COMP9311 22T3

Assignment 1

Database Systems

Last updated: **Thursday 6th October 9:34am**
Most recent changes are shown in red ... older changes are shown in brown.

[Standard ER Design]

Aims

This assignment aims to give you practice in

- analysing/refining problem requirements
- designing ER data models based on requirements
- mapping ER data models to SQL schema definitions

The ultimate goal is to build an SQL schema to represent a shared web-based calendar service.

Assignment Structure

This is an individual assignment. No group work allowed.

This assignment will run in two stages: design and implementation.

Stage 1 is a design exercise. You are encouraged to think through your problems openly on the Ed Forum (i.e., online discussion) - but no posting of the solution is allowed. The teaching team will moderate the discussion, and will not give/hint solutions on the forum. Stage 1 lasts for 10 days, after which you are expected to have:

- developed a complete ER model based on the requirements
- documented the model and any assumption made for the model

Stage 2 is an implementation exercise. I will post a “standard” ER design that captures the best aspects of the data requirement analysis discussed in the forum, plus any other components that I think were missed in the discussion. You should then develop a PostgreSQL schema (i.e., SQL DDL) to accurately implement this ER model. For Stage 2, the SQL schema is the only thing that needs to be formally submitted.

Timeline

Sunday 25th Sep/Monday 26th Sep	Stage 1 Opened
Wednesday 5th October 11:59pm	Stage 1 submission of ER design
Thursday 6th October 9am	Standard ER model for Stage 2 released
Thursday 6th October	Q+A on Standard ER model in lecture (if time permits)
Friday 14 October	Stage 2 submission of SQL schema

Submission

For Stage 1, you are expected to submit a single PDF document containing your ER design and stated assumptions/comments. Your file should be named **er.pdf** and may not exceed a 5MB submission limit.

Here is a suggested format for the Stage 1 submission - A single PDF containing the ER diagrams and comments/assumptions.

You may submit your work via the "Make Submission" tab at the top of this assignment page on WebCMS, or via give's web interface.

Alternatively, when logged into CSE, and ensuring **er.pdf** is in your current working directory, you may submit using the following **give** command:

```
$ give cs9311 a1s1 er.pdf
```

You can check your latest submission via the "Check Submission" tab at the top of this assignment page on WebCMS, or on CSE servers with:

```
$ 9311 classrun check a1s1
```

You can submit multiple times, only your last submission will be marked.

For Stage 2, you are expected to submit a single SQL file containing the schema definition statements in PostgreSQL. Submission will be setup similarly in WebCMS and further instruction provided in due course.

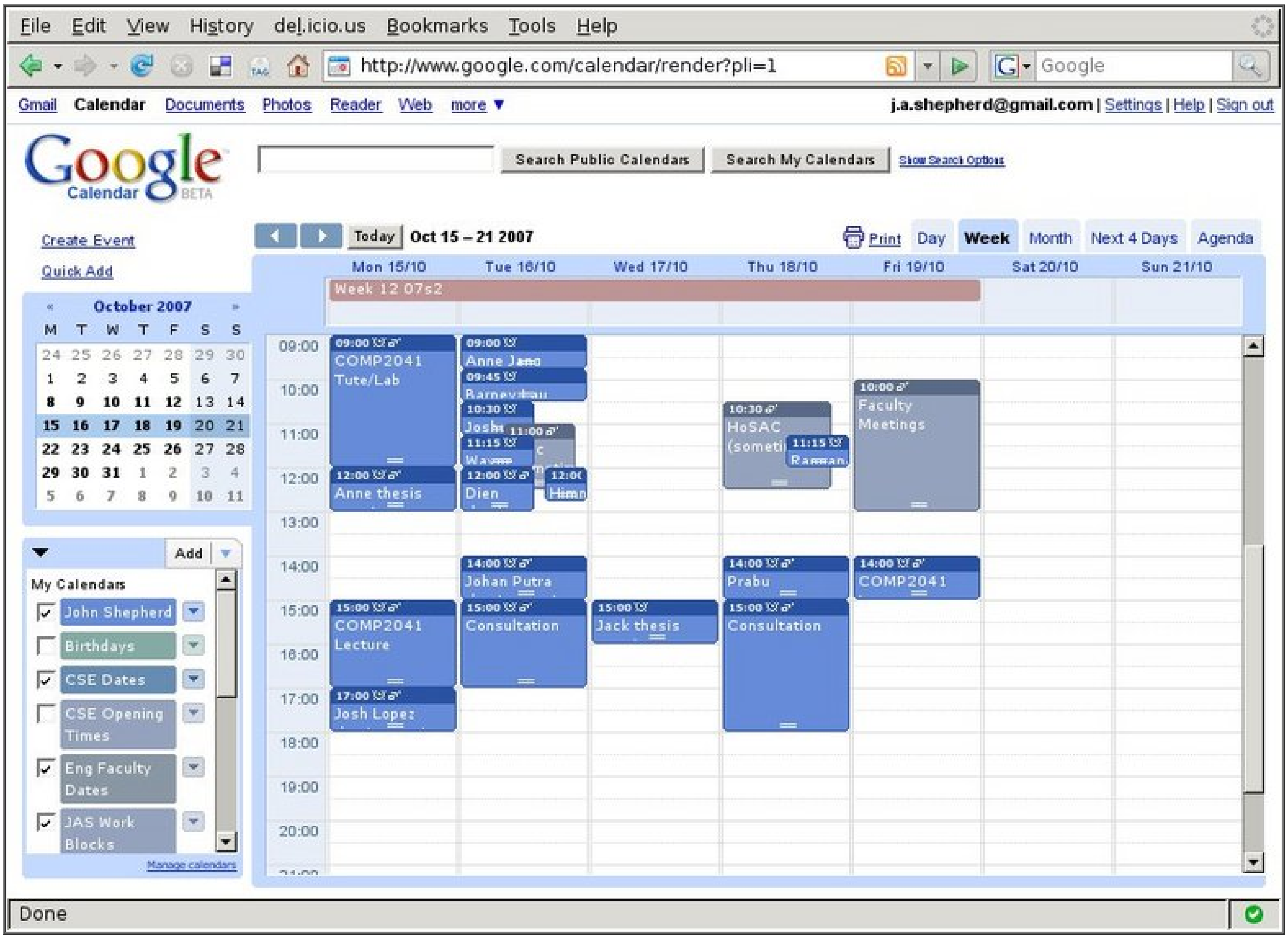
Background

CSE has decided that it needs to get its act together and organise events around the School much better. One suggestion has been to have a global shared calendar system where everyone can enter their events/availability and this can be used as a basis for:

- scheduling meetings/classes to minimise clashes
  (oh ... UNSW has a central timetabling system ... really? ... how useful is it?)
- enabling students to know when lecturers are available
- etc. etc. etc.

To maximise the usefulness/availability of the information, there naturally has to be a web interface. Nowadays this would need to be augmented with Web services to do things like include your personal calendar in your home page, SMS you when an important event is coming up, etc.

A few CSE people are already using Google's calendar system and think that this is a good starting point. Google calendar allows you to store your appointments on the Google servers and access them from any Web browser or mobile phone. Here's an example of what a Google calendar looks like:



It does a lot more than just display a calendar of events; it maintains multiple (overlaid) calendars, organises groups of people to attend specific events (invitations, RSVPs, etc.), represents recurring events, etc. etc. The functionality is similar to other calendar tools such as iCal and Outlook, but with the advantages that the data is stored at Google (i.e. doesn't take up space on your hard-drive) and you can access the data from anywhere.

Now, while CSE likes the Google interface and overall approach, they don't want to trust their precious (possibly confidential) data to Google, so they want to implement their own system. This is where you come in ...

It's your task to **design a data model** that will handle all of the things that people want to do with calendars (e.g. schedule meetings, get their Uni timetable, etc.) A goal of the system is that it will contain so much useful data about what's happening in CSE that people will use it as their primary event manager and start putting their social events in it as well, so that they have a complete picture of what's happening each week. No more "I can't go to the party on Saturday because I just remembered I have an assignment due on Sunday". When you hear about the party, you'd look at your calendar (via your phone, of course), see the assignment and realise that you needed to finish it on Friday to keep the weekend free ...

However, enough of dreaming about how wonderful the world would be if we had such a calendar ... let's build it.

A good way to get a feel for what's required is to look at any of the popular calendar/groupware/personal-information systems around these days. Most of them are based on the iCalendar data standard for interoperable calendar/scheduling data. You may get some ideas about what kind of data is needed from looking at the examples in the standard, but you're probably better off playing with some calendar program for a while to get a feel for its capabilities. One thing to remember, though, while playing with calendars ... don't get distracted by the interface; it's not relevant for this assignment, except as a way of helping you to work out what kinds of data are needed.

An important point to note on this exercise: there is no single correct answer. However, your solution should sufficiently capture the given set of requirements we specify below (i.e., your ER design should then reflect these requirements).

Here are some of the core requirements:

The CSE web-based calendar (aka CSECal) has to support the following:

- users:
    - individuals who use the calendar
    - we need to know at least their name and email address
    - they also have a username and password for authentication
- groups:
    - named collections of individuals
    - useful as shorthand for scheduling events for specific groups
- events:
    - there are various kinds of events
        - associated with a particular day/date (e.g. birthday)
        - scheduled at a given time on a given day (e.g. a meeting)
        - recurring on a regular basis (e.g. a COMP3311 lecture)
    - each event is owned by the individual user who creates it
    - each event has a title and visibility (public, private)
        - a private event is shown simply as "Busy" in the interface
    - an event may be associated with a location (where it will occur)
    - an event may be associated with a set of individual users (invitees)
    - an event may recur in a number of ways
        - on a particular day of the week (Mon,Tue,Wed,Thu,Fri,Sat,Sun)
        - weekly, every 2/3/4 weeks
        - monthly (on same date of month), every 2/3/.../11 months
        - on the first/second/third Xday of each month
        - annually
    - a recurring event will have a starting date and an ending date
    - at specified times before each event an alarm event can be triggered
    - there may be multiple alarms associated with an event
      (e.g. 15 mins before, 5 mins before, 1 minute before)
- calendars:
    - named collections of events (e.g. John's Weekly Meetings/Classes)
    - each event is attached to a specific calendar
    - each calendar has accessibility restrictions (per user and default)
      (e.g. some users have read/write, some have read-only, some have no access)
    - each calendar is owned by a user; a user may own many calendars;
    - users may subscribe to other peoples' calendars (if they can read them)

This will give you start for discussions on the data model, but needs more analysis to sort out the details.

Specific functionalities that the system must provide for users include:

- log in to the system, modify personal details
- create an event and specify its accessibility
- modify an event and specify its accessibility
- maintain a calendar and assign events to it
- view a calendar of events
- view details of individual events
- inivite other users to attend events; accept event invitations
- respond with yes/no to attendance at an event
- create a group as a shorthand for inviting people to events
- etc. etc.

You could keep adding more and more functionality, but this won't necessarily translate into a more and more complex data model. Another way of saying this is that a new functionality might not require any new data; it might simply need new methods of manipulating the existing data.

The Online Discussion in Ed Forum

You are free and, in fact, encouraged to think through the reqruiements via discussion on the forum with other class mates. The teaching staff will moderate the forum discussion.

A few things:

- Do not post a design/solution in the forum. **Do not** post chunks of relational schema in the discussions.
- The teaching staff may not answer all questions - if we think it is something you can discuss amongst yourselves.
- The teaching staff will not give a direct answer that indicates expected solution.

Assessment

This assignment is worth 17.5 marks all together. Stage 1 is worth 8.5, Stage 2 is 9.

Stage 1 will be assessed according to the following:

- On design (model drawing and documentation)

  The notations are clear and diagram itself is very easy to read ... The notations are expected to be COMP9311 lecture notes standard ... a reasonable breakdown of the diagram is expected (rather than one big diagram)

- On design (correctness of the model - syntax and semantic) Are the design elements used in the design correct (e.g., cardinality and partiipations are correctly captured per spec?) ... Are the entities and attributes chosen reasonably ... The relationships between entities are correct per the spec requirements ..., etc.

- On design (completeness)

  Are the entities and relationships in the design completely capture the requirements? ... any missing information in the design?

Stage 2 will be mainly auto-marked (and manually cross checked by tutors). You are also expected to comment the SQL code (to state any assumption or clarifications).

The submitted schema will be checked:

- whether it is syntactically correct
- how accurately it captures the standard ER design

The auto-marking will be carried out as follows:

- create an empty PostgreSQL database
- run the command `\i YOUR_SUBMISSION.sql` for your submission
  (to load a copy of schema into the empty database)
- extract a copy of your schema from the database
- compare it to a standard SQL schema

In order for this checking to work, we require that your schema meets the following criteria:

- syntactically correct (i.e. will load under PostgreSQL 13 as per setup in Lab01)
- self-contained (i.e. will load into an initially empty database)
- table and attribute names exactly match those in the standard ER design

Any schema that does not load without error into an empty PostgreSQL database will score zero marks.

Placing your name at the top of the `YOUR_SUBMISSION.sql` file is a requirement for submission. It is also a **claim that *you* are the author of the code** you are submitting. If we discover that you are not the author of any code that has your name at the top, or that you have submitted work with another person's name at the top, you will be penalised by receiving a mark of 0 for this Assignment.

Don't forget to have fun.