3.1 Languages: relational algebra

- Introduction
- Relational algebra operations
  - Union
  - Renaming
  - Intersection
  - Difference
  - Cartesian product
  - Selection
  - Projection
  - Combination (join)
- Relational algebra sequence of operations

Introduction

- The **data manipulation languages** (DML) can be classified as:
  - Languages based on relational algebra
  - Languages based on relational calculus (e.g.: SQL)

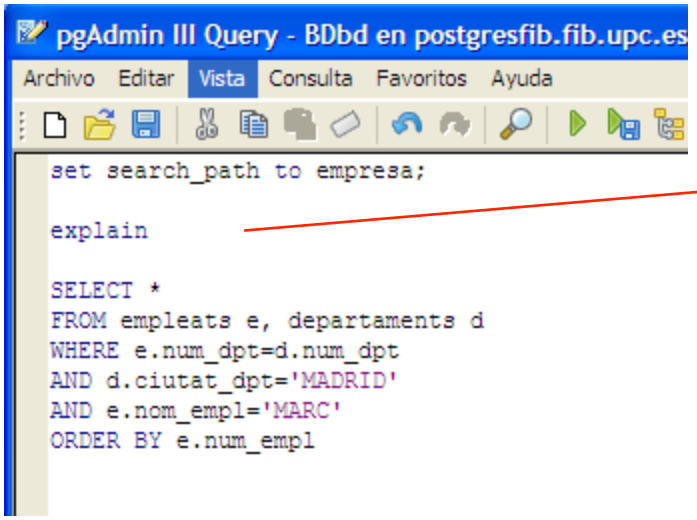
  But most of them use elements from both lines
  (SQL also incorpores elements from algebra).

- **Relational calculus:** Has its basis in the predicate calculations **Declarative What?**

- **Relational algebra:** Has its basis in the set theory
  (remember that the relations are sets) **Procedimental How?**

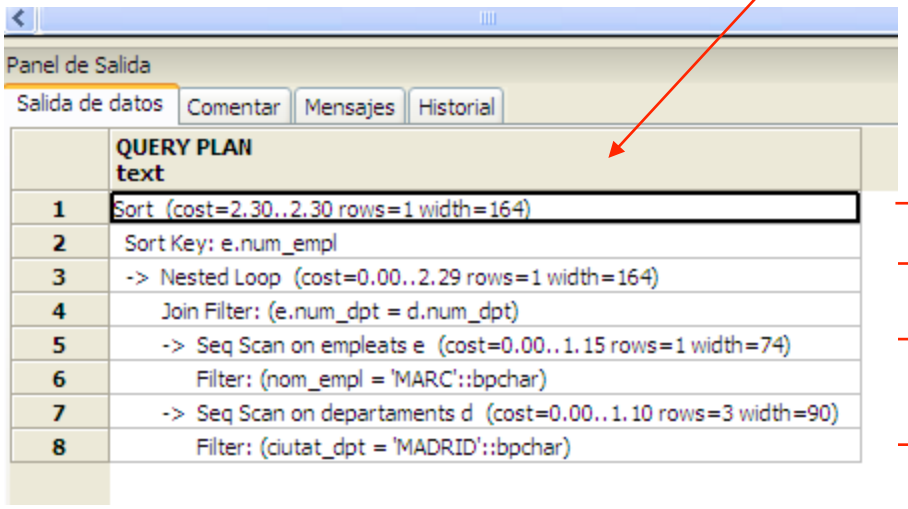
- **Interest** of relational algebra:
  - Helps to understand which querying functionalities must provide a relational language
  - The current standard version of SQL incorporates relational algebra operations
  - The DBMS process and optimize the queries based on relational algebra (remember that algebra is procedimental and, for example, SQL is declarative)

SQL: WHAT?



With this instruction 'EXPLAIN' we tell Postgres to explain us HOW it will execute this sentence

ALGEBRA: HOW?



The result contains, among other things (cost, result width, algorithms), the expression in relational algebra that represents the best possible sequence of execution to resolve the previous query.

4. Sort of result from 3

3. Join between results from 1 and 2

2. Selection over employees

1. Selection over departments

Operations of the relational algebra

- **1st classif.:** Set operations
  - Union
  - Intersection
  - Difference
  - Cartesian product

  Operations specifically relational
  - Selection
  - Projection
  - Combination (join)
  - Renaming

- **2nd classif.:** Primitive operations
  - Union
  - Difference
  - Cartesian product
  - Selection
  - Projection
  - Renaming

  Non-primitive operations
  - Intersection
  - Combination (join)

- **3rd classif.:** Binary operations
  - Union
  - Intersection
  - Difference
  - Cartesian product
  - Combination (join)

  Unary operations
  - Selection
  - Projection
  - Renaming

- **Relational closure**: Both the operands and the result of a relational algebra operation are relations

  Ex: $T = R \cup S$

Example

MODULE-CN(<u>module</u>, avg-surface)

B6	10
B2	20

OFFICE(<u>module-dp, num-dp</u>, superfície)

B6	25	10
B6	27	10
B2	25	15
B2	30	25

{module-dp} is a foreign key referencing MODULE-CN

ADM-PERSONNEL(<u>num-per</u>, name, surname, module, num)

100	Joan	Soler	B6	25
150	Clara	Bellsolà	B6	25

{module, num} is a foreign key referencing OFFICE

LAB-PERSONNEL(<u>num-per</u>, name, surname, module, num)

150	Clara	Bellsolà	B6	25
110	Núria	Nogué	B2	25
200	Jordi	Moles	B6	27
230	Pere	Roig	NULL	NULL

{module, num} is a foreign key referencing OFFICE

Union

ADM-PERSONNEL(num-per, name, surname, module, num)

100	Joan	Soler	B6	25
150	Clara	Bellsolà	B6	25

LAB-PERSONNEL(num-per, name, surname, module, num)

150	Clara	Bellsolà	B6	25
110	Núria	Nogué	B2	25
200	Jordi	Moles	B6	27
230	Pere	Roig	NULL	NULL

R =ADM-PERSONNEL ∪ LAB-PERSONNEL

R(num-per, name, surname, module, num)

100	Joan	Soler	B6	25
150	Clara	Bellsolà	B6	25
110	Núria	Nogué	B2	25
200	Jordi	Moles	B6	27
230	Pere	Roig	NULL	NULL

There are not repeated tuples!!!

- The schema attributes of the resulting relation T ∪ S are the same attributes of the T schema or S schema.
- The **body** of the resulting relation T ∪ S is the set of tuples belonging to the body of T, belonging to the body of S, or belonging to both of them
- To make the union between two relations, T and S must be **compatible relations.**
- In the case where the attributes from T and S have different names, we need to rename the attributes from one of the relations so that the relations will be compatible.

Compatible relationships

- Some relational algebra operations, such as the union, only make sense when applied to compatible relations (with "similar" tuples)

- Example: we can make the union

  ADM-PERSONNEL ∪ LAB PERSONNEL

  because the tuples from both relations are similar. However, there is no point in making the union

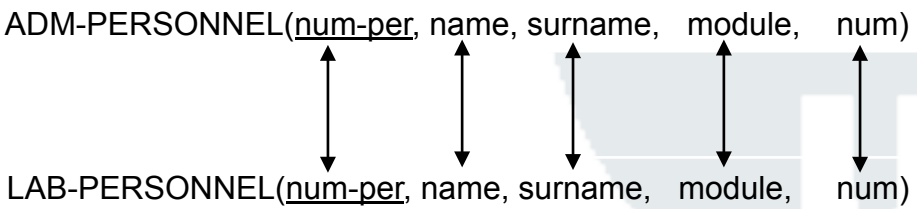
  ADM-PERSONNEL ∪ OFFICES

- **Two relations T and S are compatible** if:

  - They have schemas with an identic attribute set, and the domains for each pair of attributes are the same in T and in S.

  Example:

  ADM-PERSONNEL and ADM-PERSONNEL are clearly compatible:



Renaming

DOC-PERSONNEL(num-per, name, surname, module-dp, num-dp)				
400	Jaume	Cases	Omega	119
500	Pau	Pou	B6	123

R =DOC-PERSONNEL {module-dp -> module, num-dp -> num}

R(num-per, name, surname,		**module,**	**num)**
400	Jaume	Cases	Omega	119
500	Pau	Pou	B6	123

- The **schema attributes** of the resulting relation is the same as the initial relation except the attributes names that have been renamed.
- The **body** of the resulting relation does not change.

Intersection

ADM-PERSONNEL(num-per, name, surname, module, num)

100	Joan	Soler	B6	25
150	Clara	Bellsolà	B6	25

LAB-PERSONNEL(num-per, name, surname, module, num)

150	Clara	Bellsolà	B6	25
110	Núria	Nogué	B2	25
200	Jordi	Moles	B6	27
230	Pere	Roig	NULL	NULL

R = ADM-PERSONNEL ∩ LAB-PERSONNEL

R(num-per, name, surname, module, num)

150	Clara	Bellsolà	B6	25

- The schema attributes of the resultint relationship T ∩ S are the same attributes of the T schema or the S schema.
- The **body** of the relations resulting of T ∩ S is the set of tuples belonging to the body of both relations
- To make the intersections between two relations, T and S must be **compatible relationships**
- In the case where the attributes from T and S do have different names, we need to rename the attributes from one of the relationships so that the relations will be compatible

Difference

ADM-PERSONNEL(num-per, name, surname, module, num)

100	Joan	Soler	B6	25
150	Clara	Bellsolà	B6	25

LAB-PERSONNEL(num-per, name, surname, module, num)

150	Clara	Bellsolà	B6	25
110	Núria	Nogué	B2	25
200	Jordi	Moles	B6	27
230	Pere	Roig	NULL	NULL

R = ADM-PERSONNEL − LAB-PERSONNEL

R(num-per, nom, cognom, modul, num)

100	Joan	Soler	B6	25

- The **schema attributes** of the resultint relation T − S are the same attributes of the T schema or the S schema.
- The **body** of the relation resulting of T − S is the set of tuples belonging to the body of T but not to the body of S.
- To make the difference between two relationships, T and S must be **compatible relations.**
- In the case where T and S have different attribute names, we need to rename the attributes from one of the relationships so that the relations will be compatible

Cartesian product

MODULE-CN(module, avg-surface)

B6	10
B2	20

OFFICE(modul-dp, num-dp, surface)

B6	25	10
B6	27	10
B2	25	15
B2	30	25

R =MODULE-CN × OFFICE

R(module, avg-surface, module-dp, num-dp, surface)

B6	10	B6	25	10
B6	10	B6	27	10
B6	10	B2	25	15
B6	10	B2	30	25
B2	20	B6	25	10
B2	20	B6	27	10
B2	20	B2	25	15
B2	20	B2	30	25

• The **schema attributes** of the resulting relation T × S are all the attributes from T plus all the attributes from S.

• If T and S have attributes with the same name, previously we will need to rename one of the two relations to avoid the ambiguity.

• The **body** of the resulting relation from T × S is the set of all the tuples of the form $<v_1, v_2, ..., v_n, w_1, w_2, ..., w_m>$ where $<v_1, v_2, ..., v_n>$ belongs to the body of T and $<w_1, w_2, ..., w_m>$ belongs to the body of S.

• To make the cartesian product of two relations T and S, T and S must not be **compatible relations.**

Selection

OFFICE(moduel-dp, num-dp, surface)		
B6	25	10
B6	27	10
B2	25	15
B2	30	25

R =OFFICE(module-dp= 'B2' AND surface>16)

R(module-de, num-de, surface)		
B2	30	25

- **T(C)** denotes the selection of T with the condition C, being C the selection condition
- The condition C is composed by one or more comparisons of the form:

$$Ai\ \theta\ Vj \quad \text{or} \quad Ai\ \theta\ Aj$$

where Ai and Aj are attributes of the relation T, Vj is a constant value, and θ is a comparison operator (=, <>, <, <= , >, >=). The comparisons have to be interrelated by one of the logical operators AND ($\wedge$), OR ($\vee$).

- The **schema attributes** of the resultint relation T(C), are the same attributes of the T schema.
- The **body** of T(C) is the set of tuples that belong to the body of T that satisfy the condition C.

Projection

ADM-PERSONNEL(per-num, name, surname, module, num)				
100	Joan	Soler	B6	25
150	Clara	Bellsolà	B6	25

R =ADM-PERSONNEL[module, num]

R(module, num)	
B6	**25**

In relational algebra, **THERE ARE NOT REPEATED TUPLES**, because the result of the operations are sets.

- **T[A_i, A_j, ..., A_k]** denotes the projection of T over {A_i, A_j, ..., A_k}, being {A_i, A_j, ..., A_k} a subset of the attributes of the relation T schema.
- The **schema attributes** of the resulting relationship T[A_i, A_j, ..., A_k], are the attributes {A_i, A_j, ..., A_k}
- The **body** of the resultint relationship T[A_i, A_j, ..., A_k] is the subset of all the tuples with the form <t.A_i, t.A_j, ..., t.A_k> where t is a tuple that belongs to the body of T and where t.A_p denotes the value for the attribute A_p from the tuple t.

Combination (join)

MODULE-CN(module, avg-surface)

B6	10
B2	20

OFFICE(module-dp, num-dp, surface)

B6	25	10
B6	27	10
B2	25	15
B2	30	25

R =MODULE-CN[module=modul-dp]OFFICE

R(module, avg-surface, module-dp, num-dp, surface)

B6	**10**	**B6**	**25**	**10**
B6	**10**	**B6**	**27**	**10**
B2	**20**	**B2**	**25**	**15**
B2	**20**	**B2**	**30**	**25**

R =MODUL-ECN[module=module-dp, avg-surface<=surface]OFFICE

R(module, avg-surface, module-dp, num-dp, surface)

B6	**10**	**B6**	**25**	**10**
B6	**10**	**B6**	**27**	**10**
B2	**20**	**B2**	**30**	**25**

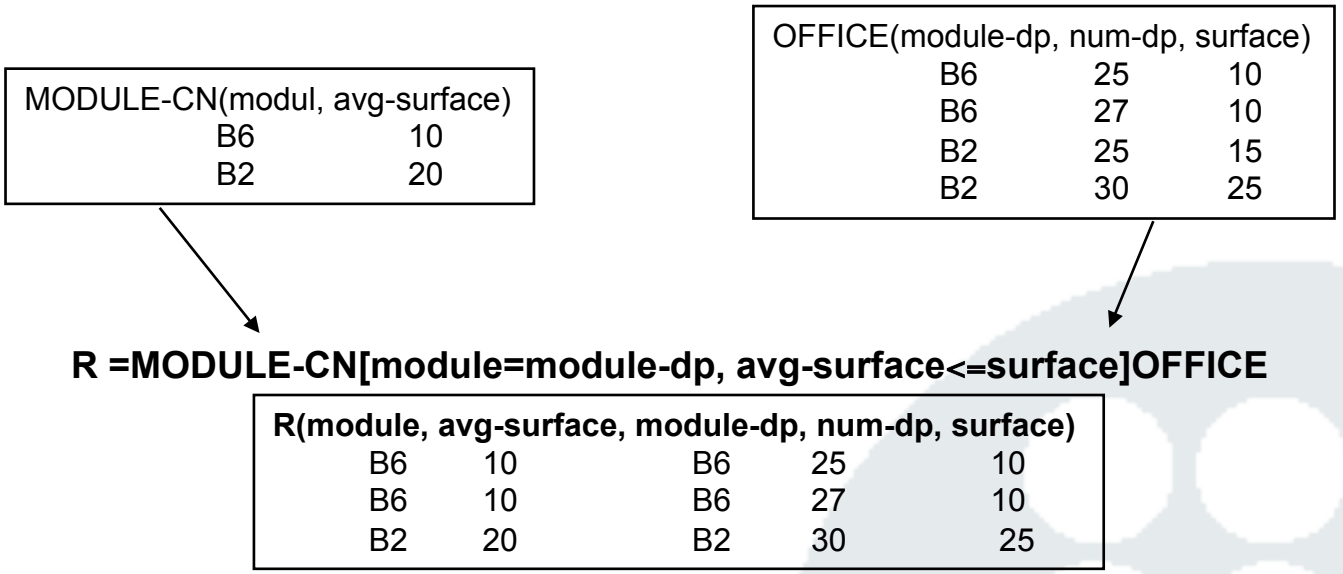
- **T[B]S** denotes the combination of T and S with the condition B
- The **condition B** of a combination **T [B] S** is composed by one or more comparisons with the form:

$$A_i \ \theta \ A_j$$

where A_i is an attribute of the relation, T, A_j is an attribute of the relation S, θ is a comparison operator ($=, \neq, <, \leq, >, \geq$) and it matches that A_i and A_j have the same domain.

- The different conditions in a combination or join are separated by commas.

Combination (join)



• The **schema attributes** of the relation resulting of T[B]S are all the attributes of T and all the attributes of S.

• If T and S have some identical attribute name, previously we will need to rename one of the two relations to avoid the ambiguity.

• The **body** of the resulting relation T[B]S is the set of tuples belonging to the extension of the cartesian product T×S that match all the comparisons that form the combination condition B.

Combination (join): Types of "joins"

- **"θ-join":** The "join" is also called "θ-join"
- **"Equi-join"**: Particular case of "join" where all the comarisons of the condition have the operator '= '.

ADM-PERSONNEL(per-num, name, surname, module, num)

100	Joan	Soler	B6	25
150	Clara	Bellsolà	B6	25

OFFICE(module-dp, num-dp, surface)

B6	25	10
B6	27	10
B2	25	15
B2	30	25

R =ADM-PERSONNEL[module=module-dp, num=num-dp]OFFICE

R(num-per, name, surname, module, num, module-dp, num-dp, surface)

100	Joan	Soler	B6	25	B6	25	10
150	Clara	Bellsolà	B6	25	B6	25	10

- **"Natural join":** Variant of the "equi-join" without all the unnecessary attributes.We denote it using *. The difference with the "equi-join" is that in the natural join schema, the second attribute of each comparison does not appear anymore in the resulting schema.

ADM-PERSONNEL(per-num, name, surname, module, num)

100	Joan	Soler	B6	25
150	Clara	Bellsolà	B6	25

OFFICE(module-dp, num-dp, surface)

B6	25	10
B6	27	10
B2	25	15
B2	30	25

R =ADM-PERSONNEL[module*module-dp, num*num-dp]OFFICE

R(num-per, name, surname, module, num, surface)

100	Joan	Soler	B6	25	10
150	Clara	Bellsolà	B6	25	10

Combination (join): implicit "Natural join"

ADM-PERSONNEL(num-per, name, surname, module, num)

100	Joan	Soler	B6	25
150	Clara	Bellsolà	B6	25

MODULE-CN(module, avg-surface)

B6	10
B2	20

R =ADM-PERSONNEL * MODULE-CN

R(num-per, name, surname, module, num, avg-surface)

100	Joan	Soler	B6	25	10
150	Clara	Bellsolà	B6	25	10

• The **implicit "natural join":** Variant of the "natural-join" where we do not specify the condition of the combination and therefore we assume (by deafult) that the combination condition corresponds to the one of a "natural join", where all the pairs of attributes with the same name on both relations are matched .

• **T * S** denotes the implicit "natural join" of T and S.

Sequences of operations of relational algebra

Example: Get the offices (module and number) from the modules that have an average surface bigger than 15.

MODULE-CN(module, avg-surface)

B6	10
B2	20

OFFICE(module-dp, num-dp, surface)

B6	25	10
B6	27	10
B2	25	15
B2	30	25

A =MODULE-CN(avg-surface >15)

B = A{module -> module-dp}

C = OFFICE * B

R = C[module-dp, num-dp]

R(modul-dp, num-dp)

B2	25
B2	30

• The **queries** of a relational DB may be expressed in terms of relation algebra **sequences of operations**.

• The sequences of operations allow us to define a relation that contains exactly what we want to query.