University Of Portsmouth
BSc (Hons) Computer Science
First Year

Database Systems Development

M30232
September 2022 - May 2023
20 Credits

Thomas Boxall
up2108121@myport.ac.uk

Contents

S.1. Introduction to module

29-09-22 13:00 Mark RB LT1

The Module Coordinator for this module is Mark (based in BK 3.09), he's assisted by Valentin and Roy in some sessions alongside others too.
Mark is using a new piece of software to make his presentations with, this is currently in the test phases and he may change back to PowerPoint if people don't like it. Slides are available on Moodle as HTML format, they can be printed to PDF files for offline viewing.

Module Aims

This module aims to help you understand where the database sits in modern systems. It does not train us to be database administrators. It gives us the skills to design a database and the knowledge of how to access it and do so safely.
This module will start from the ground up.

Learning Outcomes

- Demonstrate the fundamental principles of database design & development
- Use appropriate analysis techniques to identify the requirements of a database.
- Design and build a relational database, given a set of requirements.
- Understand how to apply data manipulation using SQL.

Historically, this module used to focus on the elements of Computer Science which relate to databases (for example, software development lifecycles). Now, it focuses on just databases.

Content Overview

This module provides an understanding of the theory of relational database design using tools standard to the industry. We will be taught how to design databases using Crows Foot Entity Relationship Diagrams and SQL to create the database. This module will also cover normalisation.

Teaching Overview

The module is a year long, worth 20 credits and has two different styles of teaching.
There will be one, one hour lecture per week. In this session, we will be taught the knowledge which we can put into practice in the following weeks practical session.
There will be one, one and a half hour practical session per week. In this session, we will practice the skills required for databases. *(N.B. This session is timetabled for two hours on the timetable, generally the lecturers will leave after an hour and a half however students can remain in the room until the end of the two hours.)*
If you are unable to make it to a lecture, you need to read the content provided on Moodle. If you are unable to make it to a practical, you need to read and do (most importantly, do) the content on Moodle; this is so you are able to complete the following practical as they all build on each other.

Resources

There are a number of resources talked through:

- Moodle - the universities Virtual Learning Environment. Notes from lectures and from practicals will be uploaded here along with quizzes and other resources.
- Google Virtual Machine - the virtual machine in which our database lives. You do not need the university VPN to access it, as it requires a SSH connection. The data is hosted by Google, the module staff have some control over the machines. More detail on this will be provided in the first practical session.
- Google Workspace
- Microsoft Office. This is available free from the university. At some point, this will include Microsoft Visio, which is useful for coursework.

Expectations

Lecturers Expectations of Students

- Turn up for lectures (from next week, the content taught in the lectures will be used in the following weeks practical sessions)
- Arrive on time (there is usually useful information given out at the beginning of sessions)
- Participate and take notes in sessions
- Catch up on sessions if you miss them
- Finish the practical work before the following weeks practical sessions
- Study for about 4 hours a week total

These things are proven to increase the likelihood that a student gets a better mark at the end of the year.

Students Expectations of Lecturers

They are nice to students; start and end sessions on time; provide students with support and feedback on work throughout the module; and to return feedback and marks on work as quickly as they can (this usually should be within two weeks).

Assessments

There are two forms of assessment in this module.

Coursework

This will be worth 50% of the overall module mark. It will be released in the next few weeks and will be due at the end of the first week after the January assessment period (probably the Friday of that week at 11pm). The content assessed will all be from the first teaching block. We will get extra marks if we include content which hasn't been taught yet.

Exam

This will be worth 50% of the overall module mark. It will take place in the May/June assessment period and be computer based. It can include anything from the entire year however we won't have to write code (probably will have to look at code and say whats wrong). It will be multiple choice questions. There will be quizzes available on Moodle which will be similar to this where we can practice.

Brief Introduction to Databases

Database

"A single, possiblely large, repository of data that can be used simultaneuously by many departments and users" *(Database Solutions: A Step by Step Guide to Databases - T Connolly & C Begg)*

Spreadsheets

Spreadsheets are not databases. This is because a spreadsheet cannot hold the amount of data which a database can and eventhough though using some software, a database could be shared with multiple people, it cannot be edited by multiple people simultaneously.
This also applies to Microsoft Access.

Database Management System (DBMS)

DBMS

"The software which interacts with the users' application programs and the database" *(Database Solutions: A Step by Step Guide to Databases - T Connolly & C Begg)*

Examples of a DBMS include PostgreSQL, MySQL, SQL Server, Oracle and Mongo DB.

Why Use a Database

An alternative to databases are file based systems.
File based systems: are old fashioned; are not necessarily digital; they often contain duplicate data; are difficult to search; are very difficult to update; have the possibility to contain different file types which may not be compatible together; are inaccessible; and security may be an issue.
A database is: a modern approach; digital; duplicates can be removed; easy to search; easy to update; comprised of only one file type; capable of having multiple levels of access control; able to limit user access.
There are times at which a Database is not suitable for the setting. In this case, it may be more suitable to use a spreadsheet.

Integrated Database Environment

In an integrated database environment, the DBMS sites as a communication hub between all nodes. The DBMS is the server on which the database is hosted.
When the database is setup correctly, you can get more information out of it than you put in.

S.2. Practical 1

29-09-22 14:00 Mark & team FTC 3rd floor

Introduction to Practical sessions

Practical documents are available on Moodle, make a copy of these and store within your university Google Drive so you can edit them during the sessions and make notes.

Access Levels

In PostgreSQL, the first level of security is that a user cannot login unless they have been given access or there is a database with the same name as their username.
We don't have `sudo` access to linux, however we have full administrative access to PostgreSQL. Don't drop the database called `upxxxxxxx` (where `xxxxxxx` is replaced with student number) or anything that is owned by `postgres` as this breaks things.

PostgreSQL

PostgreSQL is ready to accept code when the prompt ends in `=#`. If you enter part of a command and press enter, the prompt will change to `-#`, this indicates that Postgres is waiting for you to finish the command.
PostgreSQL gives some useful error messages, SQL does not.

Code Editors

A code editor should be used to write SQL into, then the SQL should be copied and pasted into the Linux machine. The only thing that should be directly entered into the shell is to connect to a different database.
This is so that a. we have a copy of what we have done and b. so that if the VM is deleted, we are able to re-build our VM with less pain than if we didn't save all the code.
A recommended setup is to use VS code, with a SQL syntax extension. VS Code comes with integrated Powershell, allowing you to ssh to the VM from the same window.

SQL

SQL works like a procedural programming language, in that it reads the code inputted line by line. This also means that long and complex lines of code can be split across many lines, making it easier to read them.

Installing The First Database

Due to an issue with the image used to build the Virtual Machines, we have to create the database which we will use for the first few sessions. The code to do this was available on Moodle, copy and paste into the code editor then copy and paste again, this time into the Postgres prompt of the linux machine. This executes and creates the database, pre-populated with some sample data.

Tasks

1. List the databases in your server

```
LANGUAGE: SQL
```

```
\l
```

LANGUAGE: Unknown

```
                                  List of databases
    Name      |     Owner      | Encoding | Collate |  Ctype  |   Access privileges
--------------+----------------+----------+---------+---------+-----------------------
dsd_22        | up2108121      | UTF8     | C.UTF-8 | C.UTF-8 |
mongo-2021-fix | mongo-2021-fix | UTF8     | C.UTF-8 | C.UTF-8 |
postgres      | postgres       | UTF8     | C.UTF-8 | C.UTF-8 |
template0     | postgres       | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres          +
              |                |          |         |         | postgres=CTc/postgres
template1     | postgres       | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres          +
              |                |          |         |         | postgres=CTc/postgres
up2108121     | up2108121      | UTF8     | C.UTF-8 | C.UTF-8 |
(6 rows)
```

2. Connect to the database

LANGUAGE: SQL

```
\c dsd_22
```

LANGUAGE: Unknown

```
You are now connected to database "dsd_22" as user "up2108121".
```

3. List everything in this database

LANGUAGE: SQL

```
\d
```

LANGUAGE: Unknown

```
                    List of relations
Schema |            Name             |   Type   |   Owner
--------+-----------------------------+----------+-----------
public | category                    | table    | up2108121
public | category_cat_id_seq         | sequence | up2108121
public | cust_order                  | table    | up2108121
public | cust_order_cust_ord_id_seq  | sequence | up2108121
public | customer                    | table    | up2108121
public | customer_cust_id_seq        | sequence | up2108121
public | manifest                    | table    | up2108121
public | manifest_manifest_id_seq    | sequence | up2108121
public | product                     | table    | up2108121
public | product_prod_id_seq         | sequence | up2108121
public | role                        | table    | up2108121
public | role_role_id_seq            | sequence | up2108121
public | staff                       | table    | up2108121
public | staff_staff_id_seq          | sequence | up2108121
(14 rows)
```

4. List just the tables

LANGUAGE: SQL

```
\dt
```

```
LANGUAGE: Unknown
```

```
            List of relations
Schema |     Name     | Type  |   Owner
--------+--------------+-------+-----------
public | category     | table | up2108121
public | cust_order   | table | up2108121
public | customer     | table | up2108121
public | manifest     | table | up2108121
public | product      | table | up2108121
public | role         | table | up2108121
public | staff        | table | up2108121
(7 rows)
```

TO BE CONTINUED.