

Illinois Institute of Technology

**CS425: Database Design and Application
Spring 2004**

SQL and PL/SQL Assignment (Individual Homework)

Date Given: February 3rd

Due Date: February 17th by no later than 11:25 am (Central Time) on courseinfo & also a hard copy for all LIVE section students.

Problem Description

A stock exchange company called “CPS”, which stands for “Company to Practice SQL” is created. Assume that the database model for creating a database for this company is already modeled based on the company requirements. Thus, your job is not to model/re-model the database, but to use the model described below in terms of different relations, to create the tables, insert data into the tables, and perform some queries. You are required to use SQL for some of the queries and PL/SQL for some other queries. Following are the relations, their description, and what you need to do:

Security

<u>ticker_symbol</u>	<u>company_name</u>	<u>open_price</u>	<u>current_price</u>
AOL	America Online	15.85	17.45
MSFT	Microsoft Corporation	55.25	60.11
CPQ	Compaq Computers	34.23	34.23
AAPL	Apple Computers	20.10	20.10
ORCL	Oracle Corporation	8.57	7.77
F	Ford Motor Company	15.89	23.22
WMT	Wal Mart Stores	56.95	56.95
BA	Boeing Co	42.53	44.99
SNE	Sony Cp	52.17	51.23
ABC	ABC Corporation	21.28	21.28

Security relation: This relation contains information about every company share that is in the stock exchange and maintains the open and current price of each company share in the market in the current day. Shares are called the securities. The attributes of this relation are:

ticker_symbol (4 char) – this is the company symbol used for trade every day in the stock exchange.

company_name (20 char)- this is the name of the company.

open_price (xxxx.xx)- this is the opening price , at which the share opens for trade.

current_price (xxxx.xx): this is the current trade price of the share.

Primary key: ticker_symbol

Candidate key: company_name

Security_History

<u>ticker_symbol</u>	<u>date sh</u>	<u>High</u>	<u>open</u>	<u>close</u>	<u>low</u>
AOL	OCT-04-2003	17.88	13.67	15.85	12.00
MSFT	OCT-12-2003	55.25	55.25	55.25	55.25
CPQ	OCT-24-2003	34.99	30.22	34.23	30
AAPL	NOV-24-2003	20.10	20.10	20.10	20.10
ORCL	NOV-29-2003	8.57	8.57	8.57	8.57
F	DEC-20-2003	25.80	15.89	15.89	13.01
WMT	JAN-2-2004	56.95	56.95	56.95	56.95
BA	JAN-4-2004	42.89	42.53	42.53	42.01
SNE	JAN-5-2004	52.17	42.14	52.17	42.14
ABC	JAN-12-2004	27.00	20.11	21.28	19.01
AOL	FEB-1-2004	16.00	15.00	16.00	13.00

Security_History relation: this relation contains the history of the securities on *open*, *low*, *high* and *close* price.

high (xxxx.xx): The highest price for which a security is traded on a particular day.

low (xxxx.xx): The lowest price for which a security is traded on a particular day.

open (xxxx.xx): The opening price for which the security is traded on a particular day.

close (xxxx.xx): The closing price for which the security is traded on a particular day.

date_sh (use data type date): This is the date for each entry.

Primary key: ticker_symbol, date_sh

Foreign key: ticker_symbol references ticker_symbol in table Security.

Customer

cust_id	customer_name	status
10111	Charles Schwab	P
10112	Datek	P
10113	DLJ Direct	R
10114	E Trade	R
10115	Fidelity.com	R
10116	JB Oxford & Company	R
10117	Merrill Lynch Direct	P
10118	Wingspan Bank	R

Customer relation: This relation stores the customer's information.

cust_id (xxxxx): Customer identification number.

customer_name (15 char): Name of the customer.

status (1 char): Premium (P) customer or Regular(R) customer. Based upon the number of trades a customer does on a particular day, the status changes. If a customer trades less than 10000 shares a day, the customer's status is set to "R" regular customer; otherwise the status is set to "P".

Primary key: cust_id

Trade

trade_id	ticker_symbol	cust_id	post	quantity	price_share	buy_sell	date_sh
13	AOL	10111	P01	9000	17.45	B	OCT-04-2003
12	MSFT	10112	P01	2000	60.11	B	OCT-12-2003
11	ORCL	10115	P99	500	7.77	S	OCT-24-2003
10	F	10115	P12	6000	23.22	B	NOV-24-2003
9	BA	10111	P44	10000	42.53	S	NOV-29-2003
8	SNE	10112	P01	7000	51.23	S	OCT-12-2003
7	SNE	10115	P76	30	52.17	S	JAN-2-2004
6	ORCL	10115	P56	1100	8.50	B	JAN-4-2004
5	MSFT	10113	P18	7600	55.25	S	JAN-5-2004
4	AOL	10117	P67	1400	16.01	B	JAN-12-2004
3	ORCL	10111	P12	8000	8.57	S	FEB-1-2004
2	F	10112	P01	3000	15.89	S	FEB-3-2004
1	AOL	10117	P01	12000	15.85	S	FEB-6-2004

Trade relation: This relation stores the trade details of a particular day. It contains the ticker symbol, cust_id, post and other details about the trade.

trade_id (xxxx): This is the primary key for each stock being traded at the stock exchange.

post: this is the place in the stock exchange where the trade happens.

quantity (xxxxxxx): Number of shares traded.

price_share(xxxx.xx): Price per share traded.

buy_sell (1 character): A flag that tells that whether the customer bought the share or sold the shares.

Primary key: trade_id

Foreign key: ticker_symbol references ticker_symbol in table Security.

Foreign key: cust_id references cust_id in table Customer.

You are required to perform the following tasks:

1. (8 pts.) Create all the tables with the specified attribute names, as listed above, with appropriate primary and foreign keys. Put all the DDL statements in a file *create.sql* and create the tables and keys running the batch of SQL statements from this file.
2. (5 pts.) Insert the data into the tables, using a batch of insert statements from *insert.sql*.
3. (5 pts.) List the name and the ticker symbol of the securities, with the decreasing order of the change in the price (high -low). Include this number in the result.
4. (5 pts.) List all companies that their *open price* is higher than their *current price*.
5. (5 pts.) What are the symbols that no shares of them are ever being traded? Give the company name and the symbol.
6. (12 pts.) How old is the oldest piece of data in history table? Show the oldest date, the most recent date, and also the total number of months between the oldest and most recent date.
7. (12 pts.) Give the *cust_id* and customer name of the customer, who bought the maximum number of shares ever.
8. (12 pts.) For each customer, give the *cust_id*, customer name and total number of shares bought for each stock (call this on your report as “Ticker Symbol”) and the total amount paid for that. On your report the column names should appear as: Customer Id, Customer Name, Ticker Symbol, Total Amount.
9. (12 pts.) Give the company name, ticker symbol, and the total number of trades for each ticker symbol, along with total shares (bought + sold) traded.
10. (12 pts.) Write a PL/SQL procedure *find_security* () that takes in the *ticker symbol* and *post* and prints the number of shares bought, the number of shares sold, indicating clearly “bought” or “sold” for that given symbol and post. Call the *find_security* () function from the *post_trade* PL/SQL block. The *post_trade* block should prompt the user to enter the ticker symbol and the post. Show the results for (AOL, P01) and (F, P12).
11. (12 pts.) Write a trigger that before entering a new trade into the trade table, updates the P/R status of the customer in the Customer table. Print Customer table before and after entering the following records.

<u>trade_id</u>	<u>ticker_Symbol</u>	<u>Cust_id</u>	<u>post</u>	<u>quantity</u>	<u>price_share</u>	<u>buy_sell</u>	<u>date_sh</u>
15	F	10112	P16	2000	17.45	B	OCT-12-2003
14	F	10114	P16	2000	17.45	B	OCT-12-2003

Remember

1. Questions from 3 – 9 should be solved using one SQL statement only.
2. Write batch of SQL statements for creating tables in *create.sql*; for inserting data in *insert.sql*; for question 3 – 9 in *query.sql*; and create files *pl-10.sql*, *pl-11.sql* for PL/SQL of questions 10 and 11.
3. Collect the output of question 1 to 11 in a spool file *spool.txt*

Deliverables

Files: *create.sql*, *insert.sql*, *query.sql*, *pl_10.sql*, *pl_11.sql*, *spool*. Zip your files into one zip folder. Your folder name should be “sqlhw_your last name”. Submit your files to the Digital Drop Box in courseinfo.

Students taking the class on main campus (Live Section) MUST ALSO submit hard copy besides the digital form.