CSC375 / BIF375
Database Management Systems

Chapter 6
The Database Language SQL
Part 2 – More Advanced Queries



Outline

- Joins
- Subqueries

Part A

Joins

Joins (product)

- We had two main types of joins between tables in relational algebra:
  - Cartesian product (x)
  - Natural join (⋈)

Cartesian product in SQL

```
SELECT <list of attributes>
FROM R,S
WHERE <list of conditions>
```

- When we use a **comma** between relation names in the “FROM” clause, it automatically performs a **Cartesian product (x)** between relation R and relation S.

- We can include more than two tables if we want.
  - Example: `FROM R,S,T`

Example

Students

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

Roster

ID	major	absence
123	EE	4
134	EE	7
1345	CCE	2
12345	ME	4

```
SELECT *
FROM students, roster
```

→ Result

name	lastname	age	ID	ID	major	absence
rim	ismail	19	123	123	EE	4
sara	kamel	18	134	123	EE	4
milad	ghantous	21	1345	123	EE	4
tarek	dib	18	12345	123	EE	4
rim	ismail	19	123	134	EE	7
sara	kamel	18	134	134	EE	7
milad	ghantous	21	1345	134	EE	7
tarek	dib	18	12345	134	EE	7
rim	ismail	19	123	1345	CCE	2
sara	kamel	18	134	1345	CCE	2
milad	ghantous	21	1345	1345	CCE	2
tarek	dib	18	12345	1345	CCE	2
rim	ismail	19	123	12345	ME	4
sara	kamel	18	134	12345	ME	4
milad	ghantous	21	1345	12345	ME	4
tarek	dib	18	12345	12345	ME	4

- Since students has 4 rows and roster has 4 rows, the product will have 16 rows

Example

In the previous example, we got **two** “ID” attributes, One from each table

name	lastname	age	ID	ID	major	absence
rim	ismail	19	123	123	EE	4

- We can refer to them using the dot operator: **`students.ID`** and **`roster.ID`**
- We can also rename them in the resulting table using the **`AS`** keyword**.**
- **Look next.**

Example

Students

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

Roster

ID	major	absence
123	EE	4
134	EE	7
1345	CCE	2
12345	ME	4

```
SELECT students.ID AS I1, roster.ID AS I2
FROM students, roster
```

Result

I1	I2
123	123
134	123
1345	123
12345	123
123	134
134	134
1345	134
12345	134
123	1345
134	1345
1345	1345
12345	1345
123	12345
134	12345
1345	12345
12345	12345

Example

Students

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

Roster

ID	major	absence
123	EE	4
134	EE	7
1345	CCE	2
12345	ME	4

```
SELECT *
FROM students, roster
WHERE name='rim'
```

Result

name	lastname	age	ID	ID	major	absence
rim	ismail	19	123	123	EE	4
rim	ismail	19	123	134	EE	7
rim	ismail	19	123	1345	CCE	2
rim	ismail	19	123	12345	ME	4

Let's analyze this query

```
SELECT *
FROM roster, students
WHERE roster.ID = students.ID
```

Q: What are we trying to do here?

Answer: Natural join

Students

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

Roster

ID	major	absence
123	EE	4
134	EE	7
1345	CCE	2
12345	ME	4

Result

ID	major	absence	name	lastname	age	ID
123	EE	4	rim	ismail	19	123
134	EE	7	sara	kamel	18	134
1345	CCE	2	milad	ghantous	21	1345
12345	ME	4	tarek	dib	18	12345

Natural join

- SQL provides a quick way to perform natural join
- Keyword: **`NATURAL JOIN`**
- It will automatically perform a product of two tables if the attributes having the same name are equal in both tables.

Example on `natural join`

Students

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

Roster

ID	major	absence
123	EE	4
134	EE	7
1345	CCE	2
12345	ME	4

```
SELECT *
FROM students NATURAL JOIN roster
```

Result

ID	name	lastname	age	major	absence
123	rim	ismail	19	EE	4
134	sara	kamel	18	EE	7
1345	milad	ghantous	21	CCE	2
12345	tarek	dib	18	ME	4

- Notice that `ID` is the common attribute and hence it appears only once in the result.
- Natural join is smart
- Each row makes sense by presenting the data of each student from both tables

Now let’s use it in something useful

- **Task**: **Get the major and number of absences of rim Ismail**.

- **Approach**:
  - Major and absence are in the roster table but roster table doesn’t have the name and last name
  - We must join the two tables so we have a row containing all the data we need.

Solution

- **Task**: Get the major and number of absences of rim Ismail.

Students

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

Roster

ID	major	absence
123	EE	4
134	EE	7
1345	CCE	2
12345	ME	4

```
SELECT major, absence
FROM students NATURAL JOIN roster
WHERE name='rim'
      AND lastname='ismail'
```

Result

major	absence
EE	4

Self-join

Students

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

- **Task:** Get all students who have the same age. Output should look like a pair <student1, student2>

- **Approach**:
  - Since vertical comparison is not available in SQL, we must do product of the table with itself → Self join
  - On one row, we will have data from two students.
  - Choose rows that have different IDs but same age → we get students who have the same age

Self-join

- **Task:** Get all students who have the same age. Output should look like a pair <student1, student2>

Students

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

- **Solution**

```
SELECT s1.name, s2.name
FROM students s1, students s2
WHERE s1.ID != S2.ID AND
s1.age = s2.age
```

we are giving the table temporary names s1 and s2

Result

name	name
tarek	sara
sara	tarek

Let’s analyze what happened.

- The self join result without conditions gives this:

name	lastname	age	ID	name	lastname	age	ID
rim	ismail	19	123	rim	ismail	19	123
sara	kamel	18	134	rim	ismail	19	123
milad	ghantous	21	1345	rim	ismail	19	123
tarek	dib	18	12345	rim	ismail	19	123
rim	ismail	19	123	sara	kamel	18	134
sara	kamel	18	134	sara	kamel	18	134
milad	ghantous	21	1345	sara	kamel	18	134
tarek	dib	18	12345	sara	kamel	18	134
rim	ismail	19	123	milad	ghantous	21	1345
sara	kamel	18	134	milad	ghantous	21	1345
milad	ghantous	21	1345	milad	ghantous	21	1345
tarek	dib	18	12345	milad	ghantous	21	1345
rim	ismail	19	123	tarek	dib	18	12345
sara	kamel	18	134	tarek	dib	18	12345
milad	ghantous	21	1345	tarek	dib	18	12345
tarek	dib	18	12345	tarek	dib	18	12345

Let’s analyze what happened.

- The self join result without conditions gives this:

name	lastname	age	ID	name	lastname	age	ID
tarek	dib	18	12345	sara	kamel	18	134
sara	kamel	18	134	tarek	dib	18	12345

- When we added the conditions s1.id!=s2.id and s1.age = s2.age

- Then project only names

name	name
tarek	sara
sara	tarek

In the previous example

- We got (sara and tarek) and (tarek and sara) even though it's the same people.
- This is due that the condition was ID1 != ID2 so of course they will appear twice after product.
- To solve this we can change the != to either < or >. This way it will also mean "different than" but will appear only once.

Solution

```
SELECT s1.name, s2.name
FROM students s1, students s2
WHERE s1.ID > s2.ID AND
s1.age = s2.age
```

Result

name	name
tarek	sara

Because
Tarek ID (12345) > Sara ID (134)


QUIZ
TIME!

Test your knowledge

Students

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

Roster

ID	major	absence
123	EE	4
134	EE	7
1345	CCE	2
12345	ME	4

→ Complete this query that finds all student names and last names for those who have **more than 3 absences**

```
SELECT name, lastname
FROM _____________________________
WHERE _____________________________
```

→ Complete this query that finds all student ID pairs that have **same number of absences.**

```
SELECT_____________________________
FROM roster r1, roster r2
WHERE _____________________________________
```

Part B

Subqueries

Subqueries

- Can be defined as query inside a query

- They can return:
  - a single constant/value
  - a relation
- They usually appear in WHERE clauses
- They can also appear in FROM clauses

Example 1

Task **Get the major of Rim Ismail**

Solution

```
SELECT major
FROM roster
WHERE ID =
```

Replace here with a query that returns the ID of Rim ismail

Students

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

Roster

ID	major	absence
123	EE	4
134	EE	7
1345	CCE	2
12345	ME	4

Example 1

Students

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

Roster

ID	major	absence
123	EE	4
134	EE	7
1345	CCE	2
12345	ME	4

Task **Get the major of Rim Ismail**

Solution

```
SELECT major
FROM roster
WHERE ID =
      (SELECT ID
        FROM students
          WHERE name='rim' AND
              lastname='ismail')
```

Result

major
EE

Alternative way

```
SELECT major
FROM students NATURAL JOIN roster
WHERE name='rim' AND lastname='ismail'
```

Example 1

Task **Get the major of Rim Ismail**

Solution

```
SELECT major
FROM roster
WHERE ID =
      (SELECT ID
        FROM students
          WHERE name='rim' AND
              lastname='ismail')
```

Students

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

Roster

ID	major	absence
123	EE	4
134	EE	7
1345	CCE	2
12345	ME	4

The above query generates an error if there are more than one student called rim ismail.
Alternative: using IN

Keywords **IN** and **NOT IN**

- The keyword 'IN' is used to specify multiple values in a WHERE clause for comparison.
- It enables filtering rows based on whether a column value matches any value in a list.
- Similarly, sometimes we need a condition that this value should not be found in a table, we use **NOT IN**

Syntax

- ***s* IN R** is true if scalar *s* = one of the values in relation R.
- → **R MUST BE a unary relation (1 column/attribute)**
- ***s* NOT IN R** is true if s is not equal to any value in R

Example 1

Students

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

Roster

ID	major	absence
123	EE	4
134	EE	7
1345	CCE	2
12345	ME	4

TASK

Get all student names who are registered in the EE program.

Solution

```
SELECT name, lastname
FROM students
WHERE ID IN
```

Replace here with a query that returns all the IDs of EE students

Example 1

Students

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

Roster

ID	major	absence
123	EE	4
134	EE	7
1345	CCE	2
12345	ME	4

TASK

Get all student names who are registered in the EE program.

Solution

```
SELECT name, lastname
FROM students
WHERE ID IN
      (SELECT ID
            FROM roster
                  WHERE major='EE')
```

ID
123
134

Result

name	lastname
rim	ismail
sara	kamel

Let's make this example a little more interesting

- We will add a student to students table **but not** to the roster. (Maybe he joined the university but didn't register anything yet)

Students

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345
Karim	Amer	18	21231

Roster

ID	major	absence
123	EE	4
134	EE	7
1345	CCE	2
12345	ME	4

Example 2

Students

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345
Karim	Amer	18	21231

Roster

ID	major	absence
123	EE	4
134	EE	7
1345	CCE	2
12345	ME	4

TASK

Get all student names who are registered **and also** are on the roster

Solution

```
SELECT name, lastname
FROM students
WHERE ID IN
            (SELECT ID
              FROM roster)
```

Result

name	lastname
rim	ismail
sara	kamel
milad	ghantous
tarek	dib

Can you solve the same task using natural join without using IN?

Example 3

Students

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345
Karim	Amer	18	21231

Roster

ID	major	absence
123	EE	4
134	EE	7
1345	CCE	2
12345	ME	4

TASK

Get all student names who are registered **but not on** the roster

Solution

```
SELECT name, lastname
FROM students
WHERE ID NOT IN
        (SELECT ID
          FROM roster)
```

Result

name	lastname
Karim	Amer

Example 4

Students

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345
Karim	Amer	18	21231

Roster

ID	major	absence
123	EE	4
134	EE	7
1345	CCE	2
12345	ME	4

IN can be used with a list of values

Solution

```
SELECT name, lastname
FROM students
WHERE ID IN(123, 134)
```

Result

name	lastname
rim	ismail
sara	kamel

Note

- IN and NOT IN can also be used on more than 1 attribute.
- Examples:
  - `... WHERE x, y IN (SELECT a, b from T)`
  - `... WHERE x, y, z NOT IN (SELECT a, b, c from T)`

Keywords **`ALL`** and **`ANY`**

- Used in conjunction with standard comparison operator (=, <>, !=, >, >=, <, or <=)
- ***`s > ALL R`*** is true if scalar s is greater than every value in unary R.
- ***`s > ANY R`*** is true if scalar s is greater than at least one value in unary R.
- The below comparisons return TRUE:

`12 > ALL R`

R

3
4
5
6

`5 > ANY R`

R

3
4
5
6

`2 < ALL R`

R

3
4
5
6

`3 <= ALL R`

R

3
4
5
6

Example

Roster

ID	major	absence
123	EE	4
134	EE	5
1345	CCE	7
12345	ME	4

TASK

Get all student IDs, from any major, for those students who have more absences than all EE students.

Solution

```
SELECT id
FROM roster
WHERE absence > ALL
```

Replace here with a query that returns all absences of EE students

Example

Roster

ID	major	absence
123	EE	4
134	EE	5
1345	CCE	7
12345	ME	4

TASK

Get all student IDs, from any major, for those students who have more absences than all EE students.

Solution

```
SELECT id
FROM roster
WHERE absence > ALL
      (select absence
           from roster
                where major='EE')
```

Result

id
1345


QUIZ
TIME!

Test your knowledge

Students

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345
Karim	Amer	18	21231

Roster

ID	major	absence
123	EE	4
134	EE	7
1345	CCE	2
12345	ME	4

→ Complete this query that finds all student names and last names for those who are 18 years old and they're registered in EE major.

```
SELECT _____________________
FROM students
WHERE age=18 AND ID
_____________________________________
```

End of part 2