CS27020: Modelling Persistent Data

Introduction: Module outline & Persistent Data

Bernie Tiddeman, Helen Miles,
& Chuan Lu

Computer Science,
Aberystwyth University

Outline

1. Admin and general information
2. Overview of the module
3. Persistent Data
4. Modelling
5. Object Relational Mapping
6. Semi-structure data & NoSQL

Teaching team

Module coordinator	
Bernie Tiddeman	bpt@aber.ac.uk
Lecturers	
Helen Miles	hem23@aber.ac.uk
Chuan Lu	cul@aber.ac.uk
Demonstrators	TBA

We will be teaching as a team. Each section, one of us will be lecturing and handling most of the questions but you can expect us all to pop up at any time.

One of us will be at each of the practical sessions, (hopefully) plus demonstrators.

Changes this year

- New teaching team! (except me)
- Back to in-person lectures (for now!)
- Removed weekly seminar that's no longer needed
- Back to in-person exam
  - But limited open book format

Note - Information Security material was removed two years ago

Module Assessment

Assessment	Ingredients	Timing
10% Quizzes	Online (BB) mini-quizzes	6 in total. Best 5 marks count
20% Practicals	5 practical worksheets worth 4% each	Weekly 2 hour sessions, starting week 2
70% Exam	A two hour exam	End semester 1

More on the assessment

Quizzes (10%): These are online mini-quizzes. You get more than one attempt at each quiz, and can do it in your own time. Each quiz is worth 2 marks. The schedule is on Blackboard.

Practicals (20%): Completed in the lab. Assessed by demonstrator sign-off. You will each be allocated to a fortnightly practical session starting in either week 2 or 3. Worth 4% each.

Exam (70%): Similar to previous years except for (a) plan to be back to in-person exam (may allow some of your own notes) and (b) no questions on information security (topic moved to a first year module).

Module Time Allocation

A 20 credit module is about 200 hours of work. This module's allocation looks a bit like this:

Spend your time on...	**How Long**
Lectures	30
Reviewing lectures	30
Independent study	**70**
Preparing for quizzes	6
Quizzes	2
Practicals	20
Preparing for practicals	10
Revision	30
The exam	2

How to find information

Blackboard - slides, announcements, practical worksheets, quizzes, discussion groups, etc.

Discord (`https://discord.gg/eb8NaQK9Rp`) alternative forum for online help both general help and discussion.

Course textbook: "Database systems: a practical approach to design, implementation, and management", Connolly, Thomas M., Beg, Carolyn E. (2015) (*available online*)

Other course textbook: "The Manga Guide to Databases", Mana Takahashi (2009)

Module study guide

I have prepared a module study guide to help you keep tab of where you are in the module.

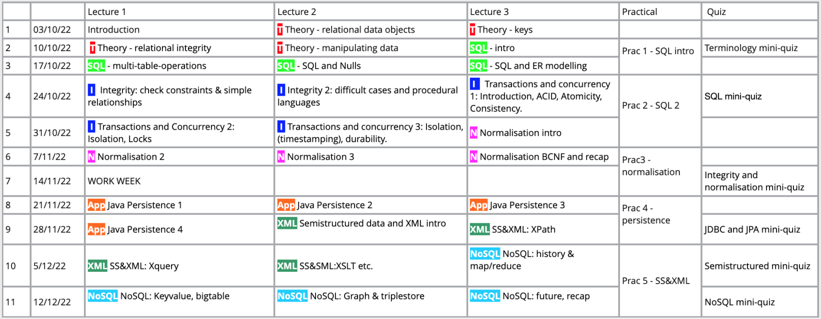
- Optional - use it if it helps you
- Get it off BB
- Tick off each item as you complete it or as you revise it

I will let you know if anything significant changes and post an update.

Outline

1. Admin and general information
2. Overview of the module
3. Persistent Data
4. Modelling
5. Object Relational Mapping
6. Semi-structure data & NoSQL

Module at a glance



		Lecture 1	Lecture 2	Lecture 3	Practical	Quiz
1	03/10/22	Introduction	T Theory - relational data objects	T Theory - keys		
2	10/10/22	T Theory - relational integrity	T Theory - manipulating data	SQL - intro	Prac 1 - SQL intro	Terminology mini-quiz
3	17/10/22	SQL - multi-table-operations	SQL - SQL and Nulls	SQL - SQL and ER modelling		
4	24/10/22	I Integrity: check constraints & simple relationships	I Integrity 2: difficult cases and procedural languages	I Transactions and concurrency 1: Introduction, ACID, Atomicity, Consistency.	Prac 2 - SQL 2	SQL mini-quiz
5	31/10/22	I Transactions and Concurrency 2: Isolation, Locks	I Transactions and concurrency 3: Isolation, (timestamping), durability.	N Normalisation intro		
6	7/11/22	N Normalisation 2	N Normalisation 3	N Normalisation BCNF and recap	Prac3 - normalisation	
7	14/11/22	WORK WEEK				Integrity and normalisation mini-quiz
8	21/11/22	App Java Persistence 1	App Java Persistence 2	App Java Persistence 3	Prac 4 - persistence	
9	28/11/22	App Java Persistence 4	XML Semistructured data and XML intro	XML SS&XML: XPath		JDBC and JPA mini-quiz
10	5/12/22	XML SS&XML: Xquery	XML SS&SML:XSLT etc.	NoSQL NoSQL: history & map/reduce	Prac 5 - SS&XML	Semistructured mini-quiz
11	12/12/22	NoSQL NoSQL: Keyvalue, bigtable	NoSQL NoSQL: Graph & triplestore	NoSQL NoSQL: future, recap		NoSQL mini-quiz

The module at a glance can also be found on Blackboard: https://blackboard.aber.ac.uk/webapps/blackboard/content/listContentEditable.jsp?content_id=_2186671_1&course_id=_33288_1&mode=reset

Module outline summary

Now you know

- What to expect from the module
- Where to find stuff
- How you're going to be assessed
- How we think you should spend your time

Next, introduction to persistent data ...

Outline

1. Admin and general information
2. Overview of the module
3. Persistent Data
4. Modelling
5. Object Relational Mapping
6. Semi-structure data & NoSQL

What is Persistent Data?

Active learning
Think - Pair - Share

What do you think persistent data is?

A minute to think,
a minute to talk with your neighbour,
a minute to share with the class
Vevox poll 195-823-886.

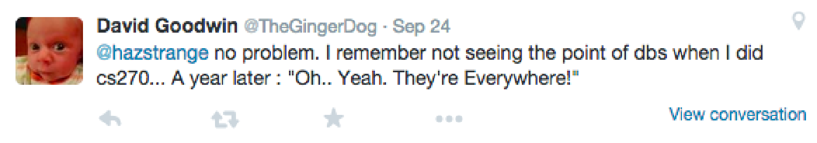
Some Key Aspects of Persistent Data

Persistent data crops up in a lot of different ways in modern life; so **what** is it? There are some common aspects:

- saved for later use
- remains (persists) after it has been used
- has an identity that remains even if its value is changed
- is data that somebody wants to keep
- often stored in a database . . .
- . . . but could be stored in a file . . .
- . . . or on paper . . .
- . . . or as something more exotic (qbits, genes,...)

Some Key Aspects of Persistent Data

Another key aspect of persistent data . . . it's everywhere



David Goodwin @TheGingerDog · Sep 24

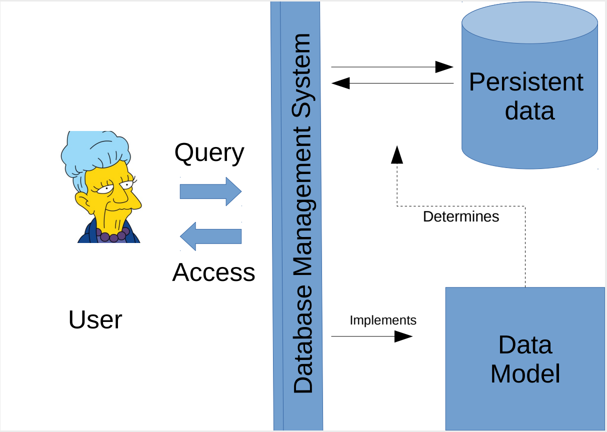
@hazstrange no problem. I remember not seeing the point of dbs when I did cs270... A year later : "Oh.. Yeah. They're Everywhere!"

Outline

1. Admin and general information
2. Overview of the module
3. Persistent Data
4. Modelling
5. Object Relational Mapping
6. Semi-structure data & NoSQL

Different Flavours of Data Model

We will often talk about persistent data, data models, and database management systems; how do they fit together?



Different Flavours of Data Model

There are lots of different ways of modelling persistent data. In CS27020 we are going to look at lots - some in passing, and some in depth:

- Relational Data Model
- Object Model
- Object-Relational Model
- Semi-structured data and XML
- NoSQL - various models

Different Flavours of Data Model

There are lots of different ways of modelling persistent data. In CS27020 we are going to look at lots - some in passing, and some in depth:

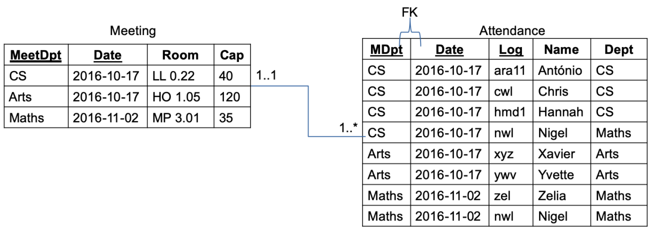
- Relational Data Model
- Object Model
- Object-Relational Model
- Semi-structured data and XML
- NoSQL - various models

OK, but are these actually used in the real world?

Relational Data Model

The relational data model is probably the most widely used data model[1]. Most enterprises will use a database management system that implements the relational data model[2].

Informally, relations are linked tables of data.



Meeting

MeetDpt	Date	Room	Cap
CS	2016-10-17	LL 0.22	40
Arts	2016-10-17	HO 1.05	120
Maths	2016-11-02	MP 3.01	35

Attendance

MDpt	Date	Log	Name	Dept
CS	2016-10-17	ara11	António	CS
CS	2016-10-17	cwl	Chris	CS
CS	2016-10-17	hmd1	Hannah	CS
CS	2016-10-17	nwl	Nigel	Maths
Arts	2016-10-17	xyz	Xavier	Arts
Arts	2016-10-17	ywv	Yvette	Arts
Maths	2016-11-02	zel	Zelia	Maths
Maths	2016-11-02	nwl	Nigel	Maths

Facebook are one of the most high-profile users of the relational data model through their use of MySQL.

[1] Don't quote me on that

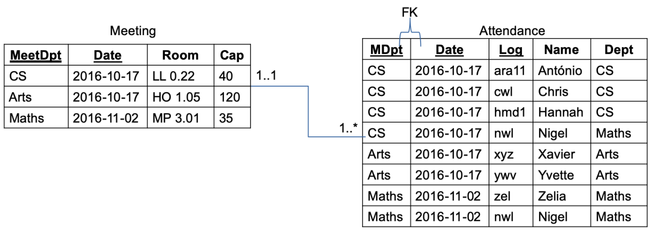
[2] See footnote above

Relational Data Model Design

Deciding which tables (headings & keys) and how they're linked is important for:

- Efficiency (storage & access)
- Reducing risk of errors / conflicts
- Isolating errors

This is covered in the database design and normalisation lectures



Meeting

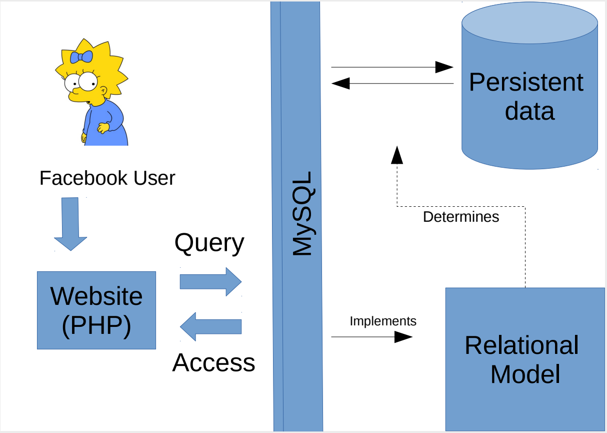
MeetDpt	**Date**	**Room**	**Cap**
CS	2016-10-17	LL 0.22	40
Arts	2016-10-17	HO 1.05	120
Maths	2016-11-02	MP 3.01	35

Attendance

MDpt	**Date**	**Log**	**Name**	**Dept**
CS	2016-10-17	ara11	António	CS
CS	2016-10-17	cwl	Chris	CS
CS	2016-10-17	hmd1	Hannah	CS
CS	2016-10-17	nwl	Nigel	Maths
Arts	2016-10-17	xyz	Xavier	Arts
Arts	2016-10-17	ywv	Yvette	Arts
Maths	2016-11-02	zel	Zelia	Maths
Maths	2016-11-02	nwl	Nigel	Maths

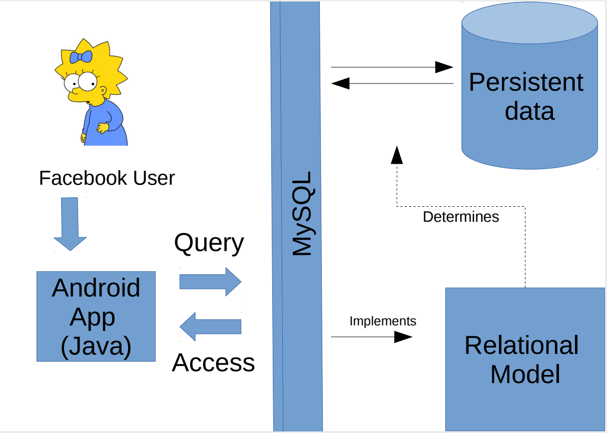
Persistent data is key

A key thing about modelling data and doing persistence well is that it doesn't matter what the front end is.



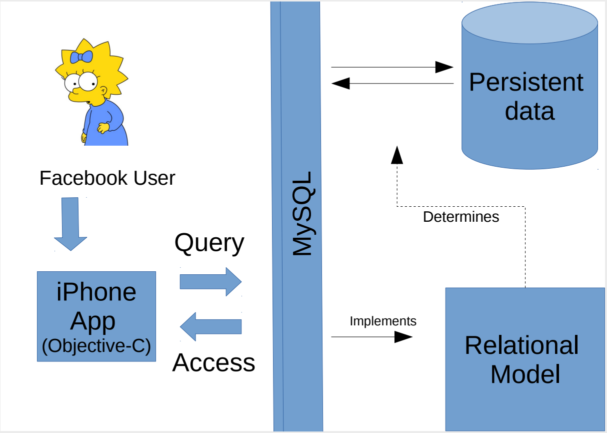
Same model, different interface

A key thing about modelling data and doing persistence well is that it doesn't matter what the front end is.



Even works on iPhone...

A key thing about modelling data and doing persistence well is that it doesn't matter what the front end is.



Outline

1. Admin and general information
2. Overview of the module
3. Persistent Data
4. Modelling
5. **Object Relational Mapping**
6. Semi-structure data & NoSQL

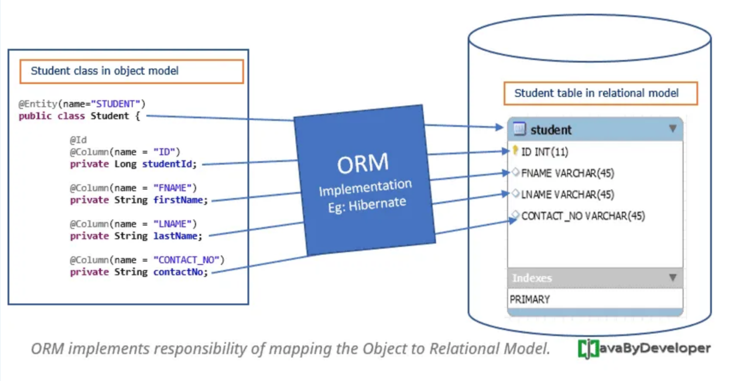
Object Model

The object model represents information in the form of objects (as in OOP) as opposed to “tables” (as seen in the relational data model).

There are various commercial and open-source object-oriented DBMSs e.g. InterSystems Caché .

So, how can we access a persistent data store from code? Could just construct SQL commands in text and send them to the DB, but the are better ways...

Object-Relational Mapping: Example



```
@Entity(name="STUDENT")
public class Student {

        @Id
        @Column(name = "ID")
        private Long studentId;

        @Column(name = "FNAME")
        private String firstName;

        @Column(name = "LNAME")
        private String lastName;

        @Column(name = "CONTACT_NO")
        private String contactNo;
```

```
student
ID INT(11)
FNAME VARCHAR(45)
LNAME VARCHAR(45)
CONTACT_NO VARCHAR(45)
Indexes
PRIMARY
```

Student class in object model
ORM
Implementation
Eg: Hibernate
Student table in relational model
JavaByDeveloper

ORM implements responsibility of mapping the Object to Relational Model.

Image taken from `https://javabydeveloper.com/orm-object-relational-mapping/`

Java Persistence

Programming languages offer ways of persisting data. We'll cover Java Persistence, but the general principles apply to other languages.

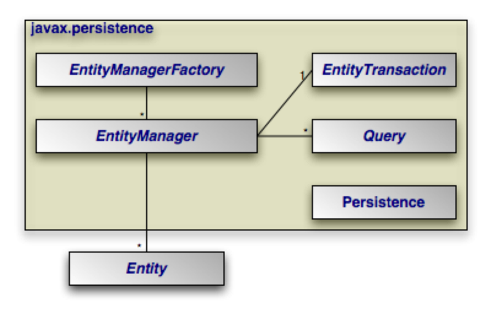


Image from https://openjpa.apache.org/

This lets you isolate the application logic from the database engine or technology. The use of persistence in the programming language takes the level of abstraction nearer to code than to databases.

Outline

1. Admin and general information
2. Overview of the module
3. Persistent Data
4. Modelling
5. Object Relational Mapping
6. **Semi-structure data & NoSQL**

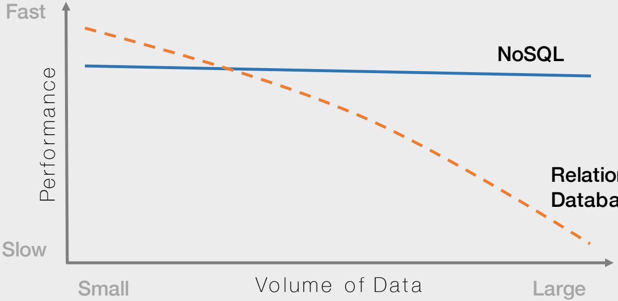
Semi-structured Data and XML

Structured data conforms to a **schema** (i.e. the Relational Data Model). Unstructured data has no schema and is difficult to query and process. However, very little data is truly unstructured. Given the underlying XML representation, even a Word document can be said to be **semi-structured**.

The use of semi-structured data is a rapidly growing area and has been for a long time. You can view all the web as semi structured data, so . . . let us say that search engines are a good example of this kind of application.

NoSQL

NoSQL is a completely different framework for databases that enables high-performance, agile processing **at scale**. It is said to "scale horizontally", adding more data? Just add more hardware!



Sketch modified from DataJobs.com

NoSQL

NoSQL is a completely different framework for databases that enables high-performance, agile processing **at scale**. It is said to “scale horizontally”, adding more data? Just add more hardware!

Spotify is a prominent user of NoSQL first through the Apache Cassandra DBMS, and more recently also through Google Bigtable

So that's the introductory lecture

Now you know

- Module structure and assessment
- What persistent data is
- Why modelling data is important
- Overview of key topics:
    - the relational model
    - SQL
    - Java persistence
    - semi-structured data
    - NoSQL