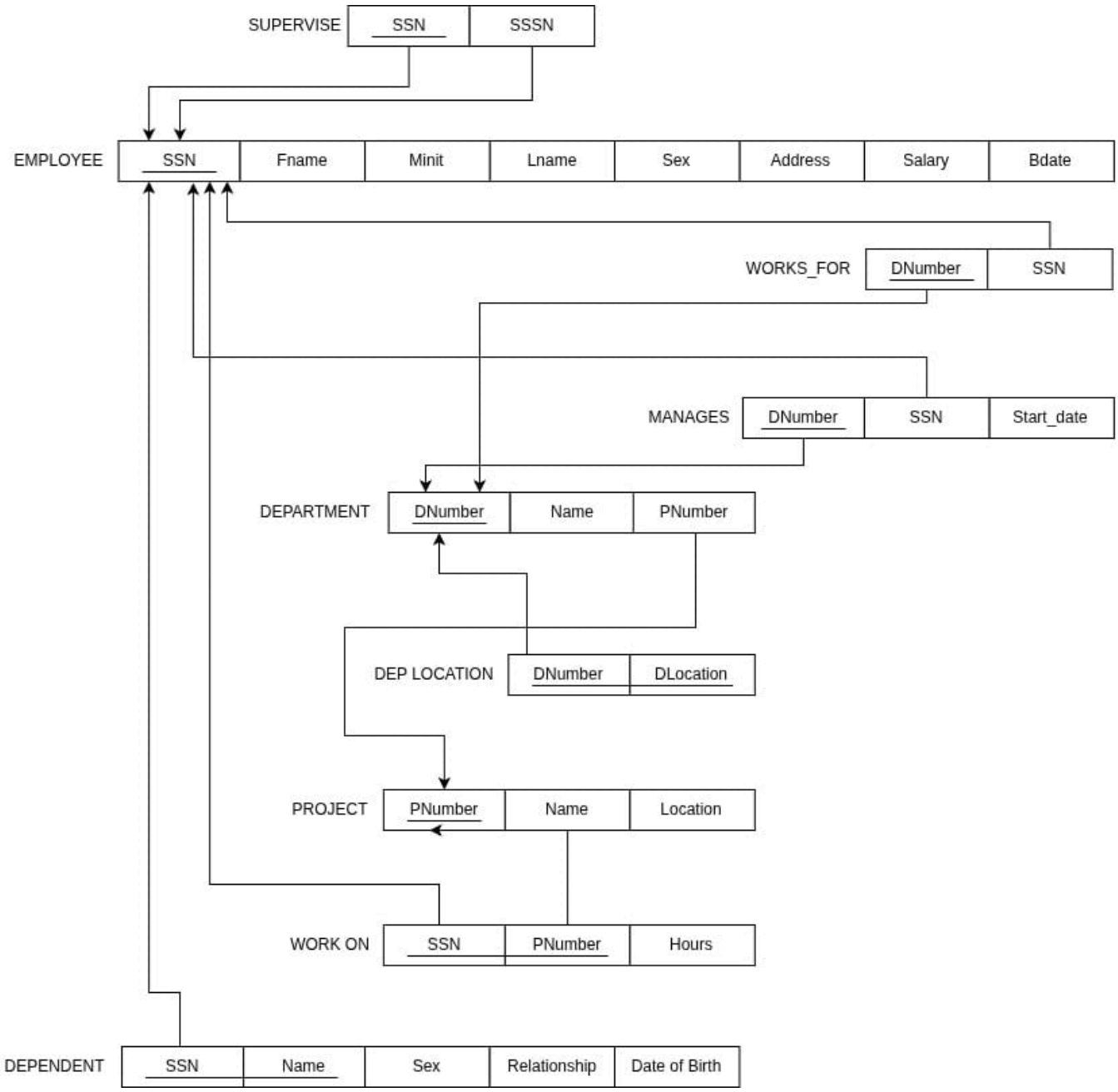
DATABASE SYSTEMS LAB

REPORT

LAB 4

Trần Đình Đăng Khoa - ID: 2211649
Lab Instructor: Võ Thị Kim Anh

Relational Diagram



Relational Algebra Expressions with SQL Equivalents

1. **Find SSN of all employees**
   - Relational Algebra: $\pi_{Ssn}(EMP)$
   - SQL:

```sql
SELECT Ssn FROM EMPLOYEE;
```

2. **Find name and address of employees working in Research department**
   - Relational Algebra: $\pi_{Fname,Lname,Address}(\sigma_{Name="Research"}(EMP \bowtie_{Dno=Number} DEPT))$

- SQL:

```sql
SELECT E.Fname, E.Lname, E.Address
FROM EMPLOYEE E, DEPARTMENT D
WHERE E.Dno = D.Number AND D.Name = 'Research';
```

3. **Find names of employees with salary over 40,000**
   - Relational Algebra: $\pi_{Fname,Lname}(\sigma_{Salary>40000}(EMP))$
   - SQL:

```sql
SELECT Fname, Lname
FROM EMPLOYEE
WHERE Salary > 40000;
```

4. **Find project names and hours worked by employees**
   - Relational Algebra: $\pi_{P.Name,W.Hours}(PROJP \bowtie_{P.Number=W.Pno} WORKSW)$
   - SQL:

```sql
SELECT P.Name, W.Hours
FROM PROJECT P, WORKS_ON W
WHERE P.Number = W.Pno;
```

5. **Find names of employees working on ProductX**
   - Relational Algebra: $\pi_{E.Fname,E.Lname}(EMPE \bowtie_{E.Ssn=W.Essn} WORKSW \bowtie_{W.Pno=P.Number} \sigma_{Name="ProductX"}(PROJP))$
   - SQL:

```sql
SELECT E.Fname, E.Lname
FROM EMPLOYEE E, WORKS_ON W, PROJECT P
WHERE E.Ssn = W.Essn AND W.Pno = P.Number AND P.Name = 'ProductX';
```

6. **Find names of employees who manage projects they work on**
   - Relational Algebra: $\pi_{E.Fname,E.Lname}(EMPE \bowtie_{E.Ssn=W.Essn} WORKSW \bowtie_{W.Pno=P.Number} \sigma_{P.Dnum=D.Number \wedge D.Mgr\_ssn=}$
   - SQL:

```sql
SELECT DISTINCT E.Fname, E.Lname
FROM EMPLOYEE E, WORKS_ON W, PROJECT P, DEPARTMENT D
```

```
WHERE E.Ssn = W.Essn AND W.Pno = P.Number AND P.Dnum = D.Number AND
D.Mgr_ssn = E.Ssn;
```

7. **Find names of employees not working on any project**
   - Relational Algebra: $\pi_{Fname,Lname}(EMP) - \pi_{Fname,Lname}(EMP \bowtie_{Ssn=Essn} WORKS)$
   - SQL:

```
SELECT Fname, Lname
FROM EMPLOYEE
WHERE Ssn NOT IN (SELECT Essn FROM WORKS_ON);
```

8. **Find employees in Research or with salary over 40,000**
   - Relational Algebra:
     $(\pi_{Fname,Lname}(\sigma_{Name="Research"}(EMP \bowtie_{Dno=Number} DEPT))) \cup (\pi_{Fname,Lname}(\sigma_{Salary>40000}(EMP)$
   - SQL:

```
SELECT E.Fname, E.Lname
FROM EMPLOYEE E, DEPARTMENT D
WHERE E.Dno = D.Number AND D.Name = 'Research'
UNION
SELECT Fname, Lname
FROM EMPLOYEE
WHERE Salary > 40000;
```

9. **Find employees in Research with salary over 40,000**
   - Relational Algebra: $\pi_{Fname,Lname}(\sigma_{Name="Research" \wedge Salary>40000}(EMP \bowtie_{Dno=Number} DEPT))$
   - SQL:

```
SELECT E.Fname, E.Lname
FROM EMPLOYEE E, DEPARTMENT D
WHERE E.Dno = D.Number AND D.Name = 'Research' AND E.Salary > 40000;
```

10. **Find names of employees with dependent children**
    - Relational Algebra: $\pi_{Fname,Lname}(EMP \bowtie_{Ssn=Essn} \sigma_{Relationship="Child"}(DEP))$
    - SQL:

```
SELECT DISTINCT E.Fname, E.Lname
FROM EMPLOYEE E, DEPENDENT D
WHERE E.Ssn = D.Essn AND D.Relationship = 'Child';
```

11. **Find names of employees with a spouse as dependent**
    - Relational Algebra: $\pi_{Fname,Lname}(EMP \bowtie_{Ssn=Essn} \sigma_{Relationship="Spouse"}(DEP))$
    - SQL:

```sql
SELECT DISTINCT E.Fname, E.Lname
FROM EMPLOYEE E, DEPENDENT D
WHERE E.Ssn = D.Essn AND D.Relationship = 'Spouse';
```

12. **Find employees with at least 2 dependents**
    - Relational Algebra:
      $\pi_{Fname,Lname}(EMP \bowtie_{Ssn=Essn} \gamma_{Essn,COUNT(*) \text{ as NumDeps}}(DEP) \bowtie \sigma_{NumDeps>=2})$
    - SQL:

```sql
SELECT E.Fname, E.Lname
FROM EMPLOYEE E
WHERE 2 <= (SELECT COUNT(*) FROM DEPENDENT D WHERE D.Essn = E.Ssn);
```

13. **Find department name where John Smith works**
    - Relational Algebra: $\pi_{Name}(DEPT \bowtie_{Number=Dno} \sigma_{Fname="John" \wedge Lname="Smith"}(EMP))$
    - SQL:

```sql
SELECT D.Name
FROM DEPARTMENT D, EMPLOYEE E
WHERE D.Number = E.Dno AND E.Fname = 'John' AND E.Lname = 'Smith';
```

14. **Find names of direct supervisors of Franklin Wong**
    - Relational Algebra:
      $\pi_{S.Fname,S.Lname}(\sigma_{E.Fname="Franklin" \wedge E.Lname="Wong"}(EMPE) \bowtie_{E.Super\_ssn=S.Ssn} EMPS)$
    - SQL:

```sql
SELECT S.Fname, S.Lname
FROM EMPLOYEE E, EMPLOYEE S
WHERE E.Super_ssn = S.Ssn AND E.Fname = 'Franklin' AND E.Lname =
'Wong';
```

15. **List employees who directly or indirectly supervise Franklin Wong**
    - Relational Algebra: The transitive closure is not directly representable in basic relational algebra. Using recursive SQL:
    - SQL:

```sql
WITH RECURSIVE SupervisorChain AS (
    SELECT Super_ssn AS SupervisorSSN
    FROM EMPLOYEE
    WHERE Fname = 'Franklin' AND Lname = 'Wong'
    UNION ALL
    SELECT E.Super_ssn
    FROM EMPLOYEE E JOIN SupervisorChain S ON E.Ssn = S.SupervisorSSN
    WHERE E.Super_ssn IS NOT NULL
)
SELECT DISTINCT E.Fname, E.Lname
FROM EMPLOYEE E JOIN SupervisorChain SC ON E.Ssn = SC.SupervisorSSN;
```

16. **Find employees with December birthdays**
    - Relational Algebra: $\pi_{Fname,Lname}(\sigma_{MONTH(Bdate)=12}(EMP))$
    - SQL:

```sql
SELECT Fname, Lname
FROM EMPLOYEE
WHERE EXTRACT(MONTH FROM Bdate) = 12;
```

17. **Find departments with at least 3 employees**
    - Relational Algebra:
      $\pi_{D.Name}(DEPTD \bowtie_{D.Number=E.Dno} \gamma_{Dno,COUNT(*) \text{ as NumEmps}}(EMPE) \bowtie \sigma_{NumEmps>=3})$
    - SQL:

```sql
SELECT D.Name
FROM DEPARTMENT D
WHERE 3 <= (SELECT COUNT(*) FROM EMPLOYEE E WHERE E.Dno = D.Number);
```

18. **Find projects with no employees working on them**
    - Relational Algebra: $\pi_{Name}(PROJ) - \pi_{Name}(PROJ \bowtie_{Number=Pno} WORKS)$
    - SQL:

```sql
SELECT Name
FROM PROJECT
WHERE Number NOT IN (SELECT Pno FROM WORKS_ON);
```

19. **Find employees with the highest salary**
    - Relational Algebra:
      $\pi_{Fname,Lname}(EMP) - \pi_{Fname,Lname}(EMPE1 \bowtie_{E1.Salary<E2.Salary} EMPE2)$

- SQL:

```
SELECT Fname, Lname
FROM EMPLOYEE
WHERE Salary = (SELECT MAX(Salary) FROM EMPLOYEE);
```

20. **Find employees working on all projects**
    - Relational Algebra:
      $\pi_{Fname,Lname}(EMPE)$ WHERE NOT EXISTS $((\pi_{Number}(PROJ)) - \pi_{Pno}(\sigma_{Essn=E.Ssn}(WORKS)$
    - SQL:

```
SELECT E.Fname, E.Lname
FROM EMPLOYEE E
WHERE NOT EXISTS (
    SELECT P.Number FROM PROJECT P
    WHERE NOT EXISTS (
        SELECT * FROM WORKS_ON W
        WHERE W.Essn = E.Ssn AND W.Pno = P.Number
    )
);
```

Analysis of Relational Algebra vs SQL

Advantages of Relational Algebra:

- Provides a formal theoretical foundation for relational database operations
- More concise representation of operations
- Helpful for query optimization and analysis

Advantages of SQL:

- More readable and accessible for non-technical users
- Includes additional capabilities not easily expressed in relational algebra (recursive queries, aggregations)
- Industry standard with broad implementation support

Differences:

- Relational algebra is a mathematical notation while SQL is a programming language
- SQL includes built-in functions (EXTRACT, COUNT) that must be expressed differently in relational algebra

- Complex operations like recursive relationships (query 15) are difficult to express in basic relational algebra