NittanyMarket Relational Schema

Please find the schemas (totally 12 tables) which you will use for the NittanyMarket prototype below. You should use your judgement for assigning proper datatypes and integrity constraints (recommended primary keys in red). You may also augment the schemas if you feel that helps with the implementation of new features. In that case, you will also need to prepare for the corresponding data for testing and final demonstration.

Users(email, password)

- All the users in the system and their passwords.

Buyers(email, first_name, last_name, gender, age, home_address_id, billing_address_id)

- Buyers is a subset of the Users.

Credit_Cards(credit_card_num, card_code, expire_month, expire_year, card type, Owner_email)

- Credit cards stored in the system.
- A credit card is owned by one buyer.
- A buyer may have one or multiple credit cards stored in the system.

Address(address_ID, zipcode, street_num, street_name)

- The Home address and Billing address in Buyers and Local_Vendors tables store address_ID, existing in this Address table (i.e., foreign key).

Zipcode_Info(zipcode, city, state_id, population, density, county_name, timezone)

- This table stores information about a zipcode area.
- Only city and state-id are really useful in this application.

Sellers(email, routing_number, account_number, balance)

- Sellers consist of a subset of buyers (i.e., student users) and all local vendors.

Local_Vendors(Email, Business_Name, Business_Address_ID, Customer_Service_Number)

- Local_vendors is a subclass of Sellers, i.e., all local vendors are sellers.

Categories(parent_category, category_name)

- This table stores the links between a category and its parent category in the category hierarchy.
- A category has only one parent category, but its parent category may have multiple children.

Product_Listings(Seller_Email, Listing_ID, Category, Title, Product_Name, Product_Description, Price, Quantity)

- Each seller has her/his own counter of Listing_ID, i.e., Listing_ID needs to be unique under the same seller.
- After a product_listing is sold out, it's still kept in the table but its quantity is 0.

Orders(Transaction_ID, Seller_Email, Listing_ID, Buyer_Email, Date, Quantity, Payment)

- Transaction_ID uniquely identifies an order.
- The product listings in the Orders table, which are identified by (Seller_Email, Listing_ID), are a subset of those in Product_Listings table.

Reviews(Buyer_Email, Seller_Email, Listing_ID, Review_Desc)

- A buyer needs to order a product listing, identified by (Seller_Email, Listing_ID), in order to make a review on the listed product.

Rating(Buyer_Email, Seller_Email, Rating, Rating_Desc)

- A buyer needs to buy a product listing from a seller in order to rate the seller.