**Core**

SQL

Final Project - Database Functionality and Reports

© FDM Group Ltd 2014. All Rights Reserved.

Any unauthorized reproduction or distribution in part  
or in whole will constitute an infringement of copyright.

# 

# 1. Project specification

It is recommended that you read the entire specification before you begin.

## Basic Reports

### Views

Two versions of two queries are provided. Refer to: QueriesProvided.sql

1. Current\_Shareholder\_Shares – Two queries are provided in QueriesProvided.sql. Both of these queries list shareholder id, shareholder type, stock id, and the total shares currently held by the shareholder. Create a view called CURRENT\_SHAREHOLDER\_SHARES using the more efficient query and include the view in your script. Please place a comment explaining why you chose the query.
2. Current\_Stock\_Stats – These queries list each stock id, the number of shares currently authorized, and the total number of shares currently outstanding. Create a view called CURRENT\_STOCK\_STATS using the more efficient query and include it in your script. Please place a comment explaining why you chose the query.

### Queries

1. Write a query which lists the name of every company that has authorized stock, the number of shares currently authorized, the total shares currently outstanding, and % of authorized shares that are outstanding.  
   Shares outstanding is the number of shares owned by external share holders.   
   Shares\_Authorized = Shares\_Outstanding + Shares\_UnIssued
2. For every direct holder: list the name of the holder, the names of the companies invested in by this direct holder, number of shares currently held, % this holder has of the shares outstanding, and % this holder has of the total authorized shares. Sort the output by direct holder last name, first name, and company name and display the percentages to two decimal places.
3. For every institutional holder (companies who hold stock): list the name of the holder, the names of the companies invested in by this holder, shares currently held, % this holder has of the total shares outstanding, and % this holder has of that total authorized shares. For this report, include only the external holders (not treasury shares). Sort the output by holder name, and company owned name and display the percentages to two decimal places.
4. Write a query which displays all trades where more than 50000 shares were traded on the secondary markets. Please include the trade id, stock symbol, name of the company being traded, stock exchange symbol, number of shares traded, price total (including broker fees) and currency symbol.
5. For each stock listed on each stock exchange, display the exchange name, stock symbol and the date and time when that the stock was last traded. Sort the output by stock exchange name, stock symbol. If a stock has not been traded show the exchange name, stock symbol and null for the date and time.
6. Display the trade\_id, name of the company and number of shares for the single largest trade made on any secondary market (in terms of the number of shares traded). Unless there are multiple trades with the same number of shares traded, only one record should be returned.

## Data Manipulation

Write the necessary INSERT, UPDATE and/or DELETE statements to complete the following data changes.

### Add a Direct Holder

1. Add “Jeff Adams” as a new direct holder. You will have to insert a record into the shareholder table and make a separate statement to insert into the direct\_holder table.

### Add an Institutional Holder

1. Add “Makoto Investing” as a new institutional holder that has its head office in Tokyo, Japan. Makoto does not currently have a stock id. A record must be inserted into the shareholder table and a corresponding record must be inserted into the company table.

### Initial Public Offering (IPO)

1. “Makoto Investing” would like to declare stock. As of today’s date, they are authorizing 100,000 shares at a starting price of 50 yen.   
   To complete the work, you will need to update the company table to give Makoto its own stock id, and insert a new entry in the shares\_authorized table.

### Listing on an Exchange

1. “Makoto Investing” would like to list on the Tokyo Stock Exchange under the stock symbol “Makoto”. You will need to insert into the stock\_listing table and the stock\_price table.

Submit your answers to questions 1 through 12 to your trainer now. All of your queries, and statements should be contained within a single .sql file separated by descriptive comments. While your trainer is evaluating your answers, continue with the problems below.

## Stored Procedures

Write the necessary CREATE OR REPLACE PROCEDURE statements and statements which test your procedures.

### Add a Direct Holder

1. Write a PL/SQL procedure called INSERT\_DIRECT\_HOLDER which will be used to insert new direct holders. Create a sequence object on the database to automatically generate shareholder\_ids. Use this sequence in your procedure.  
   -Input parameters: first\_name, last\_name

### Add an Institutional Holder

1. Write a PL/SQL procedure called INSERT\_COMPANY which will be used to insert new companies. The stock\_id for new companies will be null. Use the sequence object that you created in problem 13 to get new shareholder\_ids.   
   -Input parameters: company\_name, city, country

### Declare Stock (Initial Public Offering)

1. Write a PL/SQL procedure called DECLARE\_STOCK which will be used when a company declares it is issuing shares.  
   -Input parameters: company name, number of shares authorized, starting price (in the designated currency), and currency name.   
   -Check to ensure the company has not already been given a stock id.  
   -If the company already has a stock id then do not perform any data changes.  
   -Otherwise, the company must be assigned a stock id (create a sequence object to generate new stock\_ids) and the date of issue (current system date), number of shares authorized, the starting price and currency id must be recorded.

### Listing on an Exchange

1. Write a PL/SQL procedure called LIST\_STOCK which will be used when stock is listed on a stock exchange.  
   -Input parameters: stock\_id, stock\_ex\_id, stock\_symbol.  
   -The stock\_id, stock\_ex\_id and stock\_symbol must be recorded in the stock\_listing table.  
   -The starting price from company must be copied to the stock price list for the stock exchange. The current system time will be used for the time\_start and the time\_end will be null. The procedure must be able to convert currencies as needed.

### Stock Split

1. Write a PL/SQL procedure called SPLIT\_STOCK.  
   -input parameters: stock id, split\_factor  
   -The split\_factor must be greater than 1 and can be fractional. (The number of shares will be multiplied by the split\_factor.)  
   -The total shares outstanding cannot exceed the authorized amount. Your procedure should raise an application error if the split would cause the shares outstanding to exceed the shares authorized.  
   -Every shareholder must receive (is buyer of) an additional "trade" equal to the additional shares to which they are entitled. For example, if the split\_factor is 2 then each shareholder will be entitled to an additional “trade” that is equal to the number of shares that they owned before the split. (Use the Current\_Shareholder\_Shares view to determine the number of shares owned). These "trades" will not take place at a stock exchange, the price total will be null, and there will be no brokers involved.

### Reverse Split

1. Write a PL/SQL procedure called REVERSE\_SPLIT.  
   -input parameters: stock id, merge\_factor  
   -The merge\_factor must be greater than 0 and less than 1. (The number of shares will be multiplied by the merge\_factor.)  
   -Every shareholder must "sell" some of the stock it currently owns. (Use the Current\_Shareholder\_Shares view to determine the number of shares owned). If the merge\_factor is 1/3 then adjustments must be made to indicate the 2/3 of each shareholder’s stock has been removed. (The database can handle fractions of a share.) These "trades" will not take place at a stock exchange, the price total will be null, and there will be no brokers involved.

## Additional Queries

1. Display the trade id, the stock id and the total price (in US dollars) for the secondary market trade with the highest total price. Convert all prices to US dollars.
2. Display the name of the company and trade volume for the company whose stock has the largest total volume of shareholder trades worldwide. [Example calculation: A company declares 20000 shares, and issues 10000 on the new issue market (primary market), and 1000 shares is sold to a stockholder on the secondary market. Later that stockholder sells 500 shares to another stockholder (or back to the company itself). The number of shareholder trades is 2 and the total volume of shareholder trades is 1500.]
3. For each stock exchange, display the symbol of the stock with the highest total trade volume. Show the stock exchange name, stock symbol and total trade volume. Sort the output by the name of the stock exchange and stock symbol.
4. List the top 5 companies (in terms of shareholder trade volume) on the New York Stock Exchange. Display the company name, shareholder trade volume, the current price and the percentage change for the last price change, and sort the output in descending order of shareholder trade volume. The sample data in the database contains information for only 3 companies but your query must continue to list only the top 5 companies even when there is data for more companies.

# 2. Final Project submission

All of your queries, and statements for ALL sections (1 through 22) should be contained within a single .sql file separated by descriptive comments. For the reports, include only one query in your answer. For stored procedures, include both the stored procedure and one call which tests the procedure.

Final project marks will be delivered to you once all projects have been marked.