WebCo

Functionality

- Create and maintain customer profiles.
- Create receipts for customers orders.
- Track and maintain inventory.
- Track customers orders in-transit.

Customer Table

- Customer assigned ID when created.
- Customer information is stored for future orders.
- Customer information is linked using the ID assigned.

Orders, Receipts, Lines

Order	Receipt	Line
Ord_Num Cust_ID Ord_Date	Receipt_Num Cust_ID Ord_Num Ord_Subtotal Ord_Total	Line_Num Receipt_Num Prod_Num Prod_QTY Prod_Price Line_Subtotal

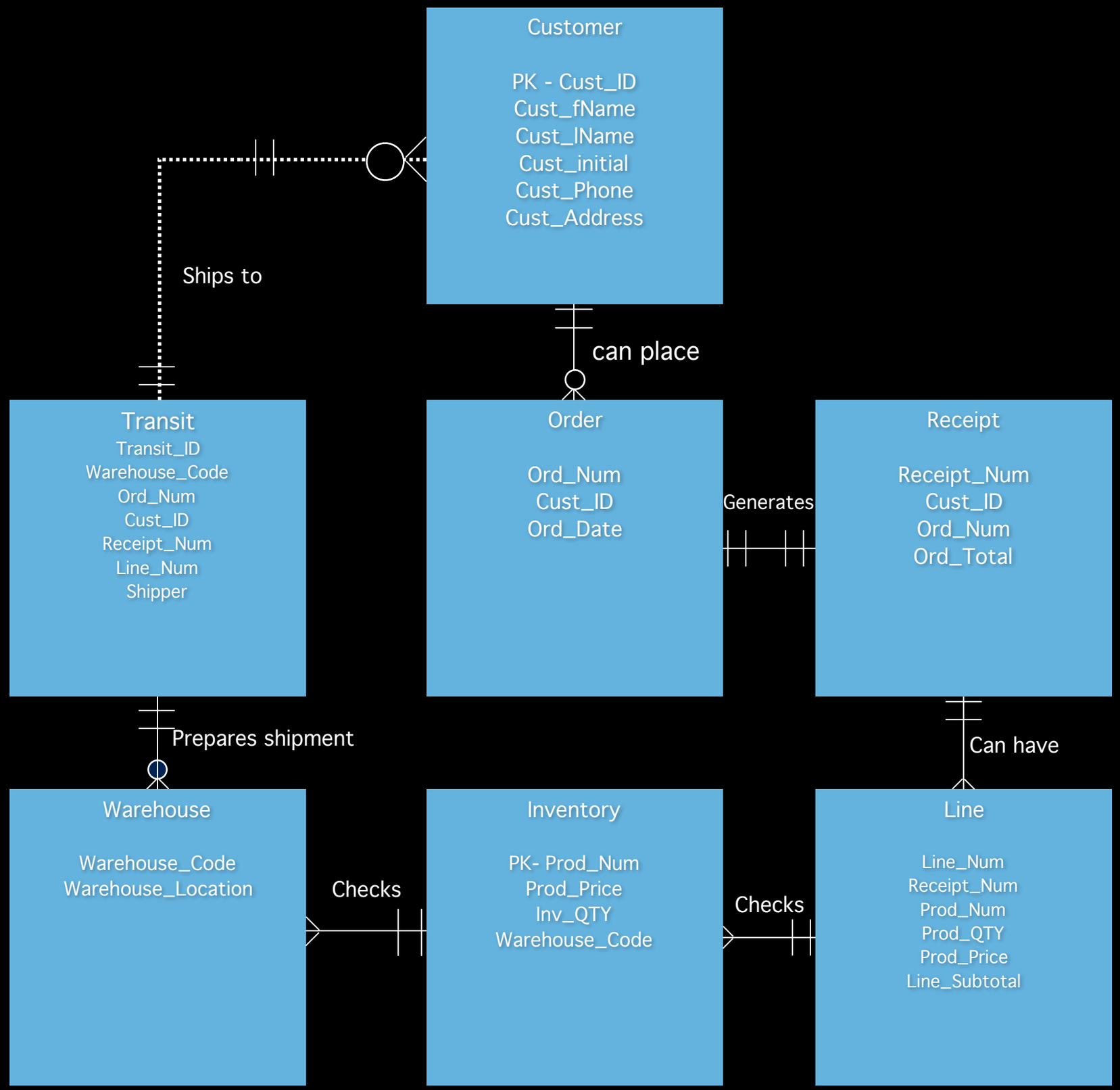
- Orders generate receipts. There's one receipt per order.
- Each receipt represents one and only one order.
- There are many lines per receipt depending on what was ordered.

Inventory, Warehouse, Transit

Inventory	Warehouse	Transit
PK- Prod_Num	Warehouse_Code	Transit_ID
Prod_Price	Warehouse_Location	Warehouse_Code
Inv_QTY		Ord_Num
Warehouse_Code		Cust_ID
		Receipt_Num
		Line_Num
		Shipper

- Inventory keeps track of price, and the quantity of products at different warehouses.
- The warehouse table details the location of the warehouse.
- Transit pulls all the details of the order together as well as tracks the shipper.

ERM


Customer
PK - Cust_ID
Cust_fName
Cust_lName
Cust_initial
Cust_Phone
Cust_Address
Ships to
can place
Transit
Transit_ID
Warehouse_Code
Ord_Num
Cust_ID
Receipt_Num
Line_Num
Shipper
Order
Ord_Num
Cust_ID
Ord_Date
Generates
Receipt
Receipt_Num
Cust_ID
Ord_Num
Ord_Total
Prepares shipment
Can have
Warehouse
Warehouse_Code
Warehouse_Location
Checks
Inventory
PK- Prod_Num
Prod_Price
Inv_QTY
Warehouse_Code
Checks
Line
Line_Num
Receipt_Num
Prod_Num
Prod_QTY
Prod_Price
Line_Subtotal

SQL

Create Tables

```
CREATE TABLE `Webco`.`Customer` ( `Cust_ID` VARCHAR(4) NOT NULL , `Cust_fName` VARCHAR(35) NOT NULL , `Cust_lName` VARCHAR(35) NOT NULL , `Cust_initial` CHAR(1) NOT NULL , `Cust_phone` CHAR(11) NOT NULL , `Cust_address` VARCHAR(50) NOT NULL , PRIMARY KEY (`Cust_ID`)) ENGINE = InnoDB;

CREATE TABLE `Webco`.`Order` ( `Ord_Num` VARCHAR(10) NOT NULL , `Cust_ID` VARCHAR(4) NOT NULL , `Ord_Date` DATE NOT NULL DEFAULT CURRENT_TIMESTAMP , PRIMARY KEY (`Ord_Num`), INDEX `FOREIGN` (`Cust_ID`)) ENGINE = InnoDB;

CREATE TABLE `Webco`.`Receipt` ( `Receipt_Num` INT NOT NULL , `Cust_ID` VARCHAR(4) NOT NULL , `Ord_Num` VARCHAR(10) NOT NULL , PRIMARY KEY (`Receipt_Num`), INDEX `FOREIGN` (`Cust_ID`), INDEX `FOREIGN2` (`Ord_Num`)) ENGINE = InnoDB;

CREATE TABLE `Webco`.`Inventory` ( `Prod_Num` VARCHAR(8) NOT NULL , `Prod_Price` DECIMAL(9,2) NOT NULL , `Inv_Qty` INT(5) NOT NULL , `Warehouse_Code` INT(5) NOT NULL , PRIMARY KEY (`Prod_Num`)) ENGINE = InnoDB

CREATE TABLE `Webco`.`Transit` ( `Transit_ID` VARCHAR(8) NOT NULL , `Warehouse_Code` INT(2) NOT NULL , `Ord_Num` VARCHAR(10) NOT NULL , `Cust_ID` VARCHAR(4) NOT NULL , `Reciept_Num` INT NOT NULL , `Line_Num` INT(2) NOT NULL , `Shipper` VARCHAR NOT NULL , PRIMARY KEY (`Transit_ID`), INDEX `FOREIGN` (`Warehouse_Code`), INDEX `FOREIGN` (`Ord_Num`), INDEX `FOREIGN` (`Cust_ID`), INDEX `FOREIGN` (`Reciept_Num`), INDEX `FOREIGN` (`Line_Num`)) ENGINE = InnoDB;
```

SQL

Insert/Update Statements

```
INSERT INTO Receipt VALUES(12345678, 1111,  20);

INSERT INTO Line VALUES(1, 12345678, 214, ‘1’, 200);

UPDATE Receipt
SET Ord_Total = (Select Line_Subtotal from Line where Receipt_Num = 12345678) * 0.0819;

UPDATE Inventory
SET Inv_qty = Inv_qty - 200;
WHERE Prod_num = ‘1’;
```

Limitations

End


A SQL QUERY
GOES INTO A BAR,
WALKS UP TO TWO TABLES
AND ASKS, "CAN I JOIN YOU?"