Databases - Introduction

Thomas Heinis

t.heinis@imperial.ac.uk
Scale Lab - scale.doc.ic.ac.uk

SCALE LAB

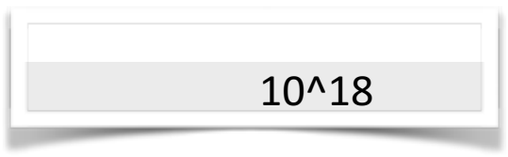
Imperial College
London

Data, data, everywhere

Barclays Bank	
Guy's Hospital	
Imperial College	
British Airways	
Met Office	
NASA	
Google	
Amazon	

Data: Types, Sizes, Lifetime

Organised collection of data. Data types?

Bytes ➔ Terabytes  ➔ Petabytes  ➔ 

Seconds ➔ Days ➔ Months ➔ Years ➔ Forever?

Data: Types, Sizes, Lifetime

Organised collection of data. Data types?

Numbers | Strings | Currency | Dates | Times | Places

Images | Video | Audio | Arrays | Graphs | Binary

Bytes ➔ Terabytes 10^12 ➔ Petabytes 10^15 ➔ Exabyte 10^18

Seconds ➔ Days ➔ Months ➔ Years ➔ Forever?

Databases: Why?

- **Organised** - Easier to model and manage. Databases are typically managed by a single Database Management System (DBMS)
- **Efficient** - Fast to search and update. Memory/Disk usage.
- **Integration** - Minimise duplication of data in an organisation.
- **Concurrency** - Support concurrent actions on the database.
- **Multi-user -** Support two more or users accessing the data at the same time.
- **Access Control -** Who/what can access which data.
- **Recovery** - Support recovery from failures.
- **Transactional**.

Space/Time tradeoffs

What about Privacy?

What kind of failures?

Other Properties?

Some Data Management History…

The Database

What about…

Reliability?
Security?
Consistency?
Response time?
Scalability?

Transactions – Data Integrity

Why Concurrent Access to Data must be managed?

John and Jane withdraw $50 and $100 from a shared account…

John:
1. get balance
2. if balance > $50
3. balance = balance - $50
4. update balance

Jane:
1. get balance
2. if balance > $100
3. balance = balance - $100
4. update balance

Initial balance $300. Final balance = ?

It depends…

Need to order operations ➔ transactions!

Transactions and ACID Properties

Transactions are sequences of database actions that are executed in a coherent and reliable way, e.g. transferring money from one bank account to another. The classical properties of transactions are known as the ACID properties:

- **Atomicity** - All or nothing execution. If one part of a transaction fails, the whole transaction fails.
- **Consistency** - Transactions do not leave the database in an *inconsistent* state. The DBMS must provide mechanisms to ensure that constraints on the database are satisfied.
- **Isolation** - Each transaction is executed as if no other transaction is executing. In some cases, a transaction may need to wait for another to complete.
- **Durability** - Results of a successful transaction are not lost. The DBMS must provide mechanisms to recover from system failures (hardware or software).

Example

Transfer of funds between accounts A and B, e.g.:

T1: A=A-100; B=B+100 | T2: B=B-100; A=A+100

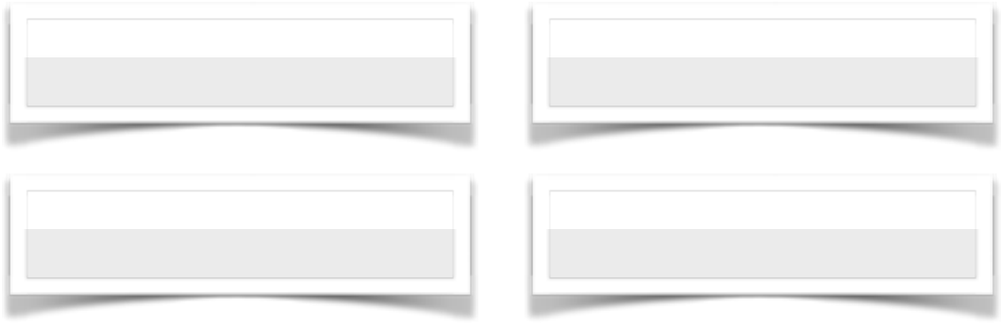
- **Atomicity**. On completion of T1 either (i) A'=A-100 & B'=B+100, or (ii) in the case of a failure e.g. hardware, network, application, A and B have their original values i.e. A'=A & B'=B
- **Consistency**. Sum of the balances remains the same, i.e. A'+B' = A + B. We might have a constraint like A'>=0 i.e. cannot have a negative balance, or | A' - A | < 1000 i.e. Account cannot change by more than 1000 in a single transaction.
- **Isolation**. Given two concurrent transactions T1 (Transfer 100 from account A to account B), and T2 (Transfer 100 from account B to account A). One of the transactions must wait until the other completes.

  No guarantee if not transaction: A=A-100; B=B+100 | A=A+100; B=B-100
- **Durability**. New values of A and B must persist after a successful completion of a transaction even if the system subsequently fails, i.e. ,disk fails before changes written.

Database Management Systems (DBMS)

- Create new databases using a **DDL** (Data Definition Language) used for defining the logical structure of a database (**schema**)
- Query (search, sort, ...) and manipulate (insert, delete, update, ...) data using a **DML** (Data Manipulation Language)

- Commercial



- Free/Opensource

- NoSQL

Not Only SQL databases
http://nosql-database.org/

Database Management Systems (DBMS)

- Create new databases using a **DDL** (Data Definition Language) used for defining the logical structure of a database (**schema**)
- Query (search, sort, ...) and manipulate (insert, delete, update, ...) data using a **DML** (Data Manipulation Language)

- Commercial
  - Oracle
  - SQL Server
  - DB2
  - Sybase
- Free
  - PostgreSQL
  - MySQL
  - SQLite
  - Derby
- NoSQL

Not Only SQL databases
http://nosql-database.org/

SQL Databases

- Most widespread database technology.
- Lets us define, query, manipulate databases using a high-level declarative language called **SQL** (Structured Query Language).
- SQL is standardised by ISO (the International Standards Organisation). SQL standards include SQL-92, SQL-1999, SQL-2003, SQL-2006, SQL-2008. **Warning**: Each SQL DBMS implements its own variations of these standards. Moving a complex database from one SQL DBMS to another is costly and time-consuming.
- SQL is pronounced "**es-queue-el**", except by Americans who prefer "sequel" !
- SQL is inspired by **E. F. Codd**'s ***relational model*** (1970) but deviates from it in many details. Purists don't consider SQL databases as relational. Codd was awarded the ACM Turing Award (the most prestigious award in Computing) in 1981 in recognition of his work on the relational model.

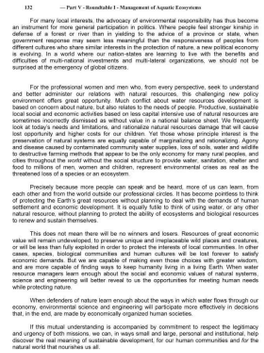
Goals of this Course

The goal of the course is that students can:

- define a schema for a use case (ER-model & relational model)
- normalize the schema (relational algebra & normalization)
- implement the schema (SQL - DDL)
- write SQL queries to insert and query the database (SQL - DML)
- accelerate queries on a database (indexes)
- transactions & scaling databases
- big data topics

Outline

- Brief overview of the relational model which is at the core of this course
- Model an abstract use case
- Turn into E-R model
- Convert to relational model
- Implement on database system
- Write queries: e.g., select * from table
- Accelerate queries & database
- Big data aspects


132 — Part V - Roundtable I - Management of Aquatic Ecosystems

For many local interests, the advocacy of environmental responsibility has thus become an instrument for more general participation in politics. Where people feel stronger kinship in defense of a forest or river than in yielding to the advice of a province or state, when government response may seem less meaningful than the responsiveness of peoples from different cultures who share similar interests in the protection of nature, a new political economy is evolving. In a world where our nation-states are learning to live with the benefits and difficulties of multi-national investments and multi-lateral organizations, we should not be surprised at the emergency of global citizens.

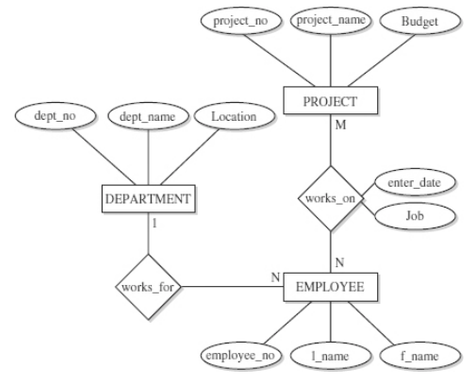
For the professional women and men who, from every perspective, seek to understand and better administer our relations with natural resources, this challenging new policy environment offers great opportunity. Much conflict about water resources development is based on concern about nature, but also relates to the needs of people. Productive, sustainable local social and economic activities based on less capital intensive use of natural resources are sometimes incorrectly dismissed as without value in a national balance sheet. We frequently look at today's needs and limitations, and rationalize natural resources damage that will cause lost opportunity and higher costs for our children. Yet those whose principle interest is the preservation of natural systems are equally capable of marginalizing and rationalizing. Agony and disease caused by contaminated community water supplies, loss of soils, water and wildlife to destructive farming methods that appear to be the only economy for many rural peoples, and cities throughout the world without the social structure to provide water, sanitation, shelter and food to millions of men, women and children, represent environmental crises as real as the threatened loss of a species or an ecosystem.

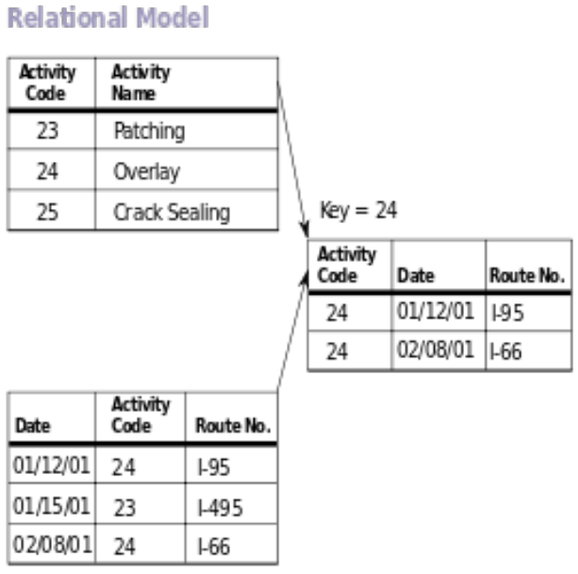
Precisely because more people can speak and be heard, more of us can learn, from each other and from the world outside our professional circles. It has become pointless to think of protecting the Earth's great resources without planning to deal with the demands of human settlement and economic development. It is equally futile to think of using water, or any other natural resource, without planning to protect the ability of ecosystems and biological resources to renew and sustain themselves.

This does not mean there will be no winners and losers. Resources of great economic value will remain undeveloped, to preserve unique and irreplaceable wild places and creatures, or will be less than fully exploited in order to protect the interests of local communities. In other cases, species, biological communities and human cultures will be lost forever to satisfy economic demands. But we are capable of making even those choices with greater wisdom, and are more capable of finding ways to keep humanity living in a living Earth. When water resource managers learn enough about the social and economic values of natural systems, science and engineering will better reveal to us the opportunities for meeting human needs while protecting nature.

When defenders of nature learn enough about the ways in which water flows through our economy, environmental science and engineering will participate more effectively in decisions that, in the end, are made by economically organized human societies.

If this mutual understanding is accompanied by commitment to respect the legitimacy and urgency of both missions, we can, in ways small and large, personal and institutional, help discover the real meaning of sustainable development, for our human communities and for the natural world that nourishes us all.





The Relational Model

ATTRIBUTES (the columns)
name:type

HEADING

BODY

title:string	year:int	length:int	genre:string
Gone with the Wind	1939	231	Drama
Star Wars	1977	124	SF
Wayne's World	1992	95	Comedy

TUPLES
(the rows)

Movies RELATION

Relations

Relation = Heading plus Body

Heading = (**Unordered**) **Set** of Attributes

Attribute= Attribute Name plus Type (normally simple indivisible types).

Body = (**Unordered**) **Set** of Tuples

Attribute Types are also known as Domains

Tuple = **Set** of attribute values, one for each attribute and of the attribute's type.

Schema for Relation = Name of relation plus heading, for example:

movies(title:string, year:int, length:int, genre:string)

Database = One or more Relations

Schema for Database = Schemas for all relations.

Why not lists, arrays, graphs, relations?

Relations II

In mathematics:

Given sets S_1, S_2, ... S_n

A relation R is a set of tuples (T_1, T_2, ... T_n) where $T_k \in S_k$

R is a subset of S_1 x S_2 ... x S_n.

Tuples are said to be R-related or 'in R'. e.g. Movies-related or in Movies

In the relational model we have *attributed* rather than *ordered* tuples:

R is the set of tuples (A_1:S_1=T_1, ... A_n:S_n=T_n) where $T_k \in S_k$

n is the **degree** of the relation, while the number of tuples is the **cardinality**

Drawing a relation

Important. In the relational model, the ordering of attributes and tuples is unimportant. We can present relations using whatever permutation of attributes and tuples we like, for example:

Movies

year:int	genre:string	title:string	length:int
1977	SF	Star Wars	124
1992	Comedy	Wayne's World	95
1939	Drama	Gone with the Wind	231

Drawing a relation II

Relations are not 2-dimensional tables. Although this a convenient way of presenting them on paper.

A better way to think of them is as a *set of n-dimensional values*.

For example:

(title:string=StarWars, year:int=1977, length:int=127, genre:string=SF)

is a 4-dimensional movie value.

Organization

- Weekly lectures
- Four short but assessed courseworks (already on CATe)
- Any questions: t.heinis@imperial.ac.uk

Recommended Books

The following textbooks provide excellent coverage of the course and provide many of the examples that are used in the course:

- **Database Systems: The Complete Book**, 2nd Edition, 1203 pages.
  Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, Pearson, 2009.

  Materials at http://infolab.stanford.edu/~ullman/dscb.html#slides

- **Database System Concepts**, 6th Edition, 1349 pages.
  Abraham Silberschatz, Henry K. Korth, S. Sudarshan, McGraw-Hill, 2011.

  Materials at http://codex.cs.yale.edu/avi/db-book/

- **Database Management Systems**, 3rd Edition
  - Raghu Ramakrishnan and Johannes Gehrke
  - Information at http://pages.cs.wisc.edu/~dbbook/