INFO20003 Database Systems

Week 12

Revise on key concepts

Key Concepts:

- What are NoSQL databases?
- Types of NoSQL database
  - Graph databases
  - Key-value stores
  - Column-family stores
  - Document stores

NoSQL databases (non-relational database)

- store and retrieve non-tabular data
- use a more flexible model

- the needs of next-generation data storage and analysis, and requirements of intensive but flexible data analysis

- Examples of unstructured but exponentially-growing data:
  - chat data
  - Messaging
  - large objects such as videos, images
  - Many types of business documents

NoSQL types

- Graph databases
- Key-value stores
- Column-family stores
- Document stores

Graph databases

- Use a **graph** to store, connect, and query data
- Focus on relationships between data items

- **Nodes** are equivalent to **rows**/records
- **Edges** corresponds to the **relationship**

NoSQL types

- Graph databases
- Key-value stores
- Column-family stores
- Document stores

Key-value store

- Most flexible
- Least structured

- **Key = unique identifier**
  - Can be ANYTHING, but DBMS can impose limitations such as the data type and size

- No schema: **Value can be anything**
  - E.g. images, long text, videos, binary data, JSON data, numbers, etc

NoSQL types

- Graph databases
- Key-value stores
- Column-family stores
- Document stores

Column-family store

- also referred to as wide-column stores or extensible record stores
- a type of **key-value** database
- use tables, rows and columns
- **Columns are created for each row** instead of being predefined for a table
- schema-free structure
- Allows to store and query semi-structured data

NoSQL types

- Graph databases
- Key-value stores
- Column-family stores
- Document stores

Document stores

- Like a key-value store

- Value is a **semi-structured or structured document**
    - JSON, XML, BSON, industry-specific data formats

Q1)

Choosing a NoSQL database

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.

Q1a)

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.

- **Column-family database**
- Each row in a column-family table may have a different set of columns associated with it

Q1b)

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.

```
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.

- **Document store**
- use a modern data interchange format e.g. JSON / XML
- Industry-specific data formats such as MARC can be used with specialised document store systems

Q1c)

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.

- **Key-value stores**
- Can store any kind of data

Q1d)

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

- **Graph database**
- Store Relationships between papers
  - Papers as nodes, references as edges

Advantages of NoSQL

- Flexible modelling
- Scalability
- Performance
- High availability

Flexible modelling

- Facilitates the implementation of flexible data models

- Suited to coping with less structured data sources

Scalability

- Capacity can be added and removed quickly using a horizontal scale-out methodology

- More efficient when handling big data

Performance

- NoSQL databases are typically stored in partitions
- Users can be routed to closest data centre

High Availability

- NoSQL databases are typically stored in partitions
- if nodes fail, the database can continue its read and write operations on a different node

CAP theorem

Consistency

- All the servers hosting the database will have the same data
- Different to “consistency” in ACID

Availability

- The system will always respond to a request

Partition tolerance

- Partition = network partition
- System continues to operate even if individual servers fail/cannot be reached

CAP theorem = a system can achieve only two out of the three principles

NoSQL & CAP theorem

- the choice is between AP or CP

- AP: The database always answers, but possibly with outdated or wrong data
  - achieve eventual consistency
- CP: The database stops all the operations until the latest copy of data is available on all nodes

- Most NoSQL databases choose AP over CP

Review Questions

Review question 1: ER modelling

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.

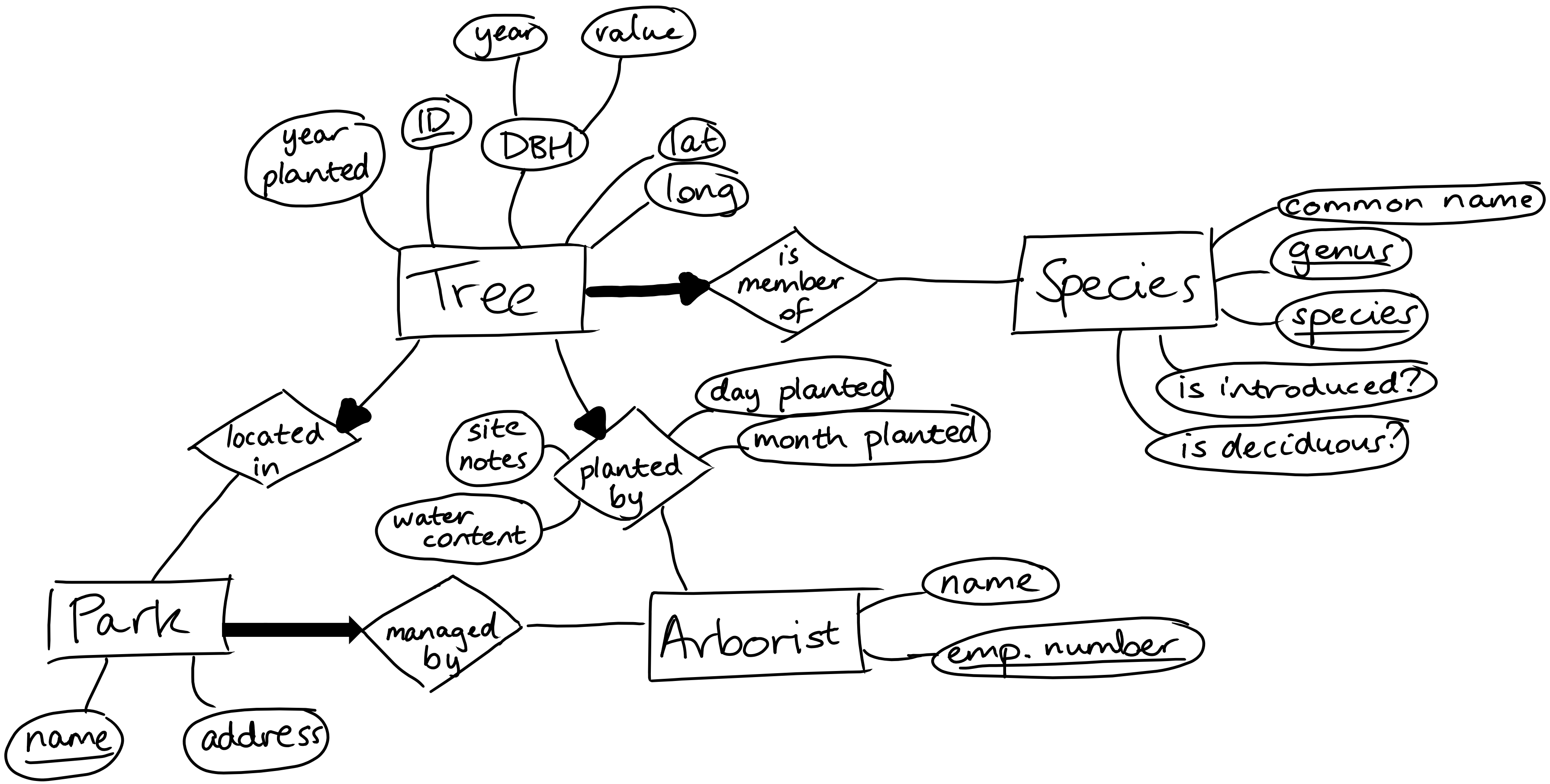
The species of each tree needs to be stored. Each species has a common name as well as a botanical genus and species; it may be native or introduced, and it may be evergreen or deciduous.

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.

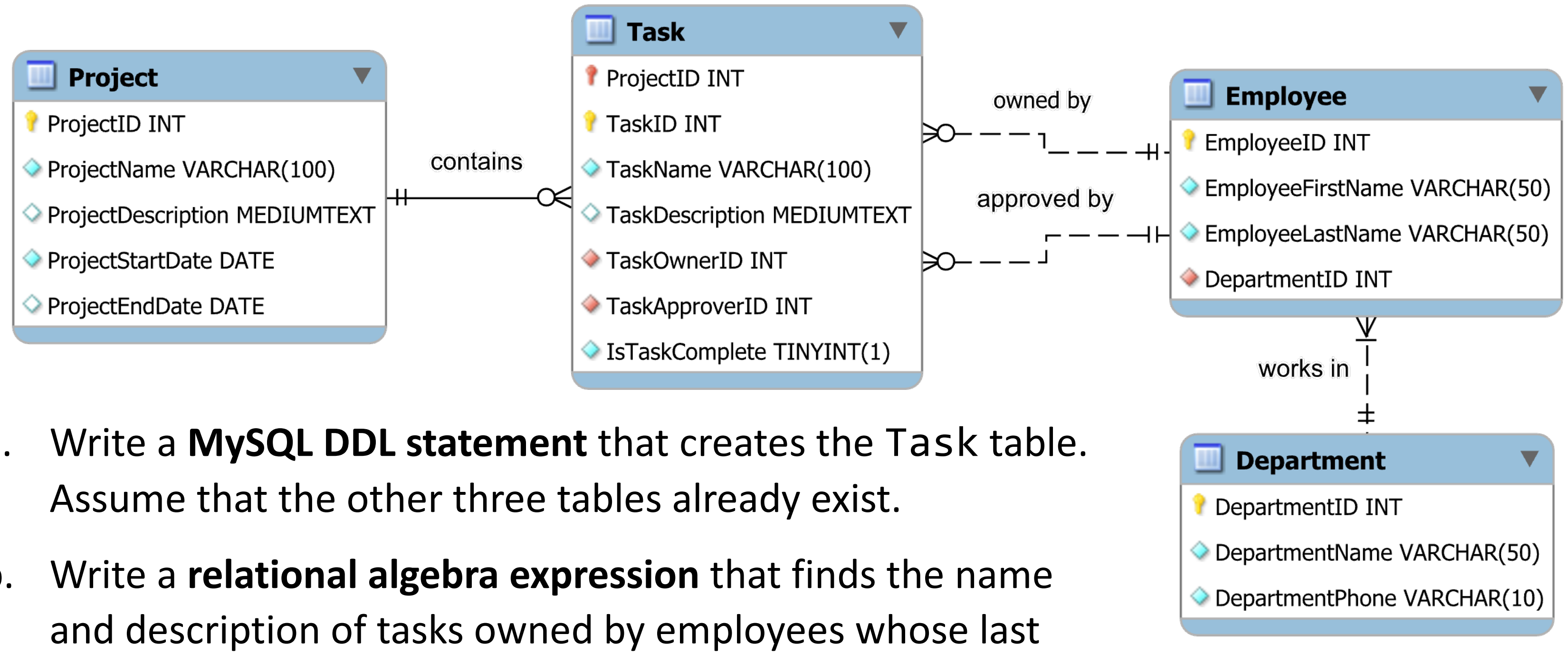
- Develop a **conceptual ER model** for this case study. You may use either Chen's or crow's-foot notation, but do not mix the two.

Suggested solution for arborist model



Review question 2: Relational algebra and SQL

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




```
CREATE TABLE Account (
  AccountID              int             auto_increment,
  AccountName            varchar(100)    NOT NULL,
  OutstandingBalance     DECIMAL(10,2)   NOT NULL,
  CustomerID             smallint        NOT NULL,
  PRIMARY KEY  (AccountID),
  FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
        ON DELETE RESTRICT
        ON UPDATE CASCADE
);
```

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.

c. Write an **SQL query** that fetches the ID, first name and last name of employees in the Costumes department.

d. Write an **SQL query** that finds the name of each department, the number of employees in that department who own at least one task, and the total number of tasks owned by those employees.

e. Write an **SQL query** to find the number of incomplete tasks approved by an employee from the Costumes department which belong to a project that started before May 2010.

Suggested solutions to SQL review questions

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

c.
```
SELECT EmployeeID, EmployeeFirstName, EmployeeLastName
FROM Department NATURAL JOIN Employee
WHERE DepartmentName = 'Costumes';
```

d.
```
SELECT DepartmentName, COUNT(DISTINCT EmployeeID), COUNT(TaskID)
FROM Department NATURAL JOIN Employee INNER JOIN Task ON TaskOwnerID = EmployeeID
  GROUP BY DepartmentID;
```

e.
```
SELECT COUNT(*)
FROM Department NATURAL JOIN Employee INNER JOIN Task ON TaskApproverID = EmployeeID
  NATURAL JOIN Project
WHERE DepartmentName = 'Costumes'
  AND IsTaskComplete = 0
  AND ProjectStartDate < '2010-05-01';
```