

Beyond Relational Databases

Future of Database

2/28

Core "database" goals:

- deal with very large amounts of data (terabytes, petabytes, ...)
- very-high-level languages (deal with big data in uniform ways)
- query optimisation (evaluation too slow \Rightarrow useless)

At the moment (and for the last 20 years) RDBMSs dominate ...

- simple, clean data model, backed up by theory
- high-level language for accessing data
- more than 30 years development work on RDB engine technology

RDBMSs work well in domains with uniform, structured data.

... Future of Database

3/28

Limitations/pitfalls of RDBMSs:

- NULL is ambiguous: unknown, not applicable, not supplied
 - "limited" support for constraints/integrity and rules
 - no support for uncertainty (data represents *the* state-of-the-world)
 - data model too simple (e.g. no support for complex objects)
 - query model too rigid (e.g. no approximate match)
 - continually changing data sources not well-handled
 - data must be "molded" to fit a single rigid schema
 - database systems must be manually "tuned"
 - do not scale well to some data sets (e.g. Google, Telco's)
-

... Future of Database

4/28

How to overcome (some of) these limitations?

Extend the relational model ...

- add new data types and query ops for new applications
- deal with uncertainty/inaccuracy/approximation in data

Replace the relational model ...

- object-oriented DBMS ... OO programming with persistent objects
- XML DBMS ... all data stored as XML documents, new query model
- application-effective data model (e.g. *(key,value)* pairs)

Performance: DBMSs that "tune" themselves ...

Big Data

5/28

Some modern applications have massive data sets (e.g. Google)

- far too large to store on a single machine/RDBMS
- query demands far too high even if could store in DBMS

Approach to dealing with such data

- distribute data over large collection of nodes (redundancy)
- provide computational mechanisms for distributing computation

Often this data does not need full relational selection

- represent data via *(key,value)* pairs
- unique *key* values can be used for addressing data
- *values* can be large objects (e.g. web pages, images, ...)

... Big Data

6/28

Popular computational approach to Big Data: *map/reduce*

- suitable for widely-distributed, very-large data
- allows parallel computation on such data to be easily specified
- distribute (map) parts of computation across network
- compute in parallel (possibly with further map'ing)
- merge (reduce) multiple results for delivery to requestor

Some Big Data proponents see no future need for SQL/relational ...

- depends on application (e.g. hard integrity vs eventual consistency)

Information Retrieval

7/28

DBMSs generally do precise matching (although `like`/regexps)

Information retrieval systems do approximate matching.

E.g. documents containing these words (Google, etc.)

Also introduces notion of "quality" of matching
(e.g. tuple T_1 is a *better* match than tuple T_2)

Quality also implies *ranking* of results.

Much activity in incorporating IR ideas into DBMS context.

Goal: support database exploration better.

Multimedia Data

8/28

Data which does not fit the "tabular model":

- image, video, music, text, ... (and combinations of these)

Research problems:

- how to specify queries on such data? ($image_1 \cong image_2$)
- how to "display" results? (synchronize components)

Solutions to the first problem typically:

- extend notions of "matching"/indexes for querying
- require sophisticated methods for capturing data features

Sample query: find other songs *like* this one?

Uncertainty

9/28

Multimedia/IR introduces approximate matching.

In some contexts, we have approximate/uncertain data.

E.g. witness statements in a crime-fighting database

"I think the getaway car was red ... or maybe orange ..."

"I am 75% sure that John carried out the crime"

- extends the relational model (ULDB)
- extends the query language (TriQL)

Stream Management Systems

10/28

Makes one addition to the relational model

- *stream* = infinite sequence of tuples, arriving one-at-a-time

Applications: news feeds, telecomms, monitoring web usage, ...

RDBMSs: run a variety of queries on (relatively) fixed data

StreamDBs: run fixed queries on changing data (stream)

Approaches:

- *window* = relation formed from a stream via a rule
- *stream data type* = build new stream-specific operations

Semi-structured Data

11/28

Uses *graphs* rather than tables as basic data structure tool.

Applications: complex data representation, via "flexible" objects, e.g. XML

Graph nature of data changes query model considerably.

(e.g. Xquery language, high-level like SQL, but different operators, etc.)

Implementing graphs in RDBMSs is often inefficient.

Research problem: query processing for XML data.

Dispersed Databases

12/28

Characteristics of dispersed databases:

- very large numbers of small processing nodes
- data is distributed/shared among nodes

Applications: environmental monitoring devices, "intelligent dust", ...

Research issues:

- query/search strategies (how to organise query processing)
- distribution of data (trade-off between centralised and diffused)

Less extreme versions of this already exist:

- grid and cloud computing
- database management for mobile devices

Looking ahead

13/28

Every so often, DBMS researchers meet to consider the field:

- Laguna Beach, 1989 ... <http://doi.acm.org/10.1145/382272.1367994>
- Asilomar, 1998 ... <http://doi.acm.org/10.1145/306101.306137>
- Claremont, 2008 ... <http://doi.acm.org/10.1145/1462571.1462573>
- Beckman, 2016 ... <http://doi.acm.org/10.1145/2845915>

Regular attendees: Rakesh Agrawal (IBM), Phil Bernstein (MS), Mike Carey (BEA), Stefano Ceri (Pisa), David deWitt (MS), Michael Franklin (UCB), Hector Garcia-Molina (Stanford) Jim Gray (MS), Laura Haas (IBM), Alon Halevy (Google) Joe Hellerstein (UCB), Mike Lesk

Beyond COMP9311

14/28

COMP9315 Database Systems Implementation

- comprehensive study of DBMS internals

COMP4317 XML and Databases

- all about XML + relationship to DBMSs

COMP9318 Data Warehousing and Data Mining

- data summarisation/discovery techniques

COMP9319 Web Data Compression and Search

- compression and searching algorithms

COMP6714 Information Retrieval and Web Search

- finding information in unstructured text
-

Course Revision and The Exam

COMP9311 Course Aims

16/28

At the end of this course you should be able to:

- develop accurate, non-redundant data models;
 - realise data models as relational database schemas;
 - formulate queries via the full range of SQL constructs;
 - use stored procedures and triggers to extend DBMS capabilities;
 - understand principles and techniques for administering RDBMSs;
 - understand performance issues in relational database applications;
 - understand the overall architecture of relational DBMSs;
 - understand the concepts behind transactions and concurrency control;
 - appreciate query and transaction processing techniques within RDBMSs;
 - appreciate the past, present and future of database technology.
-

Syllabus Overview

17/28

1. Data modelling and database design
 - Entity-relationship (ER) design, relational data model
 - Relational theory (algebra, dependencies, normalisation)
2. Database application development
 - SQL for querying, data definition and modification (PostgreSQL's version)
 - Extending SQL Queries, Functions, Aggregates, Triggers
 - PostgreSQL, `psql` (an SQL shell), `PLpgSQL` (procedural SQL)
 - PHP (DB access)
3. DBMS technology
 - Performance tuning, catalogues, access control
 - DBMS architecture, query processing, transaction processing

Things in grey will definitely **not** be examined this semester (Session 2, 2017).

Assessment Summary

18/28

Your final mark/grade will be determined as follows:

a1	= mark for assignment 1	(out of 10)
a2	= mark for assignment 2	(out of 15)
a3	= mark for assignment 3	(out of 15)
a_tot	= a1 + a2 + a3	
exam	= mark for exam (written)	(out of 60)

mark = a_tot + exam

Final Exam

19/28

- 2 hours written exam + 10 minutes reading time
 - NO textbooks, notes, calculators, etc.
 - 5 questions (some with sub questions)
 - Answer all questions
 - Questions are worth 12 marks each
 - Total number of marks is 60
-

... Final Exam

20/28

Question 1 is related to ER Diagram

- You need to understand how to draw an ER diagram based on given specifications
-

... Final Exam

21/28

Question 2 is related to Relational Mapping and SQL Schema

- You need to understand how to convert a given ER diagram into a relational model
 - You need to understand how to write SQL statements to create tables of a relational model
-

... Final Exam

22/28

Question 3 is related to Relational Algebra and SQL Queries

- You need to understand how to relational algebra expressions for specific queries
 - You need to understand how to write SQL queries
-

... Final Exam

23/28

Question 4 is about PL/SQL (SQL Procedural Language)

- You need to understand how to write functions
 - You need to understand how to write triggers
-

... Final Exam

24/28

Question 5 is about Functional Dependency and Normalization

- You need to understand functional dependencies
 - You need to understand candidate keys
 - You need to understand how to decompose a relation into 3NF and BCNF
-

Revision

25/28

Sources for revision material:

- COMP9311 Lecture Notes, Lab Exercises, Assignments
 - *Fundamentals of Database Systems*, Elmasri/Navathe
 - *Database System Concepts*, Silberschatz/Korth/Sudarshan
 - *Database Management Systems*, Ramakrishnan/Gehrke
 - *Database Systems: Complete Book*, Garcia-M/Ullman/Widom
 - *Database Systems: App-oriented*, Kifer/Bernstein/Lewis
 - PostgreSQL/PHP Documentation (to some extent)
-

Supplementary Exams

26/28

Supplementary Exams are only available to people who

- are absent from the Final Exam with *good reason*
(good = documented, serious, clearly affects ability to do exam)
- have performed well during the rest of the semester

If you are awarded a Supp Exam ...

- you **must** make yourself available for it (late November)
- non-attendance at the Supp ⇒ mark of 0 for the exam

Some Thoughts ...

27/28

You need to learn for life, not just for the exam.

In particular, learn to find answers for yourself.

No single correct answer. (Solutions range from poor to excellent)

Take *pride* in your work. (Aim for quality, not just correctness)

Finally ...

28/28

Good Luck with the Exams
