CSIT6000P Spatial and Multimedia Databases

2022 Spring

RDBMS EXercises

Prof Xiaofang Zhou

香港科技大學
THE HONG KONG UNIVERSITY OF
SCIENCE AND TECHNOLOGY

+ Assessment

- Two assessment components
  - Two assignments (individual): 30% + 30% → out soon
    - Marking criteria can be found in the assignment specs
  - Final report and presentation (group): 40% → now out
    - Marking criteria can be found in the project spec

- No exams

- Your final score for this course is the sum of the above components

+ DBMS

- What are the main components of a database system?

- What are the main functions of a RDBMS?

- Name some RDBMS products

+ Dimensions

- What data types can a RDBMS support?

- Compare and contrast multi-attributes data and multi-dimensional data

- Identify applications with one, several and many dimensions

- Why a RDBMS cannot support spatial data (such as points, lines, and polygons)?

- Discuss linear scan based search and tree-based search for the above datasets

+ DB Design

- Database normalization
  - 1st, 2nd and 3rd normal forms: definitions and issues
  - Join and denormalization

- Integrity constraints and consistency
  - Data type constraints, system constraints, app constraints

- Three schemas and data independence

+ Query Execution

- SQL and query execution planning

- Database transactions

+ Data Types

- ■ What data types can an RDBMS support?

- ■ What does a data type mean?

+ SQL

- SQL = DDL + DML + DCL

- DDL: CREATE, ALTER and DROP tables

- DML: SELECT, INSERT, DELETE and UPDATE

- DCL: GRANT, REVOKE

+ SELECT

SELECT   *a list of expressions*

FROM   *a list of tables*

WHERE   *a list of conditions (AND, OR)*

GROUP BY   *a list of expressions to be selected*

HAVING   *conditions defined on groups*

ORDER BY   *a list of expressions to be selected*

+ SELECT Query Example

To find the departments and their managers who are female of 30 years old or younger.

SELECT e1.firstName, UPPER(e1.lastName), d.name

FROM Department d, Employee e1, Employee e2

WHERE d.id = e.dNum AND e1.ssn = e2.mgrSSN AND

e1.sex = 'female' AND

DATEDIFF(YEAR, e1.dob, GETDATE()) < 30;

```
Employee[fName, lName, ssn, dob, add, sex, salary, mgrSSN, dNum]

Department[name, id, mgrSSN,  mgrStartDate]
```

+ SELECT With Sub-query

To find the departments with salary budget over 10M and their managers who are female of 30 years old or younger.

```
SELECT e1.firstName, UPPER(e1.lastName), d.name
FROM    Department d, Employee e1, Employee e2
WHERE  d.id = e1.dNum AND e1.ssn = e2.mgrSSN AND
        e1.sex = 'female' AND
        DATEDIFF(YEAR, e1.dob, GETDATE()) < 30 AND
        d.id = (SELECT      id
                FROM        Department, Employee
                WHERE       id = dNum
                GROUP BY id
                HAVING      SUM(salary) >= 10000000);
```

```
Employee[fName, lName, ssn, dob, add, sex, salary, mgrSSN, dNum]

Department[name, id, mgrSSN,  mgrStartDate]
```

+ QEP

- **SQL queries are declarative**

- **An SQL query will be translated into an execution plan by the QEP**
  - One SQL query can have many equivalent execution plans
  - But they can have different execution costs

- **The QEP selects the "best" execution plan**
  - Optimization can be based on cost models, or heuristics
  - QEP requires data statistics and index information
  - Common heuristics: push-down selections
  - Note the pros and cons of pushing-down projections

+ QEP Discussion

To find the departments and their managers who are female of 30 years old or younger.

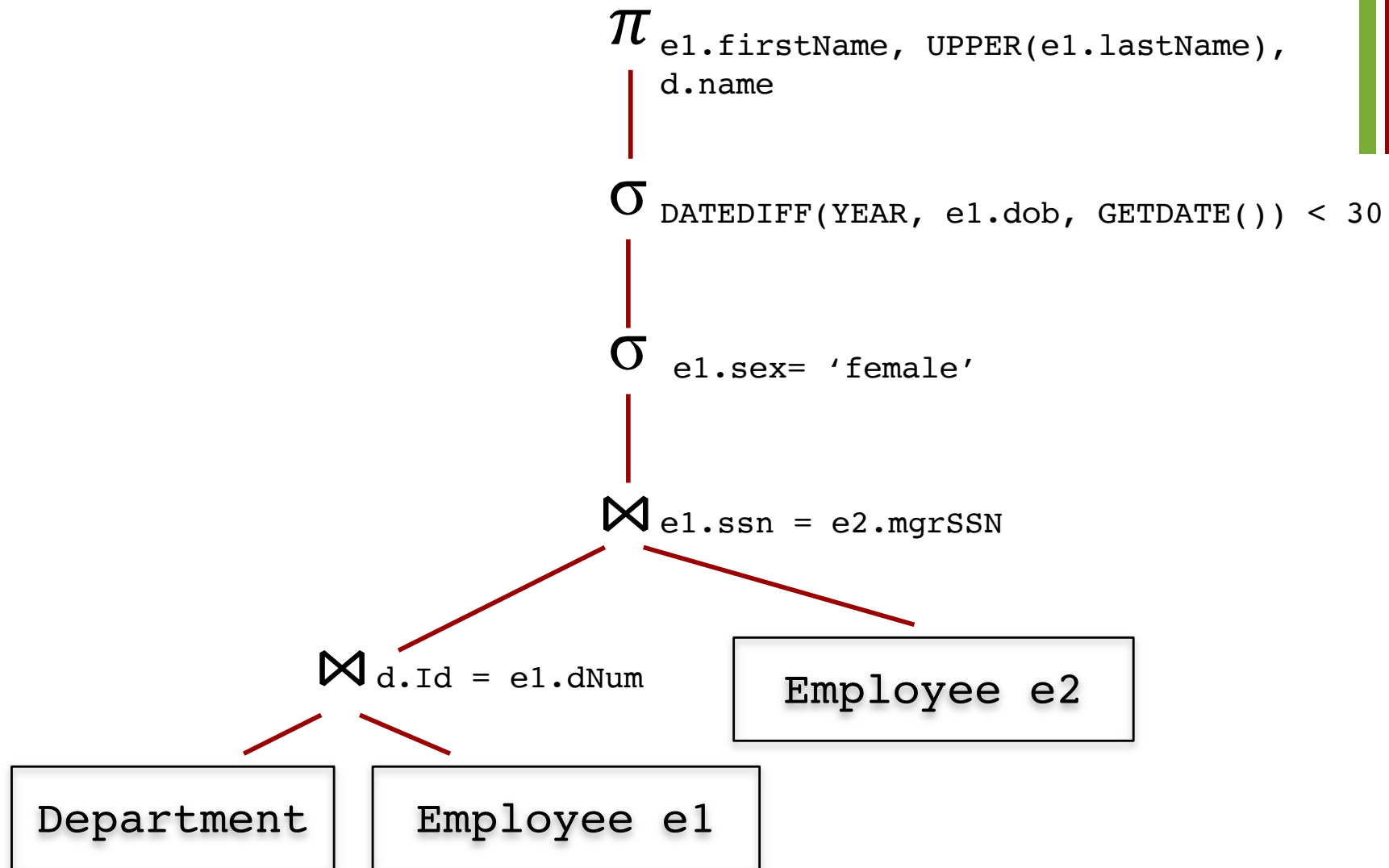**SELECT**   e1.firstName, UPPER(e1.lastName), d.name

FROM   Department d, Employee e1, Employee e2

WHERE   d.id = e1.dNum AND e1.ssn = e2.mgrSSN AND

e1.sex = 'female' AND

DATEDIFF(YEAR, e1.dob, GETDATE()) < 30;

```
Employee[fName, lName, ssn, dob, add, sex, salary, mgrSSN, dNum]

Department[name, id, mgrSSN,  mgrStartDate]
```

+ An Execution Tree: Straightforward

π e1.firstName, UPPER(e1.lastName), d.name

σ DATEDIFF(YEAR, e1.dob, GETDATE()) < 30

σ e1.sex= 'female'

$\bowtie$ e1.ssn = e2.mgrSSN

$\bowtie$ d.Id = e1.dNum

Employee e2

Department

Employee e1

Employee[fName, lName, ssn, dob, add, sex, salary, mgrSSN, dNum]

Department[name, id, mgrSSN, mgrStartDate]

+ An Execution Tree: Better

$$\pi_{\text{e1.firstName, UPPER(e1.lastName), d.name}}$$

$$\bowtie_{\text{e1.ssn = e2.mgrSSN}}$$

$$\bowtie_{\text{d.Id = e1.dNum}} \qquad \boxed{\texttt{Employee e2}}$$

$$\boxed{\texttt{Department}}$$

$$\sigma_{\text{DATEDIFF(YEAR, e1.dob, GETDATE()) < 30 AND e1.sex= 'female'}}$$
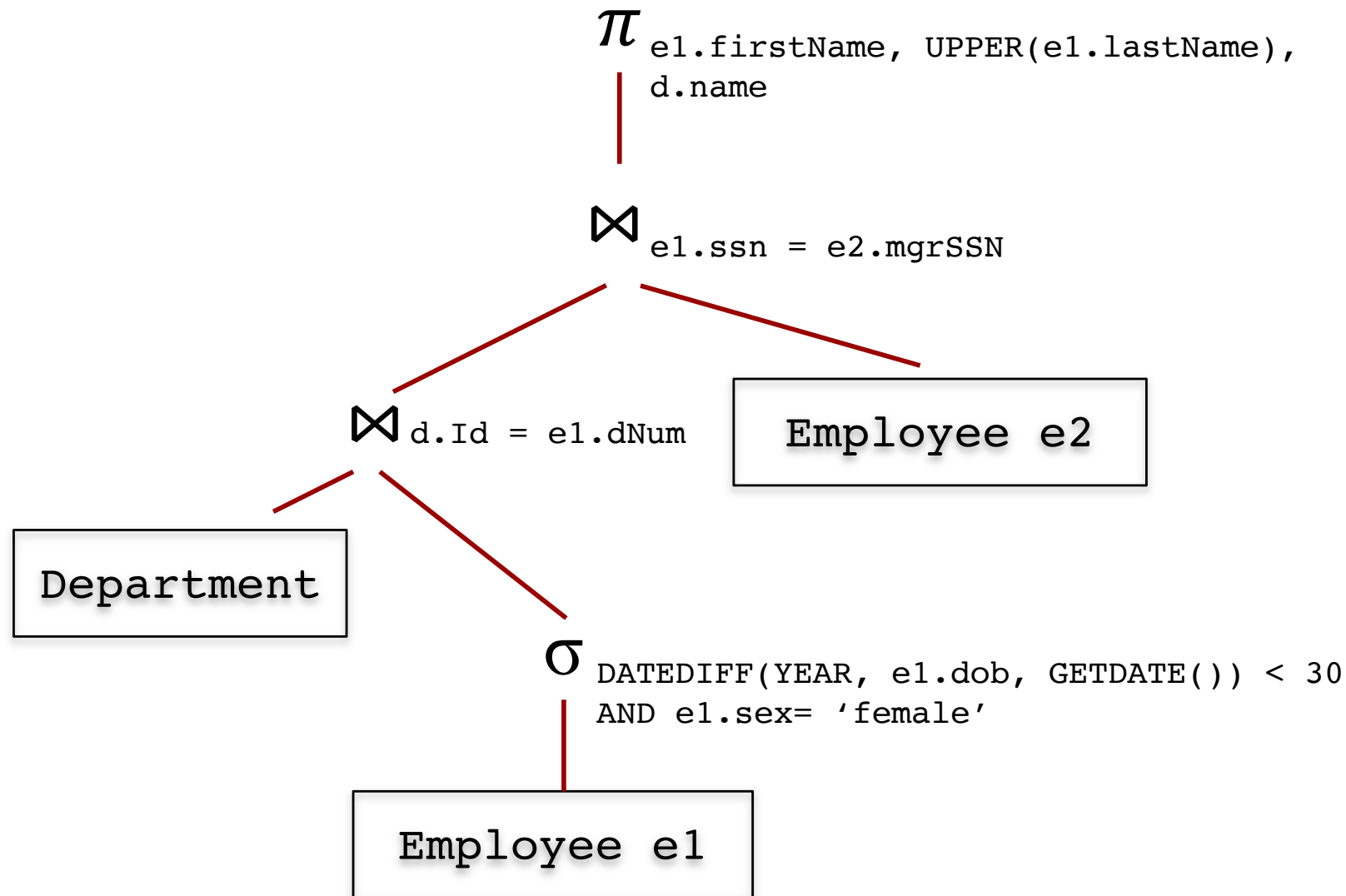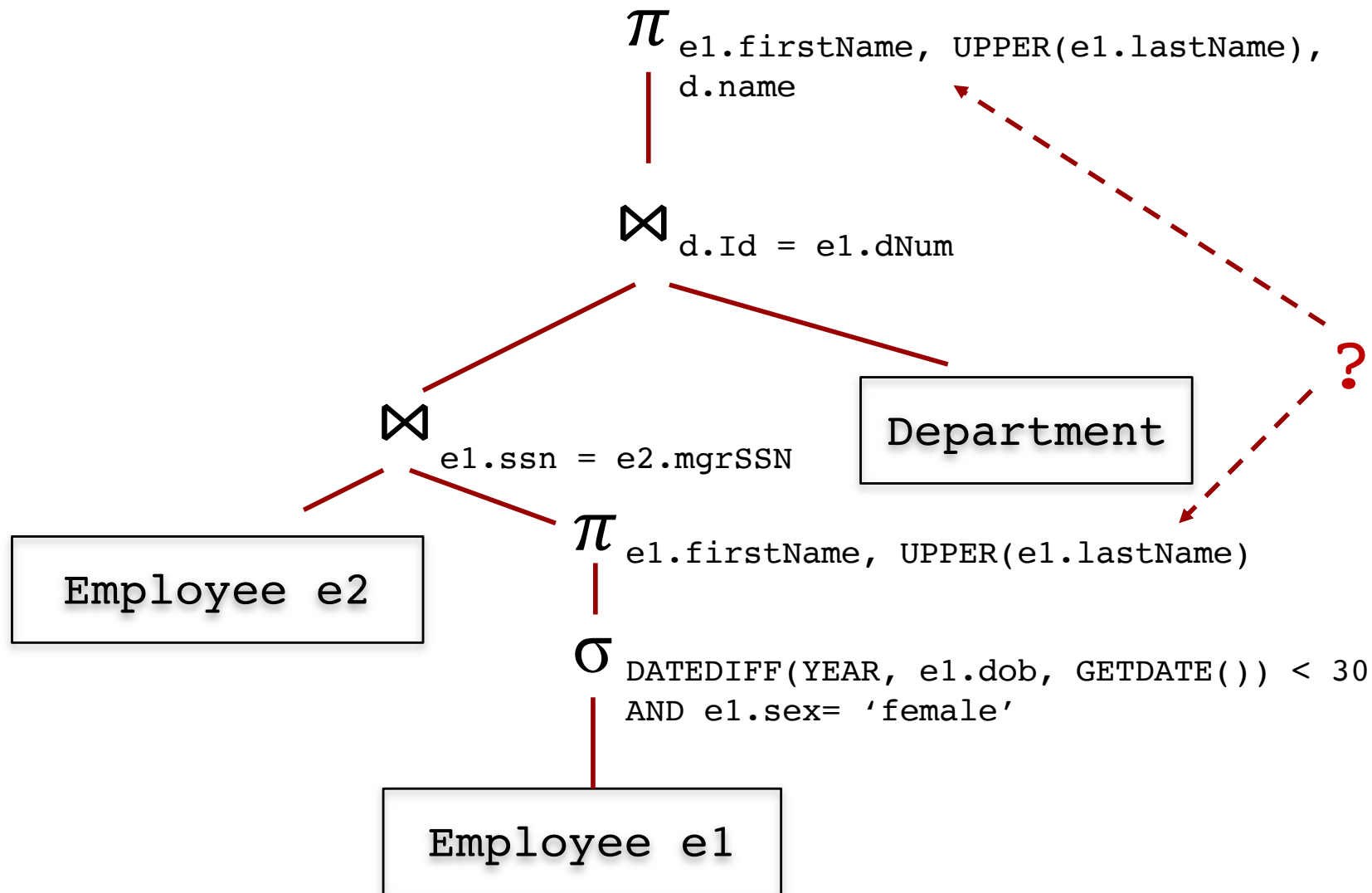
$$\boxed{\texttt{Employee e1}}$$

```
Employee[fName, lName, ssn, dob, add, sex, salary, mgrSSN, dNum]

Department[name, id, mgrSSN,  mgrStartDate]
```

+ An Execution Tree: Even Better?

π e1.firstName, UPPER(e1.lastName), d.name

$\bowtie$ d.Id = e1.dNum

$\bowtie$ e1.ssn = e2.mgrSSN

Department

?

Employee e2

π e1.firstName, UPPER(e1.lastName)

σ DATEDIFF(YEAR, e1.dob, GETDATE()) < 30 AND e1.sex= 'female'

Employee e1

Employee[fName, lName, ssn, dob, add, sex, salary, mgrSSN, dNum]

Department[name, id, mgrSSN, mgrStartDate]

+ Indexing

- B-tree and B$^+$-tree

- Bitmap index

- Hash index

+ RDBMS Summary

- **RDBMS functions**
  - Define and construct tables, data manipulation, data maintenance

- **Database design**
  - Conceptual, logical and physical design, indexing, data access control (DAC)

- **Integrity constraints**

- **SQL = DDL + DML + DCL**

- **Query optimization**

- **ACID properties**
  - Atomicity, consistency, isolation, durability