

Chapter 4:

Relational Data Manipulation Languages

Why Relational Data Manipulation Languages?

- ❑ **The structure of relations is similar to that of tables in main memory.**
- ❑ **Why were special query and update languages invented specifically for relations?**

Answer: The independence between programs and data sought in DBMSs is no longer ensured.

- ❑ Therefore, **Data Manipulation Languages (DML)** must be purely **declarative**:
 - ❑ They must refer only to the concepts of the **conceptual schema** (relations, attributes, domains).
 - ❑ They must **ignore the internal organization** of relations.

Algebraic DML and Predicate DML

- The two basic DMLs are equivalent in expressive power for the relational model, with fundamental operations:
 - **Relational algebra**
 - **Tuple calculus**
- A DML is said to be **complete** if it provides at least the same capabilities as relational algebra or tuple calculus.

Predicate DML Objective

- ❑ **Express a query simply by defining the result.**
- ❑ These languages must be **declarative**.
- ❑ Two adaptations of predicate calculus to the relational model:
 - ❑ **Tuple calculus**
 - ❑ **Domain calculus**

Predicate Calculus

- ❑ **A mathematical theory that studies formal logical formulas** (without any specific meaning).
- ❑ It is limited to logical formulas constructed using a set of predicates, the operators “**and**”, “**or**”, “**not**”, as well as constants and variables.
- ❑ A predicate has a fixed number of arguments and can be **true** or **false**.

SQL

- The most widely used relational DML, as it is the only existing standard for relational DMLs.
- It combines both **algebraic** and **predicate** features.

Part 1:

Relational Algebra.

Relational Algebra

Purpose :

- ❑ Gave rise to algebraic-type DMLs.
- ❑ Identified the fundamental operators used in a relational database (RDB).

Relational Algebra

□ **Definition:**

A set of operators that, from one or two existing relations, produce a new temporary relation as a result.

- The resulting relation has exactly the same characteristics as a relation in the database.
- The resulting relation can be manipulated again using relational algebra operators.

Relational algebra : Operators

Formally , Relational Algebra comprises the following operators :

Basic operators

- Selection
- Projection
- Union
- Difference
- Product

Syntactic Operator

- Rename

Derived Operators

- Intersection
- Natural join
- Thêta-join
- division

Example Database Schema: Education

- ❑ **Person (nP, lastname, firstname, address, gender)**: all students and all teachers of the institute.
- ❑ **Student (nP, nS, birthD)**: any person currently enrolled at the institute.
- ❑ **Student_Studies (nS, degree, year)**: previous studies of students.
- ❑ **Teacher (nTea, nP, phone, grade)**: any person currently teaching one or more courses at the institute.
- ❑ **Course (Ccode, Cname, cycle, nTea)**: any course offered by the institute.
- ❑ **Obtained (nS, Ccode, grade, year)**: student nS passed course Ccode in a given year and obtained a given grade.
- ❑ **Enrollment (nS, Ccode)**: currently, student nE is enrolled in course nomM.
- ❑ **Prerequisite (Ccode, Ccode_prerequisite)**: name of a course and one of its prerequisite courses.

Example Database Schema: Education

Relations

Person (nP, lastname, firstname, gender)

nP	Lastname	firstname	adress	gender
102	Mecheri	Mohamed	Mila	M
104	Zouaghi	Ouarda	Mila	W
107	Achouri	nihad	Mila	W
206	Ouali	Lotfi	Jijel	M
109	Zouaghi	Mohamed	Mila	M
203	Laraba	Samira	Constantine	W
.....

Example Database Schema: Education

- **Student (nP, nS, birthD)**: any person currently enrolled at the institute.

nP	nS	BirthD
102	1002	04/05/2000
104	1006	22/12/2000
107	1014	04/05/2000
109	1024	13/02/1995
.....		

Example Database Schema: Education

- **Teacher (nTea, nP, phone, grade):** any person currently teaching one or more courses at the institute.

<u>nTea</u>	nP	Phone	grade
I4	15	055555555	Mca
I10	18	066666666	Maa
i15	25	077777777	maa
m21	23	088888888	mcb
.....			

Example Database Schema: Education

- **Course (Ccode, Cname, cycle, nTea)**: any course offered by the institute.

Ccode	Cname	Cycle	nTea
Algo	algorithmic	1MI	i4
BDD	Databases	2INF	i10
OS	Operating systems	2INF	i25
.....			

Example Database Schema: Education

- ❑ **Enrollment (nS, Ccode):** currently, student nE is enrolled in course nomM.

nE	nomM
1002	BDD
1014	BDD
1014	SE
1024	SE
..	..

Example Database Schema: Education

- **Obtained (nS, Ccode, grade, year):** student nS passed course Ccode in a given year and obtained a given grade.

NS	Ccode	grade	year
1002	BDD	11	2020
1006	BDD	15	2020
1002	SE	13	2020
1014	BDD	15	2020
1014	ALGO	11	2020
1006	SE	11	2020
1024	SE	10	2020
1024	RES	12	2020
1014	AO	13	2020
1014	SM2	12	2019
1030	SM2	11	2020

Example Database Schema: Education

- ❑ **Prerequisite (Ccode, Ccode_prerequisite):** name of a course and one of its prerequisite courses.

Ccode	Ccode_prerequisite
BDD	SI
SE	SM2
SE	AO
RES	TECWEB
ALGO2	ALGO
..	..

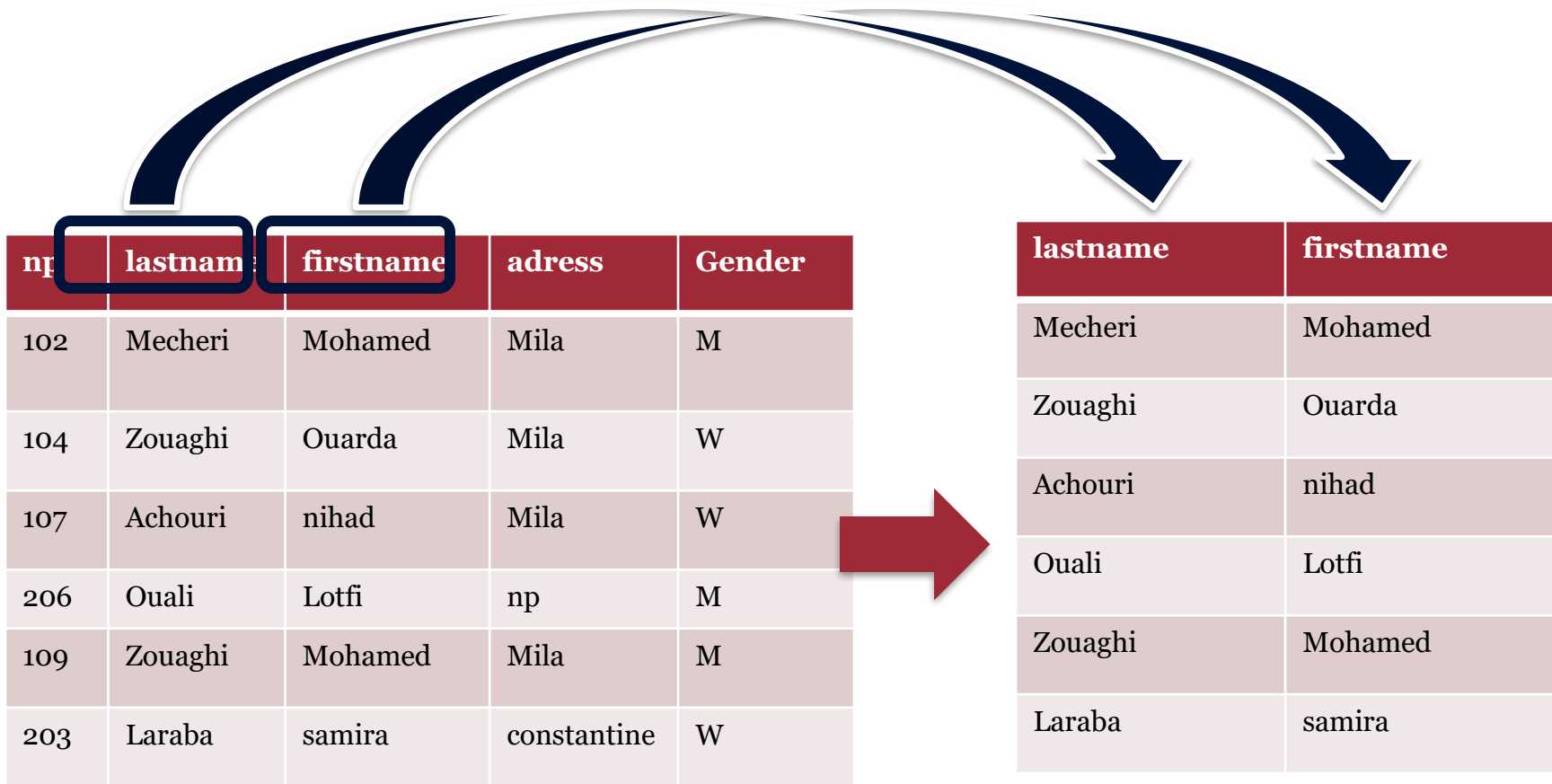
Projection Operator

- ❑ The projection operator keeps only selected attributes (columns).

Symbol: π

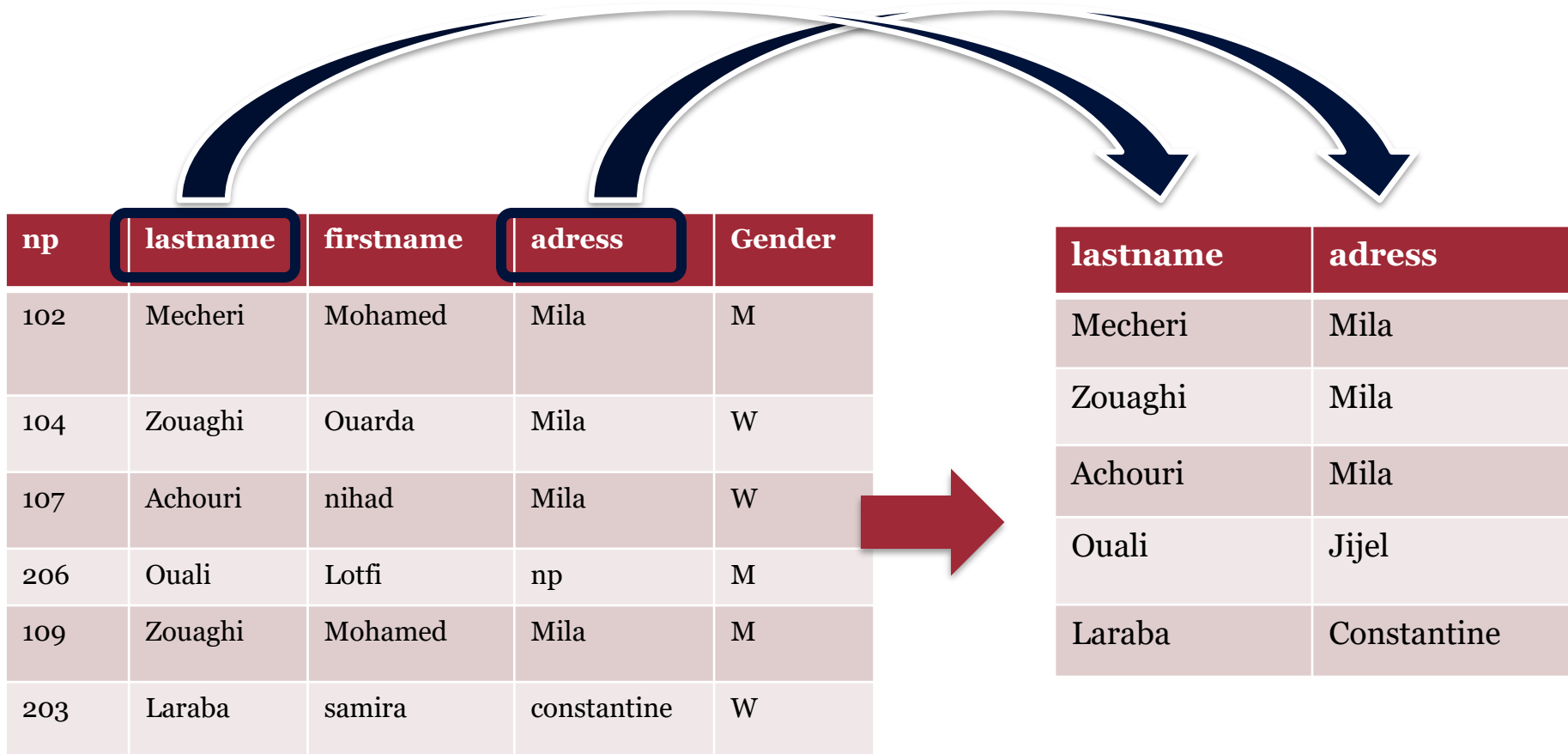
- ❑ **Number of operands:** 1 (unary operator)
- ❑ The attributes specified in the projection list are retained in the resulting relation.
- ❑ Duplicate tuples are removed.

Projection : example 1:



List of lestnames and firstnames of persons :
RES = Π [lastname, firstname] Person

Projection : example2:



List of lastnames and adresses of persons :

RES = Π [nom, Adresse] Personne

Selection Operator

- ❑ The selection operator keeps the tuples that satisfy a logical predicate.

Symbol: σ

- ❑ **Number of operands:** 1 (unary operator)
- ❑ The tuples in the resulting relation must satisfy a logical predicate.
- ❑ This predicate is expressed using comparison operators ($>$, \geq , $<$, \leq , $=$, \neq) and logical connectors (**and**, **or**, **not**).

Sélection : Example 1

- List of lastnames and firstnames of women :
- NP := Π [lastname, firstname] (σ (Gender = 'W') (person))**

np	lastname	firstname	adress	Gender
102	Mecheri	Mohamed	Mila	M
104	Zouaghi	Ouarda	Mila	W
107	Achouri	nihad	Mila	W
206	Ouali	Lotfi	np	M
109	Zouaghi	Mohamed	Mila	M
203	Laraba	samira	constantine	W



lastname	firstname
Zouaghi	Ouarda
Achouri	Nihad
Laraba	samira

Selection Predicate

- Allows comparing the value of attributes of R with other attributes of R or with constants.

- **Syntax :**

$\langle p \rangle ::= \langle \text{condition} \rangle$

| $\langle p \rangle \langle \text{logical operator} \rangle \langle p \rangle$

| $\neg \langle p \rangle$

| $(\langle p \rangle)$

- $\langle \text{logical operator} \rangle ::= \wedge \mid \vee$

- $\langle \text{condition} \rangle ::= \text{attribute-name} \langle \text{comparison-operator} \rangle \text{value}$
| $\text{attribute-name} \langle \text{comparison-operator} \rangle \text{attribute-name}$

- $\langle \text{comparison operator} \rangle ::= = \mid \geq \mid \leq \mid \neq \mid < \mid >$

Algebra Expressions

□ Relational algebra operators can be combined in expressions to express non-trivial queries.

□ **Example:**

List of names and first names of female persons living in Mila:

π (Name, FirstName) (σ (gender = 'W' \wedge address = 'Mila') person)

np	lastname	firstname	adress	Gender
102	Mecheri	Mohamed	Mila	M
104	Zouaghi	Ouarda	Mila	W
107	Achouri	nihad	Mila	W
206	Ouali	Lotfi	np	M
109	Zouaghi	Mohamed	Mila	M
203	Laraba	samira	constantine	W



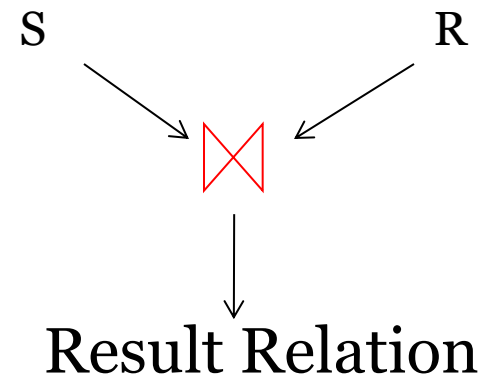
Lastname	firstname
Zouaghi	Ouarda
Achouri	Nihad

Natural Join

- ❑ The natural join is the combination of two relations that share common attributes (with the same name).
- ❑ It combines tuples from two relations that have the same value for the common attributes.
- ❑ The result of $R * S$ is the set of tuples $\langle x, y, z \rangle$ obtained by combining a tuple $\langle x, y \rangle$ from R and a tuple $\langle y, z \rangle$ from S such that both tuples have the same value for Y .

❑ **Symbol:** \bowtie or $*$

Number of operands: 2 (binary operator)



Cardinality of a Natural Join

□ The number of tuples in the result (denoted N) satisfies:

$$N \in [0 .. \text{card}(R) \times \text{card}(S)]$$

Minimum case: $N = 0$, occurs when **no tuple in R matches any tuple in S** on the common attribute Y .

👉 In other words, there is **no common value** of Y between the two relations.

Maximum case: $N = \text{card}(R) \times \text{card}(S)$

This occurs when **all tuples in R and S share the same value for Y**.

👉 In this case, the natural join behaves like a **Cartesian product**.

Natural Join : Examples

Example 1:

All information about students : Concatenate each pair of tuples from **Person** and **Student** that have the same value for **nP**.

Answer:

Student * Person

Natural Join : Examples

Example 2:

Names of students who passed the Database course (BDD).

Answer:

RES = π [lastname]

(Person * Student * σ (CCode = 'BDD') Obtained)

Natural Join : Example 2

Person

np	lastname	firstname	adress	gender
102	Mecheri	Mohamed	Mila	M
104	Zouaghi	Ouarda	Mila	F
107	Achouri	nihad	Mila	F
206	Ouali	Lotfi	Jijel	M
109	Zouaghi	Mohamed	Mila	M
203	Laraba	samira	constantine	F
.....

Student

np	NS	BirthD
102	1002	04/05/2000
104	1006	22/12/2000
107	1014	04/05/2000
109	1024	13/02/1995

Result

Lastname
Mecheri
Zouaghi
Achouri
...

σ (CCode = 'BDD') Obtained

NS	CCode	grade	year
1002	BDD	11	2020
1006	BDD	15	2020
1014	BDD	15	2020
.....		

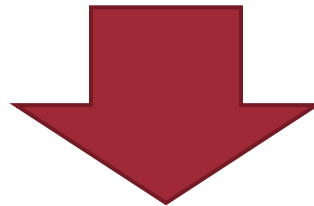
Cartesian product

- ❑ The Cartesian product of two relations **R** and **T** is the set of all possible concatenations of tuples from **R** and **T**.
- ❑ Tuples from both relations are concatenated end-to-end.
- ❑ **Symbol:** \times
- ❑ **Number of operands:** 2 (binary operator)

Cartesian product : Example

A	B
A1	B1
A2	B2

X



C	D
C1	D1
C2	D2
C3	D3

A	B	C	D
A1	B1	C1	D1
A1	B1	C2	D2
A1	B1	C3	D3
A2	B2	C1	D1
A2	B2	C2	D2
A2	B2	C3	D3

Renaming One or More Attributes of a Relation

- ❑ The rename operator, denoted α , allows changing the name of one or more attributes of a relation R .

Syntax:

α [attr1_name : new_name_for_attr1, attr2_name : new_name_for_attr2 ...] R

Role:

Useful before joins in case of **homonymy or synonymy**, or before set operators (**union, difference, intersection**).

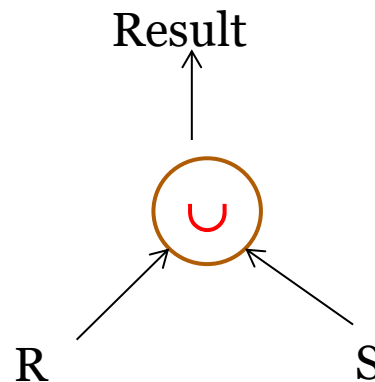
Union of Two Relations

Let **R** and **S** be two relations with the same schema

$R(A_1, A_2, \dots, A_n)$ and $S(A_1, A_2, \dots, A_n)$:

Union:

R \cup **S** creates a temporary relation with the same schema, whose tuples are those belonging to **R or S** (with duplicate tuples removed).



Union of Two Relations

$$R3 = R1 \cup R2$$

R1	
A	B
0	1
2	3

 \cup

R2	
A	B
0	1
4	5

$$R3 = R1 \cup R2$$



R3	
A	B
0	1
2	3
4	5

Union of Two Relations : Example

Query : List of person IDs who are either teachers of BDD or students enrolled in BDD.

Answer:

We create two temporary relations : **BDD-teacher** and **BDD-student:**

BDD-teacher := $\pi [nP] (\text{Teacher} * \sigma [\text{CCode} = \text{'BDD'}] \text{Course})$

BDD-student := $\pi [nP] (\text{Student} * \sigma [\text{CCode} = \text{'BDD'}] \text{Enrollment})$

Result:

BDD-teacher \cup **BDD-student**

Union: example

<u>n</u> teac	nP	phone	grade
I4	15	055555555	Mca
I10	18	066666666	Maa
i15	25	077777777	maa
m21	23	088888888	mcb

Ccode	Cname	Cycle	<u>n</u> teac
Algo	algorithmic	1MI	i4
BDD	Databases	2INF	i10
OS	Operating systems	2INF	i25

CCode	Cycle	nteac
BDD	2inf	i10

<u>n</u> Ens	nP	tel	grade	nom M	Cycle
I10	18	06666 666	Maa	BDD	2inf

BDD-Teacher

nP
18

Union: Example

nP	nS	BirthD
102	1002	04/05/2000
104	1006	22/12/2000
107	1014	04/05/2000
109	1024	13/02/1995

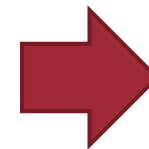
nS	CCode
1002	BDD
1014	BDD
1014	SE
1024	SE



nS	CCode
1002	BDD
1014	BDD



nP	nS	BirthD	CCode
102	1002	04/05/2000	BDD
107	1014	04/05/2000	BDD



nP
102
107

BDD-Student

Union: Example

BDD-Teacher

nP
18

BDD-Student

nP
102
107



nP
102
107
18

Result

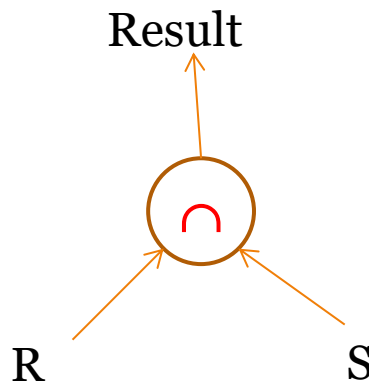
Intersection of two Relations

Let **R** and **S** be two relations with the same schema

$R(A_1, A_2, \dots, A_n)$ and $S(A_1, A_2, \dots, A_n)$:

Intersection:

$R \cap S$ creates a temporary relation with the same schema, whose tuples are those in **R that also appear in S** (i.e., tuples with identical values in both relations).



Intersection of two Relations example

$$R3 = R1 \cap R2$$

R1	
A	B
0	1
2	3

 \cap

R2	
A	B
0	1
4	5

$$R3 = R1 \cap R2$$



R3	
A	B
0	1

Intersection of two Relations : Example

Query :

List of person IDs who are both teachers and students (assistants, PhD students, etc.).

Answer:

$$\mathbf{RES} = (\pi [nP] \text{ Student}) \cap (\pi [nP] \text{ Teacher})$$

Intersection of two Relations

example

<u>ntea</u>	nP	phone	grade
I4	15	055555555	Mca
I10	18	066666666	Maa
i15	25	077777777	maa
m21	23	088888888	mcb



nP
15
18
25
23



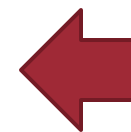
No person is both a
student and a
teacher.

\emptyset

nP	nS	BirthD
102	1002	04/05/2000
104	1006	22/12/2000
107	1014	04/05/2000
109	1024	13/02/1995



nP
102
104
107
109

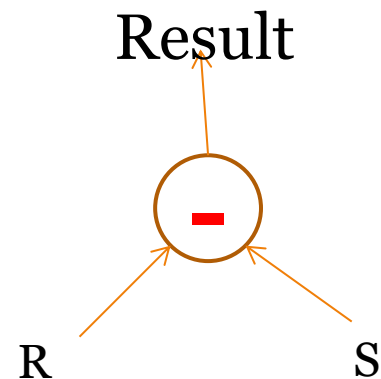


Difference of two relations

Definition: Let **R** and **S** be two relations with the same schema $R(A_1, A_2, \dots, A_n)$ and $S(A_1, A_2, \dots, A_n)$:

Difference:

R – S creates a temporary relation with the same schema, whose tuples are those in **R but not in S** (i.e., tuples that belong to R and do not appear in S).



Difference of two relations

$$R3 = R1 - R2$$

R1			R2	
A	B	−	A	B
0	1		0	1
2	3		4	5

$$R3 = R1 - R2$$



R3	
A	B
2	3

Difference of two relations example

List of person IDs who are neither teachers nor students of the BDD course:

Answer :

$$\Pi[nP]Person - (BDDTeacher \cup BDDStudent)$$

Difference of two relations : example

nP	lastname	firstname	adress	gende r
102	Mecheri	Mohamed	Mila	M
104	Zouaghi	Ouarda	Mila	F
107	Achouri	nihad	Mila	F
206	Ouali	Lotfi	Jijel	M
109	Zouaghi	Mohamed	Mila	M
203	Laraba	Samira	Consta ntine	F
.....



nP
102
104
107
206
109
203

BDDTeacher

nP
18

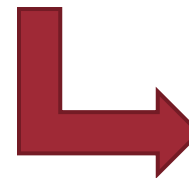


BDDStudent

nP
102
107



nP
102
107
18



nP
104
206
109
203

Theta-Join

- ❑ The theta-join is a selection of certain tuples from the result of the Cartesian product.

- ❑ **Symbol:** \bowtie_{θ}

Number of operands: 2 (binary operator)

- ❑ Let **R** and **S** be two relations with no attributes having the same name.

- ❑ **Syntax:**

$R \bowtie_{\theta} S = \sigma_{\theta} (R \times S)$, where θ is a logical predicate.

- ❑ **Result (Population):**

Population($R \bowtie_{\theta} S$) = {tuples from **R** and **S** concatenated that satisfy θ }

Theta-Join Predicate

- The predicate has the same form as a selection predicate, except that elementary conditions compare an attribute of **R** with an attribute of **T**:

$\langle \theta \rangle ::= \text{attribute-name-R} \langle \text{comparison-operator} \rangle \text{attribute-name-T}$

Theta-Join : Example

□ Query:

List of pairs of student IDs such that the two students were born on the same day.

□ Answer:

/ first, create another relation Student with renamed attributes */*

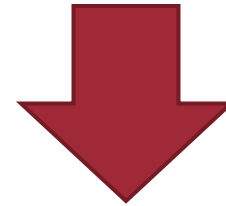
$\text{Student2} := \alpha [\text{nS} : \text{nS2}, \text{birthD} : \text{birthD } 2] \pi [\text{nS}, \text{birthD}] \text{Student}$

$\text{RES} = \pi [\text{nS}, \text{nS2}] (\text{Student} \bowtie [\text{nS} < \text{nS2} \wedge \text{birthD} = \text{birthD2}] \text{Student2})$

Theta-Join : Example

Student		
nP	nS	BirthD
102	1002	04/05/2000
104	1006	22/12/2000
107	1014	04/05/2000
109	1024	13/02/1995

Student2	
nS2	BirthD2
1002	04/05/2000
1006	22/12/2000
1014	04/05/2000
1024	13/02/1995



nP	nS	BirthD	nS2	BirthD2
102	1002	04/05/2000	1014	04/05/2000



Result:

nS	nS2
1002	1014

Division : Definition

□ Division is an operation that identifies the tuples of a relation that are associated with **all tuples of another relation**.

□ **Symbol:** \div

Number of operands: **2** (binary operator)

□ Let two relations: $R(A_1, A_2, A_3, B_1, B_2)$ and $S(B_1, B_2)$

$$R \div S = T (A_1, A_2, A_3)$$

such that, for every tuple (a_1, a_2, a_3) in **T**, and for every tuple (b_1, b_2) in **S**, there **exists a tuple** $(a_1, a_2, a_3, b_1, b_2)$ in **R**

Division : Example

❑ Query :

List of students who can enroll in the course SE (i.e., those who have passed all its prerequisites):

❑ Answer:

❑ Prerequisite courses for 'SE':

$\text{ReqSys} := \pi [\text{Ccode_prerequisite}] \sigma [\text{Ccode} = \text{'SE'}] \text{Prerequisite}$

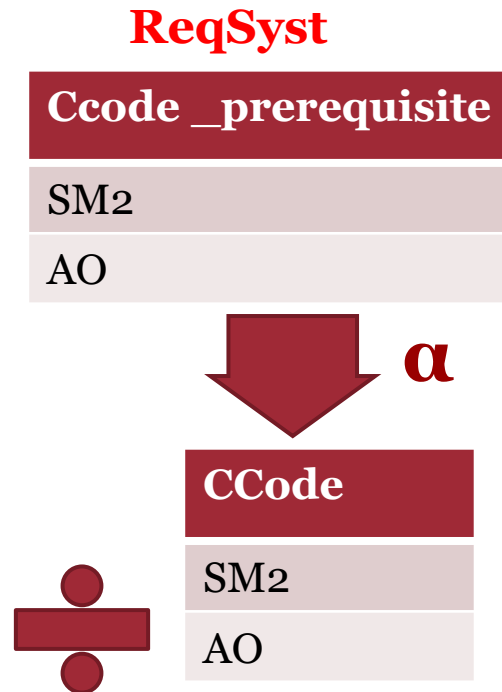
❑ Student IDs who can enroll in 'SE':

$(\pi [\text{Ccode}, \text{nS}] \text{Obtained}) \div (\alpha [\text{Ccode_prerequisite} : \text{Ccode}] \text{ReqSys})$

Division : Example

(π [nS,Ccode] Obtained)

NS	CCode
1002	BDD
1006	BDD
1002	SE
1014	BDD
1014	ALGO
1006	SE
1024	SE
1024	RES
1014	AO
1014	SM2
1030	SM2



Prerequisite

CCode	Ccode _prerequisite
BDD	SI
SE	SM2
SE	AO
RES	TECWEB
ALGO2	ALGO

Result

NE
1014

How to Write a Complex Query?

- ❑ Represent the query visually on the relational schema:
- ❑ **Identify the relevant relations** needed to express the query.
- ❑ **Reproduce the schema of these relations**, and indicate on them:
 - ✓ The attributes that are part of the query result
 - ✓ The conditions on the attributes
 - ✓ The links between the relations
- ❑ **Translate this diagram into a relational algebra expression:**
 - ✓ Apply **selections** based on attribute conditions
 - ✓ Apply **joins** (natural or theta) according to the links between relations (one join per link)
 - ✓ Apply **projection** on the attributes that are part of the result

How to Write a Complex Query?

❑ **Example:**

Names of students who are taking one of the courses taught by teacher number 200.

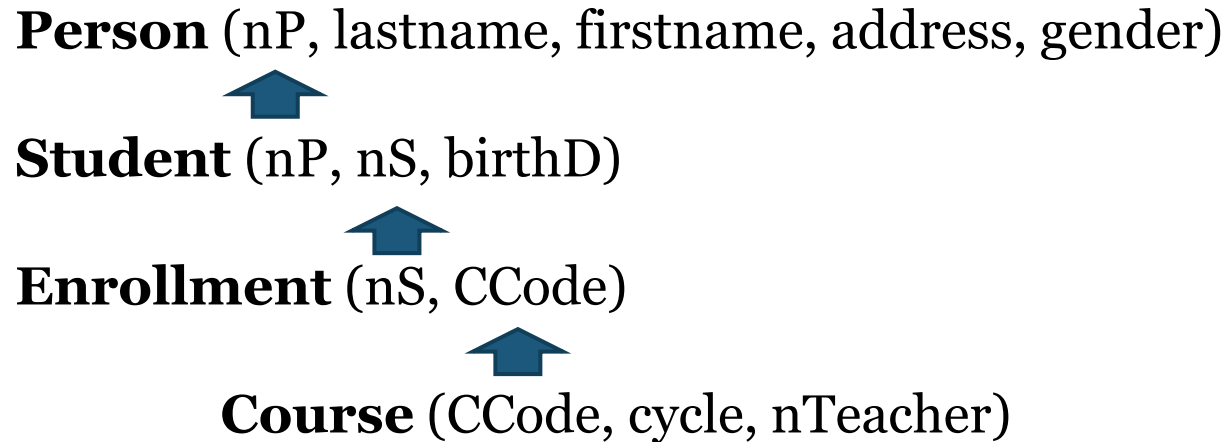
❑ **Solution:**

❑ **First step : Useful relations:**

- ✓ **Person:** for the student's name
- ✓ **Course:** for the courses taught by teacher number 200
- ✓ **Enrollment** and **Student:** to link these first two relations

How to Write a Complex Query?

□ **Step 2: We obtain the following diagram:**



□ **Step 3 : Translation of this diagram into a relational algebra expression:**

Result = π [name] (Person * Student * Enrollment * (σ [nEns = 200] Course))

Some Equivalences

Let R and S be two relations:

$$R \cap S = R - (R - S)$$

$$\sigma(\theta_1 \text{ and } \theta_2)(R) = \sigma(\theta_1)(R) \cap \sigma(\theta_2)(R)$$

$$\sigma(\theta_1 \text{ or } \theta_2)(R) = \sigma(\theta_1)(R) \cup \sigma(\theta_2)(R)$$

$$R \bowtie_{\theta} S = \sigma_{\theta}(R \times S)$$

End of part 1

Part2 to Follow: SQL