

Database Proposal

Group Member: Wen Shen (ws2543), Qianhui Yu (qy2226)

1. Description of the application

a. A high-level description of the general domain of the application.

Online shopping is now a firmly entrenched component of modern lifestyles. With barely any time to shop, the convenience of one-click ordering and rapid delivery to the customer's doorstep is pushing the e-commerce and m-commerce markets to new heights. However, sometimes it could be annoying to afford the expensive shipping, especially when you only purchase a small parcel but the shipping costs more than the actual goods. It also makes you feel "losing a million dollars" when there are discounts only applied to orders exceed a certain number. Then it comes to our project idea that we can develop a Group Purchasing Application for a certain shopping platform, for example, Freshgogo, an online Asian supermarket, to get shared benefits of shipping and discounts for customers. Customers living close to each other can actually "share" an order to save. For example, one customer may want to buy a packet of carrots on Freshgogo but find the delivery fee is \$9.99, which costs more than the carrots. Then he can submit a group purchasing request through our web front-end by selecting a pick-up station and including all the items he would like to buy. Those pick-up stations are managed by one or several managers, who are responsible to collect group purchasing requests received at certain time periods from different customers and place "shared" orders.

b. An idea of what entities and relationship sets you will have, including attributes and constraints.

This database should have at least the following entities sets:

Customer: username, password, c_phone, c_email, c_address

PickUpStation: ps_id, ps_address, detail, city, state, zip_code, bank_account, ps_phone

Manager: mid, first_name, last_name, ssn, gender, m_phone, m_email

Order: oid, otime_from, otime_to, total_price, cnum, inum, o_status, d_status

Item: iid, iname, unit_price

Cart: Cid

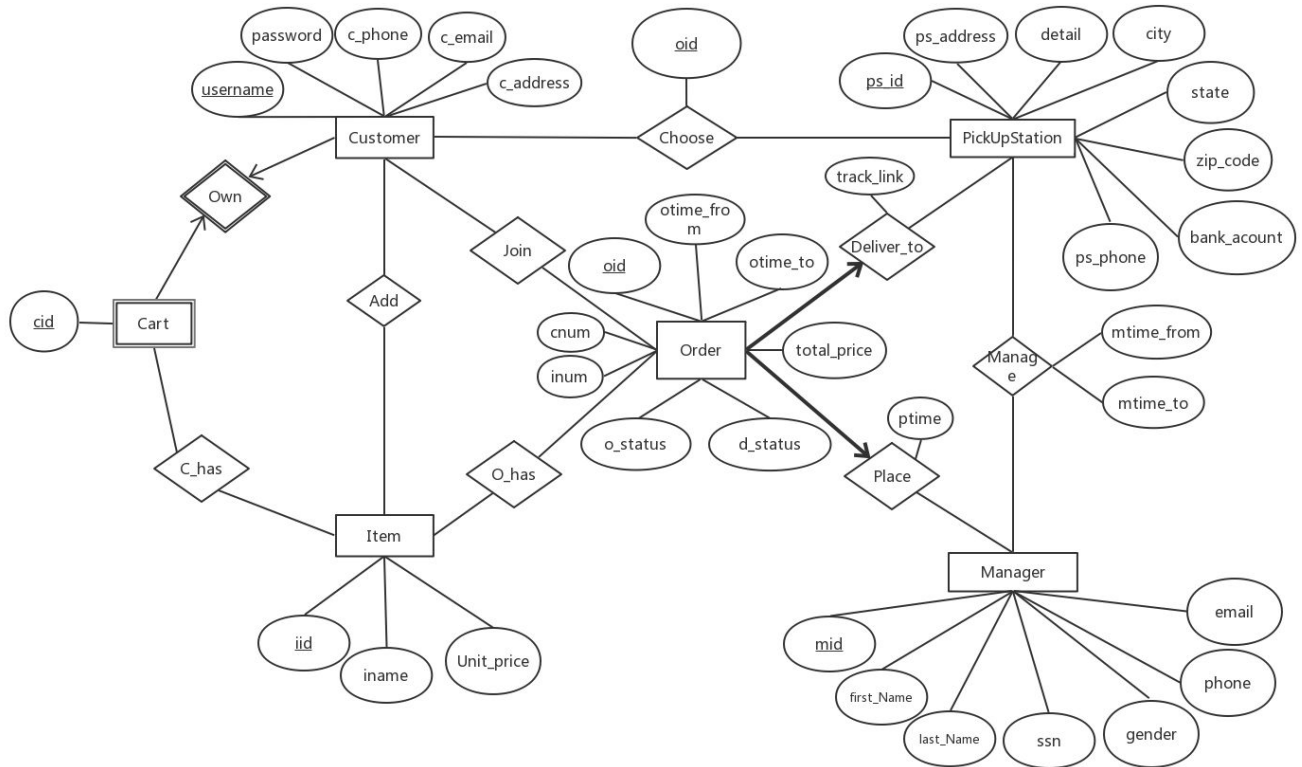
This database should have the following relationship sets:

Customer Choose PickUpStation	N-N
Customer Add Item	N-N
Customer Own Cart	1-1
Cart C_Has Item	N-N
Order O_Has Item	N-N
Manager Manage PickUpStation	N-N
Manager Place Order	1-N with total participation of Order
Order DeliverTo PickUpStation	N-1 with total participation of Order
Customer Join Order	N-N

c. Data resources.

There actually exist real-world “Order Sharing” groups. For example, many Chinese students would join “Order Sharing” chatting groups on Wechat, asking others if they would like to purchase things together and placing shared orders to save money. We will contact them to see if they could provide data for us.

2. E/R diagram



3. SQL schema

```

CREATE TABLE Customer
( username VARCHAR(20),
  password VARCHAR(20) NOT NULL,
  c_phone VARCHAR(15) NOT NULL,
  c_email VARCHAR(50),
  c_address VARCHAR(50),
  PRIMARY KEY (username));
    
```

```

CREATE TABLE PickUpStation
( ps_id INT,
  ps_address VARCHAR(20) NOT NULL,
  detail VARCHAR(50)
  city VARCHAR(20) NOT NULL,
  state VARCHAR(20) NOT NULL,
  zip_code CHAR(5) NOT NULL,
    
```

```
bank_account VARCHAR(50) NOT NULL,  
ps_phone VARCHAR(15) NOT NULL,  
PRIMARY KEY (ps_id));
```

```
CREATE TABLE Manager  
( mid INT,  
first_name VARCHAR(20) NOT NULL,  
last_name VARCHAR(20) NOT NULL,  
ssn CHAR(11),  
gender CHAR(1),  
phone VARCHAR(15) NOT NULL,  
email VARCHAR(50),  
PRIMARY KEY (mid));
```

```
CREATE TABLE Order  
( oid CHAR(10),  
otime_from DATETIME NOT NULL,  
otime_to DATETIME NOT NULL,  
total_price FLOAT NOT NULL,  
cnum INT,  
inum INT,  
o_status CHAR(1),  
d_status CHAR(1),  
PRIMARY KEY (oid));
```

```
CREATE TABLE Item  
( iid CHAR(10),  
iname VARCHAR(50) NOT NULL,  
unit_price FLOAT NOT NULL,  
PRIMARY KEY (iid));
```

```
CREATE TABLE Cart  
( cid CHAR(10),  
PRIMARY KEY (cid),  
ON DELETE CASCADE);
```

```
CREATE TABLE Choose  
( oid CHAR(10),  
username VARCHAR(20),  
ps_id INT NOT NULL,  
PRIMARY KEY (oid, username),  
FOREIGN KEY (oid) REFERENCES Order(oid),  
FOREIGN KEY (username) REFERENCES Customer(username));
```

```
CREATE TABLE Add
( oid CHAR(10),
  username VARCHAR(20),
  iid CHAR(10),
  quantity INT NOT NULL,
  PRIMARY KEY (oid, username, iid),
  FOREIGN KEY (oid) REFERENCES Order(oid),
  FOREIGN KEY (username) REFERENCES Customer(username),
  FOREIGN KEY (iid) REFERENCES Item(iid));
```

```
CREATE TABLE Own
( username VARCHAR(20) NOT NULL,
  cid CHAR(10),
  PRIMARY KEY (cid),
  FOREIGN KEY (cid) REFERENCES Cart(cid));
```

```
CREATE TABLE C_has
( cid CHAR(10),
  iid CHAR(10),
  quantity INT NOT NULL,
  PRIMARY KEY (cid, iid),
  FOREIGN KEY (cid) REFERENCES Cart(cid),
  FOREIGN KEY (iid) REFERENCES Item(iid));
```

```
CREATE TABLE O_has
( oid CHAR(10),
  iid CHAR(10),
  quantity INT NOT NULL,
  PRIMARY KEY (oid, iid),
  FOREIGN KEY (oid) REFERENCES Order(oid),
  FOREIGN KEY (iid) REFERENCES Item(iid));
```

```
CREATE TABLE Manage
( mid INT,
  ps_id INT,
  mtime_from DATETIME,
  mtime_to DATETIME,
  PRIMARY KEY (mid, ps_id),
  FOREIGN KEY (mid) REFERENCES Manager(mid),
  FOREIGN KEY (ps_id) REFERENCES PickupStation(ps_id));
```

```
CREATE TABLE Place
( oid CHAR(10),
mid INT NOT NULL,
ptime DATETIME NOT NULL,
PRIMARY KEY (oid),
FOREIGN KEY (oid) REFERENCES Order(oid));
```

```
CREATE TABLE Deliver_to
( oid CHAR(10),
ps_id INT NOT NULL NOT NULL,
track_link VARCHAR(100) NOT NULL,
PRIMARY KEY (oid),
FOREIGN KEY (oid) REFERENCES Order(oid));
```

```
CREATE TABLE Join
( oid CHAR(10),
username VARCHAR(20),
join DATETIME,
PRIMARY KEY (oid, username),
FOREIGN KEY (oid) REFERENCES Order(oid),
FOREIGN KEY (username) REFERENCES Customer(username));
```

4. Description of expansion plan (expand-design option)

N/A, we decided to choose the Front-End Option.

5. Contingency plan (indicate how you will "downgrade" the project to a simpler one in such a case)

Downgrade the project to the customer-only version which means the application doesn't have the manager's vision of checking orders and other functions.

6. A copy of the proposal with an approval signature

See next page.

交易安全问题? manager 领导整个 team
platform 降低成 Seller 的 -

