COMP 421 Group 46
(PostgreSQL)

Project Deliverable 2

Due: February 29, 2016

Henry Lin, Harvey Yang,
Kelley Zhao, Yiwei Xia

Bonus Marks

In an attempt to reconcile our below average mark for the first deliverable, we attempted to do some things to earn bonus points on this deliverable.

1. We imported real, current data to populate our Team and Player tables using scripts written in Java by extracting real data from the NBA website.
2. In questions 5-7 we used many SQL features that were not taught in class. Some of these include:
    a. wild cards
    b. rules for views
    c. aliases
    d. loops
    e. an enum type definition
    f. use of functions and use of variables

Question 1:

Customer(customerID, firstName, lastName, email, address)

Orders(orderID, payment_method, orderDate)

Product(productID, productName, retailPrice, manufacturerPrice)

TeamMerchandise(productID) where productID REFERENCES Product

PlayerMerchandise(productID) where productID REFERENCES Product

Team(teamName, accountNumber)

Player(playerNumber, accountNumber, firstName, lastName, weakEntity(teamName))
where teamName REFERENCES Team

Warehouse(warehouseID, address)

Shipment(shipmentID, shipmentDate)

Supplier(supplierName, address)

CustomerOrder(customerID, orderID)
where customerID REFERENCES(Customer) and orderID REFERENCES(Order)

ProductOrderWarehouse(quantity,productID,orderID,warehouseID)
where productID REFERENCES(Product)
orderID REFERENCES(Order)
warehouseID REFERENCES(Warehouse)

ShipmentSupplier(supplierName,shipmentID)
where supplierName REFERENCES(Supplier)
shipmentID REFERENCES(Shipment)

ShipmentWarehouse(shipmentID,warehouseID)
where warehouseID REFERENCES(Warehouser)
shipmentID REFERENCES(Shipment)

ShipmentProduct(productID,shipmentID,quantity)
where productID REFERENCES(Product)
shipmentID REFERENCES(Shipment)

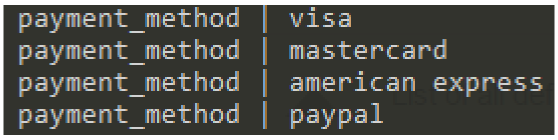
WarehouseProduct(quantity,warehouseID,productID)
where productID REFERENCES(Product)
shipmentID REFERENCES(Shipment)

TeamMerchandiseTeam(productID,teamName)
where productID REFERENCES(TeamMerchandise)
teamName REFERENCES(Team)

PlayerMerchandisePlayer(productID,playerNumber,teamName)
where productID REFERENCES(PlayerMerchandise)
(playerNumber,teamName) REFERENCES(Player)

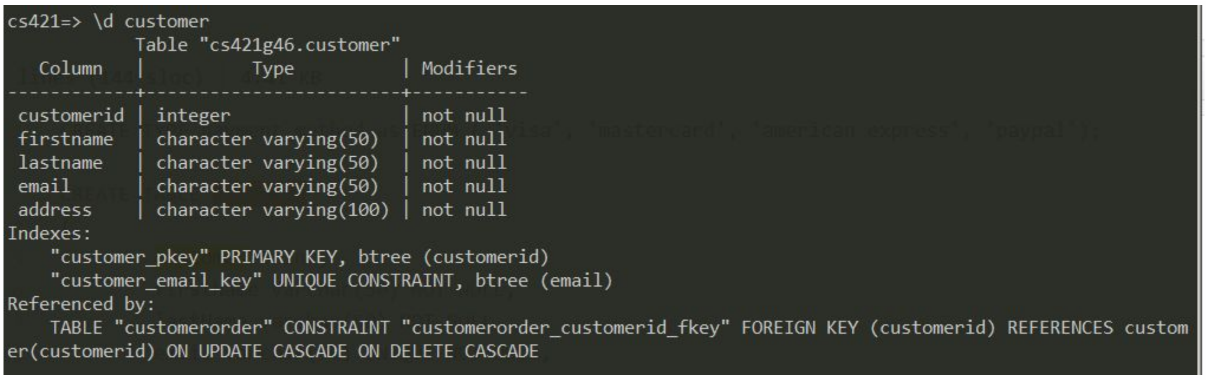
Question 2:

```
CREATE TYPE payment_method as ENUM ( 'visa', 'mastercard', 'american express', 'paypal');
```



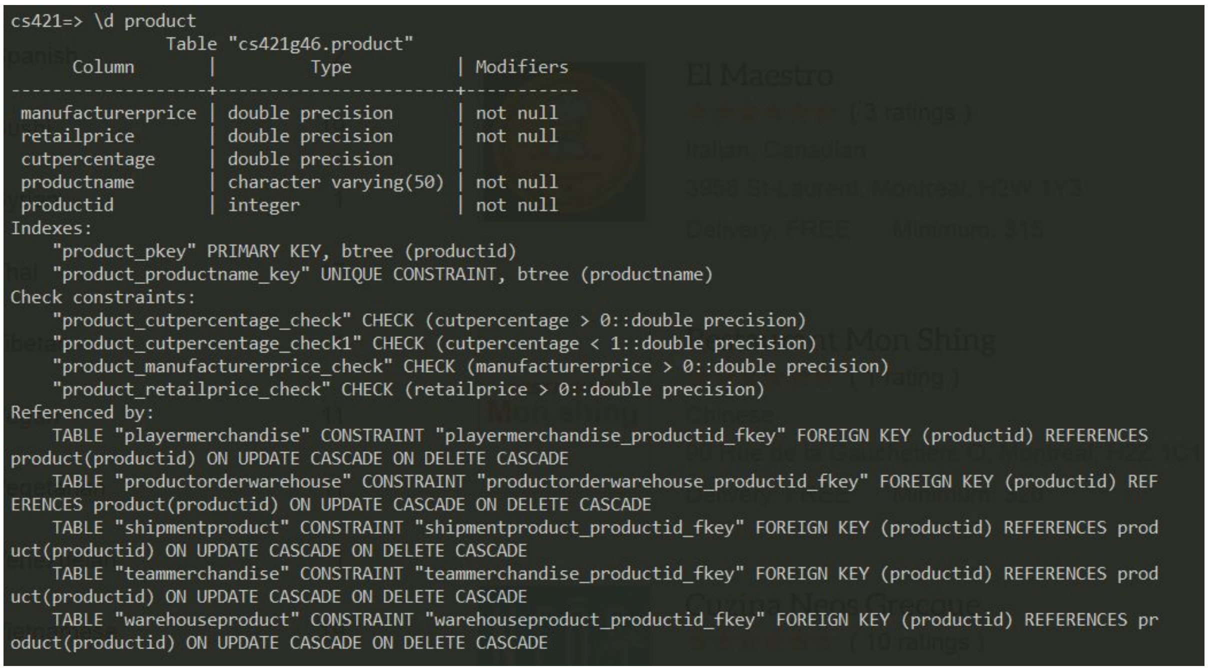
```
 payment_method | visa
 payment_method | mastercard
 payment_method | american express
 payment_method | paypal
```

```
CREATE TABLE Customer
(
        customerID int,
        firstName varchar(50) NOT NULL,
        lastName varchar(50) NOT NULL,
        email varchar(50) UNIQUE NOT NULL,
        address varchar(100) NOT NULL,
        PRIMARY KEY (customerID)
);
```

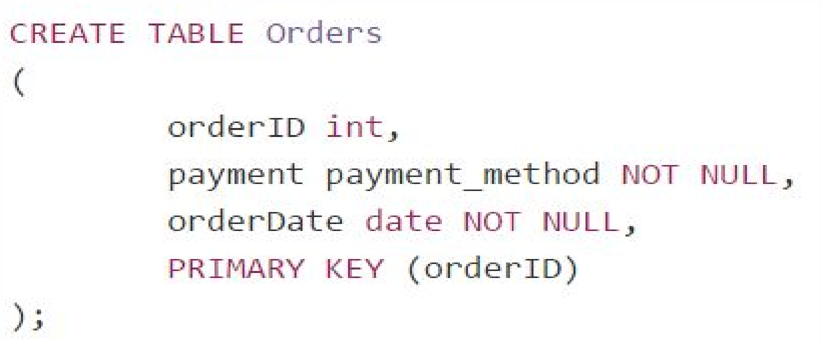


```
cs421=> \d customer
            Table "cs421g46.customer"
  Column   |          Type          | Modifiers
-----------+------------------------+-----------
 customerid | integer                | not null
 firstname  | character varying(50)  | not null
 lastname   | character varying(50)  | not null
 email      | character varying(50)  | not null
 address    | character varying(100) | not null
Indexes:
    "customer_pkey" PRIMARY KEY, btree (customerid)
    "customer_email_key" UNIQUE CONSTRAINT, btree (email)
Referenced by:
    TABLE "customerorder" CONSTRAINT "customerorder_customerid_fkey" FOREIGN KEY (customerid) REFERENCES custom
er(customerid) ON UPDATE CASCADE ON DELETE CASCADE
```

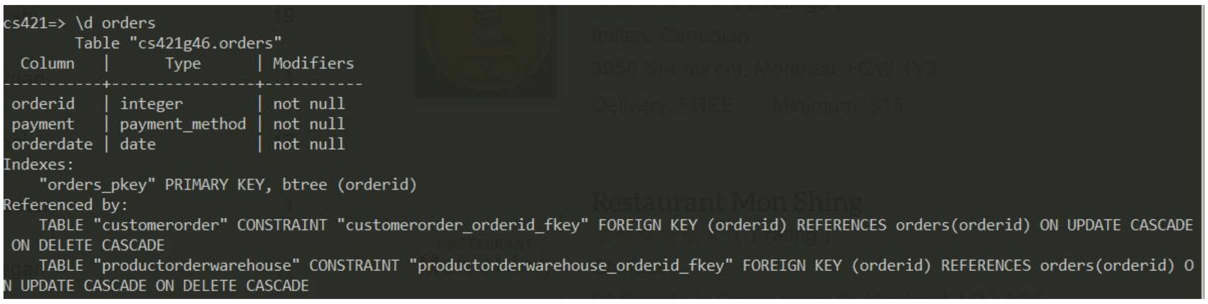
```sql
CREATE TABLE Product
(
        manufacturerPrice double precision NOT NULL CHECK (manufacturerPrice > 0),
        retailPrice double precision NOT NULL CHECK (retailPrice > 0),
        cutPercentage double precision,
        CHECK (cutPercentage > 0),
        CHECK (cutPercentage < 1),
        productName varchar(50) UNIQUE NOT NULL,
        productID int,
        PRIMARY KEY (productID)
);
```



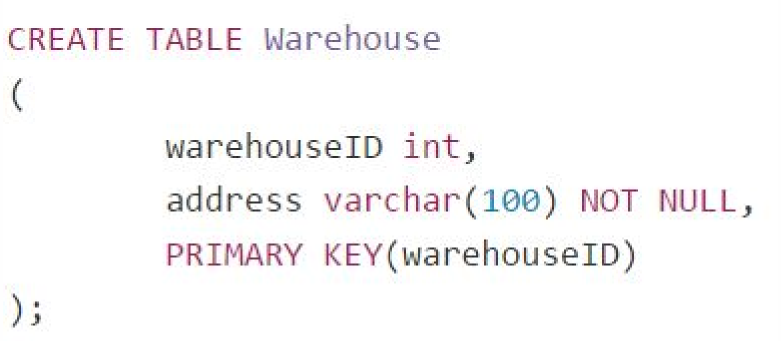
```
cs421=> \d product
              Table "cs421g46.product"
      Column       |         Type          | Modifiers
-------------------+-----------------------+-----------
 manufacturerprice | double precision      | not null
 retailprice       | double precision      | not null
 cutpercentage     | double precision      |
 productname       | character varying(50) | not null
 productid         | integer               | not null
Indexes:
    "product_pkey" PRIMARY KEY, btree (productid)
    "product_productname_key" UNIQUE CONSTRAINT, btree (productname)
Check constraints:
    "product_cutpercentage_check" CHECK (cutpercentage > 0::double precision)
    "product_cutpercentage_check1" CHECK (cutpercentage < 1::double precision)
    "product_manufacturerprice_check" CHECK (manufacturerprice > 0::double precision)
    "product_retailprice_check" CHECK (retailprice > 0::double precision)
Referenced by:
    TABLE "playermerchandise" CONSTRAINT "playermerchandise_productid_fkey" FOREIGN KEY (productid) REFERENCES
product(productid) ON UPDATE CASCADE ON DELETE CASCADE
    TABLE "productorderwarehouse" CONSTRAINT "productorderwarehouse_productid_fkey" FOREIGN KEY (productid) REF
ERENCES product(productid) ON UPDATE CASCADE ON DELETE CASCADE
    TABLE "shipmentproduct" CONSTRAINT "shipmentproduct_productid_fkey" FOREIGN KEY (productid) REFERENCES prod
uct(productid) ON UPDATE CASCADE ON DELETE CASCADE
    TABLE "teammerchandise" CONSTRAINT "teammerchandise_productid_fkey" FOREIGN KEY (productid) REFERENCES prod
uct(productid) ON UPDATE CASCADE ON DELETE CASCADE
    TABLE "warehouseproduct" CONSTRAINT "warehouseproduct_productid_fkey" FOREIGN KEY (productid) REFERENCES pr
oduct(productid) ON UPDATE CASCADE ON DELETE CASCADE
```



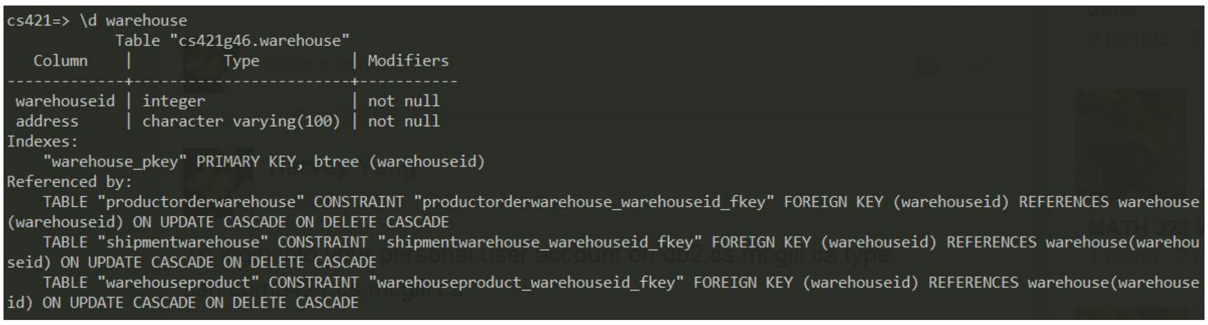
```sql
CREATE TABLE Orders
(
        orderID int,
        payment payment_method NOT NULL,
        orderDate date NOT NULL,
        PRIMARY KEY (orderID)
);
```



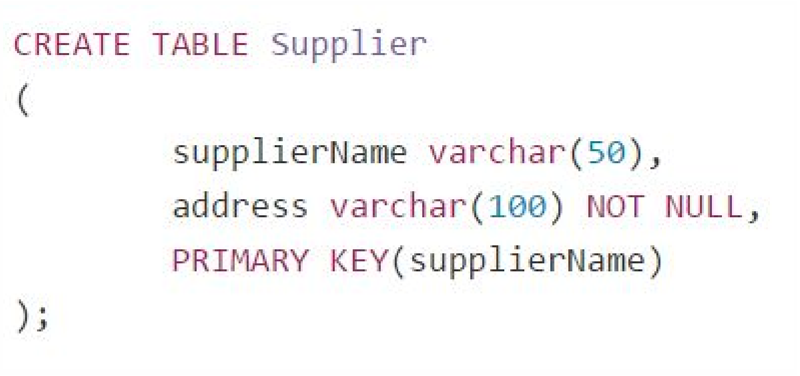
```
cs421=> \d orders
        Table "cs421g46.orders"
  Column   |      Type      | Modifiers
-----------+----------------+-----------
 orderid   | integer        | not null
 payment   | payment_method | not null
 orderdate | date           | not null
Indexes:
    "orders_pkey" PRIMARY KEY, btree (orderid)
Referenced by:
    TABLE "customerorder" CONSTRAINT "customerorder_orderid_fkey" FOREIGN KEY (orderid) REFERENCES orders(orderid) ON UPDATE CASCADE
 ON DELETE CASCADE
    TABLE "productorderwarehouse" CONSTRAINT "productorderwarehouse_orderid_fkey" FOREIGN KEY (orderid) REFERENCES orders(orderid) O
N UPDATE CASCADE ON DELETE CASCADE
```



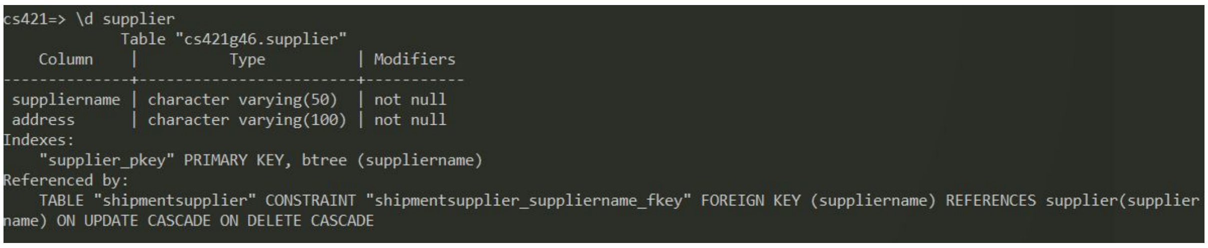
```sql
CREATE TABLE Warehouse
(
        warehouseID int,
        address varchar(100) NOT NULL,
        PRIMARY KEY(warehouseID)
);
```



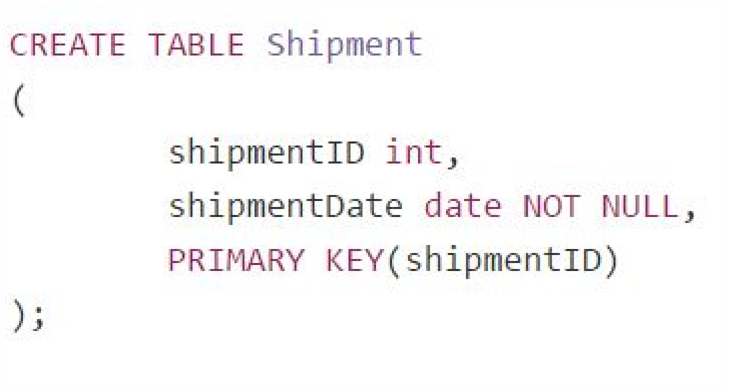
```
cs421=> \d warehouse
            Table "cs421g46.warehouse"
   Column    |          Type          | Modifiers
-------------+------------------------+-----------
 warehouseid | integer                | not null
 address     | character varying(100) | not null
Indexes:
    "warehouse_pkey" PRIMARY KEY, btree (warehouseid)
Referenced by:
    TABLE "productorderwarehouse" CONSTRAINT "productorderwarehouse_warehouseid_fkey" FOREIGN KEY (warehouseid) REFERENCES warehouse
(warehouseid) ON UPDATE CASCADE ON DELETE CASCADE
    TABLE "shipmentwarehouse" CONSTRAINT "shipmentwarehouse_warehouseid_fkey" FOREIGN KEY (warehouseid) REFERENCES warehouse(warehou
seid) ON UPDATE CASCADE ON DELETE CASCADE
    TABLE "warehouseproduct" CONSTRAINT "warehouseproduct_warehouseid_fkey" FOREIGN KEY (warehouseid) REFERENCES warehouse(warehouse
id) ON UPDATE CASCADE ON DELETE CASCADE
```



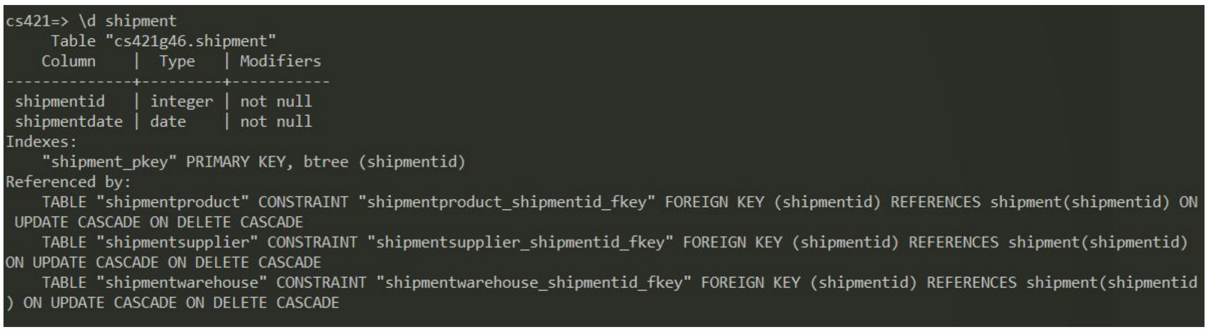
```
CREATE TABLE Supplier
(
        supplierName varchar(50),
        address varchar(100) NOT NULL,
        PRIMARY KEY(supplierName)
);
```



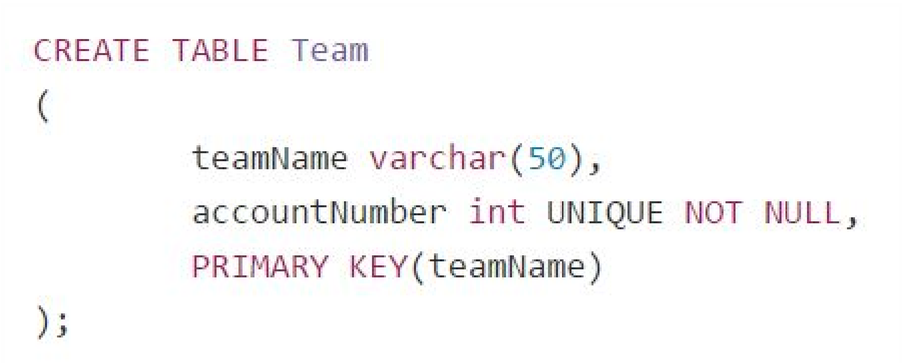
```
cs421=> \d supplier
             Table "cs421g46.supplier"
    Column    |          Type          | Modifiers
--------------+------------------------+-----------
 suppliername | character varying(50)  | not null
 address      | character varying(100) | not null
Indexes:
    "supplier_pkey" PRIMARY KEY, btree (suppliername)
Referenced by:
    TABLE "shipmentsupplier" CONSTRAINT "shipmentsupplier_suppliername_fkey" FOREIGN KEY (suppliername) REFERENCES supplier(supplier
name) ON UPDATE CASCADE ON DELETE CASCADE
```



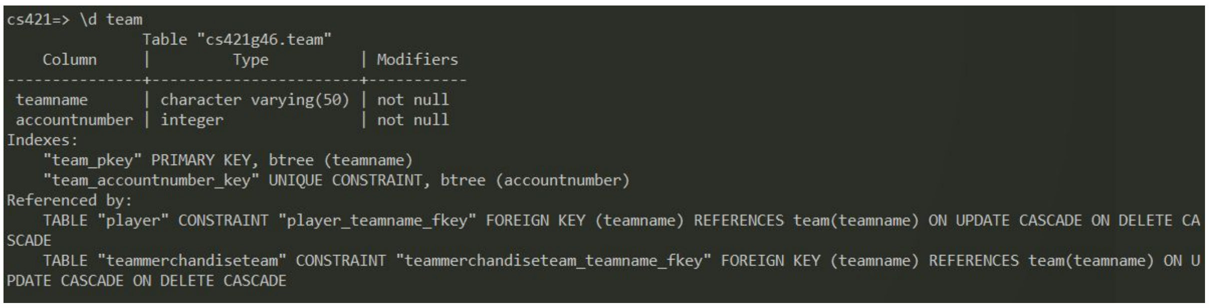
```
CREATE TABLE Shipment
(
        shipmentID int,
        shipmentDate date NOT NULL,
        PRIMARY KEY(shipmentID)
);
```



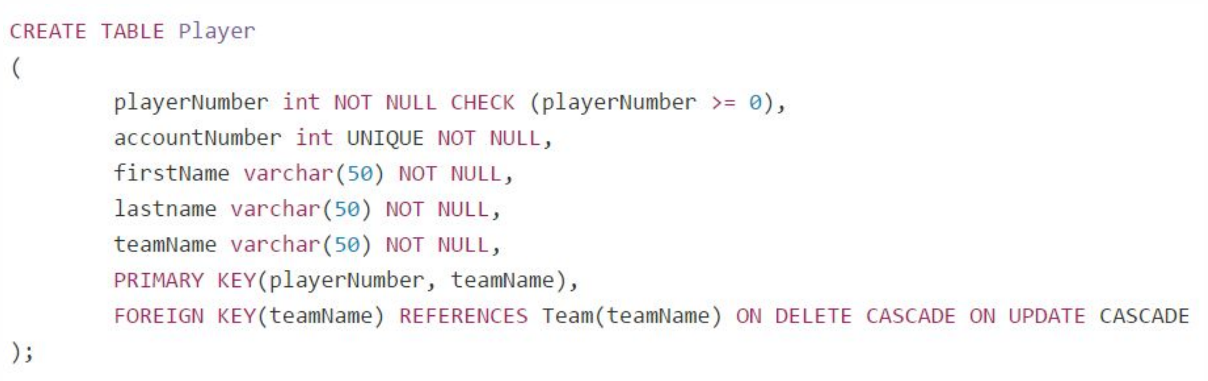
```
cs421=> \d shipment
     Table "cs421g46.shipment"
    Column    |  Type   | Modifiers
--------------+---------+-----------
 shipmentid   | integer | not null
 shipmentdate | date    | not null
Indexes:
    "shipment_pkey" PRIMARY KEY, btree (shipmentid)
Referenced by:
    TABLE "shipmentproduct" CONSTRAINT "shipmentproduct_shipmentid_fkey" FOREIGN KEY (shipmentid) REFERENCES shipment(shipmentid) ON
 UPDATE CASCADE ON DELETE CASCADE
    TABLE "shipmentsupplier" CONSTRAINT "shipmentsupplier_shipmentid_fkey" FOREIGN KEY (shipmentid) REFERENCES shipment(shipmentid)
ON UPDATE CASCADE ON DELETE CASCADE
    TABLE "shipmentwarehouse" CONSTRAINT "shipmentwarehouse_shipmentid_fkey" FOREIGN KEY (shipmentid) REFERENCES shipment(shipmentid
) ON UPDATE CASCADE ON DELETE CASCADE
```



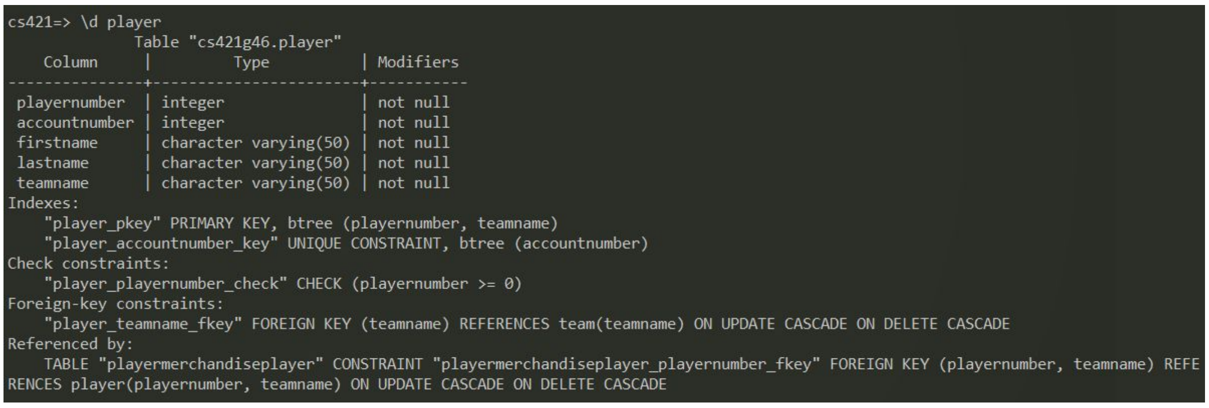
```sql
CREATE TABLE Team
(
        teamName varchar(50),
        accountNumber int UNIQUE NOT NULL,
        PRIMARY KEY(teamName)
);
```



```
cs421=> \d team
              Table "cs421g46.team"
    Column     |         Type          | Modifiers
---------------+-----------------------+-----------
 teamname      | character varying(50) | not null
 accountnumber | integer               | not null
Indexes:
    "team_pkey" PRIMARY KEY, btree (teamname)
    "team_accountnumber_key" UNIQUE CONSTRAINT, btree (accountnumber)
Referenced by:
    TABLE "player" CONSTRAINT "player_teamname_fkey" FOREIGN KEY (teamname) REFERENCES team(teamname) ON UPDATE CASCADE ON DELETE CA
SCADE
    TABLE "teammerchandiseteam" CONSTRAINT "teammerchandiseteam_teamname_fkey" FOREIGN KEY (teamname) REFERENCES team(teamname) ON U
PDATE CASCADE ON DELETE CASCADE
```

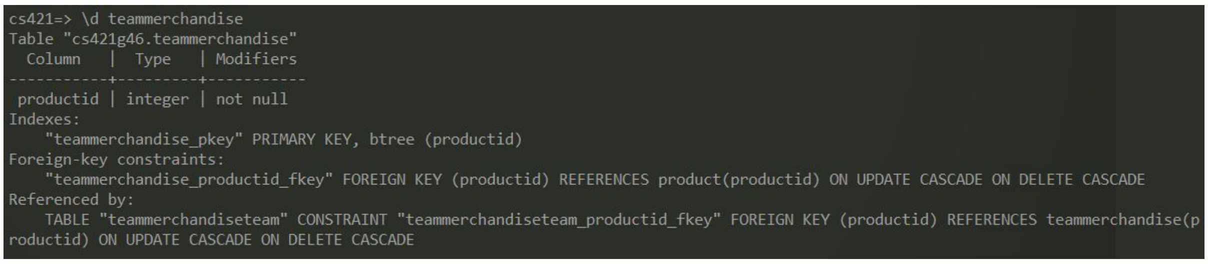


```sql
CREATE TABLE Player
(
        playerNumber int NOT NULL CHECK (playerNumber >= 0),
        accountNumber int UNIQUE NOT NULL,
        firstName varchar(50) NOT NULL,
        lastname varchar(50) NOT NULL,
        teamName varchar(50) NOT NULL,
        PRIMARY KEY(playerNumber, teamName),
        FOREIGN KEY(teamName) REFERENCES Team(teamName) ON DELETE CASCADE ON UPDATE CASCADE
);
```



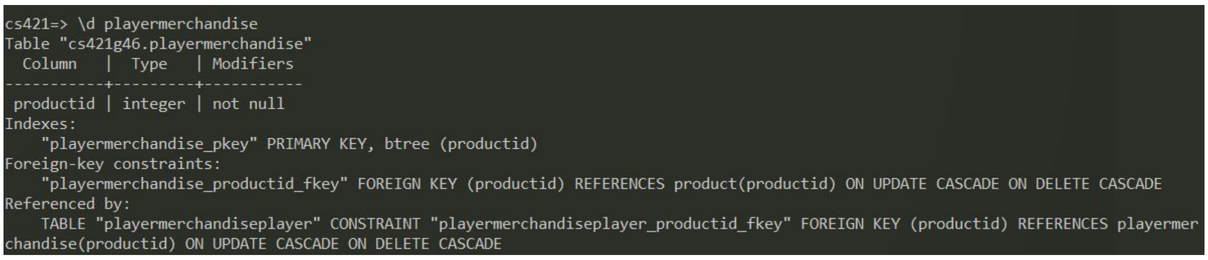
```
cs421=> \d player
             Table "cs421g46.player"
    Column     |         Type          | Modifiers
---------------+-----------------------+-----------
 playernumber  | integer               | not null
 accountnumber | integer               | not null
 firstname     | character varying(50) | not null
 lastname      | character varying(50) | not null
 teamname      | character varying(50) | not null
Indexes:
    "player_pkey" PRIMARY KEY, btree (playernumber, teamname)
    "player_accountnumber_key" UNIQUE CONSTRAINT, btree (accountnumber)
Check constraints:
    "player_playernumber_check" CHECK (playernumber >= 0)
Foreign-key constraints:
    "player_teamname_fkey" FOREIGN KEY (teamname) REFERENCES team(teamname) ON UPDATE CASCADE ON DELETE CASCADE
Referenced by:
    TABLE "playermerchandiseplayer" CONSTRAINT "playermerchandiseplayer_playernumber_fkey" FOREIGN KEY (playernumber, teamname) REFE
RENCES player(playernumber, teamname) ON UPDATE CASCADE ON DELETE CASCADE
```

```
CREATE TABLE TeamMerchandise
(
        productID int,
        PRIMARY KEY(productID),
        FOREIGN KEY (productID) REFERENCES Product(productID) ON DELETE CASCADE ON UPDATE CASCADE
);
```



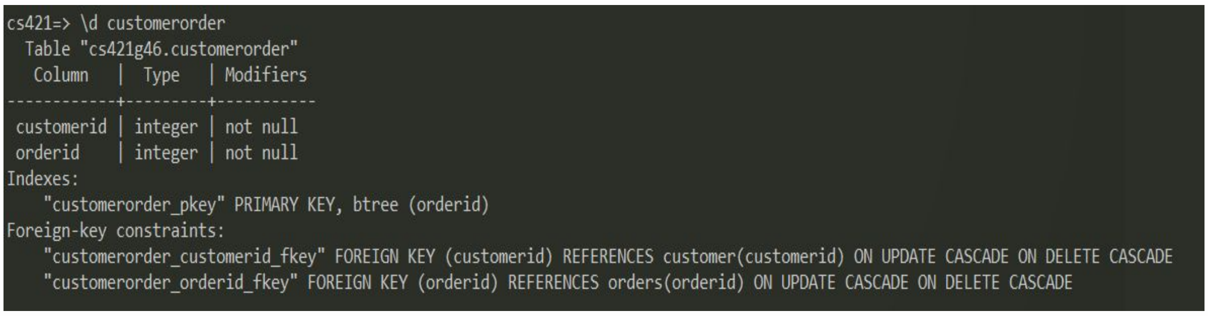
```
cs421=> \d teammerchandise
Table "cs421g46.teammerchandise"
  Column   |  Type   | Modifiers
-----------+---------+-----------
 productid | integer | not null
Indexes:
    "teammerchandise_pkey" PRIMARY KEY, btree (productid)
Foreign-key constraints:
    "teammerchandise_productid_fkey" FOREIGN KEY (productid) REFERENCES product(productid) ON UPDATE CASCADE ON DELETE CASCADE
Referenced by:
    TABLE "teammerchandiseteam" CONSTRAINT "teammerchandiseteam_productid_fkey" FOREIGN KEY (productid) REFERENCES teammerchandise(p
roductid) ON UPDATE CASCADE ON DELETE CASCADE
```

```
CREATE TABLE PlayerMerchandise
(
        productID int,
        PRIMARY KEY (productID),
        FOREIGN KEY (productID) REFERENCES Product(productID) ON DELETE CASCADE ON UPDATE CASCADE
);
```

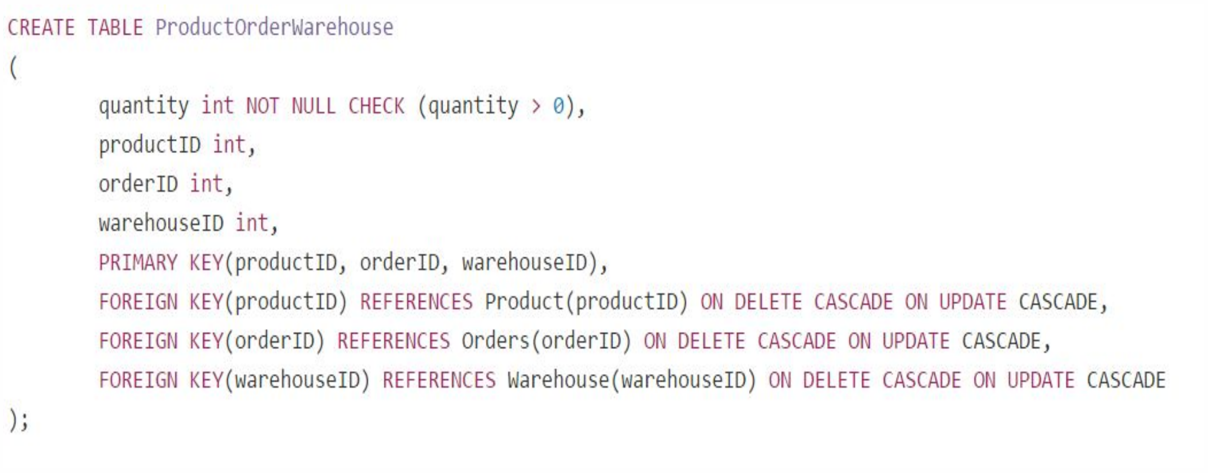


```
cs421=> \d playermerchandise
Table "cs421g46.playermerchandise"
  Column   |  Type   | Modifiers
-----------+---------+-----------
 productid | integer | not null
Indexes:
    "playermerchandise_pkey" PRIMARY KEY, btree (productid)
Foreign-key constraints:
    "playermerchandise_productid_fkey" FOREIGN KEY (productid) REFERENCES product(productid) ON UPDATE CASCADE ON DELETE CASCADE
Referenced by:
    TABLE "playermerchandiseplayer" CONSTRAINT "playermerchandiseplayer_productid_fkey" FOREIGN KEY (productid) REFERENCES playermer
chandise(productid) ON UPDATE CASCADE ON DELETE CASCADE
```

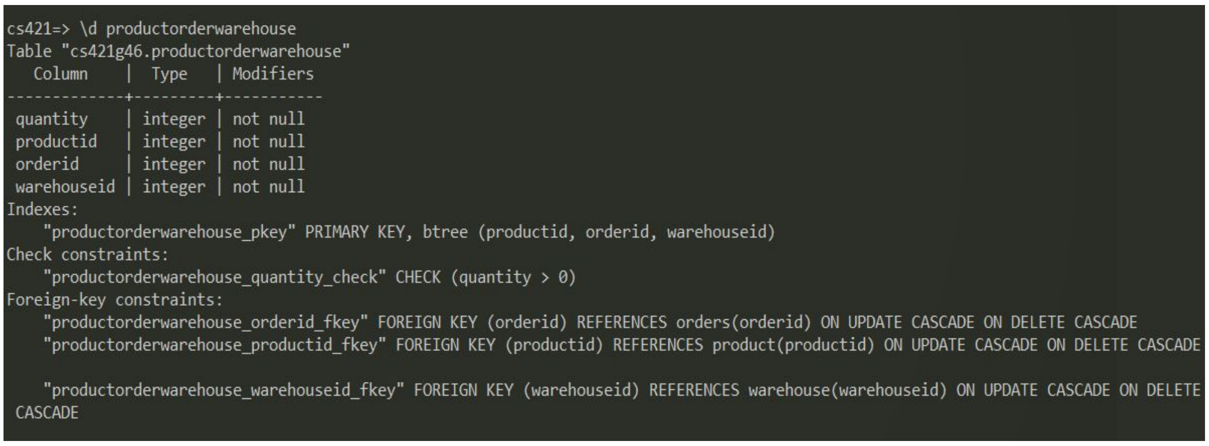
```
CREATE TABLE CustomerOrder
(
        customerID int NOT NULL,
        orderID int,
        PRIMARY KEY(orderID),
        FOREIGN KEY(customerID) REFERENCES Customer(customerID) ON DELETE CASCADE ON UPDATE CASCADE,
        FOREIGN KEY(orderID) REFERENCES Orders(orderID) ON DELETE CASCADE ON UPDATE CASCADE
);
```



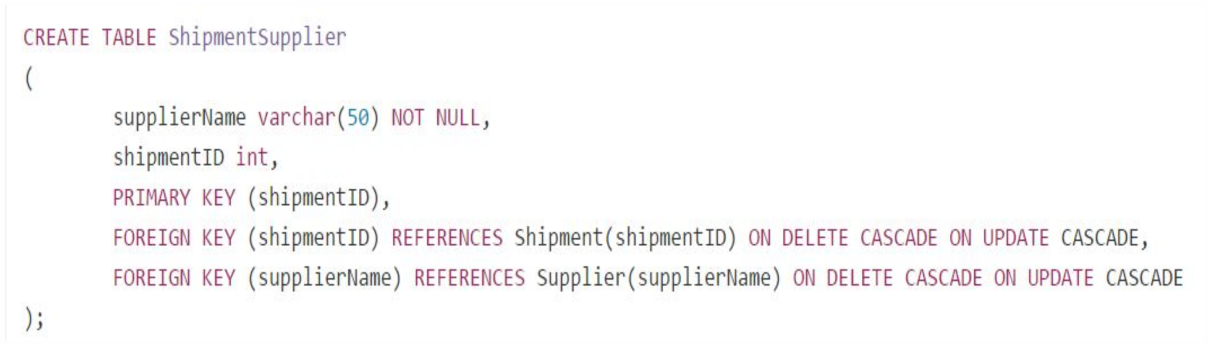
```
cs421=> \d customerorder
  Table "cs421g46.customerorder"
   Column   |  Type   | Modifiers
------------+---------+-----------
 customerid | integer | not null
 orderid    | integer | not null
Indexes:
    "customerorder_pkey" PRIMARY KEY, btree (orderid)
Foreign-key constraints:
    "customerorder_customerid_fkey" FOREIGN KEY (customerid) REFERENCES customer(customerid) ON UPDATE CASCADE ON DELETE CASCADE
    "customerorder_orderid_fkey" FOREIGN KEY (orderid) REFERENCES orders(orderid) ON UPDATE CASCADE ON DELETE CASCADE
```



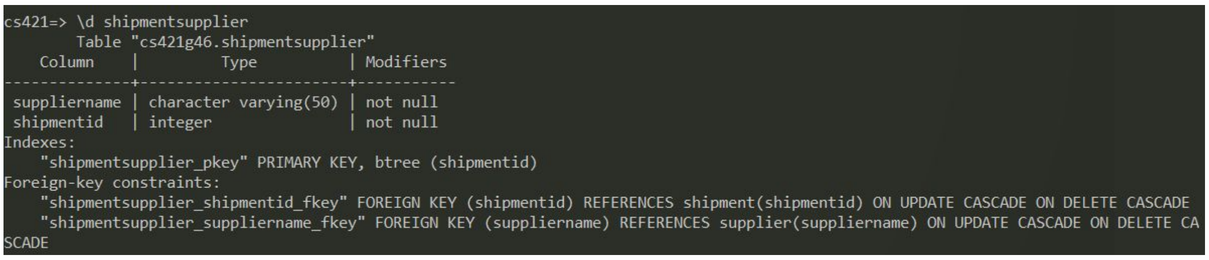
```
CREATE TABLE ProductOrderWarehouse
(
        quantity int NOT NULL CHECK (quantity > 0),
        productID int,
        orderID int,
        warehouseID int,
        PRIMARY KEY(productID, orderID, warehouseID),
        FOREIGN KEY(productID) REFERENCES Product(productID) ON DELETE CASCADE ON UPDATE CASCADE,
        FOREIGN KEY(orderID) REFERENCES Orders(orderID) ON DELETE CASCADE ON UPDATE CASCADE,
        FOREIGN KEY(warehouseID) REFERENCES Warehouse(warehouseID) ON DELETE CASCADE ON UPDATE CASCADE
);
```



```
cs421=> \d productorderwarehouse
Table "cs421g46.productorderwarehouse"
   Column    |  Type   | Modifiers
-------------+---------+-----------
 quantity    | integer | not null
 productid   | integer | not null
 orderid     | integer | not null
 warehouseid | integer | not null
Indexes:
    "productorderwarehouse_pkey" PRIMARY KEY, btree (productid, orderid, warehouseid)
Check constraints:
    "productorderwarehouse_quantity_check" CHECK (quantity > 0)
Foreign-key constraints:
    "productorderwarehouse_orderid_fkey" FOREIGN KEY (orderid) REFERENCES orders(orderid) ON UPDATE CASCADE ON DELETE CASCADE
    "productorderwarehouse_productid_fkey" FOREIGN KEY (productid) REFERENCES product(productid) ON UPDATE CASCADE ON DELETE CASCADE

    "productorderwarehouse_warehouseid_fkey" FOREIGN KEY (warehouseid) REFERENCES warehouse(warehouseid) ON UPDATE CASCADE ON DELETE
 CASCADE
```

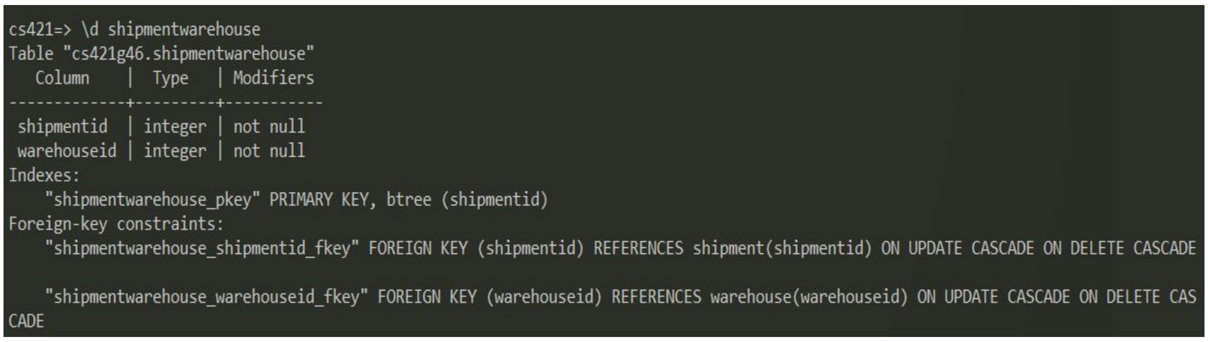


```
CREATE TABLE ShipmentSupplier
(
        supplierName varchar(50) NOT NULL,
        shipmentID int,
        PRIMARY KEY (shipmentID),
        FOREIGN KEY (shipmentID) REFERENCES Shipment(shipmentID) ON DELETE CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (supplierName) REFERENCES Supplier(supplierName) ON DELETE CASCADE ON UPDATE CASCADE
);
```



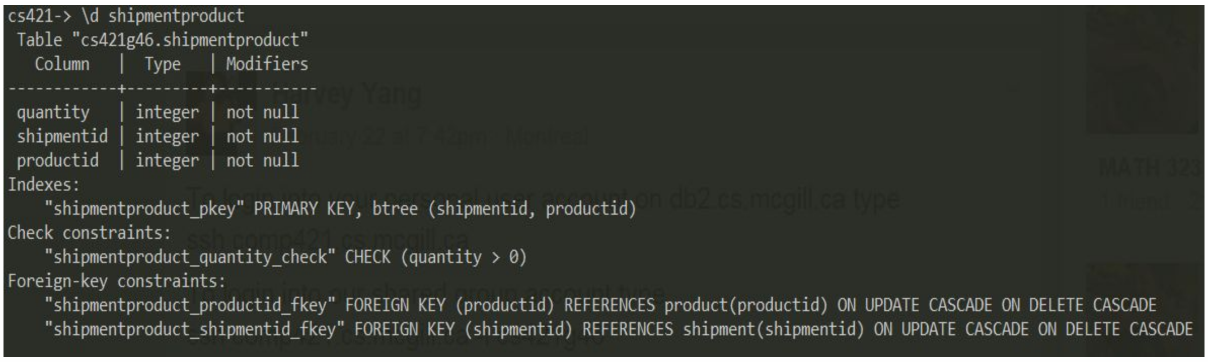
```
cs421=> \d shipmentsupplier
        Table "cs421g46.shipmentsupplier"
    Column    |         Type          | Modifiers
--------------+-----------------------+-----------
 suppliername | character varying(50) | not null
 shipmentid   | integer               | not null
Indexes:
    "shipmentsupplier_pkey" PRIMARY KEY, btree (shipmentid)
Foreign-key constraints:
    "shipmentsupplier_shipmentid_fkey" FOREIGN KEY (shipmentid) REFERENCES shipment(shipmentid) ON UPDATE CASCADE ON DELETE CASCADE
    "shipmentsupplier_suppliername_fkey" FOREIGN KEY (suppliername) REFERENCES supplier(suppliername) ON UPDATE CASCADE ON DELETE CA
SCADE
```

```sql
CREATE TABLE ShipmentWarehouse
(
        shipmentID int,
        warehouseID int NOT NULL,
        PRIMARY KEY(shipmentID),
        FOREIGN KEY(shipmentID) REFERENCES Shipment(shipmentID) ON DELETE CASCADE ON UPDATE CASCADE,
        FOREIGN KEY(warehouseID) REFERENCES Warehouse(warehouseID) ON DELETE CASCADE ON UPDATE CASCADE
);
```



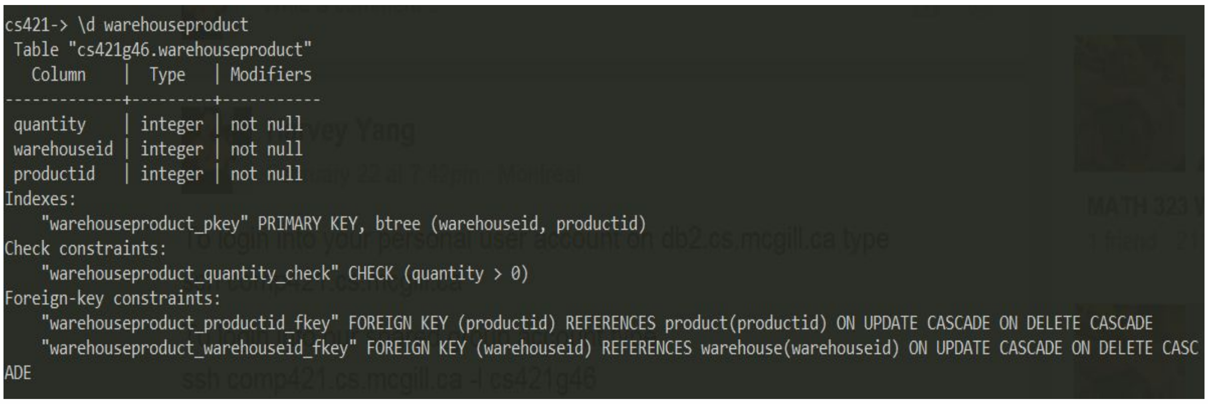
```
cs421=> \d shipmentwarehouse
Table "cs421g46.shipmentwarehouse"
   Column    |  Type   | Modifiers
-------------+---------+-----------
 shipmentid  | integer | not null
 warehouseid | integer | not null
Indexes:
    "shipmentwarehouse_pkey" PRIMARY KEY, btree (shipmentid)
Foreign-key constraints:
    "shipmentwarehouse_shipmentid_fkey" FOREIGN KEY (shipmentid) REFERENCES shipment(shipmentid) ON UPDATE CASCADE ON DELETE CASCADE

    "shipmentwarehouse_warehouseid_fkey" FOREIGN KEY (warehouseid) REFERENCES warehouse(warehouseid) ON UPDATE CASCADE ON DELETE CAS
CADE
```

```sql
CREATE TABLE ShipmentProduct
(
        quantity int NOT NULL CHECK (quantity > 0),
        shipmentID int,
        productID int,
        PRIMARY KEY (shipmentID, productID),
        FOREIGN KEY (shipmentID) REFERENCES Shipment(shipmentID) ON DELETE CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (productID) REFERENCES Product(productID) ON DELETE CASCADE ON UPDATE CASCADE
);
```



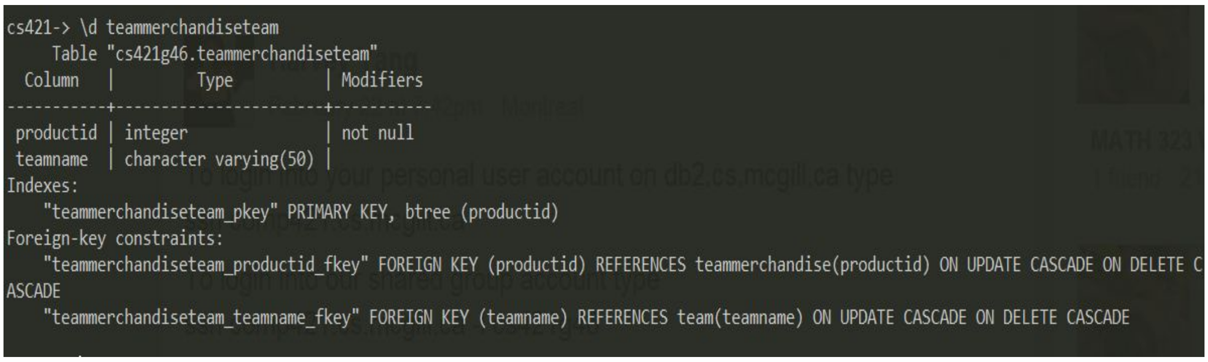
```
cs421-> \d shipmentproduct
 Table "cs421g46.shipmentproduct"
   Column   |  Type   | Modifiers
------------+---------+-----------
 quantity   | integer | not null
 shipmentid | integer | not null
 productid  | integer | not null
Indexes:
    "shipmentproduct_pkey" PRIMARY KEY, btree (shipmentid, productid)
Check constraints:
    "shipmentproduct_quantity_check" CHECK (quantity > 0)
Foreign-key constraints:
    "shipmentproduct_productid_fkey" FOREIGN KEY (productid) REFERENCES product(productid) ON UPDATE CASCADE ON DELETE CASCADE
    "shipmentproduct_shipmentid_fkey" FOREIGN KEY (shipmentid) REFERENCES shipment(shipmentid) ON UPDATE CASCADE ON DELETE CASCADE
```

```sql
CREATE TABLE WarehouseProduct
(
	quantity int NOT NULL CHECK (quantity > 0),
	warehouseID int,
	productID int,
	PRIMARY KEY(warehouseID, productID),
	FOREIGN KEY (warehouseID) REFERENCES Warehouse(warehouseID) ON DELETE CASCADE ON UPDATE CASCADE,
	FOREIGN KEY (productID) REFERENCES Product(productID) ON DELETE CASCADE ON UPDATE CASCADE
);
```

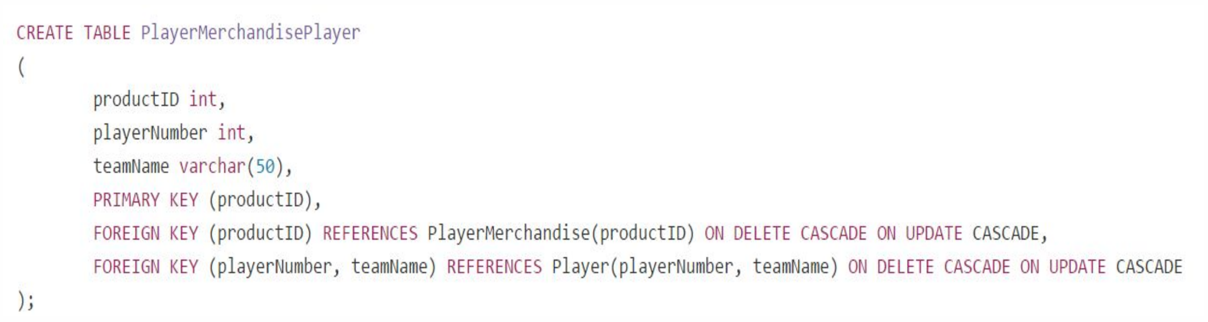


```
cs421-> \d warehouseproduct
 Table "cs421g46.warehouseproduct"
   Column    |  Type   | Modifiers
-------------+---------+-----------
 quantity    | integer | not null
 warehouseid | integer | not null
 productid   | integer | not null
Indexes:
    "warehouseproduct_pkey" PRIMARY KEY, btree (warehouseid, productid)
Check constraints:
    "warehouseproduct_quantity_check" CHECK (quantity > 0)
Foreign-key constraints:
    "warehouseproduct_productid_fkey" FOREIGN KEY (productid) REFERENCES product(productid) ON UPDATE CASCADE ON DELETE CASCADE
    "warehouseproduct_warehouseid_fkey" FOREIGN KEY (warehouseid) REFERENCES warehouse(warehouseid) ON UPDATE CASCADE ON DELETE CASC
ADE
```

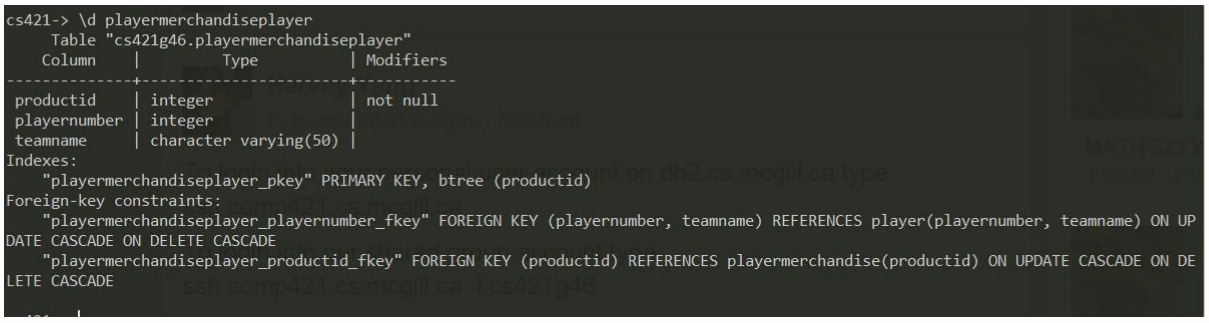
```sql
CREATE TABLE TeamMerchandiseTeam
(
	productID int,
	teamName varchar(50),
	PRIMARY KEY (productID),
	FOREIGN KEY (productID) REFERENCES TeamMerchandise(productID) ON DELETE CASCADE ON UPDATE CASCADE,
	FOREIGN KEY (teamName) REFERENCES Team(teamName) ON DELETE CASCADE ON UPDATE CASCADE
);
```



```
cs421-> \d teammerchandiseteam
     Table "cs421g46.teammerchandiseteam"
  Column   |         Type          | Modifiers
-----------+-----------------------+-----------
 productid | integer               | not null
 teamname  | character varying(50) |
Indexes:
    "teammerchandiseteam_pkey" PRIMARY KEY, btree (productid)
Foreign-key constraints:
    "teammerchandiseteam_productid_fkey" FOREIGN KEY (productid) REFERENCES teammerchandise(productid) ON UPDATE CASCADE ON DELETE C
ASCADE
    "teammerchandiseteam_teamname_fkey" FOREIGN KEY (teamname) REFERENCES team(teamname) ON UPDATE CASCADE ON DELETE CASCADE
```

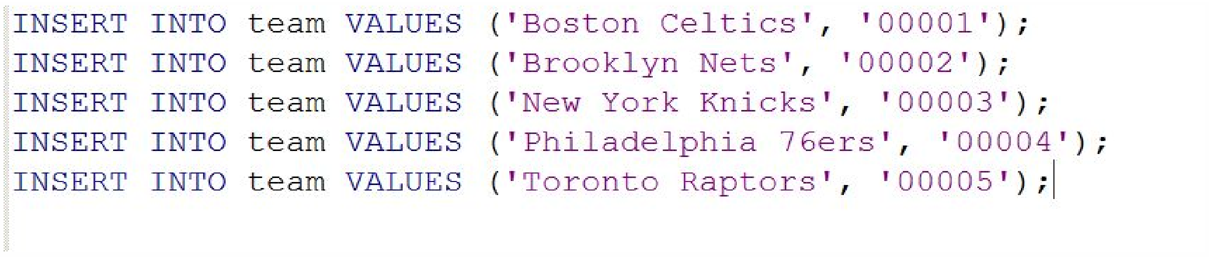


```
CREATE TABLE PlayerMerchandisePlayer
(
        productID int,
        playerNumber int,
        teamName varchar(50),
        PRIMARY KEY (productID),
        FOREIGN KEY (productID) REFERENCES PlayerMerchandise(productID) ON DELETE CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (playerNumber, teamName) REFERENCES Player(playerNumber, teamName) ON DELETE CASCADE ON UPDATE CASCADE
);
```

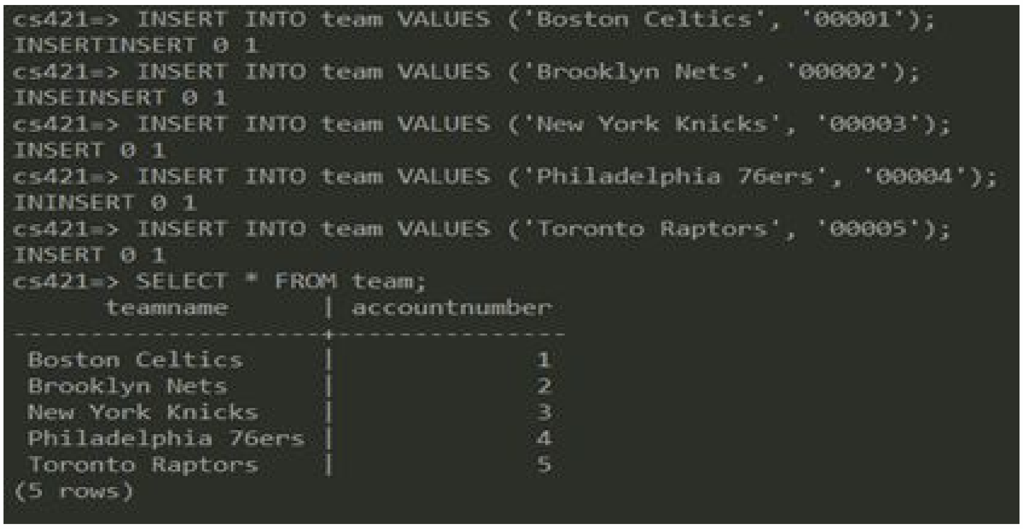


```
cs421-> \d playermerchandiseplayer
     Table "cs421g46.playermerchandiseplayer"
    Column    |         Type          | Modifiers
--------------+-----------------------+-----------
 productid    | integer               | not null
 playernumber | integer               |
 teamname     | character varying(50) |
Indexes:
    "playermerchandiseplayer_pkey" PRIMARY KEY, btree (productid)
Foreign-key constraints:
    "playermerchandiseplayer_playernumber_fkey" FOREIGN KEY (playernumber, teamname) REFERENCES player(playernumber, teamname) ON UP
DATE CASCADE ON DELETE CASCADE
    "playermerchandiseplayer_productid_fkey" FOREIGN KEY (productid) REFERENCES playermerchandise(productid) ON UPDATE CASCADE ON DE
LETE CASCADE
```

Question 3:



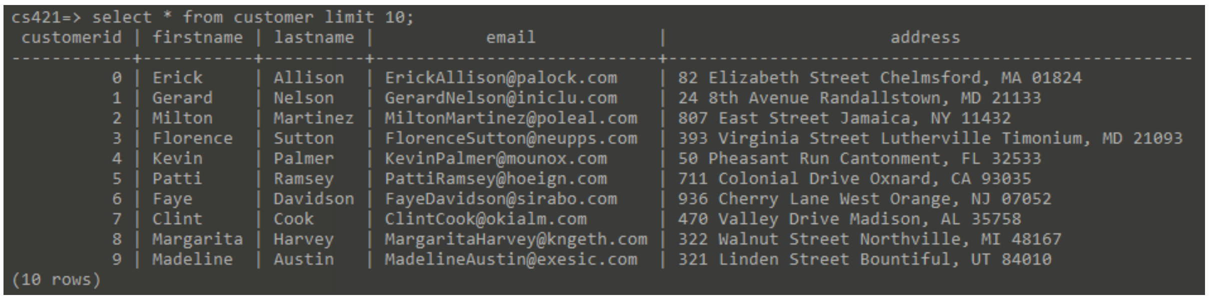
```
INSERT INTO team VALUES ('Boston Celtics', '00001');
INSERT INTO team VALUES ('Brooklyn Nets', '00002');
INSERT INTO team VALUES ('New York Knicks', '00003');
INSERT INTO team VALUES ('Philadelphia 76ers', '00004');
INSERT INTO team VALUES ('Toronto Raptors', '00005');
```



```
cs421=> INSERT INTO team VALUES ('Boston Celtics', '00001');
INSERTINSERT 0 1
cs421=> INSERT INTO team VALUES ('Brooklyn Nets', '00002');
INSEINSERT 0 1
cs421=> INSERT INTO team VALUES ('New York Knicks', '00003');
INSERT 0 1
cs421=> INSERT INTO team VALUES ('Philadelphia 76ers', '00004');
ININSERT 0 1
cs421=> INSERT INTO team VALUES ('Toronto Raptors', '00005');
INSERT 0 1
cs421=> SELECT * FROM team;
      teamname      | accountnumber
--------------------+---------------
 Boston Celtics     |             1
 Brooklyn Nets      |             2
 New York Knicks    |             3
 Philadelphia 76ers |             4
 Toronto Raptors    |             5
(5 rows)
```

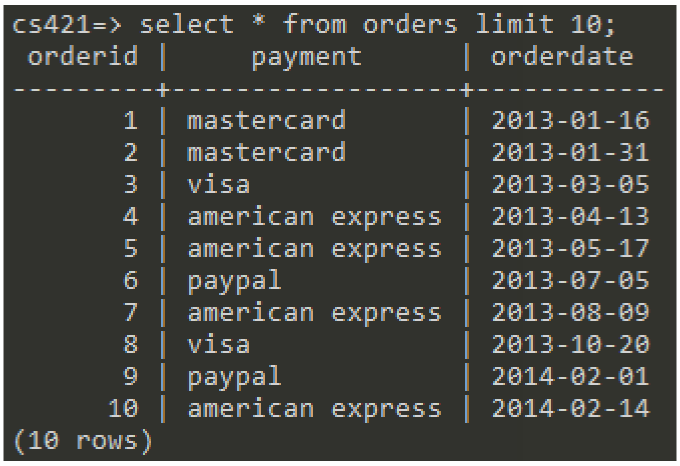
Question 4:

Customer



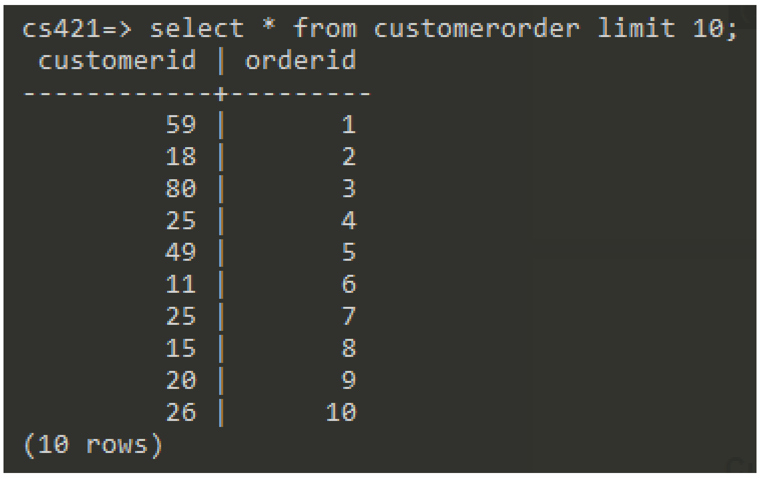
```
cs421=> select * from customer limit 10;
 customerid | firstname | lastname |            email             |                      address
------------+-----------+----------+------------------------------+----------------------------------------------------
          0 | Erick     | Allison  | ErickAllison@palock.com      | 82 Elizabeth Street Chelmsford, MA 01824
          1 | Gerard    | Nelson   | GerardNelson@iniclu.com      | 24 8th Avenue Randallstown, MD 21133
          2 | Milton    | Martinez | MiltonMartinez@poleal.com    | 807 East Street Jamaica, NY 11432
          3 | Florence  | Sutton   | FlorenceSutton@neupps.com    | 393 Virginia Street Lutherville Timonium, MD 21093
          4 | Kevin     | Palmer   | KevinPalmer@mounox.com       | 50 Pheasant Run Cantonment, FL 32533
          5 | Patti     | Ramsey   | PattiRamsey@hoeign.com       | 711 Colonial Drive Oxnard, CA 93035
          6 | Faye      | Davidson | FayeDavidson@sirabo.com      | 936 Cherry Lane West Orange, NJ 07052
          7 | Clint     | Cook     | ClintCook@okialm.com         | 470 Valley Drive Madison, AL 35758
          8 | Margarita | Harvey   | MargaritaHarvey@kngeth.com   | 322 Walnut Street Northville, MI 48167
          9 | Madeline  | Austin   | MadelineAustin@exesic.com    | 321 Linden Street Bountiful, UT 84010
(10 rows)
```

Orders



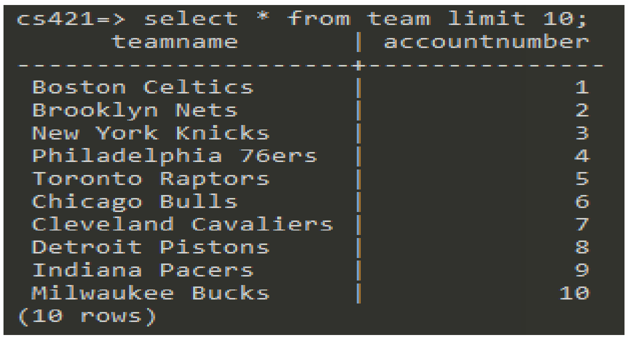
```
cs421=> select * from orders limit 10;
 orderid |      payment       | orderdate
---------+------------------+------------
       1 | mastercard       | 2013-01-16
       2 | mastercard       | 2013-01-31
       3 | visa             | 2013-03-05
       4 | american express | 2013-04-13
       5 | american express | 2013-05-17
       6 | paypal           | 2013-07-05
       7 | american express | 2013-08-09
       8 | visa             | 2013-10-20
       9 | paypal           | 2014-02-01
      10 | american express | 2014-02-14
(10 rows)
```

CustomerOrder



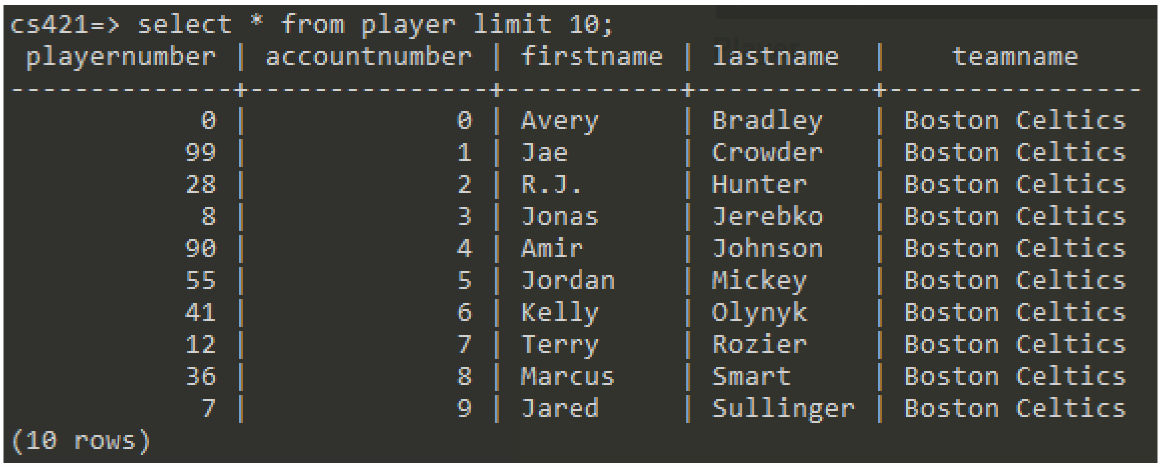
```
cs421=> select * from customerorder limit 10;
 customerid | orderid
------------+---------
         59 |       1
         18 |       2
         80 |       3
         25 |       4
         49 |       5
         11 |       6
         25 |       7
         15 |       8
         20 |       9
         26 |      10
(10 rows)
```

Team



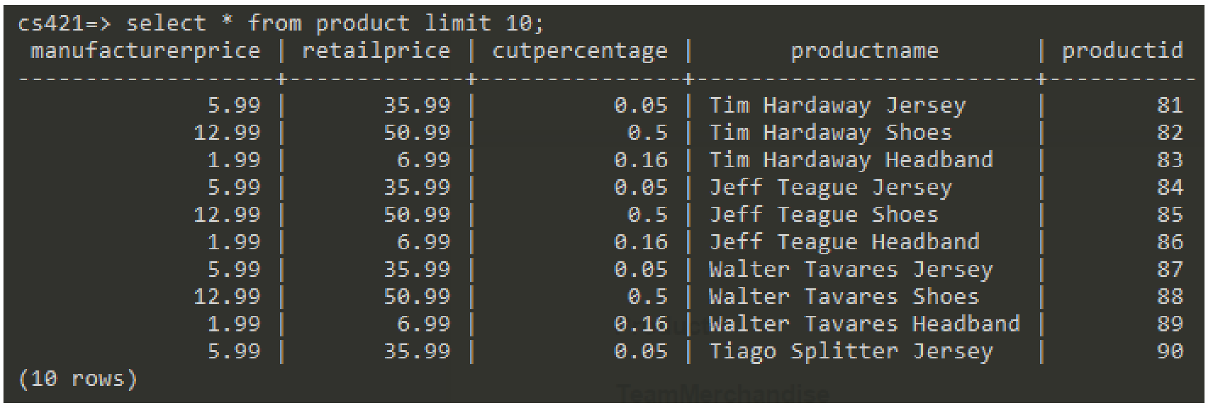
```
cs421=> select * from team limit 10;
      teamname       | accountnumber
---------------------+---------------
 Boston Celtics      |             1
 Brooklyn Nets       |             2
 New York Knicks     |             3
 Philadelphia 76ers  |             4
 Toronto Raptors     |             5
 Chicago Bulls       |             6
 Cleveland Cavaliers |             7
 Detroit Pistons     |             8
 Indiana Pacers      |             9
 Milwaukee Bucks     |            10
(10 rows)
```

Player



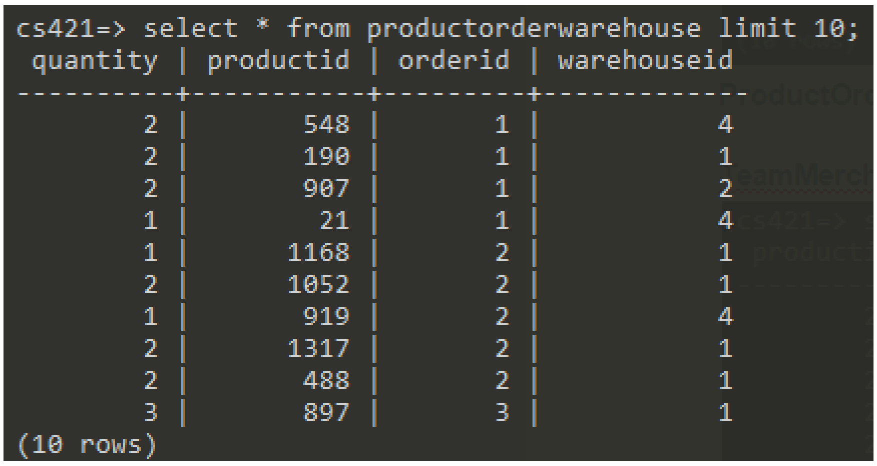
```
cs421=> select * from player limit 10;
 playernumber | accountnumber | firstname | lastname  |    teamname
--------------+---------------+-----------+-----------+----------------
            0 |             0 | Avery     | Bradley   | Boston Celtics
           99 |             1 | Jae       | Crowder   | Boston Celtics
           28 |             2 | R.J.      | Hunter    | Boston Celtics
            8 |             3 | Jonas     | Jerebko   | Boston Celtics
           90 |             4 | Amir      | Johnson   | Boston Celtics
           55 |             5 | Jordan    | Mickey    | Boston Celtics
           41 |             6 | Kelly     | Olynyk    | Boston Celtics
           12 |             7 | Terry     | Rozier    | Boston Celtics
           36 |             8 | Marcus    | Smart     | Boston Celtics
            7 |             9 | Jared     | Sullinger | Boston Celtics
(10 rows)
```

Product



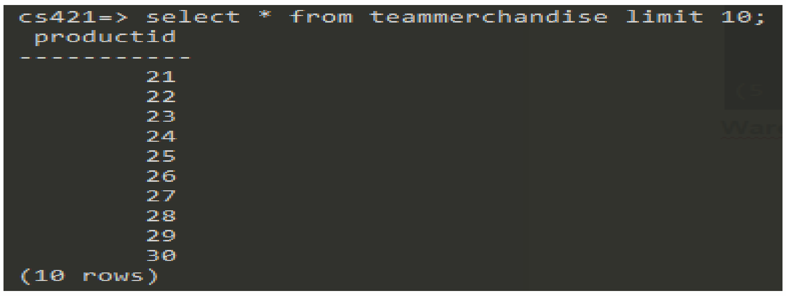
```
cs421=> select * from product limit 10;
 manufacturerprice | retailprice | cutpercentage |       productname        | productid
-------------------+-------------+---------------+--------------------------+-----------
              5.99 |       35.99 |          0.05 | Tim Hardaway Jersey      |        81
             12.99 |       50.99 |           0.5 | Tim Hardaway Shoes       |        82
              1.99 |        6.99 |          0.16 | Tim Hardaway Headband    |        83
              5.99 |       35.99 |          0.05 | Jeff Teague Jersey       |        84
             12.99 |       50.99 |           0.5 | Jeff Teague Shoes        |        85
              1.99 |        6.99 |          0.16 | Jeff Teague Headband     |        86
              5.99 |       35.99 |          0.05 | Walter Tavares Jersey    |        87
             12.99 |       50.99 |           0.5 | Walter Tavares Shoes     |        88
              1.99 |        6.99 |          0.16 | Walter Tavares Headband  |        89
              5.99 |       35.99 |          0.05 | Tiago Splitter Jersey    |        90
(10 rows)
```

ProductOrderWarehouse



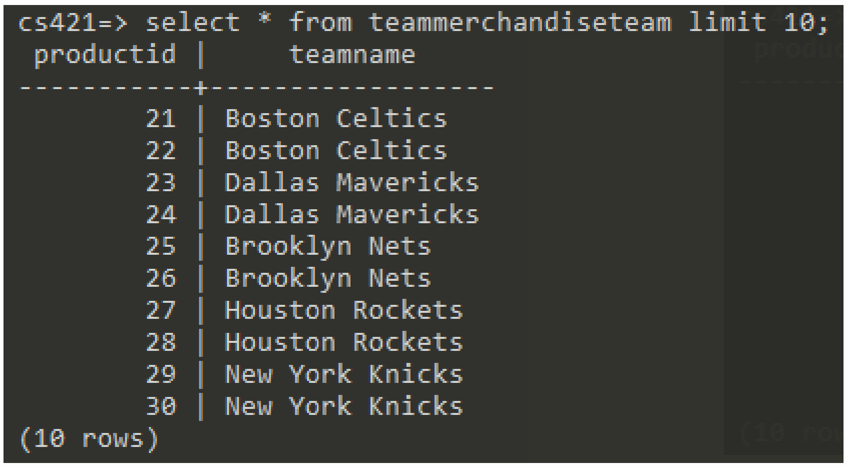
```
cs421=> select * from productorderwarehouse limit 10;
 quantity | productid | orderid | warehouseid
----------+-----------+---------+-------------
        2 |       548 |       1 |           4
        2 |       190 |       1 |           1
        2 |       907 |       1 |           2
        1 |        21 |       1 |           4
        1 |      1168 |       2 |           1
        2 |      1052 |       2 |           1
        1 |       919 |       2 |           4
        2 |      1317 |       2 |           1
        2 |       488 |       2 |           1
        3 |       897 |       3 |           1
(10 rows)
```

TeamMerchandise



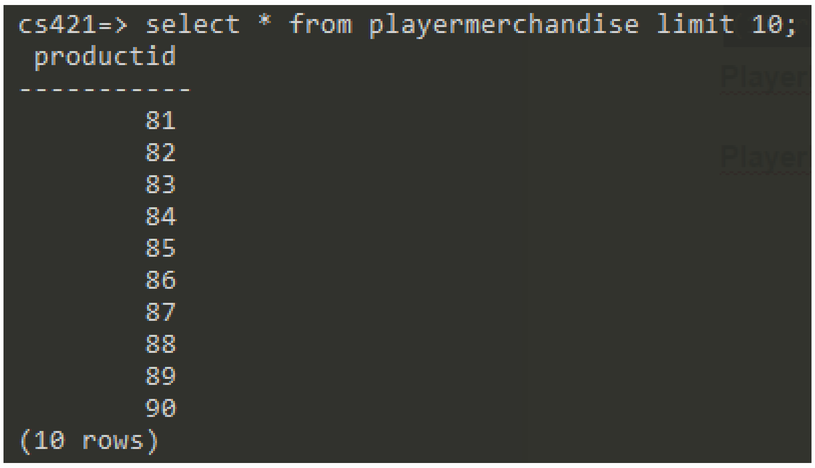
```
cs421=> select * from teammerchandise limit 10;
 productid
-----------
        21
        22
        23
        24
        25
        26
        27
        28
        29
        30
(10 rows)
```

TeamMerchandiseTeam



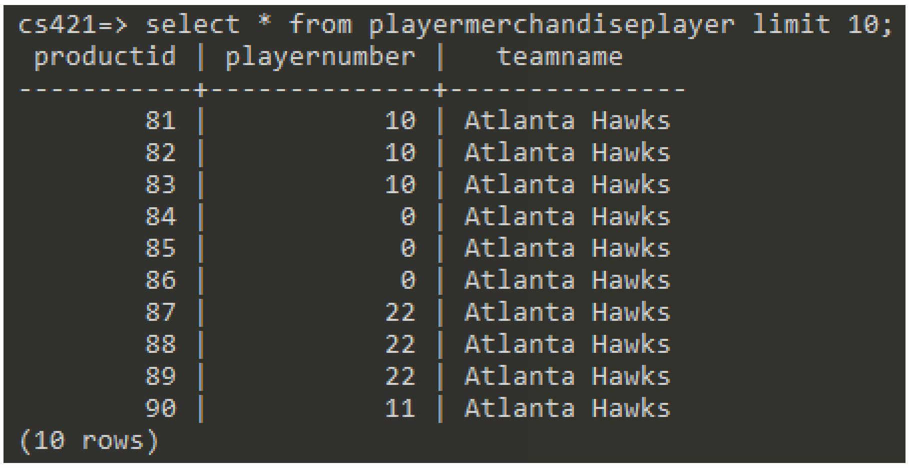
```
cs421=> select * from teammerchandiseteam limit 10;
 productid |     teamname
-----------+------------------
        21 | Boston Celtics
        22 | Boston Celtics
        23 | Dallas Mavericks
        24 | Dallas Mavericks
        25 | Brooklyn Nets
        26 | Brooklyn Nets
        27 | Houston Rockets
        28 | Houston Rockets
        29 | New York Knicks
        30 | New York Knicks
(10 rows)
```

PlayerMerchandise



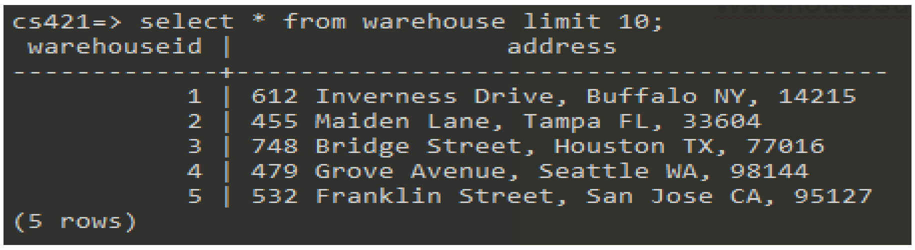
```
cs421=> select * from playermerchandise limit 10;
 productid
-----------
        81
        82
        83
        84
        85
        86
        87
        88
        89
        90
(10 rows)
```

PlayerMerchandisePlayer



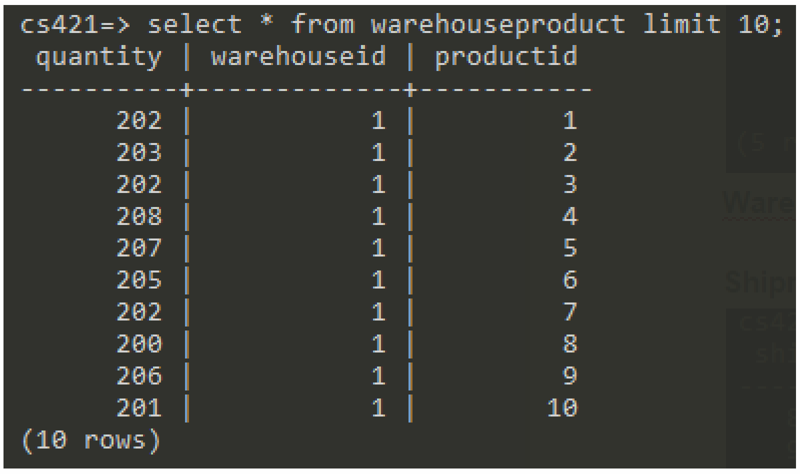
```
cs421=> select * from playermerchandiseplayer limit 10;
 productid | playernumber |    teamname
-----------+--------------+---------------
        81 |           10 | Atlanta Hawks
        82 |           10 | Atlanta Hawks
        83 |           10 | Atlanta Hawks
        84 |            0 | Atlanta Hawks
        85 |            0 | Atlanta Hawks
        86 |            0 | Atlanta Hawks
        87 |           22 | Atlanta Hawks
        88 |           22 | Atlanta Hawks
        89 |           22 | Atlanta Hawks
        90 |           11 | Atlanta Hawks
(10 rows)
```

Warehouse



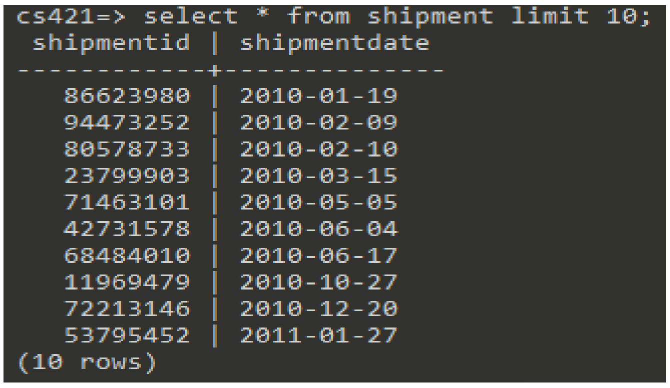
```
cs421=> select * from warehouse limit 10;
 warehouseid |                 address
-------------+-----------------------------------------
           1 | 612 Inverness Drive, Buffalo NY, 14215
           2 | 455 Maiden Lane, Tampa FL, 33604
           3 | 748 Bridge Street, Houston TX, 77016
           4 | 479 Grove Avenue, Seattle WA, 98144
           5 | 532 Franklin Street, San Jose CA, 95127
(5 rows)
```

WarehouseProduct



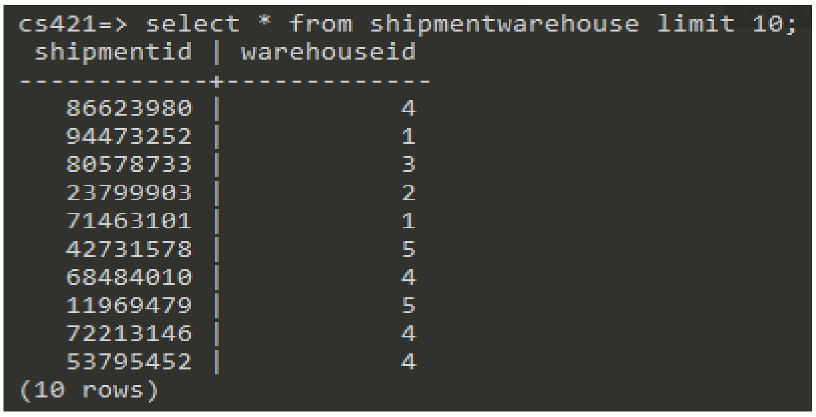
```
cs421=> select * from warehouseproduct limit 10;
 quantity | warehouseid | productid
----------+-------------+-----------
      202 |           1 |         1
      203 |           1 |         2
      202 |           1 |         3
      208 |           1 |         4
      207 |           1 |         5
      205 |           1 |         6
      202 |           1 |         7
      200 |           1 |         8
      206 |           1 |         9
      201 |           1 |        10
(10 rows)
```

Shipment



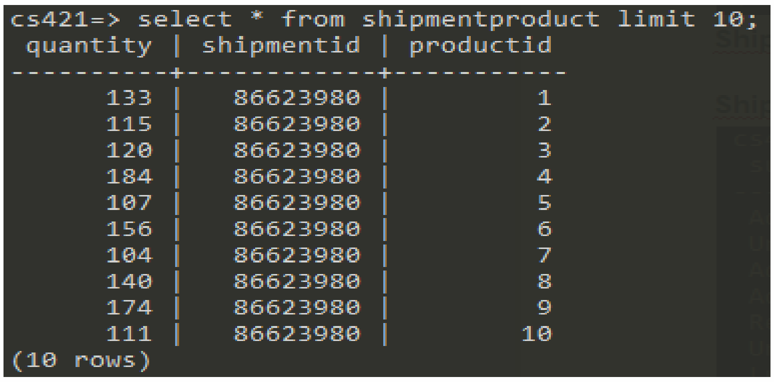
```
cs421=> select * from shipment limit 10;
 shipmentid | shipmentdate
------------+--------------
   86623980 | 2010-01-19
   94473252 | 2010-02-09
   80578733 | 2010-02-10
   23799903 | 2010-03-15
   71463101 | 2010-05-05
   42731578 | 2010-06-04
   68484010 | 2010-06-17
   11969479 | 2010-10-27
   72213146 | 2010-12-20
   53795452 | 2011-01-27
(10 rows)
```

ShipmentWarehouse



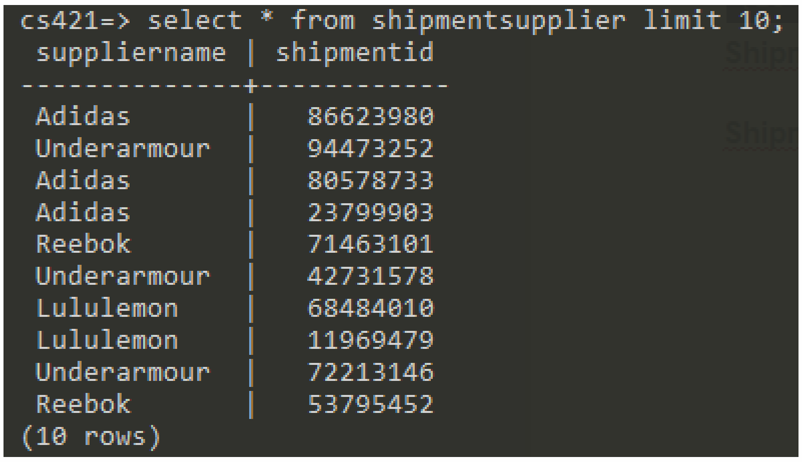
```
cs421=> select * from shipmentwarehouse limit 10;
 shipmentid | warehouseid
------------+-------------
   86623980 |           4
   94473252 |           1
   80578733 |           3
   23799903 |           2
   71463101 |           1
   42731578 |           5
   68484010 |           4
   11969479 |           5
   72213146 |           4
   53795452 |           4
(10 rows)
```

ShipmentProduct



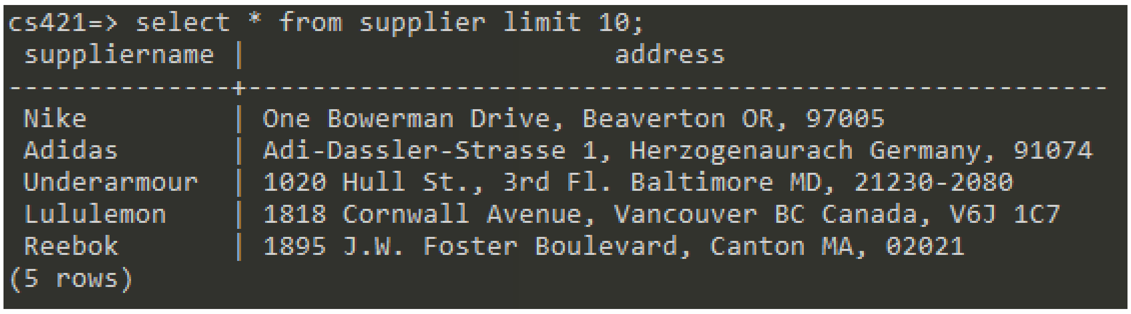
```
cs421=> select * from shipmentproduct limit 10;
 quantity | shipmentid | productid
----------+------------+-----------
      133 |   86623980 |         1
      115 |   86623980 |         2
      120 |   86623980 |         3
      184 |   86623980 |         4
      107 |   86623980 |         5
      156 |   86623980 |         6
      104 |   86623980 |         7
      140 |   86623980 |         8
      174 |   86623980 |         9
      111 |   86623980 |        10
(10 rows)
```

ShipmentSupplier



```
cs421=> select * from shipmentsupplier limit 10;
 suppliername | shipmentid
--------------+------------
 Adidas       |   86623980
 Underarmour  |   94473252
 Adidas       |   80578733
 Adidas       |   23799903
 Reebok       |   71463101
 Underarmour  |   42731578
 Lululemon    |   68484010
 Lululemon    |   11969479
 Underarmour  |   72213146
 Reebok       |   53795452
(10 rows)
```

Supplier



```
cs421=> select * from supplier limit 10;
 suppliername |                        address
--------------+------------------------------------------------------
 Nike         | One Bowerman Drive, Beaverton OR, 97005
 Adidas       | Adi-Dassler-Strasse 1, Herzogenaurach Germany, 91074
 Underarmour  | 1020 Hull St., 3rd Fl. Baltimore MD, 21230-2080
 Lululemon    | 1818 Cornwall Avenue, Vancouver BC Canada, V6J 1C7
 Reebok       | 1895 J.W. Foster Boulevard, Canton MA, 02021
(5 rows)
```

Question 5:

```
/* First query will grab the total revenue of our store, using explicit join*/
SELECT sum(quantity * (retailPrice - manufacturerPrice))
FROM Product
INNER JOIN productorderwarehouse
ON productorderwarehouse.productID = Product.productID;

/* Second query will grab the players that have made over $80 from one specific team
        The example team will be the Golden State Warriors, we use implicit join here */
SELECT playerNumber, teamName
FROM Product, ProductOrderWarehouse, PlayerMerchandisePlayer
WHERE ProductOrderWarehouse.productID = Product.productID
        AND Product.productID = PlayerMerchandisePlayer.productID
        AND PlayerMerchandisePlayer.teamName = 'Golden State Warriors'
GROUP BY playerNumber, teamName
HAVING sum(cutPercentage * quantity * (retailPrice - manufacturerPrice)) > 80

/*Third query will list all distinct productNames that were sold between 2014-01-01 and
2014-03-01*/

SELECT DISTINCT productName
FROM Product
INNER JOIN ProductOrderWarehouse
ON Product.productID = ProductOrderWarehouse.productID
INNER JOIN Orders
ON ProductOrderWarehouse.orderID = Orders.orderID
WHERE orderDate > '2014-01-01'  AND orderDate < '2014-03-01'
ORDER BY productName DESC;

/* Fourth query will grab the customers who have paid with VISA on products that have
come from Adidas*/
SELECT Customer.firstName, Customer.lastName FROM Customer
WHERE Customer.customerID IN
(SELECT DISTINCT customerID FROM CustomerOrder, Orders,
Productorderwarehouse
WHERE CustomerOrder.orderID = Orders.orderID
AND Orders.orderID = Productorderwarehouse.orderID
AND Orders.payment = 'visa'
AND Productorderwarehouse.productID
IN
(
        SELECT productID
        FROM ShipmentProduct, ShipmentSupplier
        WHERE ShipmentProduct.shipmentID = ShipmentSupplier.shipmentID
        AND ShipmentSupplier.supplierName = 'Adidas'
)
ORDER BY customerID);
```

/* **Fifth** query will get customer first and last name that have made an order that includes any stephen curry product and the total order costs more than $50 */

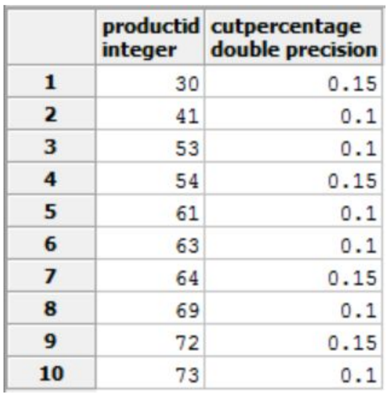
```
SELECT DISTINCT firstname, lastname
FROM productorderwarehouse
INNER JOIN Orders
ON productorderwarehouse.orderID = Orders.orderID
INNER JOIN
(
        SELECT orderID
        FROM Product
        INNER JOIN ProductOrderWarehouse
        ON Product.productID = ProductOrderWarehouse.productID
        WHERE productName LIKE 'Stephen Curry%'
        INTERSECT
        SELECT orderID
        From ProductOrderWarehouse
        INNER JOIN Product
        ON ProductOrderWarehouse.productID = Product.productID
        GROUP BY orderID
        HAVING sum(quantity * retailPrice) > 50
) AS stephenCurryOrders

ON Orders.orderID = stephenCurryOrders.orderID
INNER JOIN CustomerOrder
ON stephenCurryOrders.orderID = CustomerOrder.orderID
INNER JOIN Customer
ON CustomerOrder.customerID = Customer.customerID;
```

Question 6:

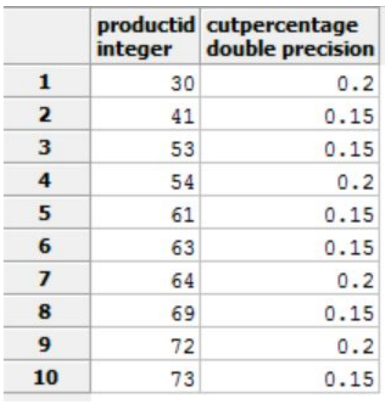
All the screenshots below will show the before and after of each update query (except the last one because we create the table)

1. query will update cut percentage of all teams that have made more than $3 dollars to current cut percentage + 5%

```
Update Product SET cutPercentage = cutPercentage + 0.05
WHERE productID IN
(
        SELECT Product.productID
        FROM Product, Productorderwarehouse, TeamMerchandiseTeam
        WHERE Product.productID = TeamMerchandiseTeam.productID
        AND TeamMerchandiseTeam.productID = productorderwarehouse.productID
        GROUP BY Product.productID
        HAVING sum(cutPercentage * quantity * (retailPrice - manufacturerPrice)) > 3
);
```



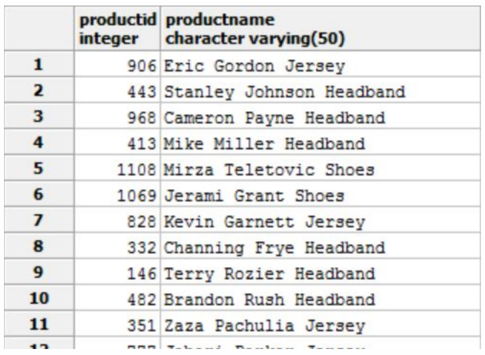
	productid integer	cutpercentage double precision
1	30	0.15
2	41	0.1
3	53	0.1
4	54	0.15
5	61	0.1
6	63	0.1
7	64	0.15
8	69	0.1
9	72	0.15
10	73	0.1



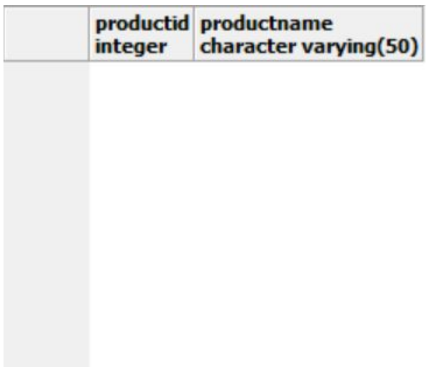
	productid integer	cutpercentage double precision
1	30	0.2
2	41	0.15
3	53	0.15
4	54	0.2
5	61	0.15
6	63	0.15
7	64	0.2
8	69	0.15
9	72	0.2
10	73	0.15

2. query will delete all player merchandise (in whole database) that have not made over $80 in total retail sales.

```
DELETE FROM Product
WHERE productID NOT IN
(
        SELECT DISTINCT Product.productID
        FROM Product, PlayerMerchandisePlayer, Productorderwarehouse
        WHERE Product.productID = PlayerMerchandisePlayer.productID
        AND PlayerMerchandisePlayer.productID = Productorderwarehouse.productID
        GROUP BY Product.productID
        HAVING sum(quantity * retailPrice) > 80
);
```



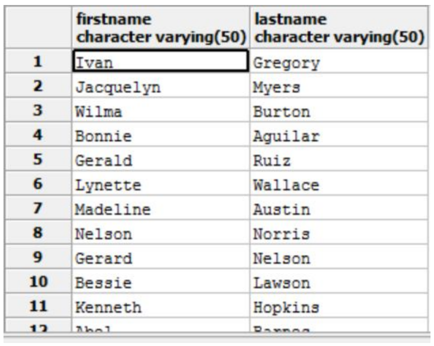
	productid integer	productname character varying(50)
1	906	Eric Gordon Jersey
2	443	Stanley Johnson Headband
3	968	Cameron Payne Headband
4	413	Mike Miller Headband
5	1108	Mirza Teletovic Shoes
6	1069	Jerami Grant Shoes
7	828	Kevin Garnett Jersey
8	332	Channing Frye Headband
9	146	Terry Rozier Headband
10	482	Brandon Rush Headband
11	351	Zaza Pachulia Jersey



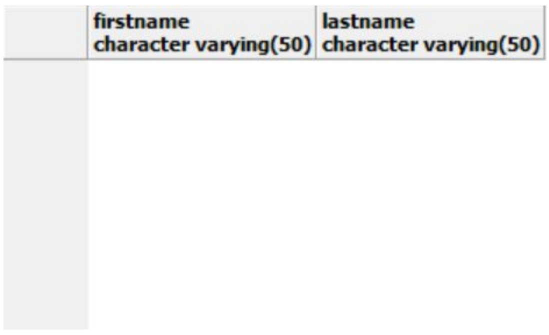
	productid integer	productname character varying(50)

3. query will delete all customers who have not ordered anything after 2014

```
DELETE FROM Customer
WHERE customerID IN
(
        SELECT Customer.customerID
        FROM Customer, Orders, CustomerOrder
        WHERE Customer.customerID = CustomerOrder.customerID
        AND Orders.orderID = CustomerOrder.orderID
        AND Orders.orderDate < '2013-12-31'
);
```



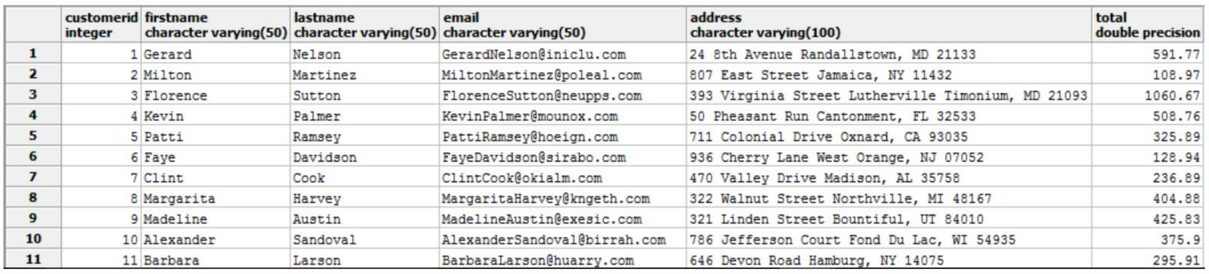
	firstname character varying(50)	lastname character varying(50)
1	Ivan	Gregory
2	Jacquelyn	Myers
3	Wilma	Burton
4	Bonnie	Aguilar
5	Gerald	Ruiz
6	Lynette	Wallace
7	Madeline	Austin
8	Nelson	Norris
9	Gerard	Nelson
10	Bessie	Lawson
11	Kenneth	Hopkins



	firstname character varying(50)	lastname character varying(50)

4. Creates a new table called ValuedCustomer and queries for customers who have spent over $50 in our store and inserts it into this table. ValuedCustomer has the same attributes as Customer with an extra field total = total that the customer has spent.

```
drop TABLE ValuedCustomer;
CREATE TABLE ValuedCustomer
(
        customerID int,
        firstName varchar(50) NOT NULL,
        lastName varchar(50) NOT NULL,
        email varchar(50) UNIQUE NOT NULL,
        address varchar(100) NOT NULL,
        total double precision,
        PRIMARY KEY (customerID)
);

CREATE OR REPLACE FUNCTION updateValuesWithLoop()
RETURNS void AS $BODY$
DECLARE
        a int;
        r double precision;
BEGIN
        FOR a IN SELECT customerID FROM Customer LOOP
                SELECT sum(retailPrice * quantity) INTO r
                FROM CustomerOrder, Orders, Product, ProductOrderWarehouse
                WHERE CustomerOrder.customerID = a
                AND Orders.orderID = ProductOrderWarehouse.orderID
                AND Orders.orderID = CustomerOrder.orderID
                AND ProductOrderWarehouse.productID = Product.productID;

                If (r > 50)
                THEN
                        INSERT INTO ValuedCustomer(customerID, firstName, lastName,
email, address)
                        SELECT * FROM Customer where Customer.customerID = a;
                        UPDATE ValuedCustomer SET total = r WHERE customerID = a;
                END IF;
        END LOOP;
END
$BODY$
LANGUAGE 'plpgsql';

SELECT updateValuesWithLoop();
SELECT * FROM ValuedCustomer;
```



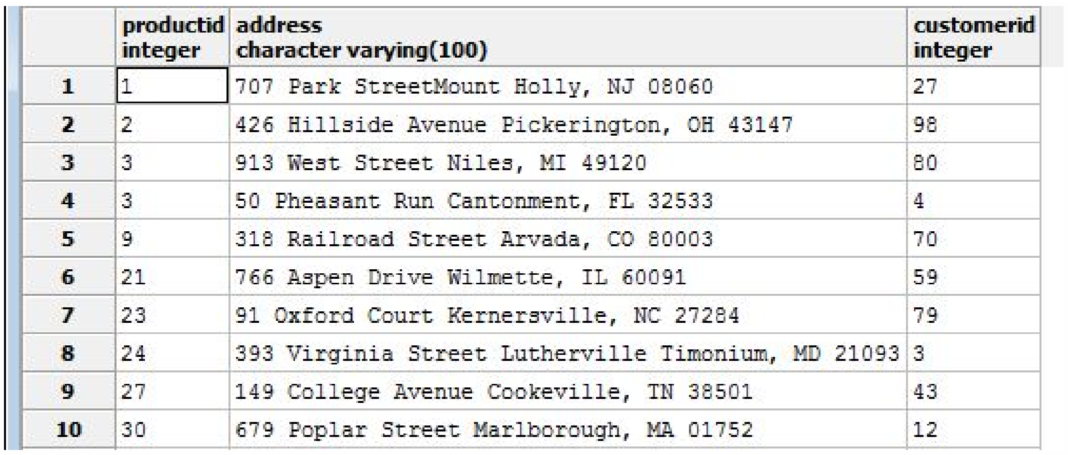
	customerid integer	firstname character varying(50)	lastname character varying(50)	email character varying(50)	address character varying(100)	total double precision
1	1	Gerard	Nelson	GerardNelson@iniclu.com	24 8th Avenue Randallstown, MD 21133	591.77
2	2	Milton	Martinez	MiltonMartinez@poleal.com	807 East Street Jamaica, NY 11432	108.97
3	3	Florence	Sutton	FlorenceSutton@neupps.com	393 Virginia Street Lutherville Timonium, MD 21093	1060.67
4	4	Kevin	Palmer	KevinPalmer@mounox.com	50 Pheasant Run Cantonment, FL 32533	508.76
5	5	Patti	Ramsey	PattiRamsey@hoeign.com	711 Colonial Drive Oxnard, CA 93035	325.89
6	6	Faye	Davidson	FayeDavidson@sirabo.com	936 Cherry Lane West Orange, NJ 07052	128.94
7	7	Clint	Cook	ClintCook@okialm.com	470 Valley Drive Madison, AL 35758	236.89
8	8	Margarita	Harvey	MargaritaHarvey@kngeth.com	322 Walnut Street Northville, MI 48167	404.88
9	9	Madeline	Austin	MadelineAustin@exesic.com	321 Linden Street Bountiful, UT 84010	425.83
10	10	Alexander	Sandoval	AlexanderSandoval@birrah.com	786 Jefferson Court Fond Du Lac, WI 54935	375.9
11	11	Barbara	Larson	BarbaraLarson@huarry.com	646 Devon Road Hamburg, NY 14075	295.91

resulting table

Question 7:

Addresses where product was shipped. This view is used to return a table with productID, address, and customerID. This is to see where each product is sold, which will aid with determining which warehouses should stock certain items for cheaper/quicker shipping without revealing sensitive customer information.

```
CREATE VIEW sale_map AS
WITH Product_customers AS ( -- gets customerID and productID
        SELECT productID, customerID
        FROM ProductOrderWarehouse INNER JOIN CustomerOrder
        ON ProductOrderWarehouse.orderID = CustomerOrder.orderID
)
SELECT productID, address, Customer.customerID
FROM Customer INNER JOIN Product_customers
ON Customer.customerID = Product_customers.customerID
ORDER BY productID;
```

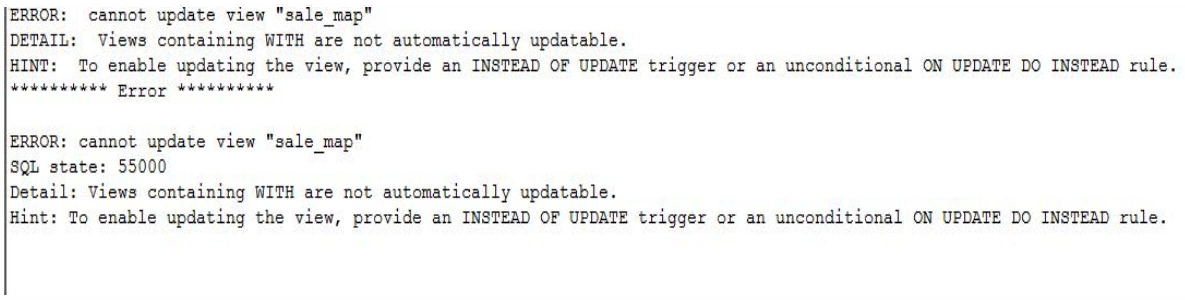


	productid integer	address character varying(100)	customerid integer
1	1	707 Park StreetMount Holly, NJ 08060	27
2	2	426 Hillside Avenue Pickerington, OH 43147	98
3	3	913 West Street Niles, MI 49120	80
4	3	50 Pheasant Run Cantonment, FL 32533	4
5	9	318 Railroad Street Arvada, CO 80003	70
6	21	766 Aspen Drive Wilmette, IL 60091	59
7	23	91 Oxford Court Kernersville, NC 27284	79
8	24	393 Virginia Street Lutherville Timonium, MD 21093	3
9	27	149 College Avenue Cookeville, TN 38501	43
10	30	679 Poplar Street Marlborough, MA 01752	12

This view is not updatable. It contains a WITH clause, and it uses a JOIN i.e. it has more than one table in its FROM list. For example, if we run the following query:

Update sale_map SET address = 'Canada' WHERE customerid = '79';

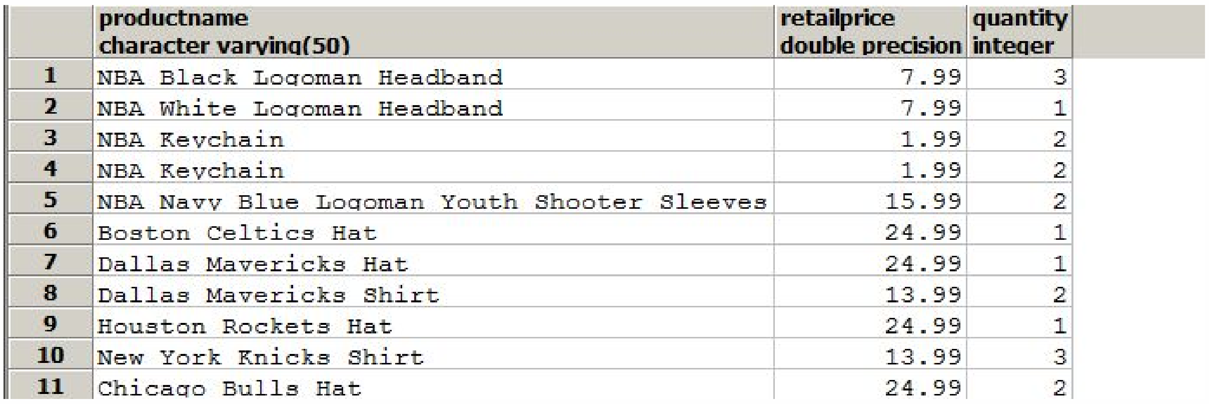
PostgreSQL returns the following error:



```
ERROR:  cannot update view "sale_map"
DETAIL:  Views containing WITH are not automatically updatable.
HINT:  To enable updating the view, provide an INSTEAD OF UPDATE trigger or an unconditional ON UPDATE DO INSTEAD rule.
********** Error **********

ERROR: cannot update view "sale_map"
SQL state: 55000
Detail: Views containing WITH are not automatically updatable.
Hint: To enable updating the view, provide an INSTEAD OF UPDATE trigger or an unconditional ON UPDATE DO INSTEAD rule.
```

This second view shows all products that have been sold in all orders along with the retail price and quantities in each order. This view is now updatable by default because we are querying from more than one table. However, it is updatable for the retail price since we have created a rule where if a user tries to update the view, it will instead update the underlying table product and set the retail price to the new price specified in the query.

CREATE VIEW ProductsSold AS
SELECT productname, retailprice, quantity
FROM Product
INNER JOIN productorderwarehouse
ON Product.productID = Productorderwarehouse.productID;
CREATE RULE visitProductsSold
AS ON UPDATE TO ProductsSold
DO INSTEAD UPDATE product SET retailprice = NEW.retailprice WHERE productname = NEW.productname;

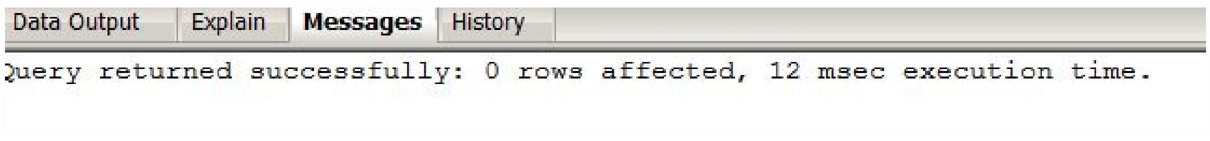


	productname character varying(50)	retailprice double precision	quantity integer
1	NBA Black Logoman Headband	7.99	3
2	NBA White Logoman Headband	7.99	1
3	NBA Keychain	1.99	2
4	NBA Keychain	1.99	2
5	NBA Navy Blue Logoman Youth Shooter Sleeves	15.99	2
6	Boston Celtics Hat	24.99	1
7	Dallas Mavericks Hat	24.99	1
8	Dallas Mavericks Shirt	13.99	2
9	Houston Rockets Hat	24.99	1
10	New York Knicks Shirt	13.99	3
11	Chicago Bulls Hat	24.99	2

If we run the following query:

Update ProductsSold SET retailprice = 100 WHERE productname = 'NBA Keychain';

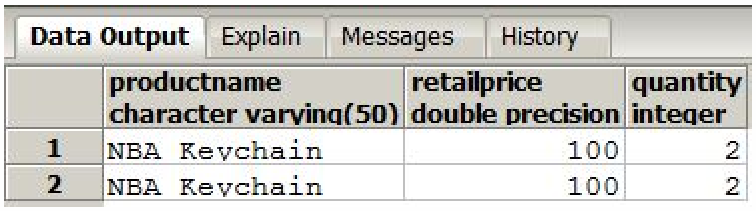
The result returned is:



```
Query returned successfully: 0 rows affected, 12 msec execution time.
```

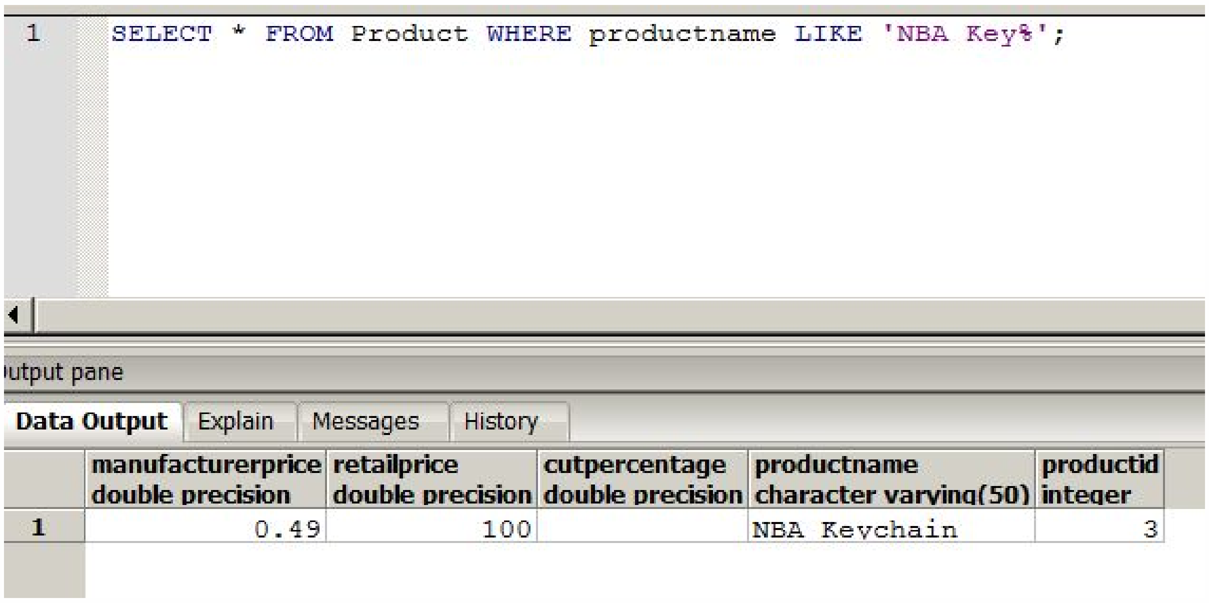
And on select:

SELECT * FROM ProductsSold WHERE productname LIKE 'NBA Key%';



	productname character varying(50)	retailprice double precision	quantity integer
1	NBA Keychain	100	2
2	NBA Keychain	100	2

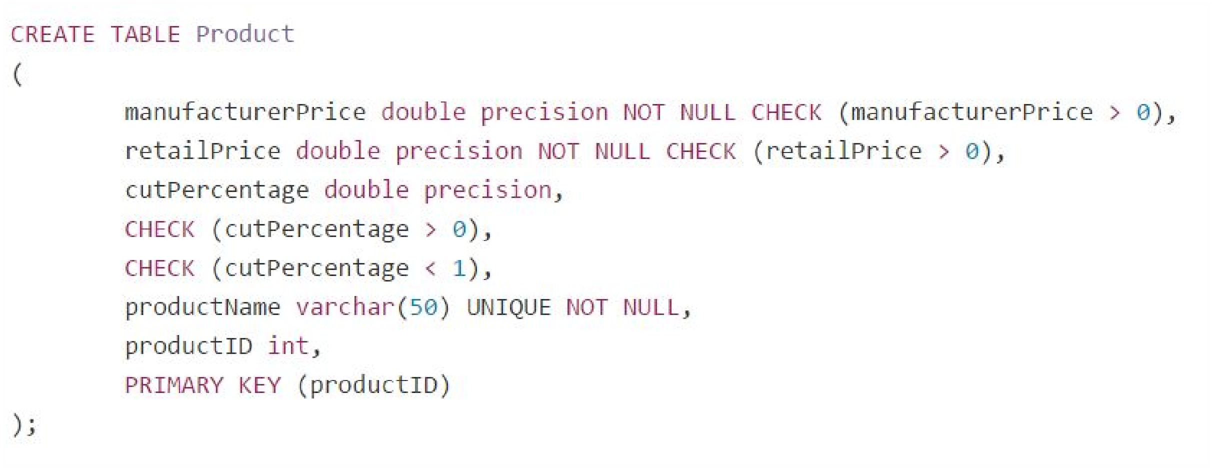
We can see that the price has changed in the view. And in the underlying table product,



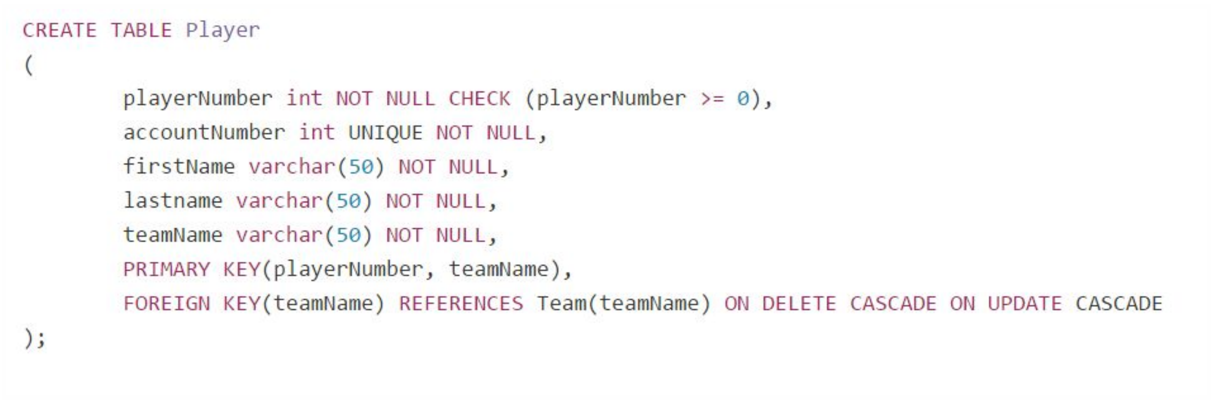
```
SELECT * FROM Product WHERE productname LIKE 'NBA Key%';
```

	manufacturerprice double precision	retailprice double precision	cutpercentage double precision	productname character varying(50)	productid integer
1	0.49	100		NBA Keychain	3

Question 8:

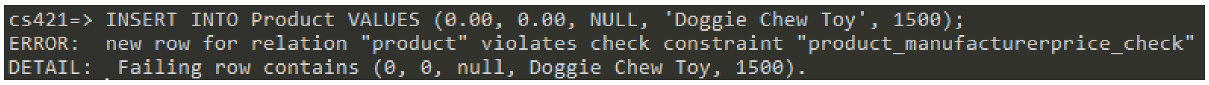


```
CREATE TABLE Product
(
        manufacturerPrice double precision NOT NULL CHECK (manufacturerPrice > 0),
        retailPrice double precision NOT NULL CHECK (retailPrice > 0),
        cutPercentage double precision,
        CHECK (cutPercentage > 0),
        CHECK (cutPercentage < 1),
        productName varchar(50) UNIQUE NOT NULL,
        productID int,
        PRIMARY KEY (productID)
);
```

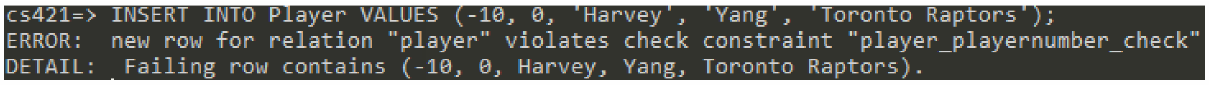


```
CREATE TABLE Player
(
        playerNumber int NOT NULL CHECK (playerNumber >= 0),
        accountNumber int UNIQUE NOT NULL,
        firstName varchar(50) NOT NULL,
        lastname varchar(50) NOT NULL,
        teamName varchar(50) NOT NULL,
        PRIMARY KEY(playerNumber, teamName),
        FOREIGN KEY(teamName) REFERENCES Team(teamName) ON DELETE CASCADE ON UPDATE CASCADE
);
```

INSERT STATEMENTS:

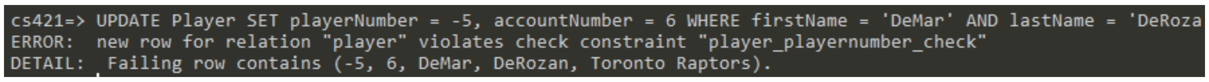


```
cs421=> INSERT INTO Product VALUES (0.00, 0.00, NULL, 'Doggie Chew Toy', 1500);
ERROR:  new row for relation "product" violates check constraint "product_manufacturerprice_check"
DETAIL:  Failing row contains (0, 0, null, Doggie Chew Toy, 1500).
```

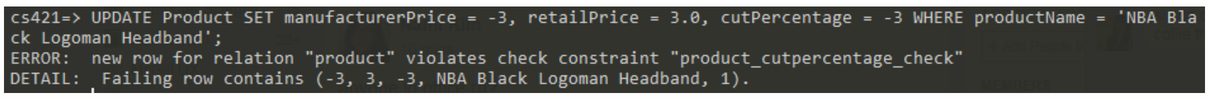


```
cs421=> INSERT INTO Player VALUES (-10, 0, 'Harvey', 'Yang', 'Toronto Raptors');
ERROR:  new row for relation "player" violates check constraint "player_playernumber_check"
DETAIL:  Failing row contains (-10, 0, Harvey, Yang, Toronto Raptors).
```

UPDATE STATEMENTS:



```
cs421=> UPDATE Player SET playerNumber = -5, accountNumber = 6 WHERE firstName = 'DeMar' AND lastName = 'DeRoza
ERROR:  new row for relation "player" violates check constraint "player_playernumber_check"
DETAIL:  Failing row contains (-5, 6, DeMar, DeRozan, Toronto Raptors).
```



```
cs421=> UPDATE Product SET manufacturerPrice = -3, retailPrice = 3.0, cutPercentage = -3 WHERE productName = 'NBA Bla
ck Logoman Headband';
ERROR:  new row for relation "product" violates check constraint "product_cutpercentage_check"
DETAIL:  Failing row contains (-3, 3, -3, NBA Black Logoman Headband, 1).
```