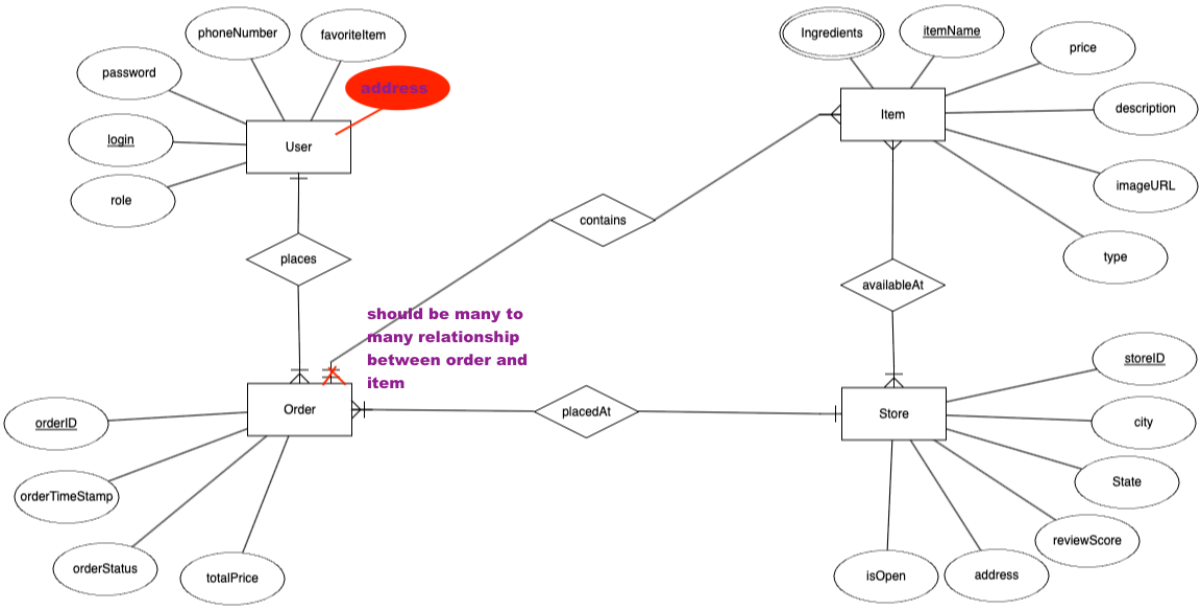
Kyle Lin 862314961 klin112@ucr.edu
Vansh Nagpal 862228203 vnagp002@ucr.edu

CS 166 Project Phase 2
SQL Relational Schema

In this document, we translate the given ER diagram into a SQL schema. We discuss assumptions and caveats related to the translation. We also briefly discuss our individual contributions to the project in this phase.

1. Translation of ER diagram to SQL

ER Diagram (provided by Prof. Salloum and her TAs):



SQL implementation:

```
DROP TABLE IF EXISTS availableAt;
DROP TABLE IF EXISTS contains;
DROP TABLE IF EXISTS Order_;
DROP TABLE IF EXISTS Item;
DROP TABLE IF EXISTS Store;
DROP TABLE IF EXISTS User_;

CREATE TABLE User_ (
   role CHAR(8) NOT NULL,
   login CHAR(16) NOT NULL,
   password CHAR(16) NOT NULL,
   phoneNumber CHAR(10) NOT NULL,
   favoriteItem CHAR(16),
   address CHAR(32) NOT NULL,
   PRIMARY KEY(login)
);

CREATE TABLE Store (
   storeID INTEGER NOT NULL,
   city CHAR(32) NOT NULL,
   state CHAR(2) NOT NULL,
   reviewScore REAL,
   address CHAR(32) NOT NULL,
   isOpen BOOLEAN NOT NULL,
   PRIMARY KEY (storeID)
);

CREATE TABLE Order_ (
   orderID INTEGER NOT NULL,
   orderTimeStamp TIMESTAMP NOT NULL,
   orderStatus CHAR(16) NOT NULL,
   totalPrice INTEGER NOT NULL,
   login CHAR(16) NOT NULL,
   storeID INTEGER NOT NULL,
```

```
   PRIMARY KEY(orderID),
   FOREIGN KEY(login) REFERENCES User_(login) ON DELETE NO ACTION,
   FOREIGN KEY(storeID) REFERENCES Store(storeID) ON DELETE NO ACTION
);

CREATE TABLE Item (
   ingredients CHAR(1024),
   itemName CHAR(32) NOT NULL,
   price INTEGER NOT NULL,
   description CHAR(64),
   imageURL CHAR(128),
   type CHAR(32) NOT NULL,
   PRIMARY KEY(itemName)
);

CREATE TABLE contains (
   itemName CHAR(32) NOT NULL,
   orderID INTEGER NOT NULL,
   FOREIGN KEY(itemName) REFERENCES Item(itemName) ON DELETE NO ACTION,
   FOREIGN KEY(orderID) REFERENCES Order_(orderID) ON DELETE NO ACTION
);

CREATE TABLE availableAt (
   itemName CHAR(32) NOT NULL,
   storeID INTEGER NOT NULL,
   FOREIGN KEY(itemName) REFERENCES Item(itemName) ON DELETE NO ACTION,
   FOREIGN KEY(storeID) REFERENCES Store(storeID) ON DELETE NO ACTION
);
```

Since the ER diagram does not mark which attributes are optional, we determine whether or not an attribute is nullable based on the specification in Project Phase 1 Requirements Document. Specifically, attributes marked “(required)” in the document become NON NULL in SQL, and all other attributes are nullable.

Additionally, we implement Item.ingredients as a comma separated string instead of a list of strings, since list-typed attributes are not supported in all versions of SQL. Thus, it has the type “CHAR[1024]”.

We must append underscores to certain identifiers. For example, the “Order” relation becomes “Order_”, because “ORDER” is a reserved word in SQL.

2. Teamwork Distribution

Kyle and Vansh split the work in this phase roughly equally.