CSE305 Assignment 2 -- ER Mapping

***Group Members**: Ray Chen, Kristy Tan, William Xiang*

***Prof**: Praveen Tripathi (**TA**: Jarin Moon)*

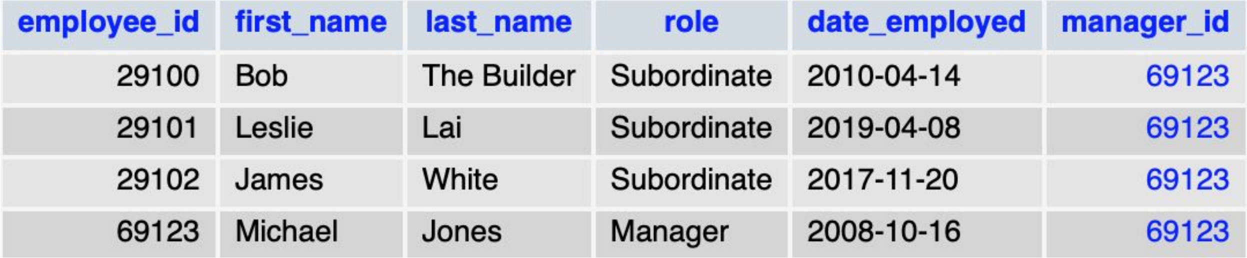
SQL Statements

```
CREATE TABLE item_stock (
     stock_id INT AUTO_INCREMENT,
     name VARCHAR(50),
     price DECIMAL(19,4),
     quantity INT,
     PRIMARY KEY (stock_id)
);

CREATE TABLE item (
     item_id INT AUTO_INCREMENT,
     stock_id INT NOT NULL,
     was_bought BOOL,
     PRIMARY KEY (item_id),
     FOREIGN KEY (stock_id)
          REFERENCES item_stock(stock_id)
          ON DELETE NO ACTION
          ON UPDATE CASCADE
);

CREATE TABLE customer (
     customer_id INT AUTO_INCREMENT,
     first_name VARCHAR(50),
     last_name VARCHAR(50),
     address VARCHAR(50),
     email VARCHAR(255),
     PRIMARY KEY (customer_id)
);

CREATE TABLE ordr (
     ordr_id INT AUTO_INCREMENT,
     shipment_id INT,
     customer_id INT NOT NULL,
     amount DECIMAL(19,4),
     card_number VARCHAR(19),
     payment_date DATE,
     card_exp_month CHAR(2),
     card_exp_year CHAR(2),
     payment_type ENUM('credit', 'paypal', 'debit'),
     PRIMARY KEY (ordr_id),
     FOREIGN KEY (customer_id)
          REFERENCES customer(customer_id)
          ON DELETE CASCADE
          ON UPDATE CASCADE
);

CREATE TABLE shipment (
     shipment_id INT AUTO_INCREMENT,
     ordr_id INT NOT NULL,
     priority CHAR(20),
     date_shipped DATE,
     date_received DATE,
     carrier VARCHAR(20),
     shipment_address VARCHAR(100),
     PRIMARY KEY (shipment_id),
     FOREIGN KEY (ordr_id)
          REFERENCES ordr(ordr_id)
          ON DELETE NO ACTION
          ON UPDATE CASCADE
);

ALTER TABLE ordr
ADD CONSTRAINT
FOREIGN KEY (shipment_id)
     REFERENCES shipment(shipment_id)
     ON DELETE NO ACTION
     ON UPDATE CASCADE;

CREATE TABLE review (
     rating ENUM('1', '2', '3', '4', '5'),
     review_text TEXT,
     stock_id INT,
     customer_id INT,
     FOREIGN KEY (stock_id)
          REFERENCES item_stock(stock_id)
          ON DELETE CASCADE
          ON UPDATE CASCADE,
     FOREIGN KEY (customer_id)
          REFERENCES customer(customer_id)
          ON DELETE CASCADE
          ON UPDATE CASCADE
```

```
);

CREATE TABLE shopping_cart (
     quantity INT,
     date_added DATE,
     stock_id INT,
     customer_id INT,
     FOREIGN KEY (stock_id)
          REFERENCES item_stock(stock_id)
          ON DELETE SET NULL
          ON UPDATE CASCADE,
     FOREIGN KEY (customer_id)
          REFERENCES customer(customer_id)
          ON DELETE CASCADE
          ON UPDATE CASCADE
);

CREATE TABLE order_contents (
     ordr_id INT,
     item_id INT,
     FOREIGN KEY (ordr_id)
          REFERENCES ordr(ordr_id)
          ON DELETE CASCADE
          ON UPDATE CASCADE,
     PRIMARY KEY (item_id),
     FOREIGN KEY (item_id)
          REFERENCES item(item_id)
          ON DELETE NO ACTION
          ON UPDATE CASCADE
);

CREATE TABLE category (
     category_name VARCHAR(30),
     PRIMARY KEY (category_name)
);

CREATE TABLE stock_category (
     stock_id INT,
     category_name VARCHAR(30),
     FOREIGN KEY (stock_id)
          REFERENCES item_stock(stock_id)
          ON DELETE CASCADE
          ON UPDATE CASCADE,
     FOREIGN KEY (category_name)
```

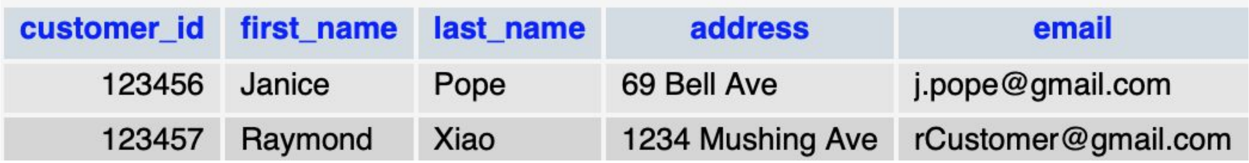
```
          REFERENCES category(category_name)
          ON DELETE CASCADE
          ON UPDATE CASCADE
);

CREATE TABLE employee (
     employee_id INT AUTO_INCREMENT,
     first_name VARCHAR(50),
     last_name VARCHAR(50),
     role VARCHAR(50),
     date_employed DATE,
     manager_id INT,
     PRIMARY KEY (employee_id),
     FOREIGN KEY (manager_id)
          REFERENCES employee(employee_id)
          ON DELETE SET NULL
          ON UPDATE CASCADE
);

/******************** Populating tables... ********************/
```

Employee Table:



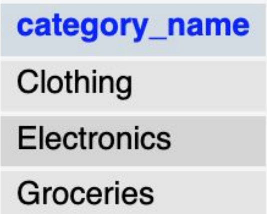
employee_id	first_name	last_name	role	date_employed	manager_id
29100	Bob	The Builder	Subordinate	2010-04-14	69123
29101	Leslie	Lai	Subordinate	2019-04-08	69123
29102	James	White	Subordinate	2017-11-20	69123
69123	Michael	Jones	Manager	2008-10-16	69123

Customer Table:



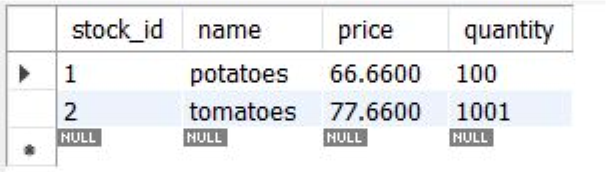
customer_id	first_name	last_name	address	email
123456	Janice	Pope	69 Bell Ave	j.pope@gmail.com
123457	Raymond	Xiao	1234 Mushing Ave	rCustomer@gmail.com

Category Table:



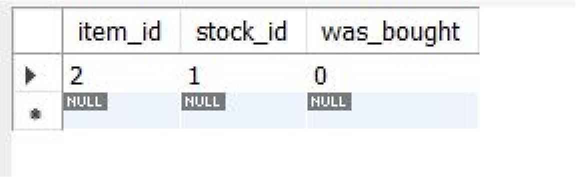
category_name
Clothing
Electronics
Groceries

Item Stock Table:



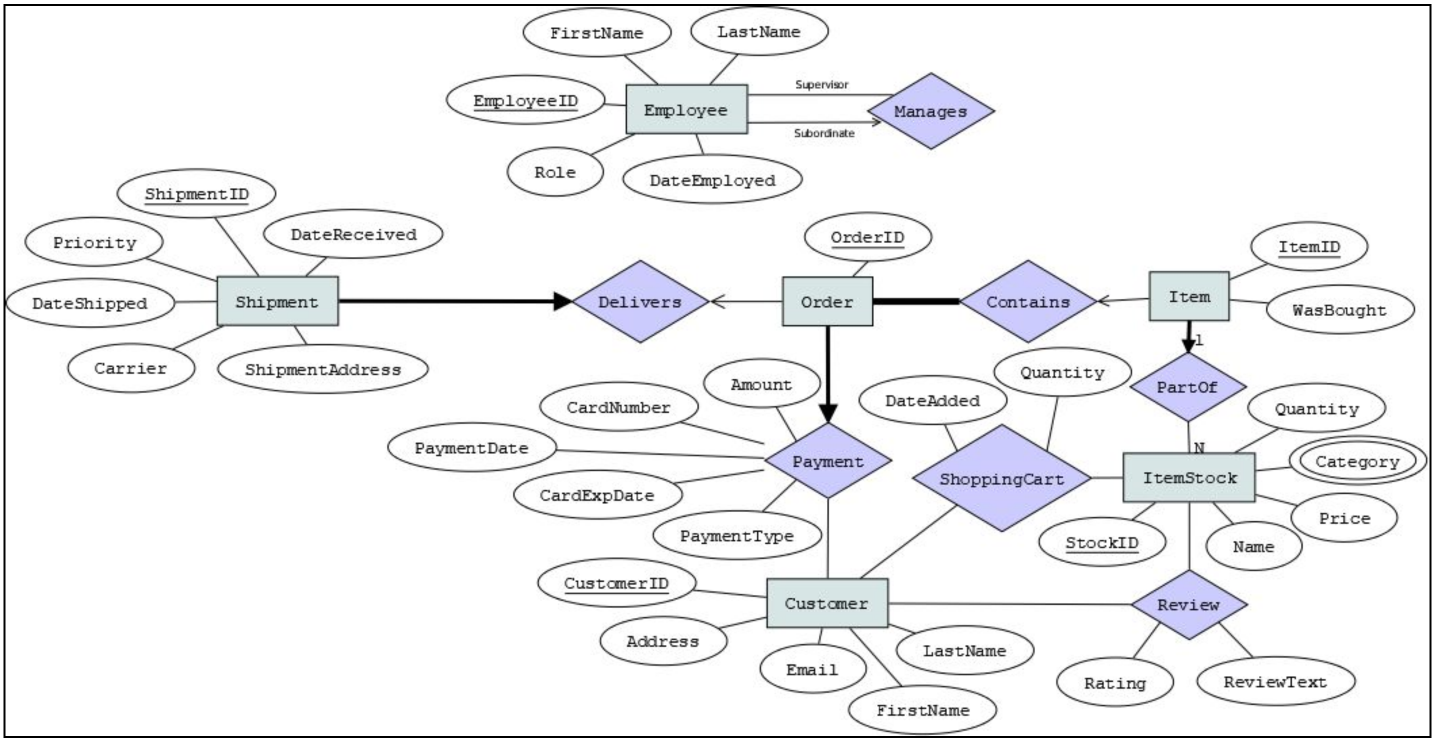
stock_id	name	price	quantity
1	potatoes	66.6600	100
2	tomatoes	77.6600	1001
NULL	NULL	NULL	NULL

```
Item Table:
```



item_id	stock_id	was_bought
2	1	0
NULL	NULL	NULL

Final E-R Model Used



Report

- If a customer id/account is deleted, reviews are deleted.
- If item gets deleted while in someone’s shopping cart, the item is lost and we plan to issue an error to the user.
- If a customer's account is deleted, we delete their purchase history.

- While the user's order has not been shipping yet, the shipping id is null.
- We used auto-increment on any id in order to generate new ones that are unique for each instance in which we require an id.
- For stock category, we decided to make a new table category to adequately insert new categories for the user to pick from.
- We decided to use 4 digits after the decimal point to represent monetary values because of rounding errors that may occur when splitting money. Using 4 digits will allow us to round and keep more exact values.
- For reactive constraints, we decided on the most reasonable for each instance. For example, if an item's stock id changes, the logical thing would be to cascade that change for the rest of the items in that stock.
- Instead of using a DATE data type for expiration date of card, we decided to use two attributes instead; each with CHAR(2) in order to store a 2 digit month and 2 digit year.
- As for payments, we give the user an option between credit, debit and paypal and store that in an enum.
- When there is a shipment, the order id may not be null because the order must exist for there to be a shipment.
- Reviews are pretty standard; each review gets the data type TEXT which holds 65,535 characters which should be enough for a user's review.