Student Name: Jiaxin Li
Student ID: 19683688

```
CREATE SCHEMA IF NOT EXISTS cs220p_hw;

--Relationships
DROP TABLE IF EXISTS cs220p_hw.Ratings;

--Entities
DROP TYPE IF EXISTS cs220P_hw.phone_kind;
DROP TABLE IF EXISTS cs220P_hw.Users;
DROP TABLE IF EXISTS cs220P_hw.Buyers;
DROP TABLE IF EXISTS cs220P_hw.Sellers;
DROP TABLE IF EXISTS cs220P_hw.Ad;
DROP TYPE IF EXISTS cs220P_hw.Picture_formats;
DROP TABLE IF EXISTS cs220P_hw.Picture;
DROP TABLE IF EXISTS cs220P_hw.Item;
DROP TABLE IF EXISTS cs220P_hw.Good;
DROP TYPE IF EXISTS cs220P_hw.Frequencies;
DROP TABLE IF EXISTS cs220P_hw.Service;
```

SQL DDLs for Entities and their supporting tables

```
CREATE TABLE cs220p_hw.Users (
        user_id text,
        email text NOT NULL,
        adress_street text,
        adress_city text,
        adress_state text,
        adress_zip integer,
        joined_date date NOT NULL,
        first_name text,
        last_name text NOT NULL,
        PRIMARY KEY(user_id)
);
CREATE TYPE phone_kind AS ENUM('Mobile', 'Home','Work');
--Multi-valued attribute “phone” for Users
CREATE TABLE cs220p_hw.Users_phone (
        user_id text,
        phone_number integer,
        kind phone_kind,
        PRIMARY KEY(user_id, phone_number),
        FOREIGN KEY(user_id) REFERENCES cs220p_hw.Users ON DELETE CASCADE
);
```

```
--Multi-valued attribute “categories” for Users
CREATE TABLE cs220p_hw.Users_categories (
        user_id text,
        interest text,
        PRIMARY KEY(user_id),
        FOREIGN KEY(user_id) REFERENCES cs220p_hw.Users ON DELETE CASCADE
);

CREATE TABLE cs220p_hw.Seller(
        user_id text,
        website text,
        PRIMARY KEY(user_id),
        FOREIGN KEY(user_id) REFERENCES cs220p_hw.Users ON DELETE CASCADE
);

--Rating attribute for seller is created after Rating table in the second part relation table

CREATE TABLE cs220p_hw.Buyer(
        user_id text,
        PRIMARY KEY(user_id),
        FOREIGN KEY(user_id) REFERENCES cs220p_hw.Users ON DELETE CASCADE
);

CREATE TABLE cs220p_hw.Item(
        item_id text,
        user_id text,
        list_date date NOT NULL,
        name text NOT NULL,
        price decimal(8,3) NOT NULL,
        category text NOT NULL,
        description text,
        purchase_date date,
        PRIMARY KEY(item_id),
        FOREIGN KEY(user_id) REFERENCES cs220p_hw.Seller ON DELETE CASCADE
);

CREATE TABLE cs220p_hw.Good(
        item_id text,
        FOREIGN KEY(item_id) REFERENCES cs220p_hw.Item ON DELETE CASCADE
);
```

```
CREATE TYPE Service_frequency AS ENUM('once', 'daily','weekly', 'monthly', 'quarter', 'yearly');

CREATE TABLE cs220p_hw.Service(
        item_id text,
        frequency Service_frequency NOT NULL,
        PRIMARY KEY(item_id),
        FOREIGN KEY(item_id) REFERENCES cs220p_hw.Item ON DELETE CASCADE
);

CREATE TYPE Pic_type AS ENUM('png', 'jpeg', 'mp4', 'eps', 'gif');

CREATE TABLE cs220p_hw.Picture(
        pic_num integer NOT NULL,
        item_id text NOT NULL, --view of ‘Item’, total participation.
        format Pic_type NOT NULL,
        url text NOT NULL,
        PRIMARY KEY(pic_num),
        FOREIGN KEY(item_id) REFERENCES cs220p_hw.Item ON DELETE CASCADE
);

CREATE TYPE ad_plan_level AS ENUM('bronze', 'silver', 'gold');

CREATE TABLE cs220p_hw.Ad(
        ad_id text,
        user_id text, --placed by ‘Users’, total participation.
        item_id text, --about ‘item’, total participation.
        pic_num integer, --use ‘picture’, total participation,
        plan ad_plan_level NOT NULL,
        content text,
        placed_date date NOT NULL,
        PRIMARY KEY(ad_id),
        FOREIGN KEY(user_id) REFERENCES cs220p_hw.Users ON DELETE CASCADE,
        FOREIGN KEY(item_id) REFERENCES cs220p_hw.Item ON DELETE CASCADE,
        FOREIGN KEY(pic_num) REFERENCES cs220p_hw.Picture ON DELETE CASCADE
);
```

SQL DDLs for Relationships

```
CREATE TYPE Rating_level AS ENUM('horrible', 'bad', 'average', 'good', 'very good', 'excellent');

CREATE TABLE cs220p_hw.Ratings(
        buyer_id text NOT NULL, -- not sure here, question
        seller_id text NOT NULL,
        quality Rating_level,
        pricing Rating_level,
        delivery Rating_level,
        rating_date date NOT NULL,
        PRIMARY KEY(buyer_id, seller_id),
        FOREIGN KEY(buyer_id) REFERENCES cs220p_hw.Buyer,
        FOREIGN KEY(seller_id) REFERENCES cs220p_hw.Seller
);

CREATE VIEW cs220p_hw.Sellerview(user_id,website,rating)
        AS SELECT R.seller_id,S.website,
        ((((SELECT COUNT(seller_id) FROM cs220p_hw.Ratings WHERE quality='bad')+
         (SELECT COUNT(seller_id) FROM cs220p_hw.Ratings WHERE quality='average')*2+
         (SELECT COUNT(seller_id) FROM cs220p_hw.Ratings WHERE quality='good')*3+
         (SELECT COUNT(seller_id) FROM cs220p_hw.Ratings WHERE quality='very good')*4+
         (SELECT COUNT(seller_id) FROM cs220p_hw.Ratings WHERE quality ='excellent')*5)/
        COUNT(seller_id)
         )+
         (((SELECT COUNT(seller_id) FROM cs220p_hw.Ratings WHERE pricing='bad')+
         (SELECT COUNT(seller_id) FROM cs220p_hw.Ratings WHERE pricing='average')*2+
         (SELECT COUNT(seller_id) FROM cs220p_hw.Ratings WHERE pricing='good')*3+
         (SELECT COUNT(seller_id) FROM cs220p_hw.Ratings WHERE pricing='very good')*4+
         (SELECT COUNT(seller_id) FROM cs220p_hw.Ratings WHERE pricing='excellent')*5)/
        COUNT(seller_id)
         )+
         (((SELECT COUNT(seller_id) FROM cs220p_hw.Ratings WHERE delivery='bad')+
         (SELECT COUNT(seller_id) FROM cs220p_hw.Ratings WHERE delivery='average')*2+
         (SELECT COUNT(seller_id) FROM cs220p_hw.Ratings WHERE delivery='good')*3+
         (SELECT COUNT(seller_id) FROM cs220p_hw.Ratings WHERE delivery='very good')*4+
         (SELECT COUNT(seller_id) FROM cs220p_hw.Ratings WHERE delivery='excellent')*5)/
        COUNT(seller_id)
         ))/3 FROM cs220p_hw.Ratings R, cs220p_hw.Seller S
        GROUP BY R.Seller_id, S.User_id;
```