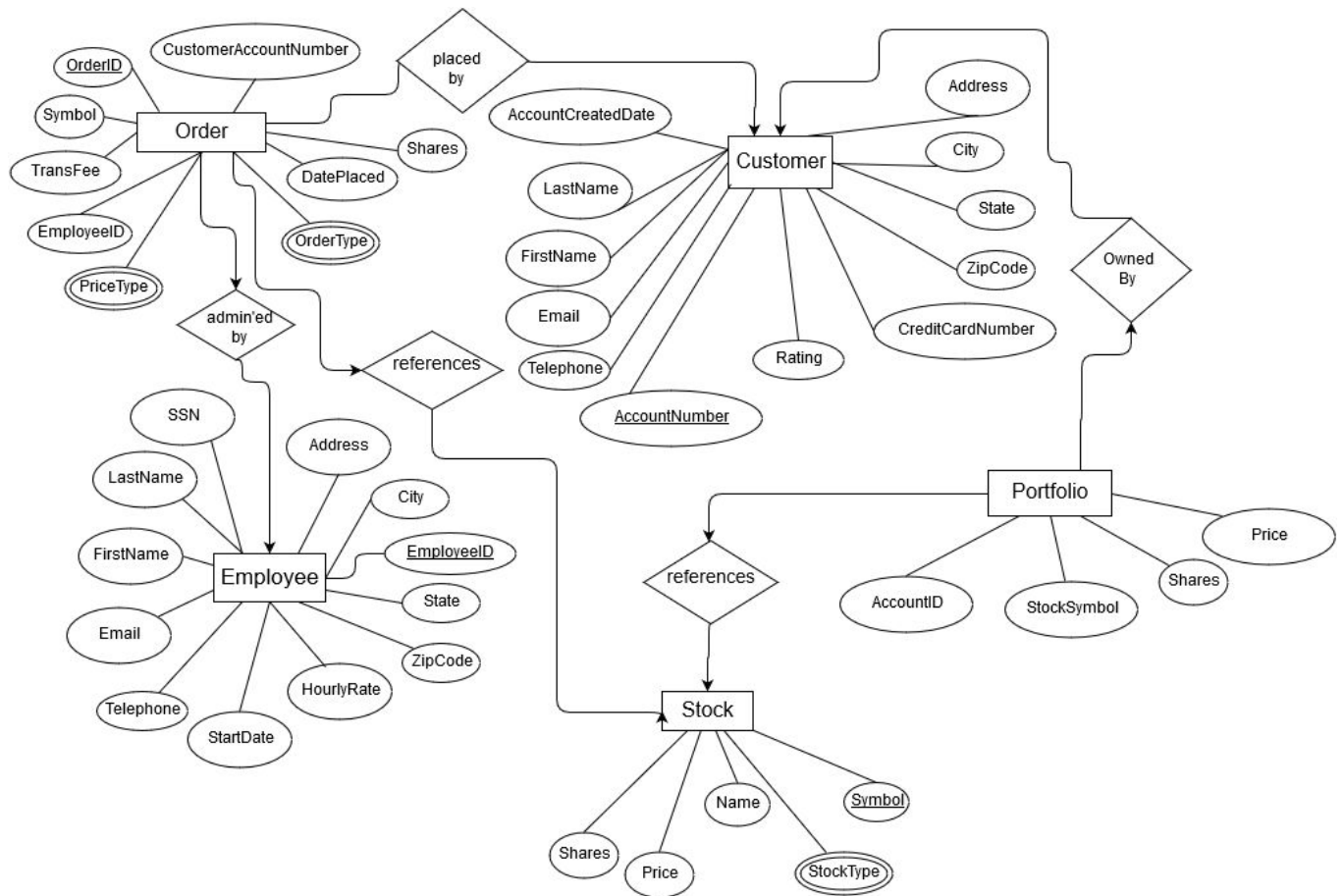


Team Name: PseudoTrade

Team members: Brandon Magaddino ([brandon.magaddino@stonybrook.edu](mailto:brandon.magaddino@stonybrook.edu))  
Matthew Deletto ([matthew.deletto@stonybrook.edu](mailto:matthew.deletto@stonybrook.edu))  
Nelson Tsui ([nelson.tsui@stonybrook.edu](mailto:nelson.tsui@stonybrook.edu))



Working Chart: <https://www.draw.io/#G0B1EmTg9SQvVCSXhRUHBBBeTRleDA>

#### Rationale:

For the Stock, Customer, and Employee data, we classified each of the structures with its corresponding data. Each of the structures have reference links to each other, whether it be directly or indirectly. All of the structures contain their corresponding identifiers which will be used as a reference to link to other structures in the database. e.g. Stock has Symbol which can be stored in Order to reference a particular stock in the Stock structure. It's important to note that we added Portfolio to hold the Stocks a Customer owns. A Customer should theoretically have many Portfolio rows.

<pre> CREATE TABLE Order (     OrderID                INTEGER,     Symbol                 CHAR(5),     OrderType              OrderTypeCons,     TransFee               FLOAT(52),     EmployeeID             INTEGER,     PriceType              PriceTypeCons     DatePlaced             DATE,     Shares                 PosInt,     CustomerAccountNumber  INTEGER,      PRIMARY KEY (OrderID),      FOREIGN KEY (EmployeeID) REFERENCES Employee(id)         ON DELETE NO ACTION         ON UPDATE CASCADE,      FOREIGN KEY (CustomerAccountNumber) REFERENCES         Employee(AccountNumber)         ON DELETE NO ACTION         ON UPDATE CASCADE )  CREATE TABLE Customer (     AccountNumber           INTEGER,     AccountCreatedDate      DATE,     LastName                CHAR(32),     FirstName               CHAR(32),     Email                   CHAR(64),     Telephone               PhoneCons,     Rating                  FLOAT(52),     CreditCardNumber        CHAR(64),     ZipCode                 ZipCons,     State                   CHAR(2),     City                    CHAR(64),     Address                  CHAR(64),      PRIMARY KEY (AccountNumber) )  CREATE TABLE Employee( </pre>	<pre> // Stock symbol links to stock // Only allowed 0-3 // percentage // Reference to employee(id) // Only allowed 0-3 // Date and Time type // Positive integer // Reference to customer(accountnumber) </pre>
---	--

EmployeeID	INTEGER,
SSN	SocialCons,
LastName	CHAR(32),
FirstName	CHAR(32),
Email	CHAR(64),
Telephone	PhoneCons,
StartDate	DATE,
ZipCode	ZipCons,
State	CHAR(2),
City	CHAR(64),
Address	CHAR(64),
HourlyRate	PosFloat,

PRIMARY KEY (EmployeeID)

)

```
CREATE TABLE Stock(
```

StockSymbol	CHAR(5),
Shares	PosInt,
Price	PosFloat,
StockType	CHAR(64),
Name	CHAR(32),

PRIMARY KEY (StockSymbol)

)

```
CREATE TABLE Portfolio(
```

AccountID	INTEGER,
StockSymbol	CHAR(5),
Shares	PosInt,
Price	PosFloat,

```
// Has an customer attached to it (owner)
// Reference to stock for data look ups
// How many we own
// Price purchased for
```

```
FOREIGN KEY (AccountID) REFERENCES Customer(AccountNumber)
ON DELETE NO ACTION
ON UPDATE CASCADE,
```

```
FOREIGN KEY (StockSymbol) REFERENCES Stock(Symbol)
ON DELETE NO ACTION
ON UPDATE CASCADE
```

)

<pre> CREATE DOMAIN SocialCons CHAR(11)     CHECK VALUE LIKE (''[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9][0-9]')  CREATE DOMAIN PhoneCons CHAR(10)     CHECK VALUE LIKE (''[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')  CREATE DOMAIN ZipCons CHAR(5)     CHECK VALUE LIKE (''[0-9][0-9][0-9][0-9][0-9]')  CREATE DOMAIN OrderTypeCons INTEGER     CHECK (VALUE =&lt; 1 AND VALUE &gt; 0)  CREATE DOMAIN PriceTypeCons INTEGER     CHECK (VALUE =&lt; 3 AND VALUE &gt; 0)  CREATE DOMAIN PosInt INTEGER     CHECK (VALUE &gt; 0)  CREATE DOMAIN PosFloat FLOAT(52)     CHECK (VALUE &gt; 0) </pre>	
---	--

#### Rationale:

The structure of the relational model follows the logic of the E-R model. Each of the structures in the E-R model we created tables for. With each of their main identifiers being set as the PrimaryKey of the system. Each of these keys will be used to link one table to another.

SocialCons domain: limits the data to only 9 digits separated by proper '-' character

PhoneCons domain: limits the data to 10 digit phone numbers. (Doesn't include formatting)

ZipCons domain: limits the data to 5 digit American ZipCodes

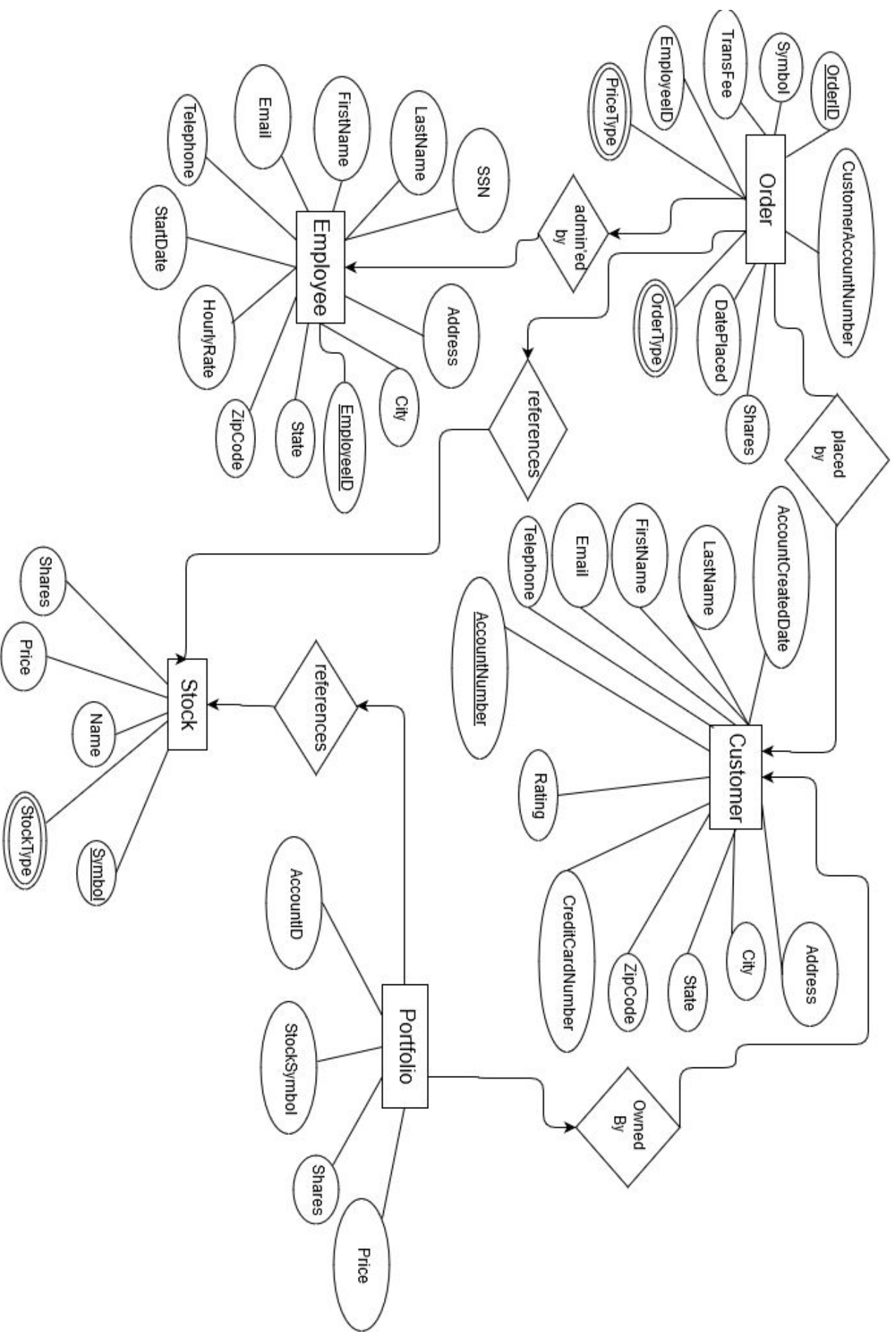
OrderTypeCons domain: limits the data to either a 1 or a 0. Could be replaced with a boolean, but leaving it a status to add flexibility later.

PriceTypeCons domain: allows data to be any of the 0-3 status' of price.

PosInt and PosFloat domain: Just limits the data to a positive decimal or float.

#### Collaboration plan:

Nelson Tsui will focus on the conditional orders. Matt DeLetto will focus on the Stock and Order data. Brandon Maggadino will focus on customer and employee data.



## QUERIES

### 3.1.1 Set the share price of a stock(for simulating market fluctuations in a stock's share price)

```
UPDATE Stock
SET PricePerShare = <pps>
WHERE StockSymbol = <stock_symbol>
```

Ex.

When <pps> = 10.2 and <stock\_symbol> = F

Then in Stock the new cell = 10.2

### 3.1.2 Add, Edit, and Delete information for employee

```
UPDATE Employee
SET <field>= <newVal>
WHERE ID = <employee_to_find>
```

Ex.

When <field> = LastName and <newVal> = DeLettoo

Then in Employee the new cell = 'DeLetto'

```
INSERT INTO Employee (ID, SSN, StartDate, HourlyRate)
VALUES(<ID>, <SSN>, <StartDate>, <HourlyRate>)
```

Ex.

When <ID> = 3 and <SSN> = 123456789 and <StartDate> = 01-13-92 and

<hourlyRate> = 15

Then a new row is added with the values 3, 123456789 , 01-13-92, 15

```
DELETE FROM Employee
WHERE ID = <Id>
```

Ex.

When <ID> = 109461292

Then the row with the column matching 109461292 is deleted

### 3.1.3 obtain a sales report for a particular month

```
SELECT *  
FROM Order AS O  
WHERE <month> = DATEPART(mm, O.DateTime)
```

Ex.

When <month> = 2

Then the row where orderID = 2 is output

### 3.1.4 Obtain a Comprehensive Listing of all Stocks

```
SELECT *  
FROM Stock AS S
```

Ex.

All rows are outputted

### 3.1.5 produce a list of orders by stock symbol or by customer name

```
SELECT *  
FROM orders AS o  
WHERE o.stockSymbol = 'GM'
```

```
SELECT *  
FROM orders AS o
```

```
WHERE o.customerID = (  
    SELECT c.customerID  
    FROM customers AS c  
    WHERE c.firstName = 'Brandon'  
)
```

Ex.

When <stock\_Symbol> = GM

Then the row where orderID = 1 is output.

3.1.6 produce a summary listing of revenue generated by a particular stock, stock type, or customer

```
SELECT SUM(o.fee)  
FROM orders AS o  
WHERE o.stockSymbol = 'GM'
```

Ex. When <particular\_stock> = GM

Then the output is the sum of the transactionFees of GM traded stocks

```
SELECT SUM(s.fee)  
FROM orders s  
WHERE s.stockSymbol = ANY (SELECT x.stockSymbol  
    FROM stocks x  
    WHERE x.stockType = 'automotive'  
)
```

Ex. When <Stock\_type> = Automotive

Then the output is the sum of the transactionFees of all orders where type is automotive

```
SELECT SUM(s.fee)  
FROM Orders s
```



WHERE s.customerID = 1

Ex. When <Client\_Id> = 111-111-1111

Then the output is the sum of the transactionFees associated with that account number

3.1.7 Determine which customer representative generated the most total revenue.

```
SELECT c.SSN, c.LastName, c.FirstName, c.Address, c.City, c.State, c.ZipCode,
c.Telephone
FROM Employees c, TotalFee t
WHERE c.EmployeeId = t.EmployeeId
ORDER BY t.TransactionFee ASC|DESC
LIMIT 1
```

```
CREATE VIEW TotalFee AS (
    SELECT O.EmployeeId, SUM(O.TransactionFee) AS TotalRev
    From Order AS O
    WHERE (SELECT *
           FROM Order AS Q
           WHERE O.EmployeeID = Q.EmployeeID
          )
)
```

Ex.

Output would be the name of the employee who is associated with the max of the sums of transactionFee's

### 3.1.8 Determine which customer generated most total revenue

```
SELECT c.ClientId, c.LastName, c.FirstName, c.Address, c.City, c.State, c.ZipCode,  
c.Telephone, c.Email, c.CreditCard, c.Rating  
FROM Client c, TotalFee t  
WHERE c.ClientId = t.ClientId  
ORDER BY t.TransactionFee ASC|DESC  
LIMIT 1
```

```
CREATE VIEW TotalFee AS (  
    SELECT O.ClientId, SUM(O.TransactionFee) AS TotalRev  
    From Order AS O  
    WHERE (SELECT *  
           FROM Order AS Q  
           WHERE O.ClientID = Q.ClientID  
          )
```

Ex.

Output would be the name of the Client who is associated with the max of the sums of transactionFee's

### 3.1.9 Produce a List of most actively traded Stocks

```
CREATE VIEW TopStocks(TSId) AS  
SELECT DISTINCT T.StockId  
FROM Trade AS T  
GROUP BY name  
ORDER BY COUNT(*) ASC  
LIMIT <Lim_Num>;
```

Ex.

Output from the sample data would be IBM

### 3.2.1 Record an order

```
INSERT INTO Orders (OrderId, Stock, OrderType, NumShares, ClientAcct, PriceType)
```

VALUES(<orderId>, <StockRef>, <Type>, <Number of Shares>, <Client account>,  
<price type>)

Ex.

When (respectively values) 1, GM, buy, 75, 4444444444, 1, market

Then data is propagated (1, GM, buy, 75, 4444444444, 1, market)

### 3.2.2 Add, Edit and Delete information for a customer

INSERT INTO clients(ClientId, LastName, FirstName, Address, City, State, ZipCode,  
Telephone, Email, CreditCard, Rating)

VALUES(<ClientId>, <LastName>, <FirstName>, <Address>, <City>, <State>,  
<ZipCode>, <Telephone>, <Email>, <CreditCard>, <Rating>)

Ex.

When (respectively values) 123456789, Smith, David, 123 College road, Stony Brook, NY,  
11790, 5162152345, 11-1-05, \$60, 1

Then data is propagated (123456789, Smith, David, 123 College road, Stony Brook, NY, 11790,  
5162152345, 11-1-05, \$60, 1)

UPDATE clients  
SET <field>= <newVal>  
WHERE ClientId = <customer id>

Ex.

When <field> = FirstName, <newVal> = 'Dan', <customer id>= 123456789

Then cell firstName is now equal to (Dan)

DELETE FROM clients  
WHERE ClientId = <Id>

Ex.

When <CustomerId> =4444444444

Then client id 4444444444 is deleted.

### 3.2.3 Produce customer mailing lists

SELECT c.address, c.city, c.state  
FROM clients c

WHERE ClientId = <Id>

Ex.

When <CustomerId> =4444444444

Then (135 Knowledge Lane | Stony Brook | NY | 11794)

3.2.4 Produce a list of stock suggestions for a given customer (based on that customer's past orders)

```
SELECT s.Stock
FROM Orders s
WHERE s.OrderType = 'BUY' AND s.ClientAcct = <CustomerId>
```

Ex.

When <CustomerId> =4444444444

Then the list of all stocks the user bought is returned, in this case (GM)

=====

### 3.3.1

A customer's current stock holdings

let <clientId> = customer's client id

```
SELECT A.Stock
FROM Accounts A
WHERE A.ClientId = <clientId>;
```

=====

### 3.3.2

The share-price and trailing-stop history for a given conditional order

let <orderId> = some given conditional order

```
SELECT S.SharePrice, O.PriceType
FROM STOCKS S, ORDERS O
WHERE O.Id = <orderId> AND
```

```
S.Symbol = O.Stock AND
O.PriceType LIKE "trailing stop%";
```

=====

### 3.3.3

The share-price and hidden-stop history for a given conditional order

let <orderId> = some given conditional order

```
SELECT S.SharePrice, O.PriceType
FROM STOCKS S, ORDERS O
WHERE      O.Id = <orderId> AND
           S.Symbol = O.Stock AND
           O.PriceType LIKE "hidden stop%";
```

=====

### 3.3.4

The share-price history of a given stock over a certain period of time (e.g., past six months)

let <currDate> = the current date

let <time> = given period of time

let <stockSymbol> = given stock symbol

```
SELECT ORDERS.Price
FROM ORDERS, TRADE
WHERE      TRADE.StockId = <stockSymbol> AND
           ORDER.Id = TRADE.OrderId AND
           ORDER.DateTime < currDate AND
           ORDER.DateTime > currDate - <time>
ORDER BY ORDERS.DateTime DESC;
```

=====

### 3.3.5

A history of all current and past orders a customer has placed

let <customers> = given clientID

```
SELECT ORDERS.Id
FROM ORDERS
WHERE ORDERS.ClientAcct LIKE "<customers> %";
```

=====

### 3.3.6

Stocks available of a particular type and most-recent order info

let <stockType> = given stock type

```
SELECT Stock.Symbol
FROM Stock
WHERE Stock.Type = <stockType>;
```

---

Most recent order information

```
SELECT Order.Id
From Order, Trade, Stock
WHERE      Order.Date = (SELECT MAX(Order.Date) FROM Order) AND
          Trade.OrderId = Order.Id
          Trade.StockId = (      SELECT Stock.Symbol
                                FROM Stock
                                WHERE Stock.Type = <stockType>;
                                );
```

=====

### 3.3.7

Stocks available with a particular keyword or set of keywords in the stock name, and most-recent order info

let <keyword> = particular keyword

```
SELECT Stock.Symbol
FROM Stock
WHERE Stock.Name = "%<stock>%"
```

---

Most recent order information

```
SELECT Order.Id
From Order, Trade, Stock
WHERE      Order.Date = (SELECT MAX(Order.Date) FROM Order) AND
          Trade.OrderId = Order.Id
          Trade.StockId = (      SELECT Stock.Symbol
                                FROM Stock
                                WHERE Stock.Name = "%<stock>%"
                                );
```

=====

### 3.3.8

Best-Seller list of stocks

```
SELECT Stock, SUM(NumShares) AS ShareSold
FROM Orders
WHERE OrderType = 'Buy'
GROUP BY Stock
ORDER BY ShareSold DESC
```

=====

### 3.3.9

Personalized stock suggestion list

```
SELECT DISTINCT T.StockId, S.CompanyName, S.Type, S.PricePerShare, S.NumShares
FROM Trade T, Stock S, Account A
WHERE T.StockId = S.StockSymbol AND A.Id = T.AccountId AND A.Client = ?
```

=====

## SQL DUMP:

```
CREATE DATABASE IF NOT EXISTS `stocks` /*!40100 DEFAULT CHARACTER SET utf8 */;
USE `stocks`;
-- MySQL dump 10.13 Distrib 5.7.9, for Win64 (x86_64)
--
-- Host: 127.0.0.1    Database: stocks
--
-- Server version 5.7.11-log
```



```

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

```

```

--
-- Table structure for table `account`
--

```

```

DROP TABLE IF EXISTS `account`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `account` (
  `Id` int(11) NOT NULL,
  `DateOpened` date DEFAULT NULL,
  `Client` int(11) DEFAULT NULL,
  `AccountNum` int(11) DEFAULT NULL,
  `Stock` char(5) DEFAULT NULL,
  `NumShares` int(11) DEFAULT NULL,
  PRIMARY KEY (`Id`),
  KEY `Client` (`Client`),
  CONSTRAINT `account_ibfk_1` FOREIGN KEY (`Client`) REFERENCES `client` (`Id`) ON
DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

```

```

--

```

-- Dumping data for table `account`

--

LOCK TABLES `account` WRITE;

/\*!40000 ALTER TABLE `account` DISABLE KEYS \*/;

INSERT INTO `account` VALUES

(0,'2010-01-06',444444444,1,'GM',250),(1,'2010-01-06',444444444,1,'Ford',100),(2,'2015-05-06',222222222,1,'IBM',50);

/\*!40000 ALTER TABLE `account` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

-- Dump completed on 2016-03-23 21:10:37

CREATE DATABASE IF NOT EXISTS `stocks` /\*!40100 DEFAULT CHARACTER SET utf8 \*/;  
USE `stocks`;

-- MySQL dump 10.13 Distrib 5.7.9, for Win64 (x86\_64)

--

-- Host: 127.0.0.1 Database: stocks

-- -----

-- Server version 5.7.11-log

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

```
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
```

--

-- Table structure for table `client`

--

```
DROP TABLE IF EXISTS `client`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `client` (
  `Email` char(32) DEFAULT NULL,
  `Rating` int(11) DEFAULT NULL,
  `Id` int(11) NOT NULL,
  `firstName` char(32) DEFAULT NULL,
  `lastName` char(32) DEFAULT NULL,
  `Address` char(64) DEFAULT NULL,
  `City` char(32) DEFAULT NULL,
  `State` char(2) DEFAULT NULL,
  `ZipCode` char(5) DEFAULT NULL,
  `Telephone` char(10) DEFAULT NULL,
  `CreditCard` char(20) DEFAULT NULL,
  PRIMARY KEY (`Id`),
  CONSTRAINT `client_ibfk_1` FOREIGN KEY (`Id`) REFERENCES `person` (`SSN`) ON
DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;
```

--

-- Dumping data for table `client`

--

```

LOCK TABLES `client` WRITE;
/*!40000 ALTER TABLE `client` DISABLE KEYS */;
INSERT INTO `client` VALUES ('syang@cs.sunysb.edu',1,111111111,'Shang','Yang','123
Success Street','Stony
Brook','NY','11790','5166328959','1234567812345678'),('vicdu@cs.sunysb.edu',1,222222222,'V
ictor','Du','456 Fortune Road','Stony
Brook','NY','11790','5166324360','5678123456781234'),('jsmith@ic.sunysb.edu',1,333333333,'J
ohn','Smith','789 Peace Blvd.','Los
Angeles','CA','93536','3154434321','2345678923456789'),('pml@cs.sunysb.edu',1,444444444,'L
ewis','Philip','135 Knowledge Lane','Stony
Brook','NY','11794','5166668888','6789234567892345');
/*!40000 ALTER TABLE `client` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2016-03-23 21:10:38

CREATE DATABASE IF NOT EXISTS `stocks` /*!40100 DEFAULT CHARACTER SET utf8 */;
USE `stocks`;
-- MySQL dump 10.13 Distrib 5.7.9, for Win64 (x86_64)
--
-- Host: 127.0.0.1    Database: stocks
--
-- Server version  5.7.11-log

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;

```

```

/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

```

```

--
-- Table structure for table `employee`
--

```

```

DROP TABLE IF EXISTS `employee`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `employee` (
  `ID` int(11) NOT NULL,
  `SSN` int(11) DEFAULT NULL,
  `StartDate` date DEFAULT NULL,
  `HourlyRate` int(11) DEFAULT NULL,
  `LastName` char(32) DEFAULT NULL,
  `FirstName` char(32) DEFAULT NULL,
  `Address` char(64) DEFAULT NULL,
  `City` char(64) DEFAULT NULL,
  `State` char(2) DEFAULT NULL,
  `Zipcode` char(5) DEFAULT NULL,
  PRIMARY KEY (`ID`),
  KEY `SSN` (`SSN`),
  CONSTRAINT `employee_ibfk_1` FOREIGN KEY (`SSN`) REFERENCES `person` (`SSN`)
  ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

```

```

--
-- Dumping data for table `employee`
--

LOCK TABLES `employee` WRITE;
/*!40000 ALTER TABLE `employee` DISABLE KEYS */;
INSERT INTO `employee` VALUES (0,123456789,'2011-01-05',60,'Smith','David','123 College
Road','Stony Brook','NY','11790'),(1,789123456,'0002-02-06',50,'Warren','David','456 Sunken
Street','Stony Brook','NY','11794');
/*!40000 ALTER TABLE `employee` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2016-03-23 21:10:38


CREATE DATABASE IF NOT EXISTS `stocks` /*!40100 DEFAULT CHARACTER SET utf8 */;
USE `stocks`;
-- MySQL dump 10.13 Distrib 5.7.9, for Win64 (x86_64)
--
-- Host: 127.0.0.1    Database: stocks
--
-- Server version 5.7.11-log

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

```

```

/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `location`
--

DROP TABLE IF EXISTS `location`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `location` (
  `ZipCode` int(11) NOT NULL,
  `City` char(20) NOT NULL,
  `State` char(20) NOT NULL,
  PRIMARY KEY (`ZipCode`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `location`
--

LOCK TABLES `location` WRITE;
/*!40000 ALTER TABLE `location` DISABLE KEYS */;
INSERT INTO `location` VALUES (11790,'Stony Brook','NY'),(11794,'Stony
Brook','NY'),(93536,'Los Angeles','CA');
/*!40000 ALTER TABLE `location` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

```

```

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2016-03-23 21:10:38

CREATE DATABASE IF NOT EXISTS `stocks` /*!40100 DEFAULT CHARACTER SET utf8 */;
USE `stocks`;
-- MySQL dump 10.13 Distrib 5.7.9, for Win64 (x86_64)
--
-- Host: 127.0.0.1    Database: stocks
--
-- Server version 5.7.11-log

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `orders`
--

DROP TABLE IF EXISTS `orders`;

```



```

/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `orders` (
  `NumShares` int(11) DEFAULT NULL,
  `PricePerShare` double DEFAULT NULL,
  `Id` int(11) NOT NULL,
  `DateTime` datetime DEFAULT NULL,
  `Percentage` double DEFAULT NULL,
  `Price` double DEFAULT NULL,
  `SOrder` char(5) DEFAULT NULL,
  `clientID` int(11) DEFAULT NULL,
  `Stock` char(5) DEFAULT NULL,
  PRIMARY KEY (`Id`),
  KEY `clientID` (`clientID`),
  CONSTRAINT `orders_ibfk_1` FOREIGN KEY (`clientID`) REFERENCES `client` (`Id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `orders`
--

LOCK TABLES `orders` WRITE;
/*!40000 ALTER TABLE `orders` DISABLE KEYS */;
INSERT INTO `orders` VALUES (75,34.23,1,'0001-01-06
00:00:00',0,0,'0',4444444444,'GM'),(NULL,91.41,2,'0002-02-06
00:00:00',10,90,'1',222222222,'IBM');
/*!40000 ALTER TABLE `orders` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;

```

```
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

-- Dump completed on 2016-03-23 21:10:38

```
CREATE DATABASE IF NOT EXISTS `stocks` /*!40100 DEFAULT CHARACTER SET utf8 */;
USE `stocks`;
```

-- MySQL dump 10.13 Distrib 5.7.9, for Win64 (x86\_64)

--

-- Host: 127.0.0.1 Database: stocks

-- -----

-- Server version 5.7.11-log

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
```

--

-- Table structure for table `person`

--

```
DROP TABLE IF EXISTS `person`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `person` (
  `SSN` int(11) NOT NULL,
  `LastName` char(20) NOT NULL,
```

```

`FirstName` char(20) NOT NULL,
`Address` char(20) DEFAULT NULL,
`ZipCode` int(11) DEFAULT NULL,
`Telephone` char(11) DEFAULT NULL,
PRIMARY KEY (`SSN`),
KEY `ZipCode` (`ZipCode`),
CONSTRAINT `person_ibfk_1` FOREIGN KEY (`ZipCode`) REFERENCES `location`
(`ZipCode`) ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `person`
--

LOCK TABLES `person` WRITE;
/*!40000 ALTER TABLE `person` DISABLE KEYS */;
INSERT INTO `person` VALUES (111111111,'Yang','Shang','123 Success
Street',11790,'5166328959'),(123456789,'Smith','David','123 College
road',11790,'5162152345'),(222222222,'Du','Victor','456 Fortune
Road',11790,'5166324360'),(333333333,'Smith','John','789 Peace
Blvd.',93536,'3154434321'),(444444444,'Philip','Lewis','135 Knowledge
Lane',11794,'5166668888'),(789123456,'Warren','David','456 Sunken
Street',11794,'6316329987');
/*!40000 ALTER TABLE `person` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

```

-- Dump completed on 2016-03-23 21:10:38

CREATE DATABASE IF NOT EXISTS `stocks` /\*!40100 DEFAULT CHARACTER SET utf8 \*/;  
USE `stocks`;

-- MySQL dump 10.13 Distrib 5.7.9, for Win64 (x86\_64)

--

-- Host: 127.0.0.1 Database: stocks

-- -----

-- Server version 5.7.11-log

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;  
/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;  
/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;  
/\*!40101 SET NAMES utf8 \*/;  
/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;  
/\*!40103 SET TIME\_ZONE='+00:00' \*/;  
/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;  
/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS,  
FOREIGN\_KEY\_CHECKS=0 \*/;  
/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE,  
SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;  
/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Dumping events for database 'stocks'

--

--

-- Dumping routines for database 'stocks'

--

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

```

/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2016-03-23 21:10:39

CREATE DATABASE IF NOT EXISTS `stocks` /*!40100 DEFAULT CHARACTER SET utf8 */;
USE `stocks`;
-- MySQL dump 10.13 Distrib 5.7.9, for Win64 (x86_64)
--
-- Host: 127.0.0.1    Database: stocks
--
-- Server version 5.7.11-log

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `stock`
--

DROP TABLE IF EXISTS `stock`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `stock` (
  `StockSymbol` char(20) NOT NULL,

```

```

`CompanyName` char(20) NOT NULL,
`Type` char(20) NOT NULL,
`PricePerShare` double DEFAULT NULL,
`NumShares` int(11) DEFAULT NULL,
PRIMARY KEY (`StockSymbol`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `stock`
--

LOCK TABLES `stock` WRITE;
/*!40000 ALTER TABLE `stock` DISABLE KEYS */;
INSERT INTO `stock` VALUES ('F','Ford','automotive',9,750),('GM','General
Motors','automotive',34.23,1000),('IBM','IBM','Computer',91.41,500);
/*!40000 ALTER TABLE `stock` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2016-03-23 21:10:39

CREATE DATABASE IF NOT EXISTS `stocks` /*!40100 DEFAULT CHARACTER SET utf8 */;
USE `stocks`;
-- MySQL dump 10.13 Distrib 5.7.9, for Win64 (x86_64)
--
-- Host: 127.0.0.1    Database: stocks
-- -----

```

-- Server version 5.7.11-log

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
```

--

-- Table structure for table `trade`

--

```
DROP TABLE IF EXISTS `trade`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `trade` (
  `AccountId` int(11) NOT NULL,
  `BrokerId` int(11) NOT NULL,
  `TransactionId` int(11) NOT NULL,
  `OrderId` int(11) NOT NULL,
  `StockId` char(20) NOT NULL,
  PRIMARY KEY (`AccountId`,`BrokerId`,`TransactionId`,`OrderId`,`StockId`),
  KEY `BrokerId` (`BrokerId`),
  KEY `TransactionId` (`TransactionId`),
  KEY `OrderId` (`OrderId`),
  KEY `StockId` (`StockId`),
  CONSTRAINT `trade_ibfk_1` FOREIGN KEY (`AccountId`) REFERENCES `account` (`Id`) ON
DELETE NO ACTION ON UPDATE CASCADE,
```

```
CONSTRAINT `trade_ibfk_2` FOREIGN KEY (`BrokerId`) REFERENCES `employee` (`ID`) ON
DELETE NO ACTION ON UPDATE CASCADE,
CONSTRAINT `trade_ibfk_3` FOREIGN KEY (`TransactionId`) REFERENCES `transaction`
(`Id`) ON DELETE NO ACTION ON UPDATE CASCADE,
CONSTRAINT `trade_ibfk_4` FOREIGN KEY (`OrderId`) REFERENCES `orders` (`Id`) ON
DELETE NO ACTION ON UPDATE CASCADE,
CONSTRAINT `trade_ibfk_5` FOREIGN KEY (`StockId`) REFERENCES `stock`
(`StockSymbol`) ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;
```

```
--
-- Dumping data for table `trade`
--
```

```
LOCK TABLES `trade` WRITE;
/*!40000 ALTER TABLE `trade` DISABLE KEYS */;
/*!40000 ALTER TABLE `trade` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
```

```
/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

```
-- Dump completed on 2016-03-23 21:10:38
```

```
CREATE DATABASE IF NOT EXISTS `stocks` /*!40100 DEFAULT CHARACTER SET utf8 */;
USE `stocks`;
```

```
-- MySQL dump 10.13 Distrib 5.7.9, for Win64 (x86_64)
```

```
--
```

```
-- Host: 127.0.0.1 Database: stocks
```



-- -----  
-- Server version 5.7.11-log

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;  
/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;  
/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;  
/\*!40101 SET NAMES utf8 \*/;  
/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;  
/\*!40103 SET TIME\_ZONE='+00:00' \*/;  
/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;  
/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS,  
FOREIGN\_KEY\_CHECKS=0 \*/;  
/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE,  
SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;  
/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--  
-- Table structure for table `transaction`  
--

DROP TABLE IF EXISTS `transaction`;  
/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  
/\*!40101 SET character\_set\_client = utf8 \*/;  
CREATE TABLE `transaction` (  
 `Id` int(11) NOT NULL,  
 `Fee` double DEFAULT NULL,  
 `DateTime` datetime DEFAULT NULL,  
 `PricePerShare` double DEFAULT NULL,  
 PRIMARY KEY (`Id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--  
-- Dumping data for table `transaction`  
--

```
LOCK TABLES `transaction` WRITE;
/*!40000 ALTER TABLE `transaction` DISABLE KEYS */;
/*!40000 ALTER TABLE `transaction` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2016-03-23 21:10:38
```

## USER GUIDE

- Customer Help
  - View your current stock holdings
  - The share-price and trailing-stop history for a given conditional order
  - The share-price and hidden-stop history for a given conditional order
  - The share-price history of a given stock over a certain period of time
  - A history of all current and past orders a customer has placed
  - Stocks available of a particular type and most-recent order info
  - Stocks available with a particular keyword or set of keywords in the stock name, and most-recent order info
  - List of best selling stocks
  - Personalized stock suggestion list
  - 
  -
- Employee Help
  - Record an order
  - Add, edit or delete customer information
  - Produce a customer mailing list
  - Produce a list of stock suggestions
- Manager Help
  - Set the share price of a stock (for simulating market fluctuations in a stock's share price)
  - Add, Edit and Delete information for an employee
  - Obtain a sales report for a particular month
  - Produce a comprehensive listing of all stocks
  - Produce a list of orders by stock symbol or by customer name
  - Produce a summary listing of revenue generated by a particular stock, stock type, or customer
  - Determine which customer representative generated most total revenue
  - Determine which customer representative generated most total revenue

- Produce a list of most actively traded stocks

## **Customer Help Menu**

### **View your current stock holdings**

To view your current stock holdings. Sign into your account through the button on the top-right corner of the window. Once you enter your information and enter your customer prompt screen. Press the 'current stock holdings' button and it should bring up a window for you to view all of your stocks, how many shares you have, how much money you made from them, and more!

### **The share-price and trailing-stop history for a given conditional order**

To view the share price and trailing stop history for a given condition order. Make sure you are signed into your account. Then on the customer homepage click the view trailing-stop history button and you will be brought to a page with your stocks and their trailing stop history. If you would like to view an individual stock just click on it.

### **The share-price and hidden-stop history for a given conditional order**

To view the share price and hidden stop history is simple. Make sure you are signed into your account. then on the customer homepage click the view hidden stop history button and you will be brought to a page with your stocks and their trailing stop history. If you would like to view an individual stock just click on it.

### **The share-price history of a given stock over a certain period of time**

To view the share price history of a given stock over a certain period of time. Make sure to be signed into your account. When you are on the customer homepage just click the share-price history and it should redirect you to another page. Once on the other page type in your given stock and the timeframe you want to inspect. Now click submit and it will appear.

### **A history of all current and past orders a customer has placed**

To view the order history of all current and past orders that you have placed. Go to the customer prompt menu then click on order history. This should direct you to a page with all of your current and past orders.

### **Stocks available of a particular type and most-recent order info**

To view the stocks that are available of a particular type. Go to the customer homepage and click on buy stocks. Once there just filter out the stocks that you do not want to view and it should narrow the results down to just the ones that you want to inspect. To view your most

recent order info. Go to order history and view the first order in the table. That will be your most-recent order info.

### **Stocks available with a particular keyword or set of keywords in the stock name, and most-recent order info**

To view stocks available with a particular keyword or set of keywords. Go to the customer homepage then click on buy stocks. Once there just filter out by typing in the keywords you want to use to filter the buy results. To view your most-recent order info refer to [here](#).

### **List of best selling stocks**

To view the best selling stocks go to the customer homepage then click on the buy stocks option. Once here you can simply click on the best selling stocks option and it should show a list of all the best selling stocks.

### **Personalized stock suggestion list**

To view a personalized stock suggestion based on your order history. Just go to the customer homepage and click on the personalized stock suggestions list.

## **Employee Help Menu**

### **Record an order**

To record an order for a customer go to the employee homepage and click the record an order button. This should redirect you to the record order page. Now fill in all of the input boxes as per the customer's requests then click submit.

### **Add, edit, or remove customer information**

To edit, add, or remove customer information. Go to the employee homepage and click the customer button. Here you should see a list of all of the customers. If you want to add a customer click the add customer button. If you want to remove a customer click the customer you wish to remove and hit the remove button. If you want to edit a customer. Edit the customer select the customer and click edit. Now it should redirect you to a page. Fill out what you want to edit about the customer

### **Produce a mailing list**

To generate a mailing list of all of the customer. Go to the employee homepage and click generate mailing list. This will redirect you to the mailing list page. Just copy and paste into your favorite email client and mail away!

## **Produce a list of stock suggestions**

To produce a list of stock suggestions for a particular customer. Go to the employee homepage then click produce stock suggestions. You will be redirected to the produce stock suggestions page. Now select the customer you want to generate the suggestion for.

## **Manager Help Menu**

### **Set the share price of a stock (for simulating market fluctuations in a stock's share price)**

To set the share price of a stock. Go to the manager homepage and click set share price. Now set share you wish to manipulate and click edit. Now type in your desired share price and click submit.

### **Add, Edit and Delete information for an employee**

To add, edit, or delete information for an employee. Go to the manager homepage and click on add employee to add an employee, select an employee then click edit to edit an employee, or click select an employee then click remove to remove the employee.

### **Obtain a sales report for a particular month**

To obtain a sales report for a particular month simply go to the manager homepage then click on sales report. Now select the particular month and you will be redirected to a page showing all the information of a sales report.

### **Produce a comprehensive listing of all stocks**

To produce a list of all stocks. Go to the manager homepage and click the produce stocks button. This will redirect you to the page with a table containing a listing of all stocks.

### **Produce a list of orders by stock symbol or by customer name**

To produce a list of orders whether by stock symbol or by customer name. Go to the manager homepage and click the list of orders button. Now select whether you want it by stock symbol or customer name. Based on the stock symbol/customer name select it should redirect you to the page that lists all orders based on your criteria.

### **Produce a summary listing of revenue generated by a particular stock, stock type, or customer**

To produce a summary listing of all the revenue generated by a particular stock. Go to the manager homepage and select the summary revenue button. Now click the dropdown menu of

the type of revenue you wish to generate. Then select the customer, stock, or stocktype and it should generate at the bottom of the window.

### **Determine which customer representative generated most total revenue**

To determine which employee has generated the most total revenue. Simply go to the manager homepage and click most revenue. Now it will redirect you to a page with either customers or customer representative. Select customer representatives. It will now show which customer representative/employee has generate the most total revenue overall.

### **Determine which customer generated most total revenue**

Follow the steps similar to here. Then instead of picking the customer representative option. Pick the customer option. It will now display the customer who has generate the most total revenue.

### **Produce a list of most actively traded stocks**

To produce a list of the most actively traded stocks. Go to the manager homepage. Now click the view stocks button. Now sort the stocks from by the amount of orders to view the most actively traded stocks.

Copyright © PsuedoTrade 2016