
- location: int, not NULL, FK
- recieve_date: date, not NULL
- is_clearence: bool, not NULL
- product_type: char, not NULL
- retail_price: int, not NULL
- Relationship: 1:m with order_item_id each product can be a part of many different order items from different orders

Order Header: Initializes order and basic details related to transaction

- order_header_id int, auto_increment, unique, NOT NULL, PK
- customer_id: int, NOT NULL, FK
- employee_id: int, NOT NULL, FK
- order_date: date, NOT NULL
- is_return: bool, NOT NULL
- is_online_order: bool, NOT NULL
  Relationships:
  1:m with order_item_id one order consists of many order_items which represent a single product in the user's order. This allows them to purchase more than one product listing.

Order Item: Keeps track of how many of a certain item are being ordered and which item is ordered, there are usually several order items per order

- order_item_id int auto_increment, unique, NOT NULL, PK
- product_id int, NOT NULL, FK1
- order_header_id int NOT NULL, int, FK2
- quantity: int , NOT NULL

  Relationships:
- 1:m with order_allocation_id one item order uses many order_allocations from many warehouses to fulfill the user's requested quantity. The sum of all allocated quantities will match requested quantity from the customer
- m:m with product_inventory_id. Many different orders of a product are sourced from many different inventories (locations)
- m:1 with product_id because a product can be a part of many different order_item_id instances

Order Allocation: Intermediate step which connects the order item and inventory location. Allocates quantity of an order item which is sourced from a specific inventory id.

- order_allocation_id int auto_increment, unique, NOT NULL, PK
- order_item_id int NOT NULL, int, FK1
- product_inventory_id, int, NOT NULL, FK2
- allocated_quantity int, NOT NULL

Relationships:

- m:1 with order_item_id because many allocations of inventory are made for a single order of an item

- m:1 with product_inventory_id because many inventories are used to fulfill a single order allocation

Inventory: Keeps track of the onhand quantity of products stored at a given location (warehouse, store, factory, etc.)

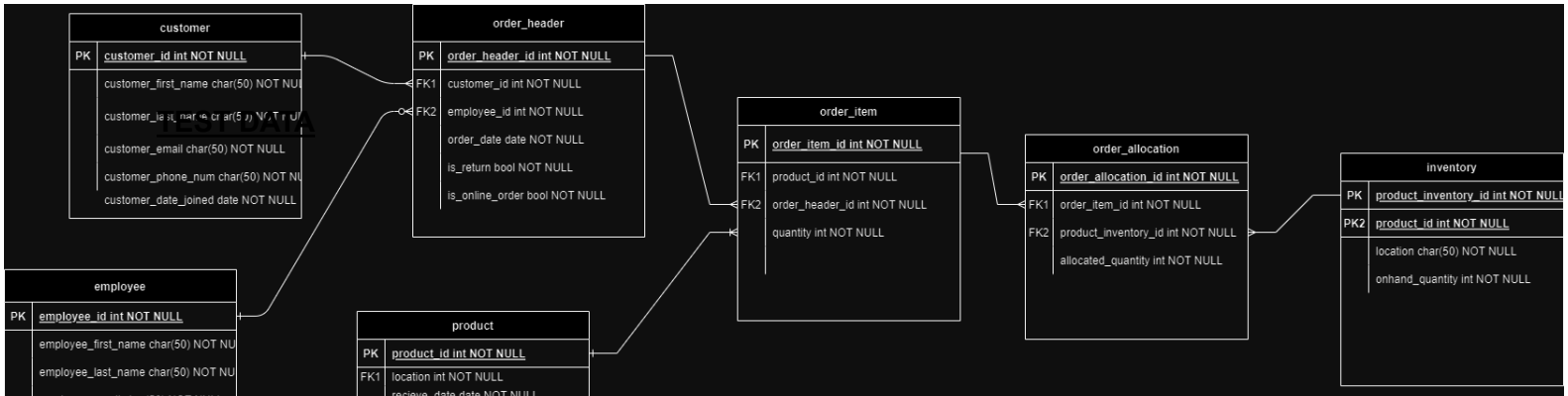
- product_inventory_id, int, auto_increment, unique, NOT NULL, PK
- product_id int, unique, NOT NULL, PK
- location: char(50), NOT NULL
- onhand_quantity: int , NOT NULL

Relationships:

- m:1 with order_allocation_id because many different orders allocated quantities from a single inventory
- m:m with order_item_id. Many different orders of a product are sourced from many different inventories(locations)

Entity Relationship Diagram

https://drive.google.com/file/d/1faHRdCRGbuX941NcLdMg6e8YQ-1OofR_/view?usp=sharing



order_allocation:

	order_allocation_id	order_item_id	product_inventory_id	allocated_quantity
Edit Copy Delete	1	1	1	9
Edit Copy Delete	2	1	2	1
Edit Copy Delete	3	1	3	2
Edit Copy Delete	4	2	2	1
Edit Copy Delete	5	2	3	1
Edit Copy Delete	6	3	2	1

customer

customer_id	customer_first_name	customer_last_name	customer_email	customer_phone_num	customer_date_joined
1	Ananya	Jaiswal	ajaiswal@hello.com	503-123-9876	2008-04-10
2	Michael	Fern	michaelf@hello.com	123-321-4564	2015-07-19
3	Abdul	Rehman	rehman@hello.com	987-012-7684	2018-02-27

employee:

employee_id	employee_first_name	employee_last_name	employee_date_hired	employee_email	employee_phone_num
1	Alex	Montgomery	2008-04-10	amont@hello.com	123-456-7892
2	Zach	Allen	2015-07-19	zallen@hello.com	123-012-7684
3	Dylan	Brehm	2018-02-27	dbrehm@hello.com	450-012-7684

products:

	product_inventory_id	product_id	location	onhand_quantity
Edit Copy Delete	1	1	Beaverton	9
Edit Copy Delete	2	1	Portland	3
Edit Copy Delete	3	2	Beaverton	20

order_header:

order_header_id	customer_id	employee_id	order_date	is_return	is_online_order
1	1	1	2020-11-19	1	0
2	2	2	2020-12-05	0	0
3	3	3	2021-01-01	0	1

order_item:

order_item_id	product_id	order_header_id	quantity
1	1	1	10
2	2	1	2
3	1	2	1
4	2	2	1
5	1	3	1

inventory:

product_inventory_id	product_id	location	onhand_quantity
1	1	Beaverton	9
2	1	Portland	3
3	2	Beaverton	20

Diagram View