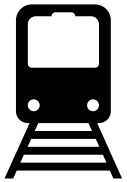
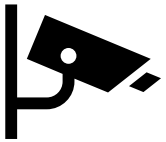
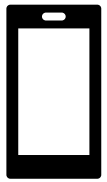
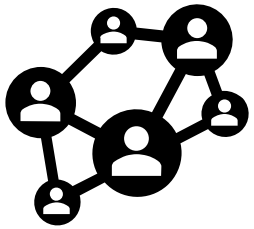
Data and serialisation

Computing & Information Sciences

W. H. Bell

Sources of data

- Distributed sensors recording values:
  - Measurements from wind turbines.
  - Monitoring for the power grid, rail network, other infrastructure.
  - Monitoring of white goods.
  - Data from mobile phones.
  - Pictures taken from CCTV cameras.
  - Documents on the Internet.
  - Data uploaded to social media.
- Frequent sampling leads to large data sets.

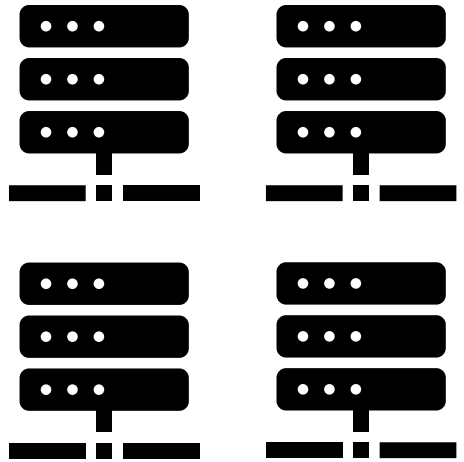
Possible applications

- Preempting maintenance.
  - Remote equipment or distributed devices.
- Traffic modelling.
  - Using mobile phone data.
- Finding criminals.
  - Face recognition.
- Analyse behaviour.
  - Recommend products and services.

Challenges to overcome

University of Strathclyde Glasgow

- Large volumes of data.
- Many different data formats.
- Rapid analysis.
- Distributed data analysis.
  - Processing bottlenecks.
- Process in parallel.
  - Must be resilient to failure.



Considerations

- Size of data sample.
- Type of data – sensitive or replaceable.
- Existing infrastructure – cluster or database server.
- Speed or performance requirements.
  - Scaleability.

Data storage approaches

University of Strathclyde Glasgow

- Database server.
  - One computer that hosts a database and accepts connections.
  - Can be overwhelmed with requests, but is robust.
- Distributed database.
  - Parallel read, single write or parallel write.
  - Many servers or files database files (SQLite).
- Distributed files containing serialised information.
  - Minimise input/output bottleneck.
- Cloud or on premises solution.

Schema constraints vs evolution

University of Strathclyde Glasgow

- Schema describes data structure.
- Data may evolve or remain similar.
  - Schema constraints may be more or less helpful.
  - Need schema constraints for more critical applications.
  - May tolerate schema changes between data records.

Serialisation approaches

University of Strathclyde Glasgow

- Relational databases or tables.
  - Data are stored across one or many tables.
  - Can enforce data to match table definitions.
- Object-based storage.
  - JavaScript Object Notation (JSON).
  - Custom binary serialisation.
  - Can enforce schema or allow schema to evolve.

Relational model

University of Strathclyde Glasgow

- Describes data as relations.
  - Defines a set of high level operations.
- Users do not need to know underlying implementation.
  - No physical pointers.
  - Different database servers may use different implementation.
- Manipulate using Structured Query Language (SQL).

Codd, E.F. (1970). "A Relational Model of Data for Large Shared Data Banks". Communications of the ACM 13 (6): 377–387

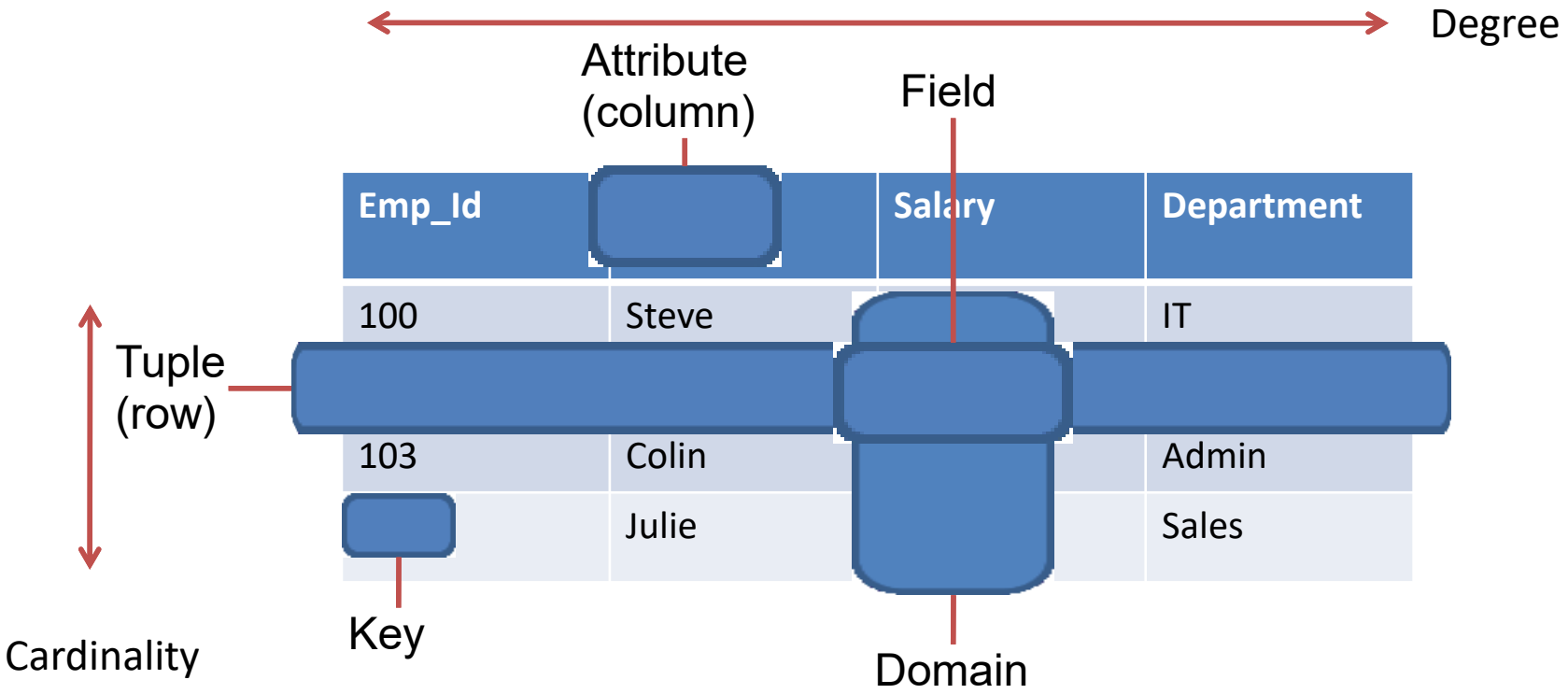
Relational databases

Hold data in tables (relations).

Id	Forename	Surname
1	John	Smith
2	Jane	Doe
3	Sally	Hughes

A relation (table)

2 dimensions: rows and columns



Employees (emp_id, name, salary, department)

Schema

- Of the relation (table).
  - Table name and set of attributes.
  - Employees(employee_id, last_name, first_name, email, ...)
  - Need type definitions too.
- Of the database
  - Consists of several relations (tables).
  - Set of constraint names, apply to tables.

Properties of a relation (table)

University of Strathclyde Glasgow

- Relation (table) has unique name in database schema.
- Each tuple (row) is distinct.
  - Database implementations may allow duplicate rows.
  - The order of the tuples is not important.
- Each attribute (column) has a distinct name.
  - The order of the columns is not important.
  - Each attribute has an elementary type.
- There is only one atomic value in each field.
  - Matching type of attribute (column).

Keys

- Comprising one or more attributes.
  - Used to identify the tuples (rows).
- Primary keys.
  - Required to be unique.
  - Can be automatically incremented.
- Foreign keys.
  - Used to refer to other tables.
  - Primary key value from other table.
  - Used to form constraints between tables.

Relational databases

Flatten arrays into normalised tables.

Id	Forename	Surname
1	John	Smith
2	Jane	Doe
3	Sally	Hughes

PurchaseId	CustomerId	ItemId
1	1	3
2	3	4

Relational databases

Flatten arrays into normalised tables.

Id	Forename	Surname
1	John	Smith
2	Jane	Doe
3	Sally	Hughes

ManagerId	EmployeeId
2	1
2	3

Relational schema constraints

University of Strathclyde Glasgow

- Field type.
  - Integer, floating point, text string, etc.
- Field length.
  - Maximum value or length of text.
- Unique values.
  - Primary key – unique, can use auto-increment.
  - Other unique values – numbers or text.

Relational schema constraints

University of Strathclyde Glasgow

- Foreign key.
  - Require value to be present in other table.
  - Prevent deletion.
  - Cascade deletion.
- Combine schema constraints to restrict data values.
  - Bank account.
  - Personal information.

Relational databases benefits

Data independence

University of Strathclyde Glasgow

- Not tightly coupled to underlying implementation.
  - Do not need to know physical storage format.
  - Application accesses database through SQL.
    - Data are associated with fields.
    - Same logic to retrieve few or many fields.

Relational databases benefits

Data security

- Resilience
  - No loss of data in case of system failure.
  - Data are serialised as part of transaction.
- Security
  - Database server user name and password, with TLS (SSL).
  - Permissions associated with database.
  - Permissions associated with views – limit access to tables.

Relational databases benefits

Data integrity

- Data Integrity
  - Prevent duplicate rows.
  - Enforce by integrity constraints.
- Efficient concurrent access.
  - Transactions do not interfere with each other.
  - Resources may be scaled according to access needs.

Relational databases benefits

Analytics and administration

University of Strathclyde Glasgow

- Database server logs actions and properties.
  - Size of tables and number of indices.
  - Number of table or column operations.
  - Query execution time.
  - Database size.
  - User interactions with database.
- Database administrator (DBA) has centralised control.
  - Design, usage.
  - Performance tweaks.

Relational databases disadvantages

University of Strathclyde Glasgow

- Complex system that has to be maintained.
  - Server and database can become large (memory, disk).
- Cost
  - Hardware costs – needed for higher performance.
  - Cost of transfer to schema or new schema.
- Scaling
  - Avoid parallel write to distributed database.

Document storage

Used for data serialisation.

- JSON (JavaScript Object Notation)
  - Web service communication and NoSQL database interface.
- BSON (Binary JSON)
  - Used internally by NoSQL database MongoDB.
- XML (Extensible markup language)
  - Java web services and some file formats (Office).
- YAML (YAML Ain't Markup Language)
  - Data formatting with minimal markup.
  - Docker files, NoSQL databases.

Document storage
JSON example

Describe nested structure within one document.

```
[
    {
        "Forename": "John",
        "Surname": "Smith",
        "Purchases": [
            {
                "PurchaseId": 1,
                "ItemId": 3
            }
        ]
    }
]
```

NoSQL

University of Strathclyde Glasgow

- Not only SQL (NoSQL).
  - Different commands with respect to SQL.
    - Some inspiration taken from SQL – read, write, queries.
  - Typically, data are not stored in tables.
- Types of NoSQL database.
  - Document database (JSON, BSON) - MongoDB
  - Key-value pair storage – Couchbase
  - Column-oriented databases - Cassandra
  - Graph databases - Neo4j

NoSQL: features

University of Strathclyde Glasgow

- Flexible schema definition.
  - Allow differences between records/documents.
  - Schema definition is possible, but more limited.
  - No foreign keys.
- Easier to distribute database.
  - Easier to scale to address loading.
  - Data held structures similar to hash tables.
  - Use cheap storage, rather than specialist servers.

NoSQL Types: Document Stores

University of Strathclyde Glasgow

- Stored as key (text) and document.
  - Similar to key and value store.
- Documents comprise serialised data.
  - XML
  - JSON
  - BSON
  - Other methods of data serialisation.
- Some schema constraints are possible.
  - Within the document schema.
  - Define index for collection.

NoSQL Types: Object Stores

University of Strathclyde Glasgow

- Key, value pair storage.
- Driven by object oriented programming.
  - Serialise and deserialise objects to and from NoSQL database.
    - Serialised data stored as text.
  - Write and read data using operator overloading.
    - Intuitive access.
  - Written in C++ or Java.
  - Provide atomicity, consistency, isolation, and durability (ACID) compliance.

NoSQL Types: Object Stores

University of Strathclyde Glasgow

```
<!- Write to database -->
<os:store key="state">
  <os:value>
    <![CDATA[#[
      output application/json
       ---
       {
          "id": attributes.queryParams.id,
          "timestamp": now(),
          "name": payload.name
        }
      ]]]>
  </os:value>
</os:store>
```

```
<!-- Query using key -->
<os:retrieve key="userId" />
```

https://docs.mulesoft.com/object-store-connector/1.2/object-store-to-store-and-retrieve

NoSQL Types: Graph Databases

University of Strathclyde Glasgow

- Each data entity (node) has pointer (relationship) to others.
  - Graph of several connected entities.
- Optimised for queries across links.
  - Much faster than equivalent SQL queries against relational database.
- Fast link processing applications:
  - Shortest distance between connected nodes.
  - Find connections between people – crime prevention.
  - Network or electricity supply analysis.

https://neo4j.com/

https://neo4j.com/blog/practical-applications-neo4j-4-0/

NoSQL Types: Graph Databases

University of Strathclyde Glasgow

Data comprises labels, values, and link relationships.

Query data structure using labels.

```
MATCH (s:Student)-[:FOLLOWING]->(c:Course)
WHERE c.name = "MSc Software Development" RETURN s.name
```

https://neo4j.com/developer/graph-db-vs-rdbms/

https://neo4j.com/developer/cypher/querying/

MongoDB

University of Strathclyde Glasgow

- MongoDB (from humongous) – document store.
  - JSON interface.
  - BSON storage.
  - Schema definition possible.
  - Written in C++, JavaScript and Python.
- Several server types.
  - Cloud instance.
  - Community edition, Percona Server for MongoDB.

https://www.mongodb.com/

https://www.percona.com/software/mongodb/percona-server-for-mongodb

MongoDB: Document

University of Strathclyde Glasgow

Key: value pairs

```
{
    full_name: "John Smith",
    reg_status: "Registered",
    reg_number: 20221012332,
    modules: ["CS982", "MS418"]
}
```

Keys are unquoted in JavaScript implementation.
Keys are quoted when using Python interface.

https://www.mongodb.com/docs/manual/core/document/

MongoDB: Structure

University of Strathclyde Glasgow

- Server hosts many databases.
  - Each database has corresponding set of files.
- A database comprises one or more collections.
- A collection comprises one or more documents.
  - Similar and related purpose.
  - No enforced document structure by default.
  - BSON schema definition is possible.

MongoDB: Features

University of Strathclyde Glasgow

- Flexible schema.
  - Query across documents – selecting matching.
- Performance.
  - Fast create and update.
  - Server-side JavaScript execution.
  - Flexible horizontal scaling – sharding.
  - Indexing – small section of data for rapid searches.
  - Aggregation
    - Summing across documents.
    - Aggregation pipeline – similar to MapReduce.

https://www.mongodb.com/advantages-of-mongodb
https://www.mongodb.com/docs/manual/indexes/ https://www.mongodb.com/docs/manual/aggregation/

CAP Theorem

University of Strathclyde Glasgow

Choose mixture of properties, depending on application.

- Relational databases:
  - Ensure Atomicity, Consistency, Isolation, Durability (ACID) compliance.
- NoSQL – trade off between:
  - Consistency
    - Every read receives the most recent write or an error.
  - Availability
    - Every request receives a (non-error) response – without guarantee that it contains the most recent write.
  - Partition tolerance
    - The system continues to operate despite an arbitrary number of messages being dropped (or delayed) by the network between nodes.

CAP Theorem

University of Strathclyde Glasgow

- Distributed systems can fail.
  - Temporary network partitioning (failure).
  - Choose consistency or availability.
- Consistency:
  - Time out or error, due to disconnection.
- Availability:
  - Return data – may not be newest information.

Polygot Persistence

University of Strathclyde Glasgow

- Use several data storage technologies together.
  - Use best features of each technology.
  - For example:
    - Relational database – personal or sensitive information.
    - MongoDB – document store – replaceable, changing data.
- Microservice per database type.
  - Combine microservices in one application.

University of Strathclyde Glasgow