

## Database Outline in Words

A local store needs a database to store information about its sales.

The store has Customers data. The customers have the following attributes: first name, last name, username, password, DOB, phone. Each customer may have zero or more orders, pay zero or more payments and has exactly one shipping and one billing address.

The store also has Payments data. Payment data include payment card number, expiration date, type of the payment. Each payment must be paid by exactly 1 customer.

The store has Orders data. Each order has order date and total that is calculated as a sum of product prices. Each order must be placed exactly by 1 customer. Order may contain 1 or more products.

The store has Shipping address ID table. It has shipping address ID and reference to address ID that's called addressID. Each shipping address may belong to zero or more customers. It also must have reference to exactly one address.

The store has Billing address ID table. It has billing address ID and reference to address ID that's called addressID. Each billing address ID may belong to zero or more customers. It also must have reference to exactly one address.

The store has Addresses data. Each address has the following attributes: street address1, street address2, city, zip code. Address may have no more than one country and no more than one state. Address is related to zero or more billing addresses and is related to zero or more shipped addresses.

The store has Countries data. Each country has country name. Each country may have zero or more addresses.

The store has States data. Each state has a state name and state abbreviation. Each state may have zero or more addresses.

The store has Products data. Each product has the following attributes: product name, retail price, wholesale price, and quantity in stock. Product may be ordered zero or many times, and it may present in order zero or more times. Each unique product may have one or more suppliers. Product quantity in the given order is an attribute of the relationship between order and product.

The store has Suppliers data. Each supplier has name, website address, and phone to contact them. Each supplier may supply zero or many products.

### **CREATE TABLE Queries:**

```
CREATE TABLE `shipAddr` (  
  `shipAddrID` int(11) NOT NULL AUTO_INCREMENT,  
  `addressID` int(11) NOT NULL,  
  PRIMARY KEY (`shipAddrID`),  
  KEY `shipAddr_ibfk_1` (`addressID`),  
  CONSTRAINT `shipAddr_ibfk_1` FOREIGN KEY (`addressID`) REFERENCES `address` (`addressID`)  
  ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB;
```

```
CREATE TABLE `billAddr` (  
  `billAddrID` int(11) NOT NULL AUTO_INCREMENT,  
  `addressID` int(11) NOT NULL,  
  PRIMARY KEY (`billAddrID`),  
  KEY `billAddr_ibfk_1` (`addressID`),  
  CONSTRAINT `billAddr_ibfk_1` FOREIGN KEY (`addressID`) REFERENCES `address` (`addressID`)  
  ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB;
```

```
CREATE TABLE `address` (  
  `addressID` int(11) NOT NULL AUTO_INCREMENT,  
  `street1` varchar(255) NOT NULL,  
  `street2` varchar(255) DEFAULT NULL,  
  `stateID` int(4) DEFAULT NULL,  
  `countryID` int(5) DEFAULT NULL,  
  `city` varchar(50) NOT NULL,  
  `zip` varchar(50) NOT NULL,  
  PRIMARY KEY (`addressID`),  
  UNIQUE KEY `address` (`street1`,`street2`,`city`,`zip`,`stateID`,`countryID`),  
  KEY `stateID` (`stateID`),  
  KEY `countryID` (`countryID`),  
  CONSTRAINT `address_ibfk_2` FOREIGN KEY (`stateID`) REFERENCES `states` (`stateID`) ON  
  DELETE CASCADE ON UPDATE CASCADE,  
  
  CONSTRAINT `address_ibfk_3` FOREIGN KEY (`countryID`) REFERENCES `countries` (`countryID`)  
  ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB;
```

```
CREATE TABLE `states` (  
  `stateID` int(4) NOT NULL AUTO_INCREMENT,  
  `stateName` varchar(50) DEFAULT NULL,  
  `abbrev` char(2) DEFAULT NULL,  
  PRIMARY KEY (`stateID`)  
) ENGINE=InnoDB;
```

```
CREATE TABLE `countries` (  
  `countryID` int(5) NOT NULL AUTO_INCREMENT,  
  `countryName` varchar(50) NOT NULL,  
  PRIMARY KEY (`countryID`)  
) ENGINE=InnoDB;
```

```
CREATE TABLE `customer` (  
  `customerID` int(11) NOT NULL AUTO_INCREMENT,  
  `firstName` varchar(255) NOT NULL,  
  `lastName` varchar(255) NOT NULL,  
  `userName` varchar(255) NOT NULL,  
  `password` varchar(255) NOT NULL,  
  `phone` varchar(255) DEFAULT NULL,  
  `email` varchar(255) DEFAULT NULL,  
  `billAddrID` int(11) NOT NULL,  
  `shipAddrID` int(11) NOT NULL,  
  PRIMARY KEY (`customerID`),  
  UNIQUE KEY `userName` (`userName`),  
  UNIQUE KEY `email` (`email`),  
  KEY `customer_ibfk_2` (`shipAddrID`),  
  KEY `customer_ibfk_1` (`billAddrID`),  
  CONSTRAINT `customer_ibfk_2` FOREIGN KEY (`shipAddrID`) REFERENCES `shipAddr`  
  (`shipAddrID`) ON DELETE CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `customer_ibfk_1` FOREIGN KEY (`billAddrID`) REFERENCES `billAddr`  
  (`billAddrID`) ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB
```

```
CREATE TABLE `payment` (  
  `paymentID` int(11) NOT NULL AUTO_INCREMENT,  
  `cardNum` varchar(255) NOT NULL,  
  `paymentType` varchar(255) NOT NULL,  
  `expDate` date NOT NULL,
```

```
`customerID` int(11) NOT NULL,  
PRIMARY KEY (`paymentID`),  
UNIQUE KEY `cardNum` (`cardNum`),  
KEY `payment_ibfk_1` (`customerID`),  
CONSTRAINT `payment_ibfk_1` FOREIGN KEY (`customerID`) REFERENCES `customer`  
(`customerID`) ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB;
```

```
CREATE TABLE `order` (  
  `orderID` int(11) NOT NULL AUTO_INCREMENT,  
  `orderDate` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  `customerID` int(11) NOT NULL,  
  PRIMARY KEY (`orderID`),  
  KEY `order_ibfk_1` (`customerID`),  
  CONSTRAINT `order_ibfk_1` FOREIGN KEY (`customerID`) REFERENCES `customer`  
  (`customerID`) ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB;
```

```
CREATE TABLE `product_order` (  
  `productID` int(11) NOT NULL,  
  `orderID` int(11) NOT NULL,  
  `quantOrdered` int(11) NOT NULL,  
  PRIMARY KEY (`orderID`,`productID`),  
  KEY `productID` (`productID`),  
  CONSTRAINT `product_order_ibfk_1` FOREIGN KEY (`productID`) REFERENCES `product`  
  (`productID`) ON DELETE CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `product_order_ibfk_2` FOREIGN KEY (`orderID`) REFERENCES `order` (`orderID`)  
  ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB;
```

```
CREATE TABLE `product` (  
  `productID` int(11) NOT NULL AUTO_INCREMENT,  
  `productName` varchar(255) NOT NULL,  
  `retailPrice` double(12,2) NOT NULL,  
  `wholePrice` double(12,2) NOT NULL,  
  `quantStock` int(11) NOT NULL,  
  PRIMARY KEY (`productID`),  
  UNIQUE KEY `productName` (`productName`,`retailPrice`,`wholePrice`)  
) ENGINE=InnoDB;
```

```
CREATE TABLE `supplier` (  
  `supplierID` mediumint(9) NOT NULL AUTO_INCREMENT,
```

```
`supName` varchar(255) NOT NULL,  
`phone` varchar(255) DEFAULT NULL,  
`website` varchar(255) DEFAULT NULL,  
PRIMARY KEY (`supplierID`),  
UNIQUE KEY `supName` (`supName`)  
) ENGINE=InnoDB;
```

### **INSERT INTO Queries:**

```
# address  
INSERT INTO `address`  
(`street1`, `street2`, `city`, `stateID`, `zip`, `countryID`)  
VALUES  
('[street1]', '[street2]', '[city]', '[stateID]', '[countryID]');
```

```
#countries  
INSERT INTO `countries` (`countryName`) VALUES  
('[countryName]');
```

```
#states  
INSERT INTO `states` VALUES ('[stateName]', '[abbrev]');
```

```
#shipAddr  
INSERT INTO `shipAddr` (addressID) values ((SELECT `addressID` FROM `address`  
WHERE (`street1` = '[street1]' AND `street2` = '[street2]' AND `city` = '[city]'  
AND `stateID` = '[stateID]' AND `zip` = '[zip]' AND `countryID` = '[countryID]') LIMIT 1));
```

```
#billAddr  
INSERT INTO `billAddr` (addressID) values ((SELECT `addressID` FROM `address`  
WHERE (`street1` = '[street1]' AND `street2` = '[street2]' AND `city` = '[city]'  
AND `stateID` = '[stateID]' AND `zip` = '[zip]' AND `countryID` = '[countryID]') LIMIT 1));
```

```
# customer  
INSERT INTO `customer`  
(`firstName`, `lastName`, `userName`, `password`, `phone`, `email`, `billAddrID`, `shipAddrID`)  
VALUES  
('[firstName]', '[lastName]', '[userName]', '[password]', '[phone]', '[email]',  
(SELECT billAddrID FROM billAddr BA  
INNER JOIN address A ON A.addressID = BA.addressID  
WHERE (street1 = '[street1]' AND street2 = '[street2]' AND
```

```

    city = '[city]' AND stateID = '[stateID]' AND zip = '[zip]' AND
    countryID = '[countryID]')LIMIT 1),
(SELECT shipAddrID FROM shipAddr SA
INNER JOIN address A ON A.addressID = SA.addressID
WHERE (street1 = '[street1]' AND street2 = '[street2]' AND
    city = '[city]' AND stateID = '[stateID]' AND zip = '[zip]' AND
    countryID = '[countryID]')LIMIT 1));

# order
INSERT INTO `order` (`customerID`) VALUES (
    (SELECT `customerID` FROM `customer` WHERE `userName` = '[userName]')
);

# product
INSERT INTO `product`
(`productName`, `retailPrice`, `wholePrice`, `quantStock`)
VALUES('[productName]', '[retailPrice]', '[wholePrice]', '[quantStock]');

# product_order
INSERT INTO `product_order`
(`productID`, `orderID`, `quantOrdered`) VALUES([productID], [orderID], [quantOrdered]);

# supplier
INSERT INTO `supplier`
(`supName`, `phone`, `website`)
VALUES
('[supName]', '[phone]', '[website]');

# payment
INSERT INTO `payment`
(`cardNum`, `paymentType`, `expDate`, `customerID`)
VALUES
('[cardNum]', '[paymentType]', '[expDate]',
    (SELECT `customerID` FROM `customer` WHERE `userName` = '[userName]'));

```

### **General Queries:**

```

# Show customers + billing address + shipping address joined table
SELECT firstName, lastName, userName, `password`, phone, email,
b_street1, b_street2, b_city, b_state, b_zip, b_country,
s_street1, s_street2, s_city, s_state, s_zip, s_country FROM

```

```

(SELECT customerID AS b_cid, firstName, lastName, userName, `password`,
phone, email, street1 AS b_street1, street2 AS b_street2, city AS b_city,
abbrev AS b_state, zip AS b_zip, countryName AS b_country
FROM customer C
INNER JOIN billAddr AS BA ON C.billAddrID = BA.billAddrID
INNER JOIN address A ON A.addressID = BA.addressID
INNER JOIN states S ON A.stateID = S.stateID
INNER JOIN countries CO ON A.countryID = CO.countryID) AS tb1
INNER JOIN
(SELECT customerID AS s_cid, street1 AS s_street1,
street2 AS s_street2, city AS s_city, abbrev AS s_state,
zip AS s_zip, countryName AS s_country
FROM customer C
INNER JOIN shipAddr AS SA ON C.shipAddrID = SA.shipAddrID
INNER JOIN address A ON A.addressID = SA.addressID
INNER JOIN states S ON A.stateID = S.stateID
INNER JOIN countries CO ON A.countryID = CO.countryID) AS tb2
ON tb1.b_cid = tb2.s_cid;

```

# Show Order and Username

```

SELECT userName, orderID, orderDate FROM `order`
LEFT JOIN customer C ON C.customerID = order.customerID;

```

#Show Product\_Order

```

SELECT userName, PO.orderID, productName, quantOrdered FROM `order`
INNER JOIN customer C ON C.customerID = order.customerID
INNER JOIN product_order PO ON order.orderID = PO.orderID
INNER JOIN product P ON P.productID = PO.productID
ORDER BY orderID;

```

# Filter:

# Show customers who shipping address is NOT IN chosen state

```

SELECT tb1.userName AS userName, abber, countryName
(SELECT firstName, lastName, userName, shipAddrID FROM customer C
WHERE C.customerID
NOT IN (
SELECT customerID
FROM customer C
INNER JOIN shipAddr SA ON C.shipAddrID = SA.shipAddrID
INNER JOIN address A ON A.addressID = SA.addressID
INNER JOIN states S ON S.stateID = A.stateID
WHERE S.abbrev = '[stateID]'
)) as tb1
INNER JOIN shipAddr SA ON tb1.shipAddrID = SA.shipAddrID

```

INNER JOIN address A ON A.addressID = SA.addressID  
INNER JOIN states S ON S.stateID = A.stateID  
INNER JOIN country CO ON A.countryID = CO.countryID

```
# UPDATE product_order
UPDATE `product_order` SET `productID` = '[productID from product]',
`orderID` = '[orderID from order]', `quantOrdered` = '[quantOrdered]'
WHERE `productID` = '[productID from product_order]'
AND `orderID` = '[productID from product_order]' LIMIT 1;
```

```
# UPDATE quantity in stock in Product after order has been made
UPDATE product SET quantStock = (quantStock - [quantOrdered])
WHERE productID = [productID];
```

```
# UPDATE Quantatity in Stock in Products if the same product was attempted to add
UPDATE product SET quantStock = (quantStock + ?)
WHERE productName = [productName] AND retailPrice = [retailPrice] AND wholePrice =
[wholePrice];
```

```
# DELETE a customer
DELETE FROM customer WHERE customerID = '[customerID]' LIMIT 1;
```

```
# DELETE a product_order
DELETE FROM product_order WHERE `productID` = '[productID]'
AND `orderID` = '[orderID]' LIMIT 1;
```

Project Website: <http://web.engr.oregonstate.edu/~goncharn/CS275/Project/Interface.php>



