

Aims

The aims of this assignment are to:

- formulate SQL queries;
- populate an RDBMS with a real dataset, and analyse the data;
- design test data for testing SQL queries;
- create SQL views;
- understand the limitations of SQL queries; and
- create SQL functions (when needed).

Description

In this assignment, the schema for a simple ASX (The Australian Securities Exchange) database is provided to you. ASX is Australia's primary securities exchange. Based on the provided schema: [asx-schema.sql](#), you are required to answer the following questions by formulating SQL queries. You may create SQL functions or PLpgSQL to help you, if and only if the standard SQL query language is not expressive and powerful enough to satisfy a particular question. To enable auto-marking, your queries should be formulated as SQL views, using the view and attribute names provided.

If any queries below require you to order/rank your result in a certain way, unless it is mentioned explicitly, they are supposed to be in **ascending** order. If order is specified, correctness of your solution will include the correct ordering of the output from your solution.

Furthermore, if information from the previous day is needed for your calculation in any queries below, please disregard the first trading when you count the total number of trading days otherwise you should include the first day into your calculation. For example, suppose that there are totally 5 trading days in the database. To calculate the average price of a given stock, please sum its prices and then divide by 5 days. To calculate its average gain in percentage, please sum its gains (only 4 of them) and then divide by 4 days.

1. List all the company names and countries that are incorporated outside Australia.

```
create or replace view Q1(Name, Country) as ...
```

2. List all the company codes that have more than five executive members on record (i.e., at least six).

```
create or replace view Q2(Code) as ...
```

3. List all the company names that are in the sector of "Technology"

```
create or replace view Q3(Name) as ...
```

4. Find the number of Industries in each Sector

```
create or replace view Q4(Sector, Number) as ...
```

5. Find all the executives (i.e., their names) that are affiliated with companies in the sector of "Technology". If an executive is affiliated with more than one company, he/she is counted if one of these companies is in the sector of "Technology".

```
create or replace view Q5(Name) as ...
```

6. List all the company names in the sector of "Services" that are located in Australia with the first digit of their zip code being 2.

```
create or replace view Q6(Name) as ...
```

7. Create a database view of the ASX table that contains previous Price, Price change (in amount, can be negative) and Price gain (in percentage, can be negative). (Note that the first trading day should be excluded in your result.) For example, if the PrevPrice is 1.00, Price is 0.85; then Change is -0.15 and Gain is -15.00 (in percentage but you do not need to print out the percentage sign).

```
create or replace view Q7("Date", Code, Volume, PrevPrice, Price, Change, Gain) as ...
```

8. Find the most active trading stock (the one with the maximum trading volume; if more than one, output all of them) on every trading day. Order your output by "Date" and then by Code.

```
create or replace view Q8("Date", Code, Volume) as ...
```

9. Find the number of companies per Industry. Order your result by Sector and then by Industry.

```
create or replace view Q9(Sector, Industry, Number) as ...
```

10. List all the companies (by their Code) that are the only one in their Industry (i.e., no competitors).

```
create or replace view Q10(Code, Industry) as ...
```

11. List all sectors ranked by their average ratings in descending order. AvgRating is calculated by finding the average AvgCompanyRating for each sector (where AvgCompanyRating is the average rating of a company).

```
create or replace view Q11(Sector, AvgRating) as ...
```

12. Output the person names of the executives that are affiliated with more than one company.

```
create or replace view Q12(Name) as ...
```

13. Find all the companies with a registered address in Australia, in a Sector where there are no overseas companies in the same Sector. i.e., they are in a Sector that all companies there have local Australia address.

```
create or replace view Q13(Code, Name, Address, Zip, Sector) as ...
```

14. Calculate stock gains based on their prices of the first trading day and last trading day (i.e., the oldest "Date" and the most recent "Date" of the records stored in the ASX table). Order your result by Gain in descending order and then by Code in ascending order.

```
create or replace view Q14(Code, BeginPrice, EndPrice, Change, Gain) as ...
```

15. For all the trading records in the ASX table, produce the following statistics as a database view (where Gain is measured in percentage). AvgDayGain is defined as the summation of all the daily gains (in percentage) then divided by the number of trading days (as noted above, the total number of days here should exclude the first trading day).

```
create or replace view Q15(Code, MinPrice, AvgPrice, MaxPrice, MinDayGain, AvgDayGain, MaxDayGain) as ...
```

16. Create a trigger on the Executive table, to check and disallow any insert or update of a Person in the Executive table to be an executive of more than one company.

17. Suppose more stock trading data are incoming into the ASX table. Create a trigger to increase the stock's rating (as Star's) to 5 when the stock has made a maximum daily price gain (when compared with the price on the previous trading day) in percentage within its sector. For example, for a given day and a given sector, if Stock A has the maximum price gain in the sector, its rating should then be updated to 5. If it happens to have more than one stock with the same maximum price gain, update all these stocks' ratings to 5. Otherwise, decrease the stock's rating to 1 when the stock has performed the worst in the sector in terms of daily percentage price gain. If there are more than one record of rating for a given stock that need to be updated, update (not insert) all these records.

18. Stock price and trading volume data are usually incoming data and seldom involve updating existing data. However, updates are allowed in order to correct data errors. All such updates (instead of data insertion) are logged and stored in the ASXLog table. Create a trigger to log any updates on Price and/or Volume in the ASX table and log these updates (only for update, not inserts) into the ASXLog table. Here we assume that Date and Code cannot be corrected and will be the same as their original, old values. Timestamp is the date and time that the correction takes place. Note that it is also possible that a record is corrected more than once, i.e., same Date and Code but different Timestamp.

Submission

Submission : Login to a CSE Linux machine such as wagner and use the give command below to submit your assignment (note that the give command does not work on grieg)::

```
give cs9311 a2 a2.sql
```

Deadline: Friday 15th September 2017 @ 23:59

Late Penalty: Late submissions will attract a 10% penalty for each day.

Assessment

This assignment is worth a total of **15 marks**.

Your submission (in a file called **a2.sql**) will be auto-marked to check:

- whether it is syntactically correct;
- if using SQL queries without creating an SQL function or PLpgSQL unless it is necessary; and of course,
- if each query produces correct results.

Queries are worth equal marks. Also, documentation and coding style (e.g., comments) will be considered while marking.

What To Do Now

Make sure you read the above description thoroughly, and review and/or test out the provided schema [asx-schema.sql](#) and sample data [asx-insert.sql](#). The sample data is provided to help you quickly get started. While the same schema will be used to test your submission, a different dataset (that may be larger, smaller, or totally different) may be used for auto-marking. Therefore, you may need to create your own or modify the provided data file to test your queries before submitting your assignment. Note that you DO NOT need to submit your data file as part of the submission.

If any aspect of this assignment specification requires further clarification, ask a question under topic "Assignment 2" on the course website Forums.

Reminder: before you submit, ensure that your solution (a2.sql) will load into PostgreSQL without error if used as follows on grieg:

```
% dropdb a2
% createdb a2
% psql a2 -f asx-schema.sql
% psql a2 -f asx-insert.sql
% psql a2 -f a2.sql
... will produce notices, but should have no errors ...
% psql a2
... can start testing your solution ...
```

Penalty: If we have to fix errors in your solution before it will load, you will incur 4 (out of a total of 15) marks "penalty". Also, if your view or attribute names are different from the names specified above, you will incur 2 marks "penalty".

Plagiarism

The work you submit must be your own work. Submission of work partially or completely derived from any other person or jointly written with any other person is not permitted. The penalties for such an offence may include negative marks, automatic failure of the course and possibly other academic discipline. Assignment submissions will be examined both automatically and manually for such submissions.

Relevant scholarship authorities will be informed if students holding scholarships are involved in an incident of plagiarism or other misconduct.

Do not provide or show your assignment work to any other person - apart from the teaching staff of this subject. If you knowingly provide or show your assignment work to another person for any reason, and work derived from it is submitted you may be penalized, even if the work was submitted without your knowledge or consent. This may apply even if your work is submitted by a third party unknown to you.