CSE 305
Spring 2012
Course Project Assignment #1
(ER & Relational Models for Online Trading System)
Solution

Feb. 16, 2012
1. ER Diagram
Rationale for E-R Model

Trading information is primarily specified using three entities: Account, Stock, and Transaction. These entities are linked through the relationship HasStock, which relates stocks with the accounts they are held in, and the relationship Trade, which represents the actual buying and selling of stocks. Notice that the roles Transaction and Order are keys of the relationship Trade and that every transaction and order must be involved in some Trade relationship. Thus, there is a one-to-one correspondence between Transaction entities and Trade relationships, and Order entities and Trade relationships.

Also notice that an order is placed at a certain date/time and a transaction occurs at certain, presumably different, date/time. Furthermore, the number of shares traded is encapsulated in the Order entity. As for the price per share, this information resides in both the Transaction and Order entities: a hidden-stop order dictates the sale price of a share of stock; for all other types of orders, the price per share will be determined at runtime, that is, when the transaction occurs. Finally, hidden- and trailing-stop conditional orders are, by definition, sell orders, while market and market-on-close orders may be buy or sell orders.

Please see Figure 4.31 (p. 109) of the course textbook for a similar E-R perspective of a stock-trading system.
CREATE TABLE Person (  
    SSN     INTEGER,  
    LastName CHAR(20) NOT NULL,  
    FirstName CHAR(20) NOT NULL,  
    Address  CHAR(20),  
    ZipCode  INTEGER,  
    Telephone INTEGER,  
    PRIMARY KEY (SSN),  
    FOREIGN KEY (ZipCode) REFERENCES Location (ZipCode)  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE )

CREATE TABLE Location (  
    ZipCode INTEGER,  
    City CHAR(20) NOT NULL,  
    State CHAR(20) NOT NULL,  
    PRIMARY KEY (ZipCode) )

CREATE TABLE Employee (  
    ID    EmpId,  
    SSN   INTEGER,  
    StartDate DATE,  
    HourlyRate INTEGER,  
    PRIMARY KEY (ID),  
    FOREIGN KEY (SSN) REFERENCES Person (SSN)  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE )
2. Relational Model (2)

CREATE TABLE Account (  
    Id INTEGER,  
    DateOpened DATE,  
    Client ClientId,  
    PRIMARY KEY (Id),  
    FOREIGN KEY (Client) REFERENCES Client (Id)  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE    )

CREATE TABLE Client (  
    Email CHAR(32),  
    Rating INTEGER,  
    CreditCardNumber INTEGER,  
    Id ClientId,  
    PRIMARY KEY (Id),  
    FOREIGN KEY (Id) REFERENCES Person (SSN)  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE    )

CREATE TABLE Transaction (  
    Id TxnId,  
    Fee CURRENCY,  
    DateTime DATETIME,  
    PricePerShare CURRENCY  
    PRIMARY KEY (Id)    )
CREATE TABLE Order (  
    NumShares INTEGER,  
    PricePerShare CURRENCY,  
    Id INTEGER,  
    DateTime DATETIME,  
    Percentage PERCENTAGE,  
    Price PriceType,  
    Order OrderType,  
    PRIMARY KEY (Id)  
)

CREATE TABLE Stock (  
    StockSymbol CHAR(20) NOT NULL,  
    CompanyName CHAR(20) NOT NULL,  
    Type CHAR(20) NOT NULL,  
    PricePerShare CURRENCY,  
    PRIMARY KEY (StockSymbol)  
)

CREATE DOMAIN PriceType CHAR(20)  
    CHECK ( VALUE IN ('Market', 'MarketOnClose', 'TrailingStop', 'HiddenStop') )

CREATE DOMAIN OrderType CHAR(5)  
    CHECK ( VALUE IN ('Buy', 'Sell') )

CREATE DOMAIN EmpId INTEGER  
    CHECK (EmpId > 0 AND EmpId < 1000000000)

CREATE DOMAIN ClientId INTEGER  
    CHECK (ClientId > 0 AND ClientId < 1000000000)
2. Relational Model (4)

CREATE TABLE Trade (
    AccountId INTEGER,
    BrokerId  EmpId,
    TransactionId INTEGER,
    OrderId INTEGER,
    StockId CHAR(20),
    PRIMARY KEY (AccountId, BrokerId, TransactionId, OrderId, StockId),
    FOREIGN KEY (AccountID) REFERENCES Account (Id)
        ON DELETE NO ACTION
        ON UPDATE CASCADE,
    FOREIGN KEY (BrokerId) REFERENCES Employee (Id)
        ON DELETE NO ACTION
        ON UPDATE CASCADE,
    FOREIGN KEY (TransactionID) REFERENCES Transaction (Id)
        ON DELETE NO ACTION
        ON UPDATE CASCADE,
    FOREIGN KEY (OrderId) REFERENCES Order (Id)
        ON DELETE NO ACTION
        ON UPDATE CASCADE,
    FOREIGN KEY (StockId) REFERENCES Stock (StockSymbol)
        ON DELETE NO ACTION
        ON UPDATE CASCADE
    )