CMPU4003: Advanced Databases

- TU856, TU857, TU858 plus Erasmus
- Dr. Deirdre Lawless
  - e: deirdre.lawless@tudublin.ie

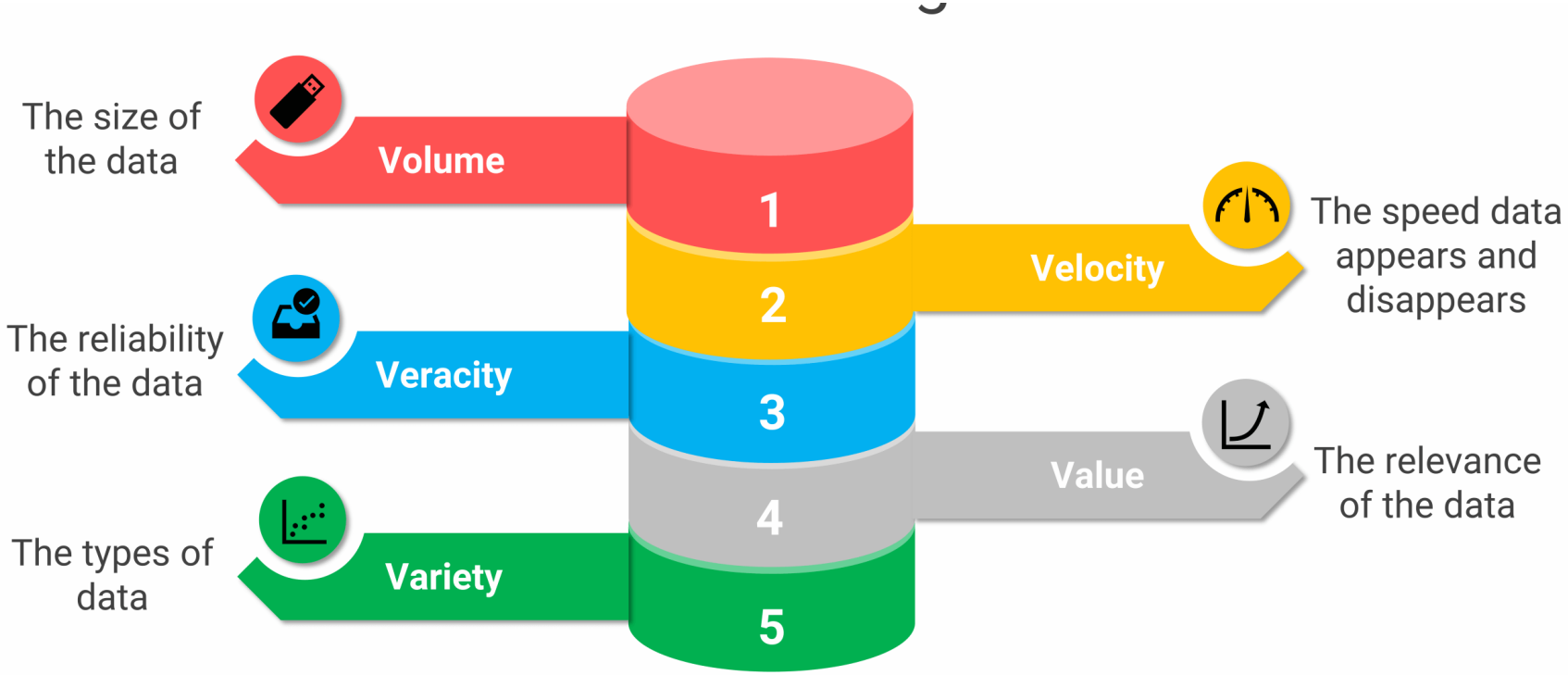
Problem?

Data (and the ability to use manage and use it) is the main focus of modern computer systems

Volumes of data are growing rapidly

And so is the variation in the type of questions analysis organisations want to conduct using this data

- Which increasingly requires multiple types of data suited to data models other than relational

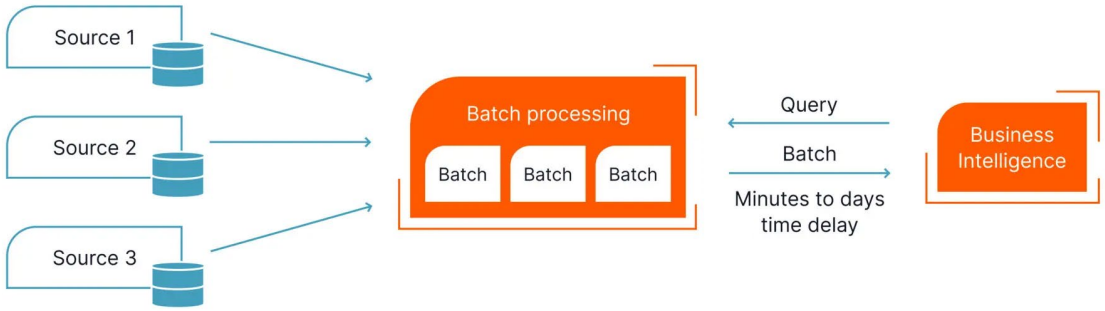

The size of the data
Volume
1
Velocity
2
The speed data appears and disappears
The reliability of the data
Veracity
3
Value
4
The relevance of the data
The types of data
Variety
5

Big Data 5 Vs

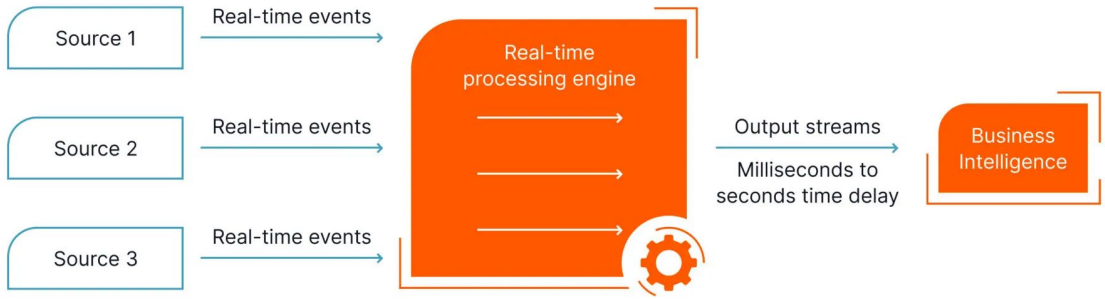
Big Data

intelliarts

Batch processing of big data



Real-time processing of big data



Big Data Trends 2025

Peripheral/Edge Integration

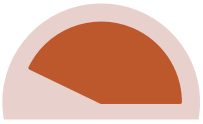
Real Time Processing and Analytics

AI, ML and Big Data Synergy

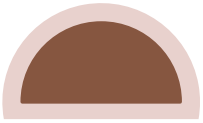
Advanced Analytics and Prediction

Blockchain

Data Lakes



Hybrid and Multi-Cloud



Problem?

Organisations have lots and lots and lots of data

- Not necessarily nicely structured and organised

And are constantly collecting more…

Big (MAD) ML, AI and Data Landscape 2024

https://mad.firstmark.com/

Features 2,011 logos in total.

Note: the first version of this in 2012 had 139 logos

What is the module about?

Making Databases Work !

- Implementing
  - Database design
  - Data storage
  - Management
  - Retrieval
  - and Architectures
- Needed to support contemporary enterprise applications

Learning Outcomes

Discuss

- Challenges of, and discriminate between approaches to, database modelling – conceptual, logical, and physical design;

Discuss

- Database considerations for data integrity, integration, security, query optimisation, performance tuning and concurrency control in contemporary enterprise applications;

Apply

- Techniques for data integrity, security, data optimisation, performance tuning and concurrency control;

Understand and apply

- Different approaches to data integration, both semantic and physical;

Understand and apply

- Data modelling techniques for logical and physical design to support contemporary enterprise applications;

Learning Outcomes

Design, create, and query

- A data warehouse;

Design, create, and query

- A distributed database;

Design, create, and query

- A document-oriented database

Critically compare

- The strengths and limitations of different database technologies used in contemporary enterprise applications;

Discuss

- Recent developments and emerging trends in database technologies and their use in contemporary enterprise applications

Module Content

Relational database design: reviewing and distinguishing between conceptual and logical design and addressing considerations and implications of implementing physical design from logical design.

Data warehouses: definitions, rationale, architectures, design, implementation, and manipulation; dimensional modelling techniques: design of star and snowflakes schemas, types of fact tables, conversion of relational model into dimensional model.

Distributed databases: definitions, rationale, architectures, design, implementation, and manipulation; approaches to fragmentation and replication (synchronous and asynchronous); the CAP theorem.

Document-oriented data stores: definitions, rationale, architectures, design, implementation, and manipulation; differentiating between document-oriented, key-value pairs, column-oriented and graph types; designing schema-less data structures;.

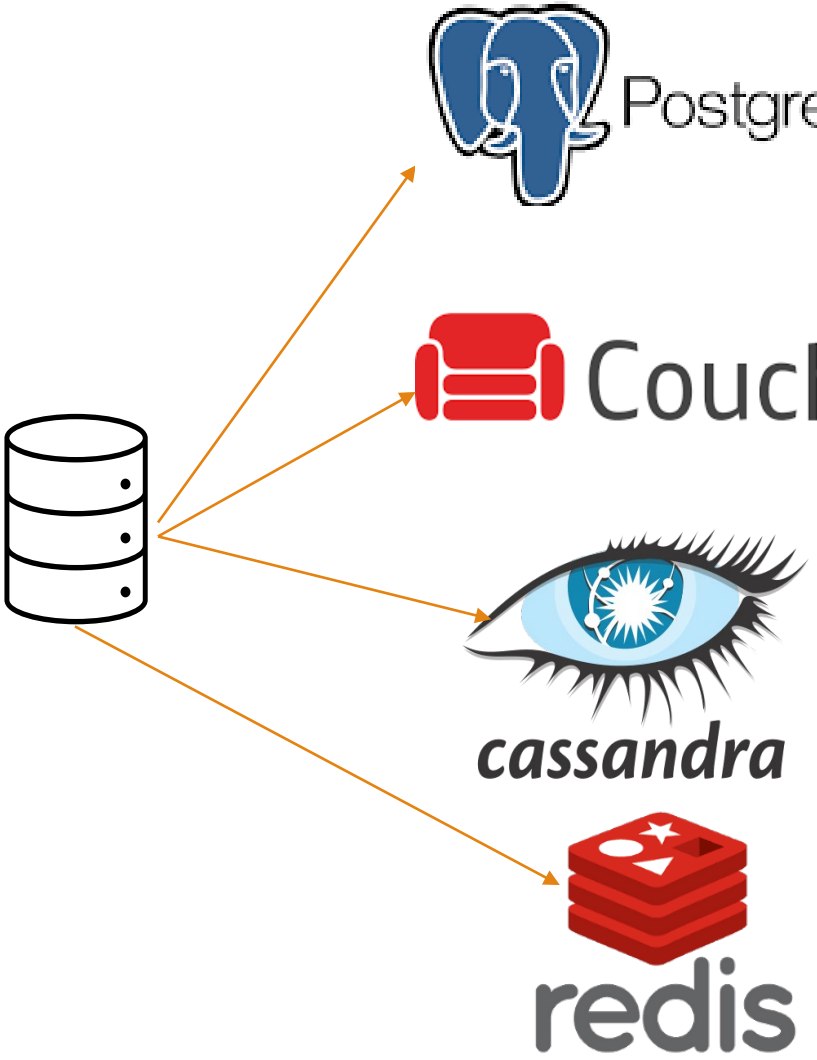
Module Content

Data and database integration: process, challenges, benefits; ETL process for different database types; standards for interoperability and integration including JSON and Web services.

Optimisation and performance tuning: understanding the reasons for and the implications of poorly performing queries; identifying poorly performing queries and structures; examining how queries are executed in various database technologies; using constraints and indexes; choosing between alternative execution plans; amending queries to perform more appropriately.

Security and integrity: requirements, e.g. confidentiality, integrity, reliability, storage, and availability; security preserving approaches, integrity preserving approaches; polyglot persistence; transactions, concurrency control; the BASE model vs. the ACID model for transactions management.;

Emerging database technologies and application areas.



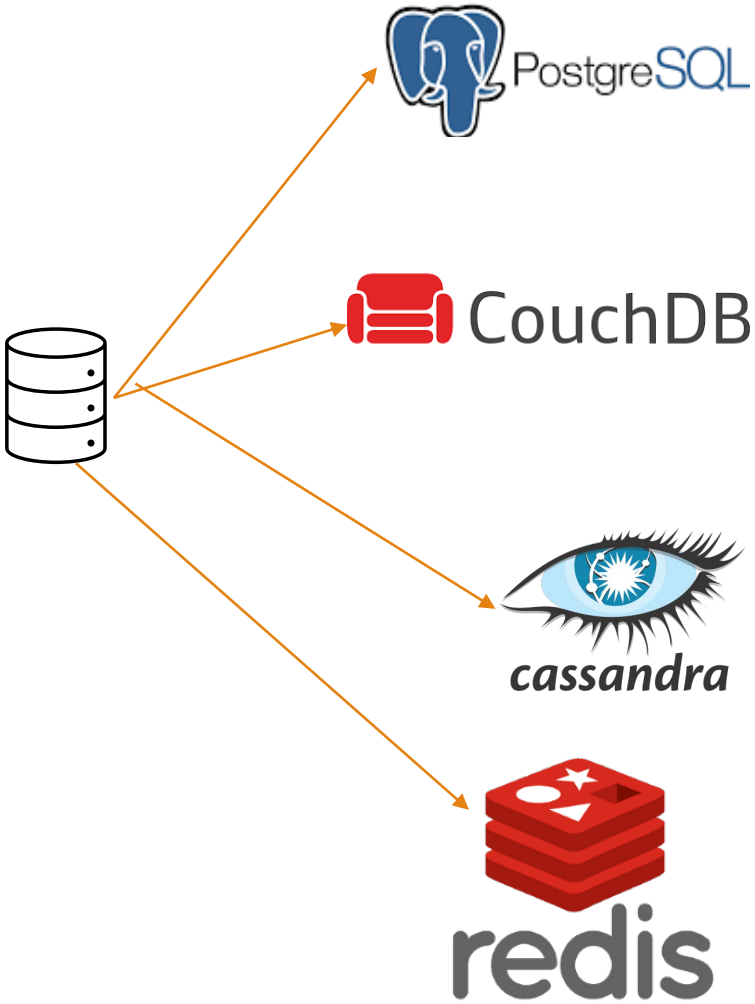
Relational

Document

Distributed

In Memory

Planned Technologies



Relational database

Open-source document-oriented NoSQL database, implemented in Erlang.

Open source, distributed and decentralized/distributed storage system (database)

Open-source, in-memory data structure store used as a database, cache, and message broker

Planned Technologies

You can use Docker on your own local machine to create containers for each database type.

You can use GitHub codespaces which will provide you with a cloud-based development environment where you can create containers for each database type.

If an online alternative exists that allows you to achieve what you need to achieve you can use that.

Planned Technologies

Exam (40%)

CA (60%)

PostgreSQL

cassandra

CouchDB

redis

What will we do in the labs?

Setup:

- a relational database (PostgreSQL)
- a document store (CouchDB)
- a column-oriented store (Cassandra)
- an in memory store (Redis)

PostgreSQL

cassandra

CouchDB

redis

What will we do in the labs?

- Implementing:
  - optimization
  - replication
  - partitioning
- Porting data between different data models

What will the CA require?

Design and implement a polyglot system that integrates both a **data warehouse** and a **production database environment**.

Develop ETL processes to transfer and transform data between the different data stores.

Formulate and execute queries across the system to address analytical and operational needs.

Evaluate and apply optimization strategies, making informed choices for query performance and system efficiency.

Implement replication and partitioning mechanisms to support scalability, fault tolerance, and availability.

Provide evidence and justification for all key design and implementation decisions.

Critically reflect on the effectiveness of your decisions, considering trade-offs and possible alternatives.