COMP9311 22T3 Database Systems

Assignment 1

Last updated: **Sunday 25th September 10:11pm**
Most recent changes are shown in red ... older changes are shown in brown.

[Standard ER Design] (TO BE ANNOUNCED AFTER STAGE 1)

Aims

This assignment aims to give you practice in

这个作业的目的是让你练习分析/精炼问题需求，根据需求将ER数据模型映射到SQL模式定义来设计ER数据模型

- 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.

最终目标是构建一个SQL模式来表示基于web的共享日历服务。

Assignment Structure

我们鼓励你在Ed论坛(即在线讨论)上公开思考你的问题，但不允许发布解决方案。教学团队将主持讨论，不会在论坛上给出/提示解决方案。第一阶段为期10天，之后您将有:

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:

根据需求开发一个完整的ER模型，记录模型和对模型所做的任何假设

- 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

第一阶段和第二阶段的提交将在适当的时候在WebCMS3中设置。对于阶段1，您需要提交一份包含您的ER设计和陈述的假设/评论的PDF文件。对于阶段2，你需要提交一个包含PostgreSQL中模式定义语句的SQL文件

Both Stage 1 and Stage 2 submissions will be setup in WebCMS3 in due course. For Stage 1, you are expected to submit a single PDF document containing your ER design and stated assumptions/comments. For Stage 2, you are expected to submit a single SQL file containing the schema definition statements in PostgreSQL.

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

Background

CSE已经决定，它需要采取行动，更好地在学校周围组织活动。一种建议是建立一个全球共享的日历系统，每个人都可以输入他们的事件/可用性，这可以作为以下操作的基础:

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.

安排会议/课程，尽量减少冲突

让学生知道老师什么时候有空等等

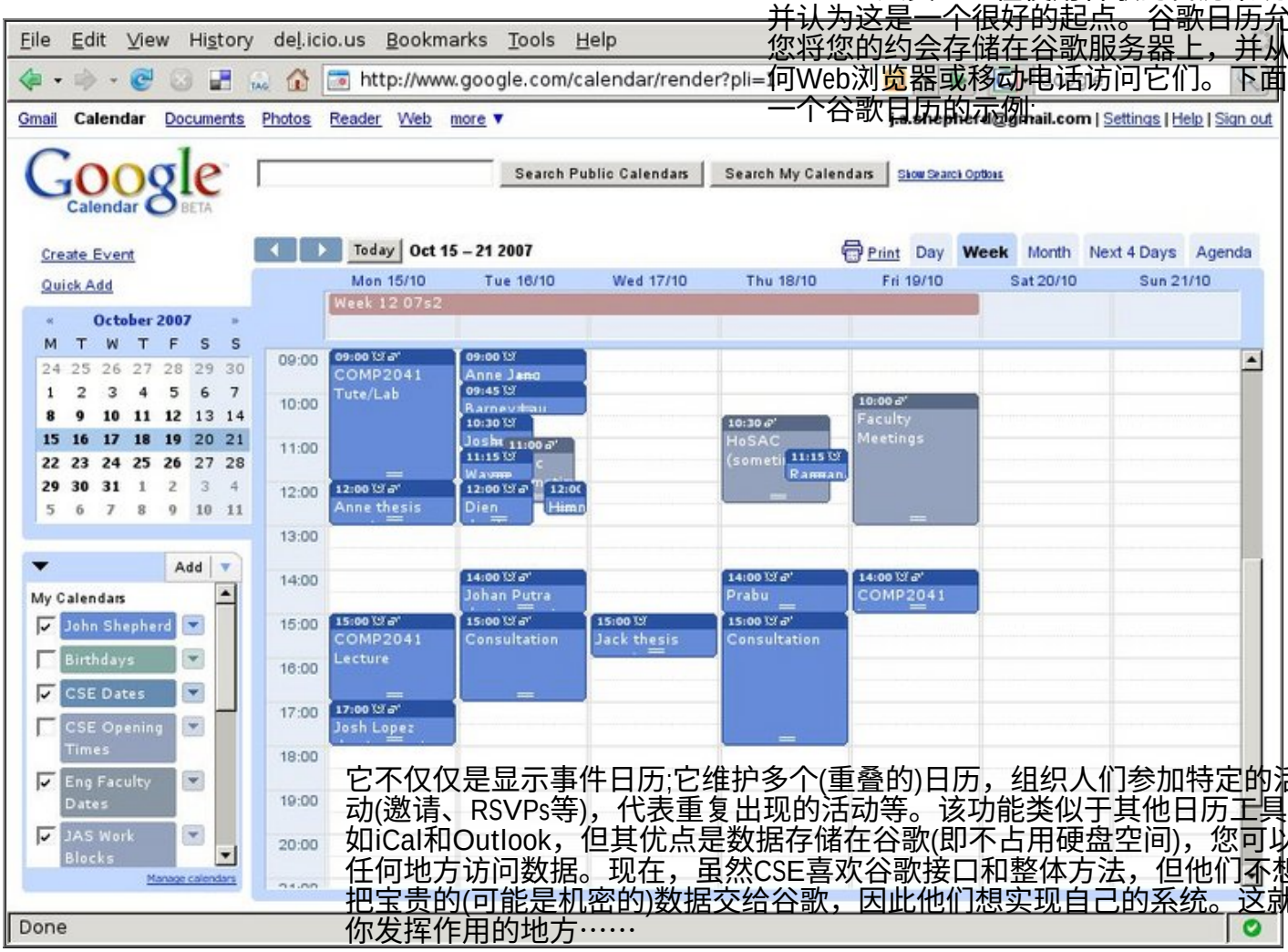
To maximise the usefulness/availability of the information, there naturally has to be a web interface. Nowadays this would need to be augmented with Web services to do things like include your personal calendar in your home page,

为了最大化信息的有用性/可用性，自然必须有一个web界面。如今，这将需要通过Web服务来增强，以完成诸如在主页中包含个人日历,有重要事件发生时给你发短信

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:

一些CSE人员已经在使用谷歌的日历系统，并认为这是一个很好的起点。谷歌日历允许您将您的约会存储在谷歌服务器上，并从任何Web浏览器或移动电话访问它们。下面是一个谷歌日历的示例:



它不仅仅是显示事件日历;它维护多个(重叠的)日历，组织人们参加特定的活动(邀请、RSVPs等)，代表重复出现的活动等。该功能类似于其他日历工具，如iCal和Outlook，但其优点是数据存储在谷歌(即不占用硬盘空间)，您可以从任何地方访问数据。现在，虽然CSE喜欢谷歌接口和整体方法，但他们不想把宝贵的(可能是机密的)数据交给谷歌，因此他们想实现自己的系统。这就是你发挥作用的地方……

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:

你的任务是设计一个数据模型,将处理所有的人想做的日历(例如安排会议、Uni时间表,等等)系统的一个目标是,它将包含很多有用的数据对CSE发生了什么,人们会把它作为他们的主要事件管理器,开始把他们的社交活动中,使他们有一个完整每周发生的事情。再也不会说“我周六不能去派对了，因为我刚想起我周日有作业要交”。当你听到派对的消息时，你会查看你的日历(当然是通过手机)，看到任务，并意识到你需要在周五完成它，以

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.

了解需求的一个好方法是看看当今流行的日历/群件/个人信息系统。它们大多是基于iCalendar数据标准实现可互操作的日历/调度数据。通过查看标准中的示例，您可能会对需要什么样的数据有一些想法，但您可能最好先玩一些日历程序，以了解它的功能。不过，在玩日历时要记住一件事……不要被界面分心;它与这个作业无关，只是用来帮助你弄清楚需要什么样的数据。这个练习需要注意的一点是:没有唯一的正确答案。然而，您的解决方案应该充分捕捉我们在下面指定的给定需求集(即，您的ER设计应该反映这些需求)。

使用日历的个人

我们至少要知道他们的姓名和电子邮件地址

它们也有用户名和密码进行身份验证

命名的个体集合

用作为特定组安排事件的简写

地点和任务以及拥有者属于关系

重复的方式有很多种，种类，这个应该作为值的限制

触发关系，多对一，属性time

从属关系

依赖关系

有条件的依赖

创建一个小组作为邀请人们参加活动的简写

你可以继续添加越来越多的功能，但这并不一定会转化为越来越复杂的数据模型。换句话说，新功能可能不需要任何新数据;它可能只需要操作现有数据的新方法。

Stage 1 will be assessed according to the following:

符号清晰，图表本身也很容易阅读。这些符号应该符合COMP9311讲义标准…对图表进行合理的分解(而不是一个大的图表)

- 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.

在设计上(模型的正确性——语法和语义)，设计中使用的设计元素是否正确(例如，每个规范都正确捕获了基数和划分?)实体和属性的选择是否合理?根据规范要求，实体之间的关系是正确的……等。

- On design (completeness)

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

设计中的实体和关系是否完全捕获了需求?…设计中有什么信息缺失吗?

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

任何不能毫无错误地加载到空PostgreSQL数据库中的模式都将得到0分

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.