DATABASE SYSTEMS

Dr. Noha Nagy

Lecture 11

Subqueries

3- Exist Subquery Operators

Operator	Meaning
Exists	Test the existence of any record in the subquery
Not Exists	The opposite of exists

Exists

- **The EXISTS operator returns true if the subquery returns one or more records.**
- Supplier(SupplierName,SupplierID)
- Products(ProductName,Price,ProductID,SupplierID)
- Get the suppliername who supply products with price<20

```
SELECT SupplierName
FROM Suppliers
WHERE EXISTS (SELECT ProductName FROM Products WHERE
Products.SupplierID = Suppliers.supplierID
AND Price < 20);
```

Example

- □ Get the names and department number of employees who earn more than 3500.”use subquery”

```
SELECT ename, deptno
FROM emp as emp1
WHERE EXISTS
            (SELECT *
            FROM emp as emp2
            WHERE sal >3500 and emp2.id=emp1.id)
```

EXISTS

Get the employee’s name that has a first name same as his dependent

```
SELECT FNAME, LNAME
        FROM EMPLOYEE
        WHERE EXISTS (SELECT *
                FROM DEPENDENT
                WHERE SSN=ESSN AND
                FNAME=DEPENDENT_NAME)
```

Example

- Write a query to display the employee’s name and hiredate for all employees in the same department as Blake. Exclude Blake.
- Emp(eid, ename, hiredate, deptno)

Answer with subquery

```
SELECT ename, hiredate
      FROM emp
      WHERE deptno =  ( SELECT deptno
                               FROM emp
                        WHERE ename = 'BLAKE')
AND  ename <> 'BLAKE';
```

Answer with join

```
□ SELECT emp1.ename, emp2.hiredate
        FROM emp as emp1, emp as emp2
        WHERE emp1.deptno =  emp2.deptno
                             and  emp2.ename = 'BLAKE'
                             and  emp1.ename <> 'BLAKE';
```

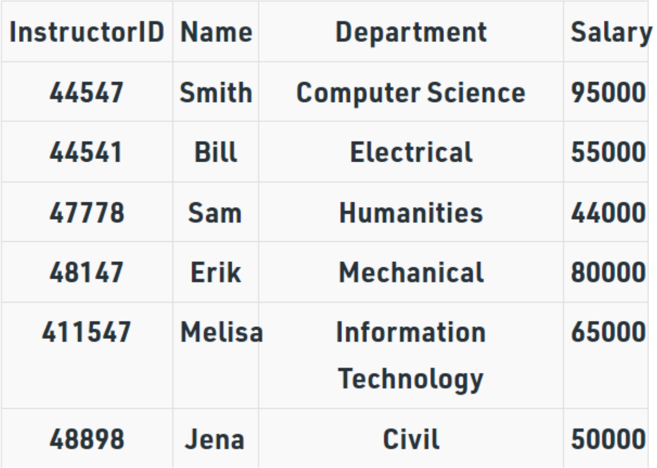
Subquery in From Clause

- □ From clause can be used to specify a sub-query expression in SQL. The relation produced by the sub-query is then used as a new relation on which the outer query is applied.

Example

- □ Find all professors whose salary is greater than 3000

- □ Select * from instructor
- □ Where salary >3000

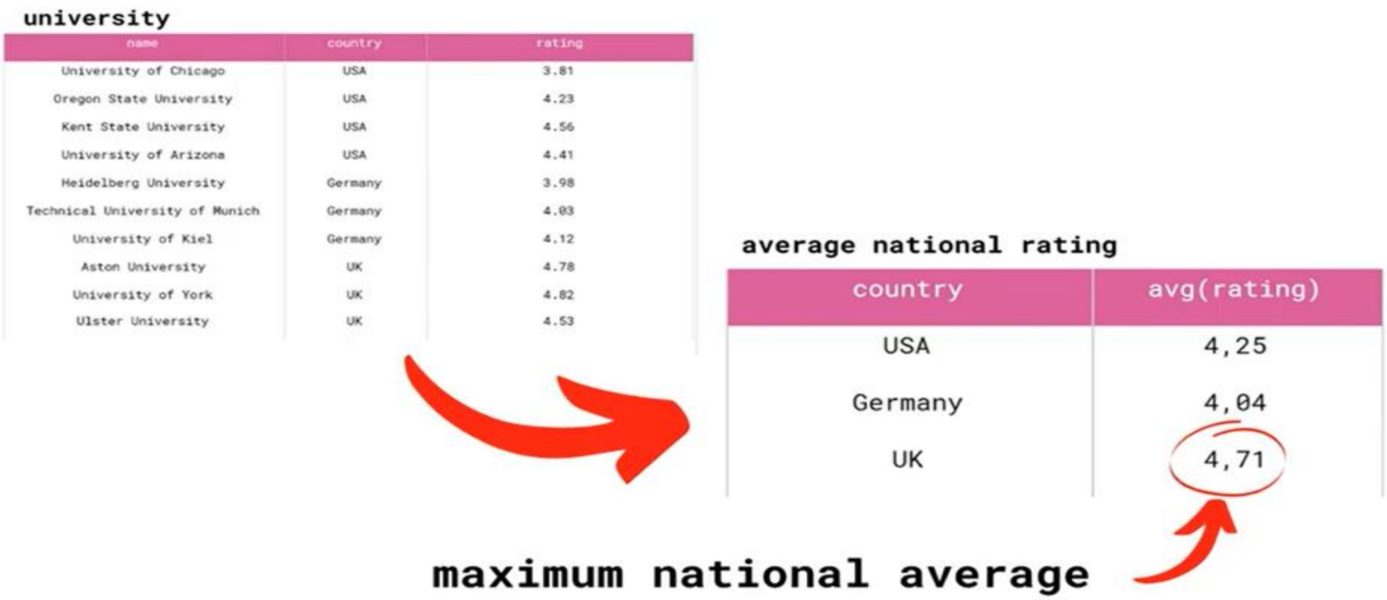


InstructorID	Name	Department	Salary
44547	Smith	Computer Science	95000
44541	Bill	Electrical	55000
47778	Sam	Humanities	44000
48147	Erik	Mechanical	80000
411547	Melisa	Information Technology	65000
48898	Jena	Civil	50000

- □ Select Name from (select * from instructor where salary >3000)

Example

- Get the maximum national average rating of universities



university

name	country	rating
University of Chicago	USA	3.81
Oregon State University	USA	4.23
Kent State University	USA	4.56
University of Arizona	USA	4.41
Heidelberg University	Germany	3.98
Technical University of Munich	Germany	4.03
University of Kiel	Germany	4.12
Aston University	UK	4.78
University of York	UK	4.82
Ulster University	UK	4.53

average national rating

country	avg(rating)
USA	4,25
Germany	4,04
UK	4,71

maximum national average

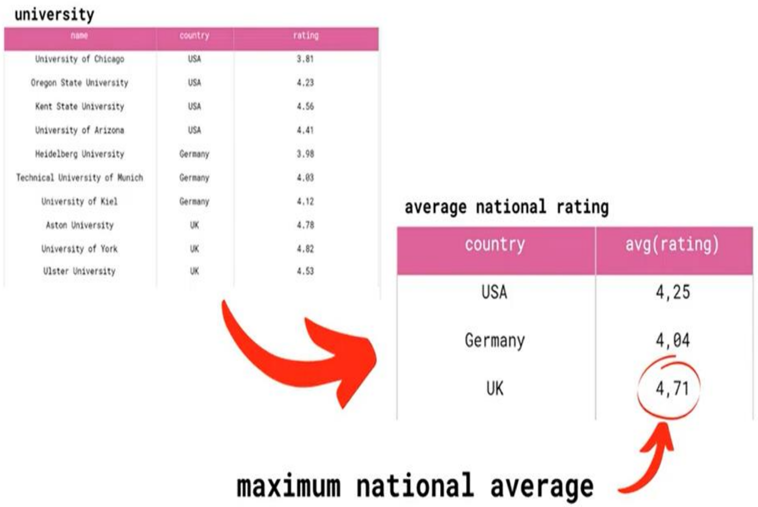
- Select max(avg(rating))

  From university

Not possible

Example

- □ Get the maximum national average rating of universities



university

name	country	rating
University of Chicago	USA	3.81
Oregon State University	USA	4.23
Kent State University	USA	4.56
University of Arizona	USA	4.41
Heidelberg University	Germany	3.98
Technical University of Munich	Germany	4.03
University of Kiel	Germany	4.12
Aston University	UK	4.78
University of York	UK	4.82
Ulster University	UK	4.53

average national rating

country	avg(rating)
USA	4,25
Germany	4,04
UK	4,71

maximum national average

- □ Select max(avg-rating)

From (select country, avg(rating) as avg-rating from university

group by country)

Plz study divide operation in SQL

Relational Algebra

Relational Query Languages

- Languages for describing queries on a relational database
- *Structured Query Language* (SQL)
  - Predominant application-level query language
  - Declarative
- *Relational Algebra*
  - Intermediate language used within DBMS
  - Procedural

Relational Algebra Operations

- Unary Operations
  - Selection σ (sigma))
  - Projection π (pi))
  - Rename ρ (rho))
- Binary Operations
  - Union ∪
  - Intersection ∩
  - Set difference –
  - Cartesian product x

Select Operator

- Produce table containing subset of **rows** of argument table satisfying condition

  $\sigma_{condition}$ *relation*

- Example:

Person

Id	*Name*	*Address*	*Hobby*
1123	John	123 Main	stamps
1123	John	123 Main	coins
5556	Mary	7 Lake Dr	hiking
9876	Bart	5 Pine St	stamps

$\sigma_{Hobby=\text{‘stamps’}}$(Person)

Id	*Name*	*Address*	*Hobby*
1123	John	23 Main	stamps
9876	Bart	Pine St	stamps

Selection Condition - Examples

- □ $\sigma_{\textit{Id>3000} \text{ Or } \textit{Hobby='hiking'}}$ (Person)
- □ $\sigma_{\textit{Id>3000} \text{ AND } \textit{Id <3999}}$ (Person)
- □ $\sigma_{\text{NOT}\textit{(Hobby='hiking')}}$ (Person)
- □ $\sigma_{\textit{Hobby}\neq\textit{'hiking'}}$ (Person)

STUDENT

ST-ID	Name	Address	Major	GPA
123	Ali	Dokki	EE	3.2
456	Maha	Nasr City	CE	1.9
789	Ahmad	Haram	Arch	2.7
341	Noha	Dokki	EE	1.0

$$\sigma_{\text{Address = “Dokki” and Major= “EE”}}(\text{STUDENT})$$

ST-ID	Name	Address	Major	GPA
123	Ali	Dokki	EE	3.2
341	Noha	Dokki	EE	1.0

STUDENT

ST-ID	Name	Address	Major	GPA
123	Ali	Dokki	EE	3.2
456	Maha	Nasr City	CE	1.9
789	Ahmad	Dokki	Arch	2.7
341	Noha	Dokki	EE	1.0

$$\sigma_{\text{Address = “Dokki” or Major= “EE”}}(\text{STUDENT})$$

ST-ID	Name	Address	Major	GPA
123	Ali	Dokki	EE	3.2
789	Ahmad	Dokki	Arch	2.7
341	Noha	Dokki	EE	1.0

Unary Relational Operations: SELECT (contd.)

- SELECT Operation Properties
  - SELECT σ is commutative:
    - $\sigma_{<condition1>}(\sigma_{< condition2>} (R)) = \sigma_{<condition2>} (\sigma_{< condition1>} (R))$
  - Because of commutativity property, a cascade (sequence) of SELECT operations may be applied in **any order:**
    - $\sigma_{<cond1>}(\sigma_{<cond2>} (\sigma_{<cond3>} (R)) = \sigma_{<cond2>} (\sigma_{<cond3>} (\sigma_{<cond1>} (R)))$
  - A cascade of SELECT operations may be replaced by a single selection with a conjunction of all the conditions:

    $\sigma_{<cond1>}(\sigma_{< cond2>} (\sigma_{<cond3>}(R)) = \sigma_{<cond1> \text{ AND } < cond2> \text{ AND } < cond3>}(R)))$

Unary Relational Operations: PROJECT

- PROJECT Operation is denoted by π (pi)
- This operation keeps certain *columns* (attributes) from a relation and discards the other columns.
- Example: To list each employee's first and last name and salary, the following is used:

$\pi_{\text{LNAME, FNAME,SALARY}}(\text{EMPLOYEE})$

Project Operator

Produces table containing subset of columns of □ argument table

$\Pi_{\mathit{attribute\ list}}(\mathit{relation})$

Example: □

Person

Id	*Name*	*Address*	*Hobby*
1123	John	123 Main	stamps
1123	John	123 Main	coins
5556	Mary	7 Lake Dr	hiking
9876	Bart	5 Pine St	stamps

$\Pi_{\mathit{Name,Hobby}}$(**Person**)

Name	*Hobby*
John	stamps
John	coins
Mary	hiking
Bart	stamps

PROJECT Operation π

STUDENT

ST-ID	Name	Address	Major	GPA
123	Ali	Dokki	EE	3.2
456	Maha	Nasr City	CE	1.9
789	Ahmad	Haram	Arch	2.7
341	Noha	Dokki	EE	1.0

$\pi_{\text{ST-ID, Major}}$ (STUDENT)

ST-ID	Major
123	EE
456	CE
789	Arch
341	EE

π_{Major} (STUDENT)

Major
EE
CE
Arch
~~EE~~