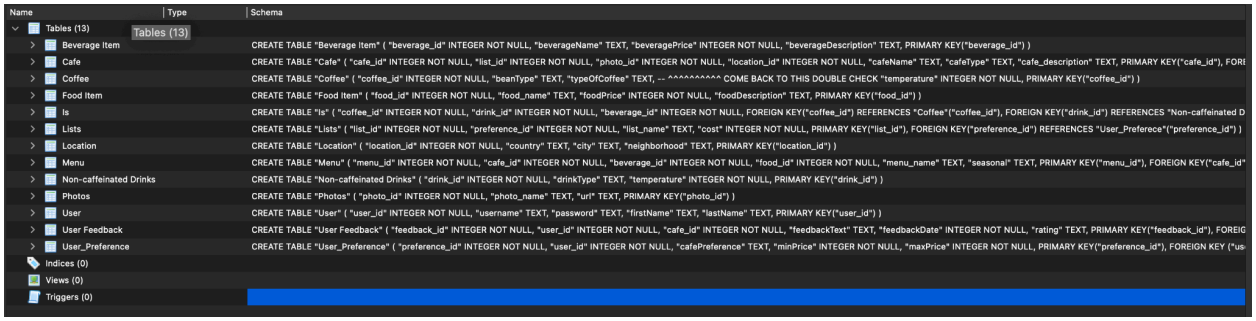
(10 pts) Create a set of SQL data definition statements for the above model and realize that schema in SQLite3 by executing the script from the SQLite3, the console or Node. You can use DB Browser to generate these statements. Show that the tables were created and conform to the constraints through screen shots or other means.



Tables:

User
- user_id: NUMBER (PK)
- username: TEXT
- password: TEXT
- firstName: TEXT
- lastName: TEXT

```
CREATE TABLE "User" (
"user_id" INTEGER NOT NULL,
"username" TEXT,
"password" TEXT,
"firstName" TEXT,
"lastName" TEXT,
PRIMARY KEY("user_id")
);
```

User Preference
- preference_id: NUMBER (PK)
- user_id: NUMBER (FK)
- cafePreference: TEXT
- minPrice: NUMBER
- maxPrice: NUMBER

```
CREATE TABLE "User_Preference" (
"preference_id" INTEGER NOT NULL,
"user_id" INTEGER NOT NULL,
"cafePreference" TEXT,
"minPrice" INTEGER NOT NULL,
"maxPrice" INTEGER NOT NULL,
PRIMARY KEY("preference_id"),
FOREIGN KEY ("user_id") REFERENCES "User" ("user_id")
);
```

User Feedback
- feedback_id: NUMBER (PK)
- user_id: NUMBER (FK)
- cafe_id: NUMBER (FK)
- feedbackText: TEXT
- feedbackDate: NUMBER
- rating: TEXT

```
CREATE TABLE "User Feedback" (
"feedback_id" INTEGER NOT NULL,
"user_id" INTEGER NOT NULL,
"cafe_id" INTEGER NOT NULL,
"feedbackText" TEXT,
"feedbackDate" INTEGER NOT NULL,
"rating" TEXT,
PRIMARY KEY("feedback_id"),
FOREIGN KEY("user_id") REFERENCES "User"("user_id"),
FOREIGN KEY("cafe_id") REFERENCES "Cafe"("cafe_id")
);
```

Cafe
- cafe_id: NUMBER (PK)
- list_id: NUMBER (FK)
- photo_id: NUMBER (FK)
- location_id: NUMBER (FK)
- cafeName: TEXT
- cafeType: TEXT
- cafeDescription: TEXT

```
CREATE TABLE "Cafe" (
"cafe_id" INTEGER NOT NULL,
"list_id" INTEGER NOT NULL,
"photo_id" INTEGER NOT NULL,
"location_id" INTEGER NOT NULL,
"cafeName" TEXT,
"cafeType" TEXT,
"cafe_description" TEXT,
PRIMARY KEY("cafe_id"),
FOREIGN KEY("list_id") REFERENCES "Liste"("lists_id"),
FOREIGN KEY("photo_id") REFERENCES "Photos"("photo_id"),
FOREIGN KEY("location_id") REFERENCES "Location"("location_id")
);
```

Lists
- list_id: NUMBER (PK)
- preference_id: NUMBER (FK)
- listName: TEXT
- cost: NUMBER

```
CREATE TABLE "Lists" (
"list_id" INTEGER NOT NULL,
"preference_id" INTEGER NOT NULL,
"list_name" TEXT,
"cost" INTEGER NOT NULL,
PRIMARY KEY("list_id"),
FOREIGN KEY("preference_id") REFERENCES "User_Preferece"("preference_id")
);
```

Photos
- photo_id: NUMBER (PK)
- photoName: TEXT
- url: TEXT

```
CREATE TABLE "Photos" (
"photo_id" INTEGER NOT NULL,
"photo_name" TEXT,
"url" TEXT,
PRIMARY KEY("photo_id")
);
```

Location
- location_id: NUMBER (PK)
- country: TEXT
- city: TEXT
- neighborhood: TEXT

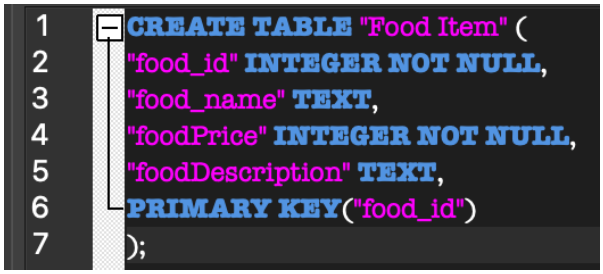
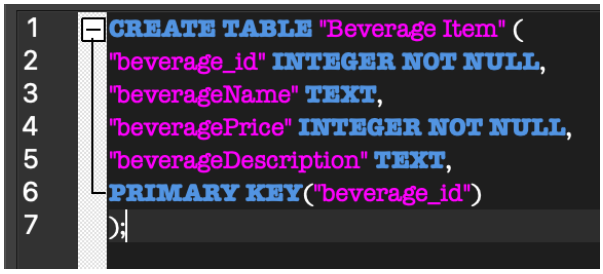
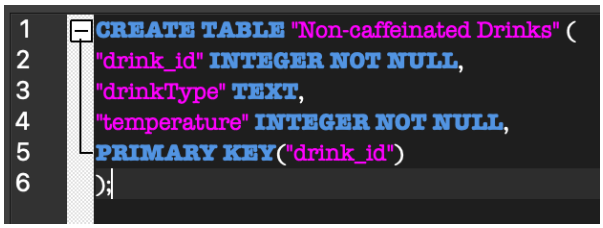
```
CREATE TABLE "Location" (
"location_id" INTEGER NOT NULL,
"country" TEXT,
"city" TEXT,
"neighborhood" TEXT,
PRIMARY KEY("location_id")
);
```

Coffee
- coffee_id: NUMBER (PK)
- beanType: TEXT
- typeOfCoffee: [cappucino, americano, machiato, latte, mocha]
- temperature: NUMBER

```
CREATE TABLE "Coffee" (
"coffee_id" INTEGER NOT NULL,
"beanType" TEXT,
"typeOfCoffee" TEXT,
-- ^^^^^^^^^^ COME BACK TO THIS DOUBLE CHECK
"temperature" INTEGER NOT NULL,
PRIMARY KEY("coffee_id")
);
```

Menu
- menu_id: NUMBER (PK)
- cafe_id: NUMBER (FK)
- beverage_id: NUMBER (FK)
- food_id: NUMBER (FK)
- menuName: TEXT
- seasonal: TEXT

```
CREATE TABLE "Menu" (
"menu_id" INTEGER NOT NULL,
"cafe_id" INTEGER NOT NULL,
"beverage_id" INTEGER NOT NULL,
"food_id" INTEGER NOT NULL,
"menu_name" TEXT,
"seasonal" TEXT,
PRIMARY KEY("menu_id"),
FOREIGN KEY("cafe_id") REFERENCES "Cafe"("cafe_id"),
FOREIGN KEY("beverage_id") REFERENCES "Beverage Item"("beverage_id"),
FOREIGN KEY("food_id") REFERENCES "Food Item"("food_id")
);
```

Food Item - food_id: NUMBER (PK) - foodName: TEXT - foodPrice: NUMBER - foodDescription: TEXT	
Beverage Item - beverage_id: NUMBER (PK) - beverageName: TEXT - beveragePrice: NUMBER - BeverageDescription: TEXT	
Non-caffeinated Drinks - drink_id: NUMBER (PK) - drinkType: TEXT - temperature: NUMBER	
Is - coffee_id: NUMBER (FK) - drink_id: NUMBER (FK) - beverage_id: NUMBER (FK)