Ethan Spiro
CS 275
Final Project Written Portions

Website Link: http://web.engr.oregonstate.edu/~spiroe/275/spiroe_html.html

Outline (5%):

My database is a miniworld that represents the surfboard manufacturing/import business. It is designed to be used by a company which designs surfboards and surfboard components, but outsources the construction of the actual board. The database consists of products, manufacturers, orders, customers and local merchants (and addresses). This is both interesting and useful because I am designing the database for an actual surfboard company. I don't know how much of the database will be used, and it is likely to be adapted within the constraints of a web commerce content management system, but the project is very useful for using as a basis for the final design.

The project is also interesting because it has many real world applications outside of the surfboard industry. It is essentially a specialized, customer, product, manufacturer database that could be adapted to other industries as well. There are a lot of moving parts and the intricacies of shipping addresses, billing addresses, card info etc… has proved to be challenging and rewarding. Specifically the orders table, which has three foreign key constraints (customers, merchants and addresses) which can all be potentially NULL. Hopefully pieces of this database will be implemented in the real world, and I have hopefully created something that I can reuse or adapt if need be.

Database Outline in Words (5%)

My database will consists of 6 different entities, and 8 different tables. The 6 primary entities are customers, products, orders, manufacturers, addresses, and merchants. The two additional tables – order details for customers (mtm_products_orders) and shipments (mtm_products_manufacturers) – are a result of the many to many relationships between products and orders and manufacturers and products respectively.

Addresses:
Addresses are represented in a couple different tables, so for efficiency I represented addresses in their own table. Each address will have a mandatory unique id, a mandatory street, a mandatory city, and a mandatory country. Each address will have optional fields for states (because not all countries the company ships to have states) and zip codes (same reason). The address id's will be represented as foreign keys in the customer table (billing address), order table (shipping address), manufacturers table (physical address) and merchants table (physical address). An address is related in a (0-1) 1:1 relationship with stores, manufacturing plants and billing addresses (for now only 1 billing address stored) but an address can be associated with 0-N orders if someone places multiple orders to the same address.

Customers:
Each customer will have to have a mandatory: unique id, first name, last name, email, phone, payment type, card number, card expiration month, and card expiration year). They will also have a mandatory billing address (which is a foreign key pointing to the id in addresses). Everything is mandatory in this table because all these fields are required upon checkout (online) for a customer, in other words none can have a null or empty string value. Customers can make 1-N orders because they won't be registered unless they have made at least one order.

Orders:
Orders are the most interesting table. Each order contains a mandatory: unique id, delivery method (local pickup, local delivery, normal shipping, or international shipping), shipping cost, total cost, tax cost, weight and date. Orders also contain an optional shipping address which is a foreign key pointing to an address id (null potential for pickup). Finally orders have two more foreign keys which can or can't be null. They have a foreign key pointing to customer id and a foreign key pointing to merchant id. If the customer orders directly from the surfing company, then merchant id is null, but if a merchant orders a bulk order from the surf company, then customer id is null. This way you can tell who has placed the order – a merchant or a customer. An order must be associated with *exactly* 1 customer OR merchant. An order can contain 1-N products in it. An order is associated with *0-1* shipping address.

Products:
Every product has a mandatory unique id, a mandatory: name, msrp, buy price, quantity (inventory amount) and an optional: text description and category (accessories, clothing, surfboard). Products can be associated with 0-N orders and 1-N manufacturers. This is because one product can actually be the result of a couple different manufacturers' shipments(different components form different plants).

Order Details (mtm_products_orders):
Since there is a m:m relationship between orders and products we have to have this table which is essentially an order details table. This table has a mandatory unique id, a mandatory: foreign key which points to product id, a foreign key which points to order and the quantity of each product. Note this is not an entity, but is a table represented in the schema.

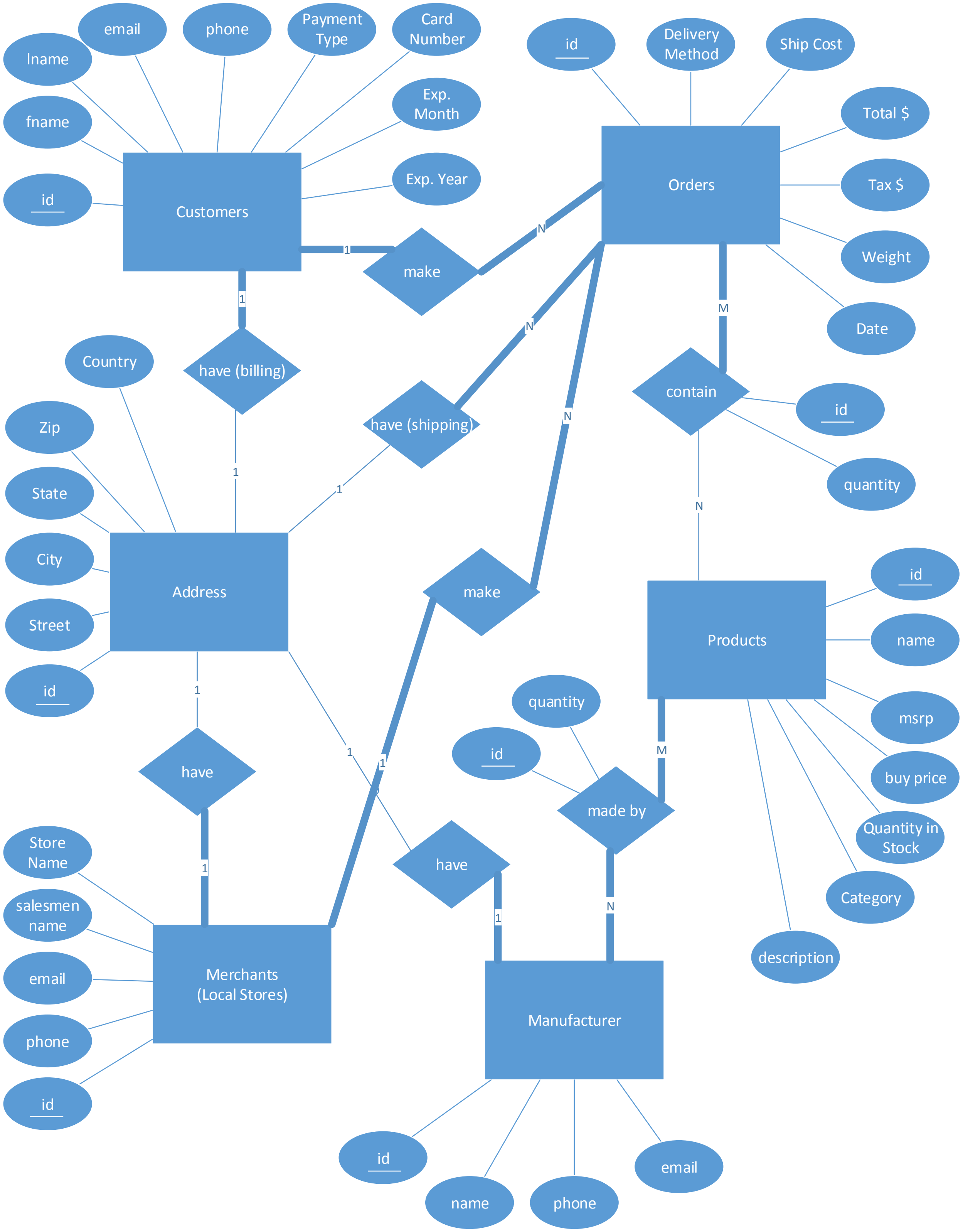
Manufacturers:

Each manufacturer will have a mandatory unique id, mandatory: name, phone, email, and physical address which is a foreign key pointing to the address id. Manufacturers make 1-N products (they have to make at least one to be in the database).

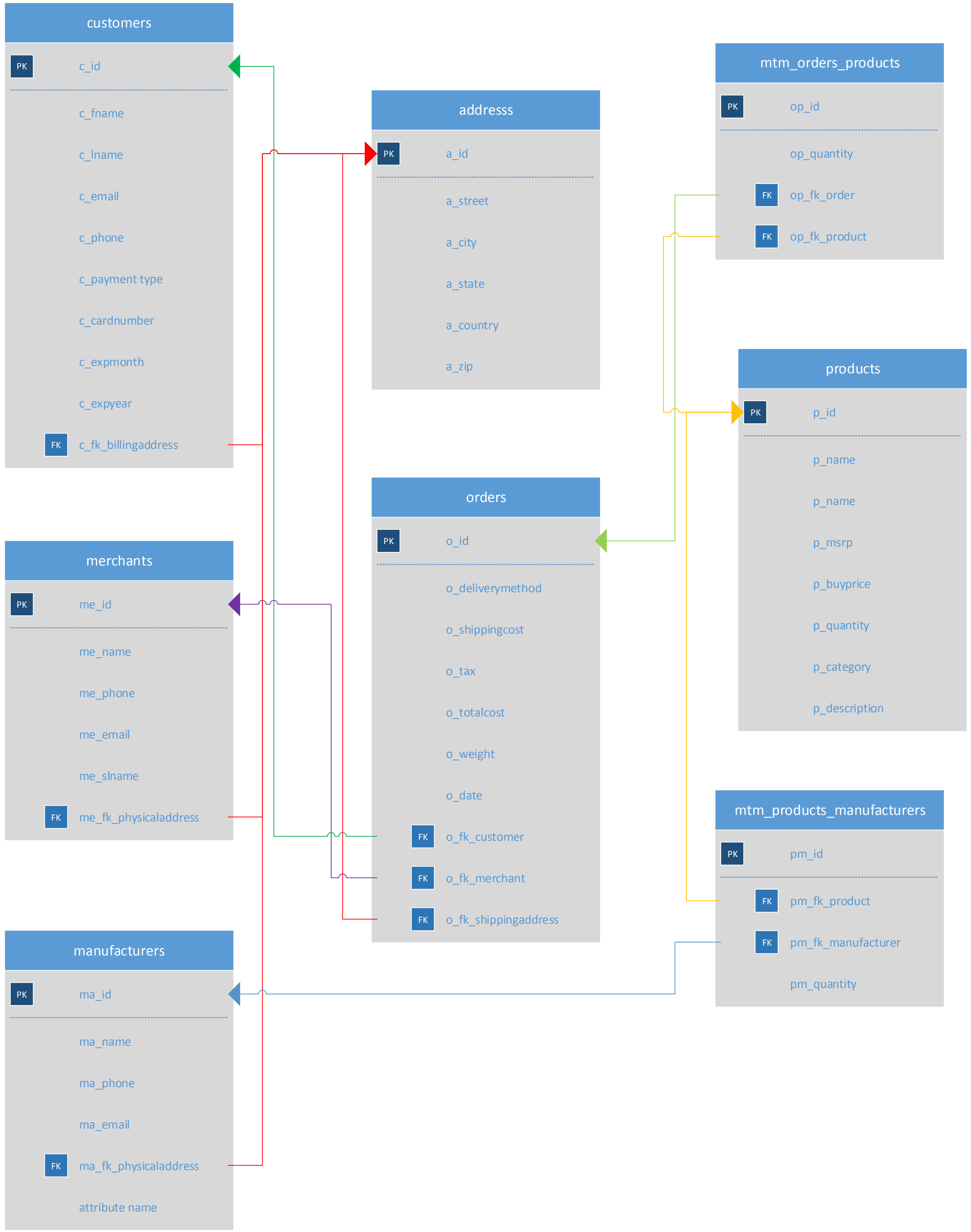
Shipments (mtm_products_manufacturers):
Since there is a m:m relationship between manufacturers and products we have to have this table which details collaboration on products and shipments. This table will have a mandatory: foreign key for product id and a foreign key for manufacturer id. It will also contain the quantity of said product in a shipment, but this doesn't have to be mandatory.

Merchants:
Each merchant (or local store) will have a mandatory unique id and mandatory: phone, email phone, and store name. There will also be a field for salesmen last name, but this won't be mandatory. Finally they will have a foreign key which represents their physical address. Names need not be unique because a name can have two different addresses (For example, jack's surfboards might have two stores). Merchants, like customers can place 1-N orders.

ER Diagram of Database (10%)



Database Schema (10%)



customers

PK c_id

c_fname

c_lname

c_email

c_phone

c_payment type

c_cardnumber

c_expmonth

c_expyear

FK c_fk_billingaddress

addresss

PK a_id

a_street

a_city

a_state

a_country

a_zip

mtm_orders_products

PK op_id

op_quantity

FK op_fk_order

FK op_fk_product

products

PK p_id

p_name

p_name

p_msrp

p_buyprice

p_quantity

p_category

p_description

orders

PK o_id

o_deliverymethod

o_shippingcost

o_tax

o_totalcost

o_weight

o_date

FK o_fk_customer

FK o_fk_merchant

FK o_fk_shippingaddress

merchants

PK me_id

me_name

me_phone

me_email

me_slname

FK me_fk_physicaladdress

mtm_products_manufacturers

PK pm_id

FK pm_fk_product

FK pm_fk_manufacturer

pm_quantity

manufacturers

PK ma_id

ma_name

ma_phone

ma_email

FK ma_fk_physicaladdress

attribute name

Table Creation Queries (10%)

```
# Ethan Spiro
# CS 275


# Table Creation Queries

SET FOREIGN_KEY_CHECKS = 0;

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Address Table
# id, street, city, state, country, zip
# primary key = id
DROP TABLE IF EXISTS `address`;
CREATE TABLE `address` (
        `a_id` int NOT NULL AUTO_INCREMENT,
        `a_street` varchar(255) NOT NULL,
        `a_city` varchar(255) NOT NULL,
        `a_state` varchar(255),
        `a_country` varchar(255) NOT NULL,
        `a_zip` int,
        PRIMARY KEY (`a_id`)
) ENGINE=InnoDB;
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Customers Table
# id, first name, last name, email, phone, payment type, card number, exp month, exp day, billing address
# primary key = id
# foreign key = billing address
DROP TABLE IF EXISTS `customers`;
CREATE TABLE `customers` (
        `c_id` int NOT NULL AUTO_INCREMENT,
        `c_fname` varchar(255) NOT NULL,
        `c_lname` varchar(255) NOT NULL,
        `c_email` varchar(255) NOT NULL,
        `c_phone` int NOT NULL,
        `c_paymenttype` varchar(255) NOT NULL,
        `c_cardnumber` bigint NOT NULL,
        `c_expmonth` int NOT NULL,
        `c_expyear` int NOT NULL,
        `c_fk_billingaddress` int NOT NULL,
        PRIMARY KEY (`c_id`),
        FOREIGN KEY (`c_fk_billingaddress`) REFERENCES address(a_id) ON DELETE RESTRICT ON UPDATE CASCADE
) ENGINE=InnoDB;
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Merchants Table
# id, phone, email, salesman last name, store name, physical address
# primary key = id
# foreign key = physical address of merchant headquarters
DROP TABLE IF EXISTS `merchants`;
CREATE TABLE `merchants` (
        `me_id` int NOT NULL AUTO_INCREMENT,
        `me_name` varchar(255) NOT NULL,
        `me_phone` int NOT NULL,
        `me_email` varchar(255) NOT NULL,
        `me_slname` varchar(255),
        `me_fk_physicaladdress` int NOT NULL,
        PRIMARY KEY (`me_id`),
        FOREIGN KEY (`me_fk_physicaladdress`) REFERENCES address(a_id) ON DELETE RESTRICT ON UPDATE CASCADE
) ENGINE=InnoDB;
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```sql


#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Orders Table
# id, delivery method, shipping $, tax $, total $, weight, date, customer, merchant, address
# primary key = id
# foreign key = customer, merchant, shipping address of order
DROP TABLE IF EXISTS `orders`;
CREATE TABLE `orders` (
        `o_id` int NOT NULL AUTO_INCREMENT,
        `o_deliverymethod` varchar(255) NOT NULL,
        `o_shippingcost` decimal(10, 2) NOT NULL,
        `o_tax` decimal(10, 2) NOT NULL,
        `o_totalcost` decimal(10, 2) NOT NULL,
        `o_weight` decimal(10, 2) NOT NULL,
        `o_date` date NOT NULL,
        `o_fk_customer` int, # null if it's a merchant order
        `o_fk_merchant` int, # null if it's a customer order
        `o_fk_shippingaddress` int, # null if its a local pickup
        PRIMARY KEY (`o_id`),
        FOREIGN KEY (`o_fk_customer`) REFERENCES customers(c_id) ON DELETE RESTRICT ON UPDATE CASCADE,
        FOREIGN KEY (`o_fk_merchant`) REFERENCES merchants(me_id) ON DELETE RESTRICT ON UPDATE CASCADE,
        FOREIGN KEY (`o_fk_shippingaddress`) REFERENCES address(a_id) ON DELETE RESTRICT ON UPDATE CASCADE
) ENGINE=InnoDB;
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Products Table
# id, name, msrp, buy price, quantity in stock, category
# primary key = id
DROP TABLE IF EXISTS `products`;
CREATE TABLE `products` (
        `p_id` int NOT NULL AUTO_INCREMENT,
        `p_name` varchar(255) NOT NULL,
        `p_msrp` decimal(10, 2) NOT NULL,
        `p_buyprice` decimal(10,2) NOT NULL,
        `p_quantity` int NOT NULL,
        `p_description` text,
        `p_category` varchar(255),
        PRIMARY KEY (`p_id`)
) ENGINE=InnoDB;
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# mtm_Orders_Products Table (order details table)
# id, product id, order id, quantity, price
# primary key = id
# foreign key = product id, order id
DROP TABLE IF EXISTS `mtm_orders_products`;
CREATE TABLE `mtm_orders_products` (
        `op_id` int NOT NULL AUTO_INCREMENT,
        `op_quantity` int NOT NULL,
        `op_fk_order` int NOT NULL,
        `op_fk_product` int NOT NULL,
        PRIMARY KEY (`op_id`),
        FOREIGN KEY (`op_fk_order`) REFERENCES orders(o_id) ON DELETE RESTRICT ON UPDATE CASCADE,
        FOREIGN KEY (`op_fk_product`) REFERENCES products(p_id) ON DELETE RESTRICT ON UPDATE CASCADE
) ENGINE=InnoDB;
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Manufacturers Table
# id, name, phone, email, address
# primary key = id
# foreign key = address
DROP TABLE IF EXISTS `manufacturers`;
```

```
CREATE TABLE `manufacturers` (
        `ma_id` int NOT NULL AUTO_INCREMENT,
        `ma_name` varchar(255) NOT NULL,
        `ma_phone` int NOT NULL,
        `ma_email` varchar(255) NOT NULL,
        `ma_fk_physicaladdress` int NOT NULL,
        PRIMARY KEY (`ma_id`),
        FOREIGN KEY (`ma_fk_physicaladdress`) REFERENCES address(a_id) ON DELETE RESTRICT ON UPDATE CASCADE
) ENGINE=InnoDB;
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# mtm_Products_Manufacturers Table
# id, manufacturer id, product id
# primary key = id
# foreign id = manufacturer id, product id
DROP TABLE IF EXISTS `mtm_products_manufacturers`;
CREATE TABLE `mtm_products_manufacturers` (
        `pm_id` int NOT NULL AUTO_INCREMENT,
        `pm_fk_product` int NOT NULL,
        `pm_fk_manufacturer` int NOT NULL,
        `pm_quantity` int,
        PRIMARY KEY (`pm_id`),
        FOREIGN KEY (`pm_fk_product`) REFERENCES products(p_id) ON DELETE RESTRICT ON UPDATE CASCADE,
        FOREIGN KEY (`pm_fk_manufacturer`) REFERENCES manufacturers(ma_id) ON DELETE RESTRICT ON UPDATE CASCADE
) ENGINE=InnoDB;
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

SET FOREIGN_KEY_CHECKS = 1;
```

Table Manual Insertions (Not required, just for testing initially 0%)

```
# Ethan Spiro
# CS 275


# Manual Insertion Queries for Populating Tables (to populate initially for testing)

SET FOREIGN_KEY_CHECKS = 0;
# Insert addresses
INSERT INTO address(a_street, a_city, a_state, a_country, a_zip)
VALUES ('4000 Point White Drive', 'Bainbridge Island', 'WA', 'United States', 98110),
('145 S Kingsley Drive', 'Los Angeles', 'CA', 'United States', 90004),
('235 Elm Street', 'Manchester', 'NH', 'United States', 90567),
('44 Pike Street', 'Calgary', NULL, 'Canada', NULL),
('49 Pine Street', 'Santa Barbara', 'CA', 'United States', 45678),
('4400 Brown Street', 'Portland', NULL, 'United States', 12345),
('5050 Maple Street', 'San Francisco', 'CA', 'United States', 56789),
('1234 Hong Street', 'Beijing', NULL, 'China', NULL),
('5678 Blue Street', 'Perth', NULL, 'Australia', NULL),
('1010 Binary Street', 'Dunedin', NULL, 'New Zealand', NULL);

# Insert 3 Customers
INSERT INTO customers(c_fname, c_lname, c_email, c_phone, c_paymenttype, c_cardnumber, c_expmonth,
c_expyear, c_fk_billingaddress)
VALUES ('Ethan', 'Spiro', 'ethanspiro@gmail.com', 2069657953, 'Visa', 4388678923456789, 12, 2017, 1),
('Bob', 'Jones', 'bjones@gmail.com', 2065687953, 'Visa', 6577778923456789, 11, 2020, 2),
('Sue', 'Greenway', 'sue@hotmail.com', 6079657953, 'Visa', 1234678923456789, 07, 2019, 3);

# Insert 3 merchants
INSERT INTO merchants(me_name, me_phone, me_email, me_slname, me_fk_physicaladdress)
VALUES ('Jacks Surfboards', 2475657899, 'jackssurf@gmail.com', 'Roberts', 4),
('Magic Seaweed', 1234567890, 'ms@gmail.com', 'Slimey', 5),
('Fools Gold', 9994567890, 'fools@gmail.com', 'Smith', 6);

```

```
# Insert 3 Orders
INSERT INTO orders(o_deliverymethod, o_shippingcost, o_tax, o_totalcost, o_weight, o_date, o_fk_customer,
o_fk_merchant,o_fk_shippingaddress)
VALUES ('Local Pickup', 0, 12.00, 125.00, 8.00, '2012-12-28', 1, NULL, NULL),
('Local Delivery', 25.00, 12.00, 1100.00, 34.50, '2013-10-14', NULL, 1, 5),
('Delivery', 0, 36.00, 450.00, 18.50, '2014-09-12', 2, NULL, 4);

# Insert 5 products
INSERT INTO products(p_name, p_msrp, p_buyprice, p_quantity, p_description, p_category)
VALUES ('Longboard', 400.00, 250.00, 15, 'A board 8-10 feet', 'surfboard'),
('Funboard', 300.00, 150.00, 25, 'A board 7-9 feet', 'surfboard'),
('Shortboard', 200.00, 100.00, 45, 'A board 5-7 feet', 'surfboard'),
('T-Shirt', 20.00, 5.00, 100, 'A short sleeve shirt', 'apparel'),
('Leash', 25.00, 5.00, 40, 'A leash for a surfboard', 'accessories');

# Insert order details for all orders
INSERT INTO mtm_orders_products(op_quantity, op_fk_order, op_fk_product)
VALUES (2, 1, 2),
(1, 1, 3),
(3, 2, 2),
(2, 3, 3);

# Insert 3 manufacturers
INSERT INTO manufacturers(ma_name, ma_phone, ma_email, ma_fk_physicaladdress)
VALUES ('Beijing Export', 0055657899, 'bejexp@gmail.com', 8),
('Australia Export', 7777567890, 'aussie@gmail.com', 9),
('NZ Manufacturing Co.', 007807890, 'kiwi@gmail.com', 10);

# Insert product_manufacturer details for all products
INSERT INTO mtm_products_manufacturers(pm_fk_product, pm_fk_manufacturer, pm_quantity)
VALUES (1, 1, 20), #order 1, 2 tshirts
(2, 1, 30), #order 1, 1 longboard
(3, 1, 40),
(4, 2, 100),
(5, 3, 200);

SET FOREIGN_KEY_CHECKS = 1;
```

General Use Queries (30%)

```
# Ethan Spiro
# CS275


# All Database Queries (Displaying tables, Filters, Non-dropdown queries)


#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Display Queries
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Display address table
# Tables: address
SELECT address.a_id, address.a_street, address.a_city, address.a_state, address.a_country, address.a_zip
FROM address;

# Display customers table
# Tables: customers, address
SELECT customers.c_id, customers.c_fname, customers.c_lname, customers.c_email, customers.c_phone,
customers.c_paymenttype,
customers.c_cardnumber, customers.c_expmonth, customers.c_expyear,address.a_street FROM customers
INNER JOIN address ON address.a_id = customers.c_fk_billingaddress;

# Display manufacturers table
# Tables: manufacturers, address
SELECT manufacturers.ma_id, manufacturers.ma_name, manufacturers.ma_phone, manufacturers.ma_email,
address.a_street FROM manufacturers
INNER JOIN address ON address.a_id = manufacturers.ma_fk_physicaladdress;
```

```sql

# Display merchants table
# Tables: merchants, address
SELECT merchants.me_id, merchants.me_name, merchants.me_phone, merchants.me_email, merchants.me_slname,
address.a_street FROM merchants
INNER JOIN address ON address.a_id = merchants.me_fk_physicaladdress;

# Display order details table
# Tables: mtm_order_products, orders, products
SELECT mtm_orders_products.op_id, mtm_orders_products.op_quantity, orders.o_id, products.p_name
FROM mtm_orders_products
INNER JOIN orders ON orders.o_id = mtm_orders_products.op_fk_order
INNER JOIN products ON products.p_id = mtm_orders_products.op_fk_product;

# Display orders table
# Tables: orders, address, customers, merchants
SELECT orders.o_id, orders.o_deliverymethod, orders.o_shippingcost, orders.o_tax, orders.o_totalcost,
orders.o_weight, orders.o_date,customers.c_lname, merchants.me_name, address.a_street FROM orders
LEFT JOIN address ON address.a_id = orders.o_fk_shippingaddress
LEFT JOIN customers ON customers.c_id = orders.o_fk_customer
LEFT JOIN merchants ON merchants.me_id = orders.o_fk_merchant;

# Display products table
# Tables: products
SELECT products.p_id, products.p_name, products.p_msrp, products.p_buyprice, products.p_quantity,
products.p_description,products.p_category FROM products;

# Display shipments table
# Tables: mtm_products_manufacturers
SELECT mtm_products_manufacturers.pm_id, products.p_name, manufacturers.ma_name,
mtm_products_manufacturers.pm_quantity FROMmtm_products_manufacturers
INNER JOIN manufacturers ON manufacturers.ma_id = mtm_products_manufacturers.pm_fk_manufacturer
INNER JOIN products ON products.p_id = mtm_products_manufacturers.pm_fk_product;
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Filter Queries
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Filter what orders are from what customer. Show the order and allow the user to use a drop down to select
customer
# Tables: orders, customers
SELECT orders.o_id, orders.o_deliverymethod, orders.o_shippingcost, orders.o_tax, orders.o_totalcost,
orders.o_weight, orders.o_date, customers.c_lname, merchants.me_name, address.a_street FROM orders
LEFT JOIN address ON address.a_id = orders.o_fk_shippingaddress
LEFT JOIN customers ON customers.c_id = orders.o_fk_customer
LEFT JOIN merchants ON merchants.me_id = orders.o_fk_merchant
WHERE customers.c_id = [?];

# Filter shipments by the manufacturer
# Tables: mtm_products_manufacturers, manufacturers, orders
SELECT mtm_products_manufacturers.pm_id, products.p_name, manufacturers.ma_name,
mtm_products_manufacturers.pm_quantity FROM mtm_products_manufacturers
INNER JOIN manufacturers ON manufacturers.ma_id = mtm_products_manufacturers.pm_fk_manufacturer
INNER JOIN products ON products.p_id = mtm_products_manufacturers.pm_fk_product
WHERE manufacturers.ma_id = [?];

# Filter shipments by product name
# Tables: mtm_products, manufacturers, products, manufacturers
SELECT mtm_products_manufacturers.pm_id, products.p_name, manufacturers.ma_name,
mtm_products_manufacturers.pm_quantity FROMmtm_products_manufacturers
INNER JOIN manufacturers ON manufacturers.ma_id = mtm_products_manufacturers.pm_fk_manufacturer
INNER JOIN products ON products.p_id = mtm_products_manufacturers.pm_fk_product
WHERE products.p_id = [?];

# Filter/show billing address by last name
# Tables: customers, address
SELECT address.a_id, address.a_street, address.a_city, address.a_state, address.a_country, address.a_zip
FROM address
```

```
INNER JOIN customers ON customers.c_fk_billingaddress = address.a_id
WHERE customers.c_id = [?];
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Other (non-dropdown queries)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Show all international merchants (NOT IN USA).
# Tables: address, merchant
SELECT merchants.me_id, merchants.me_name, merchants.me_phone, merchants.me_email, merchants.me_slname,
address.a_street FROM merchants
INNER JOIN address ON address.a_id = merchants.me_fk_physicaladdress
WHERE merchants.me_id NOT IN (
        SELECT merchants.me_id FROM merchants
        INNER JOIN address ON address.a_id = merchants.me_fk_physicaladdress
        WHERE address.a_country = "United States"
);

# Show all Customers (all fields) who ordered a t-shirt
# Tables: Address, Customers, Orders, Order Details, Products
SELECT customers.c_id, customers.c_fname, customers.c_lname, customers.c_email,
customers.c_phone, customers.c_paymenttype, customers.c_cardnumber, customers.c_expmonth,
customers.c_expyear, address.a_street FROM address
INNER JOIN customers ON customers.c_fk_billingaddress = address.a_id
INNER JOIN orders ON orders.o_fk_customer = customers.c_id
INNER JOIN mtm_orders_products ON mtm_orders_products.op_fk_order = orders.o_id
INNER JOIN products ON products.p_id = mtm_orders_products.op_fk_product
WHERE products.p_name = 'T-Shirt'
GROUP BY customers.c_id;

# Show all manufacturers (store) and their phone numbers and emails in china
# Tables: address, manufacturer
SELECT manufacturers.ma_id, manufacturers.ma_name, manufacturers.ma_phone,
manufacturers.ma_email, address.a_street FROM manufacturers
INNER JOIN address ON address.a_id = manufacturers.ma_fk_physicaladdress
WHERE address.a_country = 'China';

# Show all customers (all fields - address) who have bought a surfboard from China
# Tables: all
SELECT customers.c_id, customers.c_fname, customers.c_lname, customers.c_email,
customers.c_phone, customers.c_paymenttype, customers.c_cardnumber, customers.c_expmonth,
customers.c_expyear FROM customers
INNER JOIN orders ON orders.o_fk_customer = customers.c_id
INNER JOIN mtm_orders_products ON mtm_orders_products.op_fk_order = orders.o_id
INNER JOIN products ON products.p_id = mtm_orders_products.op_fk_product
INNER JOIN mtm_products_manufacturers ON mtm_products_manufacturers.pm_fk_product = products.p_id
INNER JOIN manufacturers ON manufacturers.ma_id = mtm_products_manufacturers.pm_fk_manufacturer
INNER JOIN address ON address.a_id = manufacturers.ma_fk_physicaladdress
WHERE address.a_country = 'China' AND products.p_category = 'surfboard'
GROUP BY customers.c_id;
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```