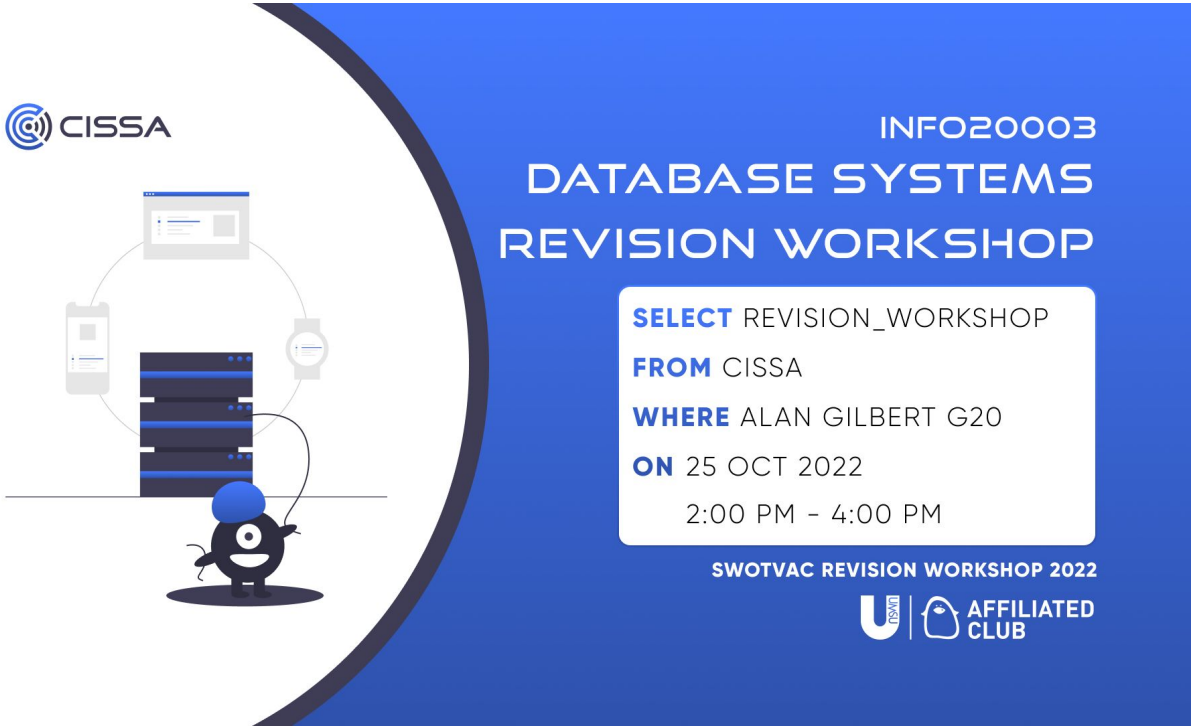
Database Systems

Tutorial Week 12

CISSA Revision Workshop



INFO20003 feedback

- SLS Feedback – please help us!
  https://www.unimelb.edu.au/sls

- Changes based on previous feedback:
  - Revamped the subject (new syllabus)
  - New lectures, new tutorials, new labs
  - Assignment feedback generator
  - Constant adjustments based on your feedback (sample exams, Peerwise, practice on your own, database playground, polls)
  - No MST in 2019, developed lecture quizzes
  - Incorporated flipped model with playlists for a quick revision
  - **We *hear you* and *thank you* for your feedback**

Please tell us which parts you enjoyed and what else would you like to see

Objectives

I. Understand the concepts of NoSQL databases
II. Choosing appropriate NoSQL database types for scenarios
III. CAP theorem with respect to NoSQL databases
IV. Revision

Key Concepts

- What are NoSQL databases?
  - “Non-relational” databases that facilitate the storage and retrieval of *non-tabular data* (e.g. JSON, XML)
  - Use a more flexible model
  - Don’t depend on a particular structure such as tables, rows, columns or schemas to organise data
  - Traditional relational databases are unable to meet performance, scalability and flexibility requirements of handling *Big Data*
    - 3Vs of Big Data:
      - Volume (lots of data)
      - Variety (different data types and formats)
      - Velocity (data comes at a very fast rate)

Key Concepts

- Four main categories of NoSQL databases
  - Graph databases
  - Key-value stores
  - Column-family stores
  - Document stores

Key Concepts

- Types of NoSQL database
  - Graph databases
    - Use a graph to store, connect and query data
    - Nodes are linked together by edges
    - Nodes = rows in relational database, and represent entities
    - Edges = relationships
    - Both nodes and edges can have properties associated with them
    - E.g. neo4J (used by Airbnb, Microsoft, IBM, eBay and Walmart)

Key Concepts

- Types of NoSQL database
  - Key-value stores
    - The most flexible and least structured NoSQL databases
    - Use a key-value structure to organise data, so basically like a massive Python dictionary
    - There is no schema
    - Support massive scalability
    - Key
      - Unique identifier
      - Can theoretically be anything but DBMS can impose limitations e.g. key size, key type to achieve better performance
    - Value
      - Can be of any data type (images, PDFs, binary files, JSON files, videos, long text etc.)
    - E.g. Berkeley DB

Key Concepts

- Types of NoSQL database
  - Column-family stores
    - A.k.a. wide-column stores or extensible record stores
    - Type of key-value database
    - Like relational databases, column-family databases use tables, rows and columns
    - But for each record, the column names, their format and record keys can vary
    - This results in a “schema-free” structure
    - Columns are created for each row instead of being pre-defined for a table
    - E.g. Cassandra

Key Concepts

- Types of NoSQL database
  - Document stores
    - Similar to key-value stores (there's a key, and a value where data is stored)
    - However, the value contains structured or semi-structured data (a *document*)
    - This data is typically stored in JSON, XML or BSON documents (mostly JSON)
      - Can also be stored in industry-specific data formats e.g. MARC
    - These documents are independent components which can be distributed more easily
    - Storage doesn't require compliance with a set schema
      - Each document can have its own structure and schema, which increases agility and flexibility
    - Still possible to create indexes within documents
      - If indexing features associated with document databases are used to their fullest, they can provide fast and efficient querying of data
    - E.g. MongoDB (see week 12 lab)

Exercise — Choosing a NoSQL Database (8 mins)

Libraries store information about their collections in their catalogue.

Match each of the following statements to the type of NoSQL database that would be best for storing that library's data. Select from the four types of NoSQL database discussed previously.

a. In one library, items are catalogued by author, title and publisher, as well as any number of other fields chosen by the cataloguer, such as physical description, subject codes and notes.
b. In another library, each catalogue record is stored in the MARC format (Figure 1), a coded text format that contains all the catalogue information for a particular item.
c. A public library wishes to store cover photos of all its items, which might be in JPEG, PNG or PDF format, or stored as a URL.
d. A university library wishes to keep track of which published academic papers reference each other in order to help researchers measure their metrics.

```
LEADER 00000nam  22000001  4500
008    730220s1955    ilu      b    00000 eng
019    55007351
050 0  QA276.5|b.R3
082    311.22
110 20 Rand Corporation.
245 12 A million random digits|bwith 100,000 normal deviates.
260 0  Glencoe, Ill.,|bFree Press|c[1955]
300    xxv, 400, 200 p.|c28 cm.
504    Bibliography: p. xxiv-xxv.
650  0 Numbers, Random.
984    |cMS T 519 R152
```

Figure 1: An example of a MARC record. MARC is a very old format that predates NoSQL, JSON and even XML by several decades, yet it remains the industry standard in library data systems.

Exercise — Choosing a NoSQL Database (8 mins)

a. In one library, items are catalogued by author, title and publisher, as well as any number of other fields chosen by the cataloguer, such as physical description, subject codes and notes.

- A column-family database would be the best choice
- Each row in a column-family table can have a different set of columns associated with it

Exercise — Choosing a NoSQL Database (8 mins)

b. In another library, each catalogue record is stored in the MARC format (Figure 1), a coded text format that contains all the catalogue information for a particular item.

- A document store would be the best choice
- Industry-specific structured data formats like MARC can be used with specialised document store systems

Exercise — Choosing a NoSQL Database (8 mins)

c. A public library wishes to store cover photos of all its items, which might be in JPEG, PNG or PDF format, or stored as a URL.

- A key-value store would be the best choice
- They can store any kind of data
- A document store wouldn’t be a good choice
  - Images are unstructured data
  - Each document in a document store should be made up of *semi-structured* or *structured* data

Exercise — Choosing a NoSQL Database (8 mins)

d. A university library wishes to keep track of which published academic papers reference each other in order to help researchers measure their metrics.

- A graph database would be the best choice
- You could store papers as nodes and references as edges joining the nodes
- A graph database can efficiently capture and answer complex queries about the relationships between papers

Key Concepts

- Advantages of NoSQL
  - 4 key advantages:
    - Flexible modelling
    - Scalability
    - High availability
    - Performance

Key Concepts

- Advantages of NoSQL
  - Flexible modelling
    - NoSQL databases aren’t restricted to a fixed schema, data types, row size and column names
    - Makes it more suited to less structured data sources e.g. chat data, videos etc. (which are exponentially growing!)

Key Concepts

- Advantages of NoSQL
  - Scalability
    - Capacity in a NoSQL database can be added and removed quickly with both horizontal and vertical scaling
      - Horizontal scaling: add more servers to connect to a database
      - Vertical scaling: add more power to existing systems (e.g. RAM, storage, CPU cores etc.)
    - Good for handling Big Data
    - Avoid the cost and complexity of scaling up a relational database into a distributed database
      - Horizontal scaling is difficult for relational databases since there are different tables and you have to consider joins

Key Concepts

- Advantages of NoSQL
  - High availability
    - NoSQL databases are typically stored in partitions
    - If sites fail, the database can continue its read and write operations at a different site

Key Concepts

- Advantages of NoSQL
  - Performance
    - NoSQL databases are typically stored in partitions
    - Big social media companies have users globally so they deploy data centres in different parts of the world and partition their users so that they’re routed to the closest data centre
      - Users get the best/fastest experience

Key Concepts

- The CAP theorem
  - 3 key components:
    - Consistency
    - Availability
    - Partition tolerance

Key Concepts

- The CAP theorem
  - Consistency
    - All servers hosting the database have the same data, so users access the same data irrespective of which server is used to answer a query
    - This is different to “consistency” in the context of ACID principles, which refers to data integrity constraints

Key Concepts

- The CAP theorem
  - Availability
    - The system will always answer a query, even if it’s not the latest data or consistent across the system

Key Concepts

- The CAP theorem
  - Partition tolerance
    - When the system continues to operate as a whole, even if individual servers fail or can’t be reached

Key Concepts

- The CAP theorem states that…
  - At any given point in time, a system can achieve two out of three principles
  - In the case of NoSQL databases, the choice is between AP or CP
    - Since NoSQL databases are typically stored in partitions, the biggest advantage of NoSQL databases compared to relational DBMSs is partition tolerance

Key Concepts

- The CAP theorem
  - AP
    - Ensures availability instead of consistency
    - The database always answers, but possibly with outdated or wrong data
    - Occurs with systems that allow reads before all sites are updated
    - Such systems eventually achieve consistency
  - CP
    - Ensures consistency instead of availability
    - The database stops all operations until the latest copy of data is available at all sites
    - Such systems become available after consistency is achieved
  - Most NoSQL databases choose AP over CP to ensure continuous availability and eventual consistency (B.A.S.E.)

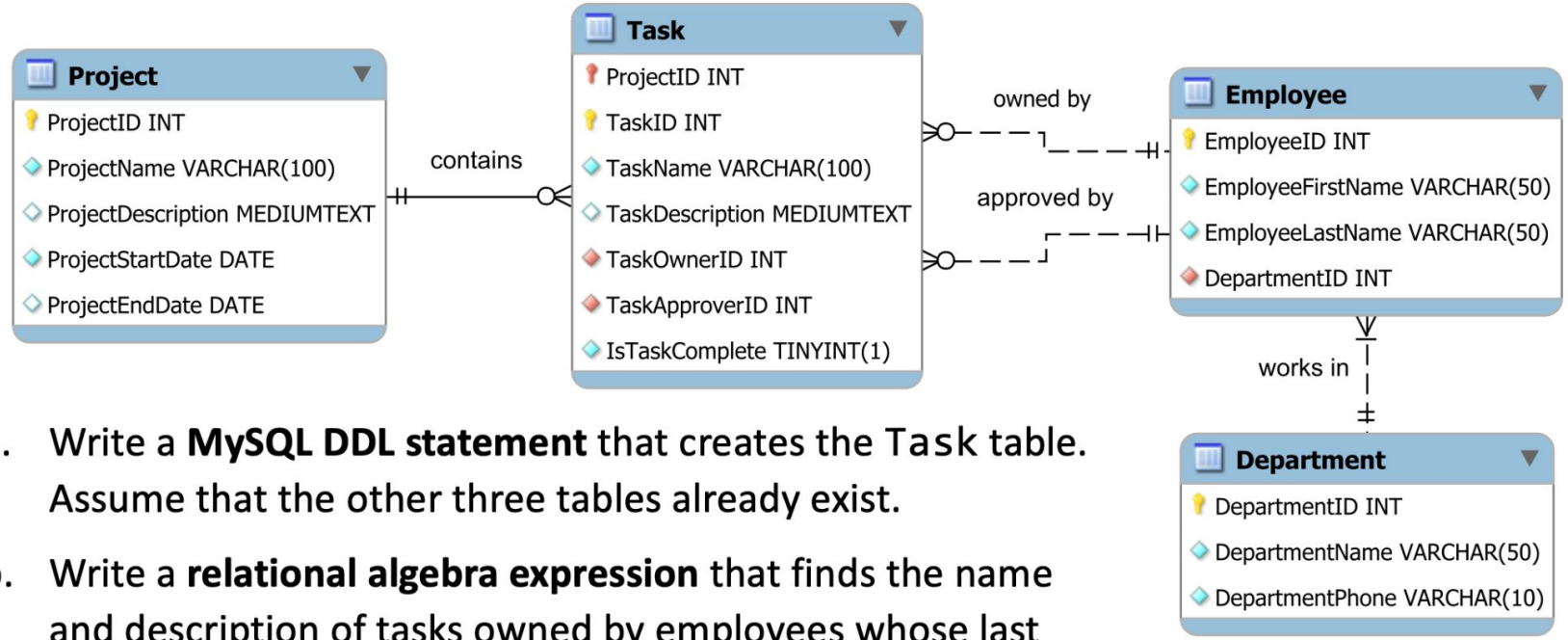
Exam Revision

(Some of) what to revise for in preparation for the exam:

1. ER diagrams
2. DMBS design (conceptual → logical → physical)
3. SQL
    - SQL CREATE TABLE statements (see week 12 revision lectures)
    - Outer joins
    - The SQL questions on the exam will probably be related to the A2 case study ;)
    - Can open up MySQL and run your queries to check them during the exam!
4. Relational algebra
5. Indexes (hash vs. B-tree indexing)
6. Query optimisation
7. Armstrong's Axioms and normalisation
8. ACID principles and transactions
9. NoSQL and CAP Theorem

Exam Revision

The following is part of a schema for a company's project management system.



a. Write a **MySQL DDL statement** that creates the Task table. Assume that the other three tables already exist.

b. Write a **relational algebra expression** that finds the name and description of tasks owned by employees whose last name is Williams.

Exam Revision

a.
```
CREATE TABLE Task (
  ProjectID INT NOT NULL,
  TaskID INT NOT NULL,
  TaskName VARCHAR(100) NOT NULL,
  TaskDescription MEDIUMTEXT,
  TaskOwnerID INT NOT NULL,
  TaskApproverID INT NOT NULL,
  IsTaskComplete TINYINT(1) NOT NULL,
  PRIMARY KEY (ProjectID, TaskID),
  FOREIGN KEY (ProjectID) REFERENCES Project(ProjectID),
  FOREIGN KEY (TaskOwnerID) REFERENCES Employee(EmployeeID),
  FOREIGN KEY (TaskApproverID) REFERENCES Employee(EmployeeID)
);
```

b. $\pi_{\text{TaskName, TaskDescription}}\left(\sigma_{\text{EmployeeLastName} = \text{'Williams'}}\left(\text{Task} \bowtie_{\text{TaskOwnerID} = \text{EmployeeID}} \text{Employee}\right)\right)$

Exam Revision

The City of Melbourne is developing a database system to store details of the trees within the municipality. The City's existing system records the year each tree was planted and the tree's diameter at breast height (DBH). The DBH value is updated every year, and the date of the most recent update is stored alongside the value itself. The latitude and longitude of the tree are also tracked.

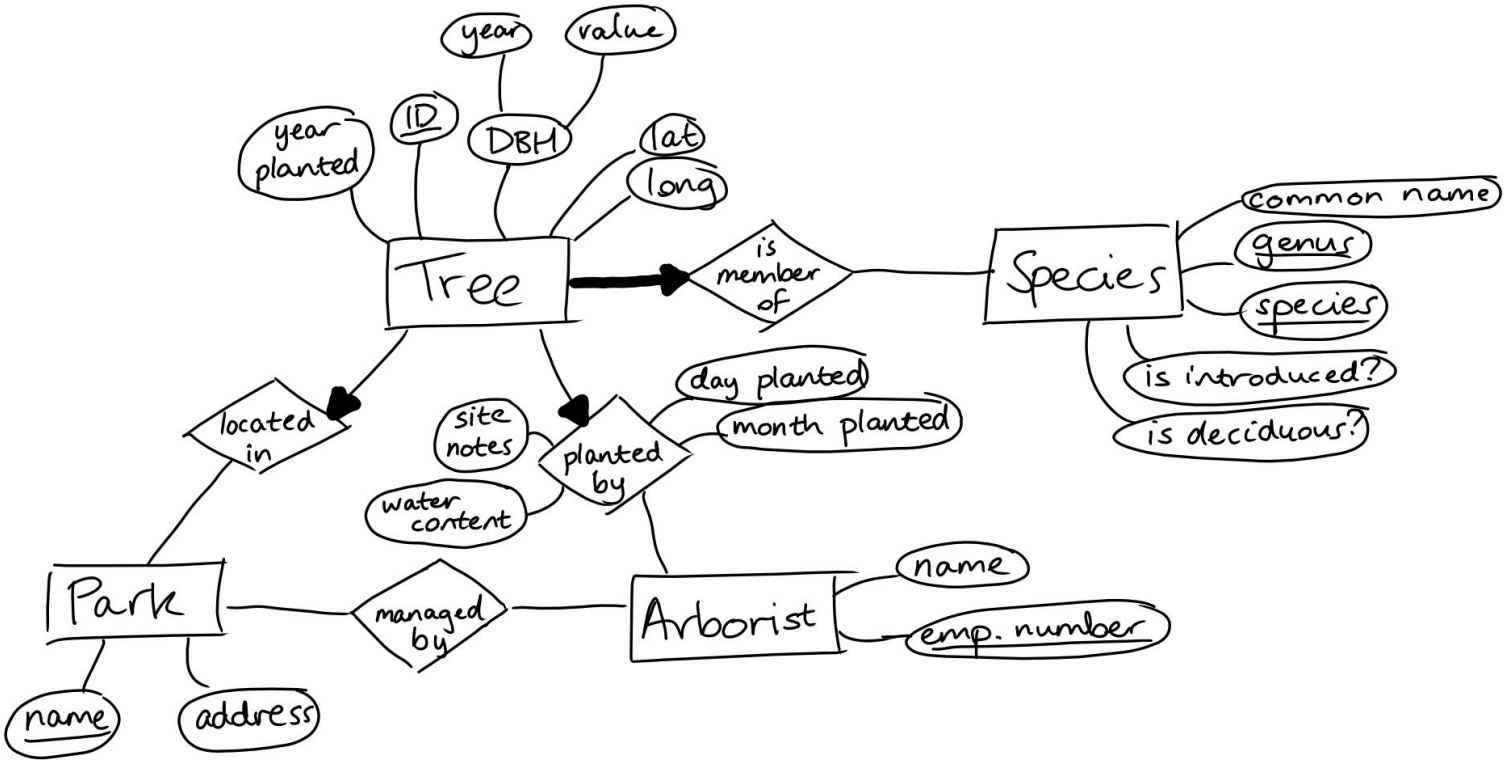
For trees that are planted after the system is implemented, the City arborist who oversaw the planting of the tree is recorded, along with the day and month they planted the tree, their notes about site conditions, and the soil water content reading taken on the day of planting.

The species of each tree needs to be stored. Each species has a common name as well as a botanical genus and species; it may be native or introduced, and it may be evergreen or deciduous.

The City manages various parks, each of which has a name and street address, and which is managed by a City arborist. The park in which each tree is located must be recorded – although some trees, such as street trees, are not situated in a park.

City arborists are known by a name and employee number.

Exam Revision



Week 12 Lab

- Canvas → Modules → Week 12 → Lab → L12 MongoDB (PDF)
- Objectives:
  - Explore the difference between an SQL and NoSQL database
  - Load data into the MongoDB instance
- Note: the mongo syntax in this lab is *not* assessable
- Breakout rooms, “ask for help” button if you need help or have any questions