

Final Project Report

Running on port 8604

Overview

The application is an e-commerce platform that tracks the sale of products between customers and sellers and manages information about the products and where they are stored. The system allows users to query information about sales, track the location of where sellers have products stored, and record information about completed sales to track revenue.

Entity and Relationships

Entities:

- Seller (sid, name, country)
- Manufacturer (mid, name)
- Product (serial_num, name, category, description, price)
- Warehouse (name, capacity, items_stored, address)
- Customer (cid, name, surname, address)
- Time(timestamp)
- Inventory (iid, quantity)

Relationship Set:

- **Transaction** - A transaction is a relationship between a customer, seller, product, and time. All 4 entities are required to record a transaction.
- **Producer** - A manufacturer produces a product. A product is produced by exactly one manufacturer.
- **Stored_in** - When a product is sold, it is shipped from a warehouse and this relationship represents where the product was originally stored.
- **Stock** - A product is in stock of an inventory. Each product must be stocked in at least one inventory. An inventory may contain no stock of a product.
- **Manage** - A seller manages the inventory. A seller is not required to have any inventory, but each inventory must be managed by exactly one seller.

Business Rules

A customer and a seller transact a product at a specific timestamp. We record every time they transact and the specific price they transact the product at.

Each product has a unique serial number. Each product is produced by only one manufacturer that contains a unique id. A product is only recorded in our database if it has been sold.

A seller manages an inventory that has an iid and tracks the product quantity. The inventory is managed by exactly one unique seller.

Sellers have products stored in various warehouses. When the item is sold, a record of where the product was stored is also recorded. The warehouses have a distinct name, to uniquely identify where the item was sold from.

Data Aggregation

The data in the csv files was created with several sources. The names were generated with random name generators online. For the products, we searched on Amazon by category and populated the csv files with information including product name, description, price, and manufacturer. For the data_times, we wrote a simple python script to generate this information and copied it into the csv file. For the sellers and warehouses, we retrieved a list of popular Amazon sellers and logistics providers from online, respectively.

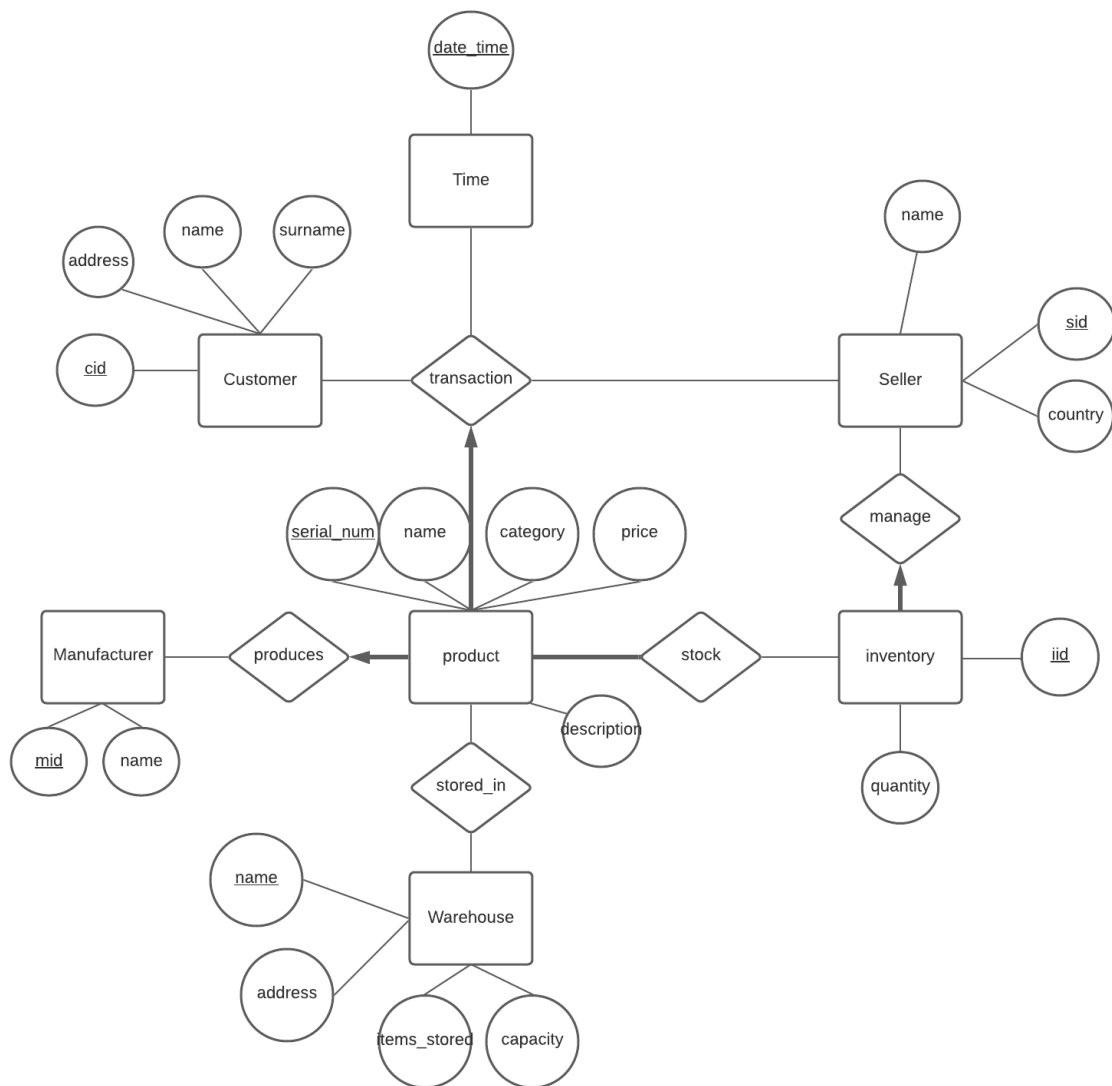
Data Loading Procedure

To load the data into the database, we created a shell script named load.sh. The shell script executes the schema.sql script to create all of the tables, and then loads the respective csv files into the tables. When executing the shell script, the user is prompted for their NYU id which will be used to add the data to the users databases.

User Interactions

- Select a range of dates to see what the total sales for that range were by seller. The user can also filter by specific seller
- Select a product to see the quantity of stock for the product for each seller.
- Select a manufacturer to see which products they produce.
- Select a customer ID to see all of the purchases by that customer.
- The best selling products, ranked by number of items sold.
- Seller's product storage storage by warehouse.
- Total sales by country.

ER Diagram



SQL Schema

```
drop table if exists Customers cascade;
drop table if exists Manufacturers cascade;
drop table if exists Sellers cascade;
drop table if exists Warehouses cascade;
drop table if exists Inventory_manage cascade;
drop table if exists Time cascade;
drop table if exists Product_produces_transaction cascade;
drop table if exists Stock cascade;
drop table if exists Stored_in cascade;
```

```
create table Manufacturers(
    mid integer primary key,
    name varchar (128) not null
);
```

```
create table Warehouses(
    name varchar(128) primary key,
    items_stored integer,
    capacity integer,
    address varchar(256)
);
```

```
create table Sellers(
    sid integer primary key,
    name varchar(64),
    country varchar(64)
);
```

```
create table Customers(
    cid integer primary key,
    name varchar(64),
    surname varchar(64),
    address varchar(256)
);
```

```
create table Time(
    date_time timestamp primary key
);
```

```
create table Product_produces_transaction(
    serial_num char(32) primary key,
    cid integer not null,
    sid integer not null,
    date_time timestamp not null,
    price decimal not null,
    name varchar(128) not null,
    category varchar (32),
    manufacturer integer not null,
```

```
description varchar(512),
foreign key (manufacturer) references Manufacturers(mid),
foreign key (cid) references Customers(cid),
foreign key (sid) references Sellers(sid),
foreign key (date_time) references Time(date_time)
);

create table Inventory_manage(
iid integer primary key,
manager integer not null,
quantity integer,
foreign key (manager) references Sellers(sid)
);

create table Stock(
serial_num char(32),
iid integer,
primary key(serial_num, iid),
foreign key (serial_num) references Product_produces_transaction(serial_num),
foreign key (iid) references Inventory_manage(iid)
);

create table Stored_in(
serial_num char(32),
name varchar(64),
primary key(serial_num, name),
foreign key (serial_num) references Product_produces_transaction(serial_num),
foreign key (name) references Warehouses(name)
);
```