escola d'enginyeria

UAB

ESCI upf. School of International Studies

Block 3
SIMPLE QUERIES (PART 1)

Debora Gil, Oriol Ramos, Carles Sanchez

Contents: Simple Queries

Contents: Part 1

1. Introduction

Queries:

Q1: Who has not bought any ticket?

Q2: Who has watched West Side Story and Terrific?

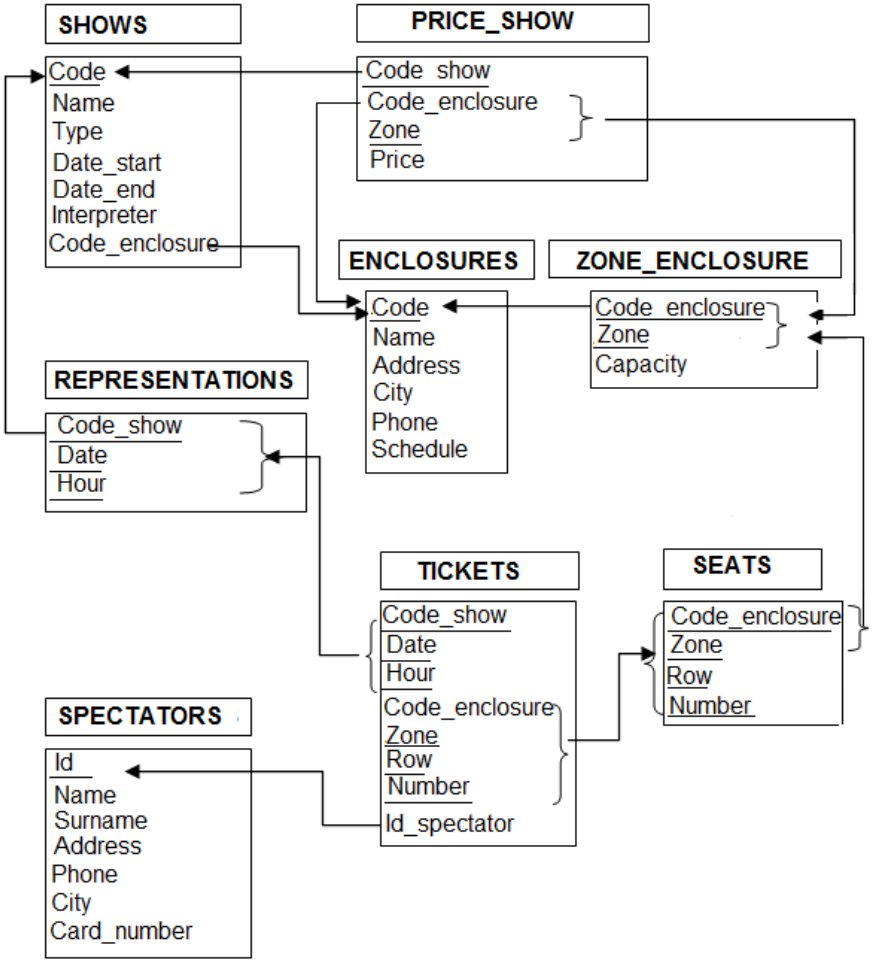


Table containing Registers and / or attributes that keep some conditions → We need operators to manipulate tables

Data Manipulation (DML)

There are two strategies to define the theoretical basis of the manipulation of the data:

- **Relational Algebra (RA) -** Constructive formulation. Apply specific operators (explicit) to existing tables to build the table to be achieved.

- **Relational Calculus (RC)**- Descriptive formulation. Give the definition of what you want to search using logical operators

This course will study the strategy defined by relational algebra.

2. Relational Algebra

Characteristics

- Set of high level operators that operate on tables.
- The searched information is expressed through concatenation of relational operators (the result of an operator is a table)
- Basic rules to build relationships and formalize searches on database (SQL base). Automatization on creation of queries
- Next to programming languages
- Allows a query scheme before programming it physically. Specially useful for complex queries
- Defined by Codd (1970)

Elements

Set of elements

Example:

Set of relations: $\mathcal{R} = \{R, S, T, \ldots\}$

Operators applied on the set elements (closed operators)

Example:

Unary: $\mathcal{R} \rightarrow \mathcal{R}$

Binary operators: $\mathcal{R} \times \mathcal{R} \rightarrow \mathcal{R}$

operators

Set Operators:

Union
Intersection
Difference
Cartesian product

Relational operators:

Restriction
Projection
Natural Join
Theta join
Division

Operators Codd

Additional operators:

Extension (extension)
Summary (group by)
General division
Outer join
Outer join theta

Codd operators

	Binary	Unary
Set Theory	Cartesian product	
	Union	
	Difference	
	Intersection	
RA specifics	Join	Restriction
	Division	Projection

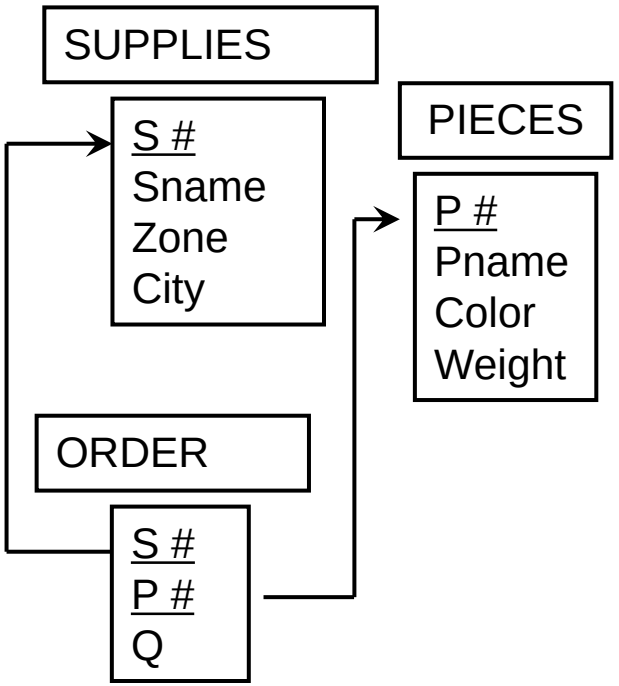
Codd operators

	Access by Content	**Tables Fusion**
Set theory	Union Difference Intersection	Cartesian Product
Specific Relational Algebra	Restriction Projection Division	Join

- 5 elementary operators generate all other operators:
  - Union
  - Difference
  - Cartesian product
  - Restriction
  - Projection

Example

Get IDs and names of suppliers who supply the piece P2



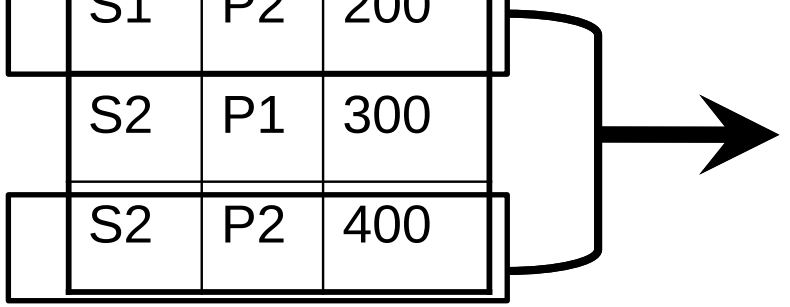
1. **Restrict** SP with tuples with pieces P2 (WHERE)
2. **Natural Join** between relations S and SP Restricted
3. **Project** the result on attributes S #, Sname

(SELECT ... FROM ... WHERE .... “SQL”)

Restrict: Take tuples having P #= P2

SP

S #	**P #**	**Q**
S1	P1	300
S1	P2	200
S2	P1	300
S2	P2	400



S #	**P #**	**Q**
S1	P2	200
S2	P2	400

Joining by common attributes

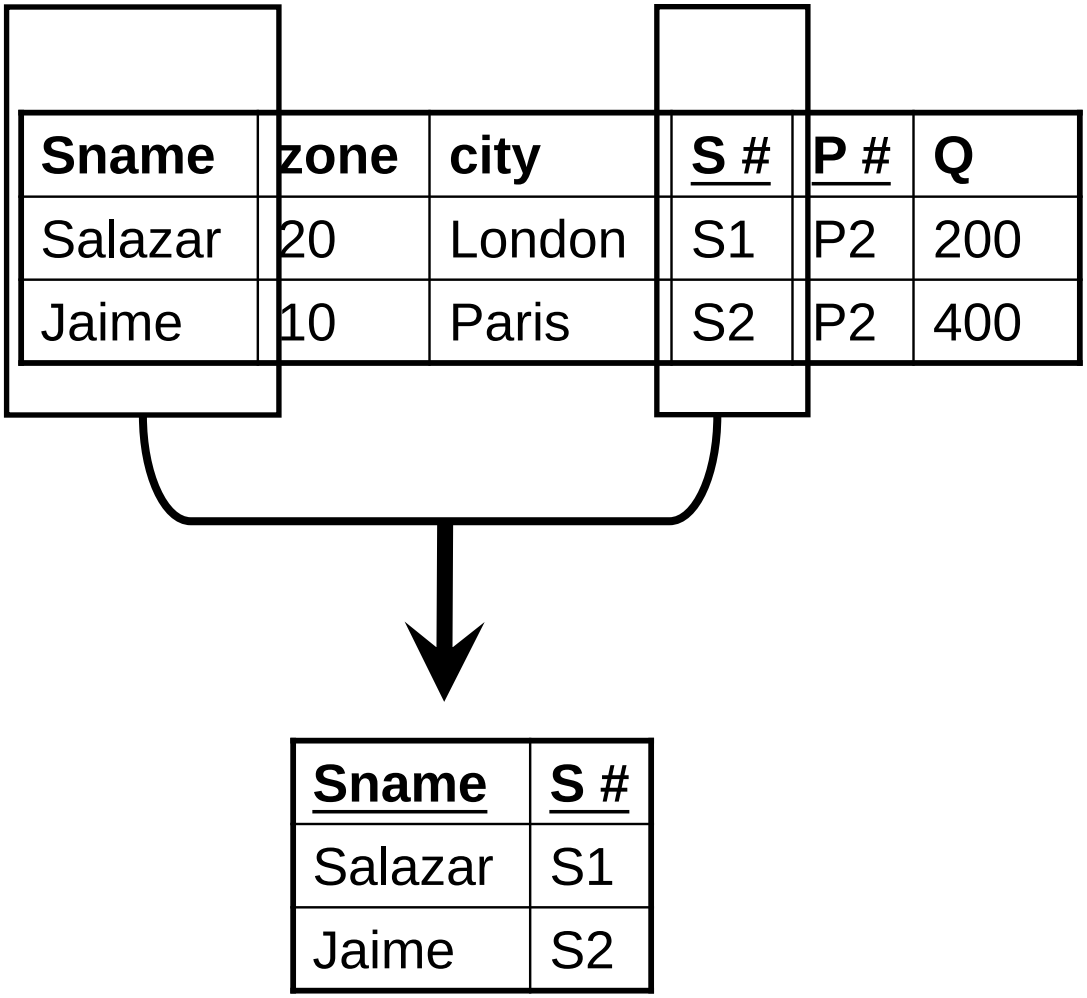
S

Sname	**zone**	**city**	**S #**
Salazar	20	London	S1
Jaime	10	Paris	S2
Bernal	30	Paris	S3

S #	**P #**	**Q**
S1	P2	200
S2	P2	400

Sname	**zone**	**city**	**S #**	**P #**	**Q**
Salazar	20	London	S1	P2	200
Jaime	10	Paris	S2	P2	400

Project: We stayed with solicited attributes

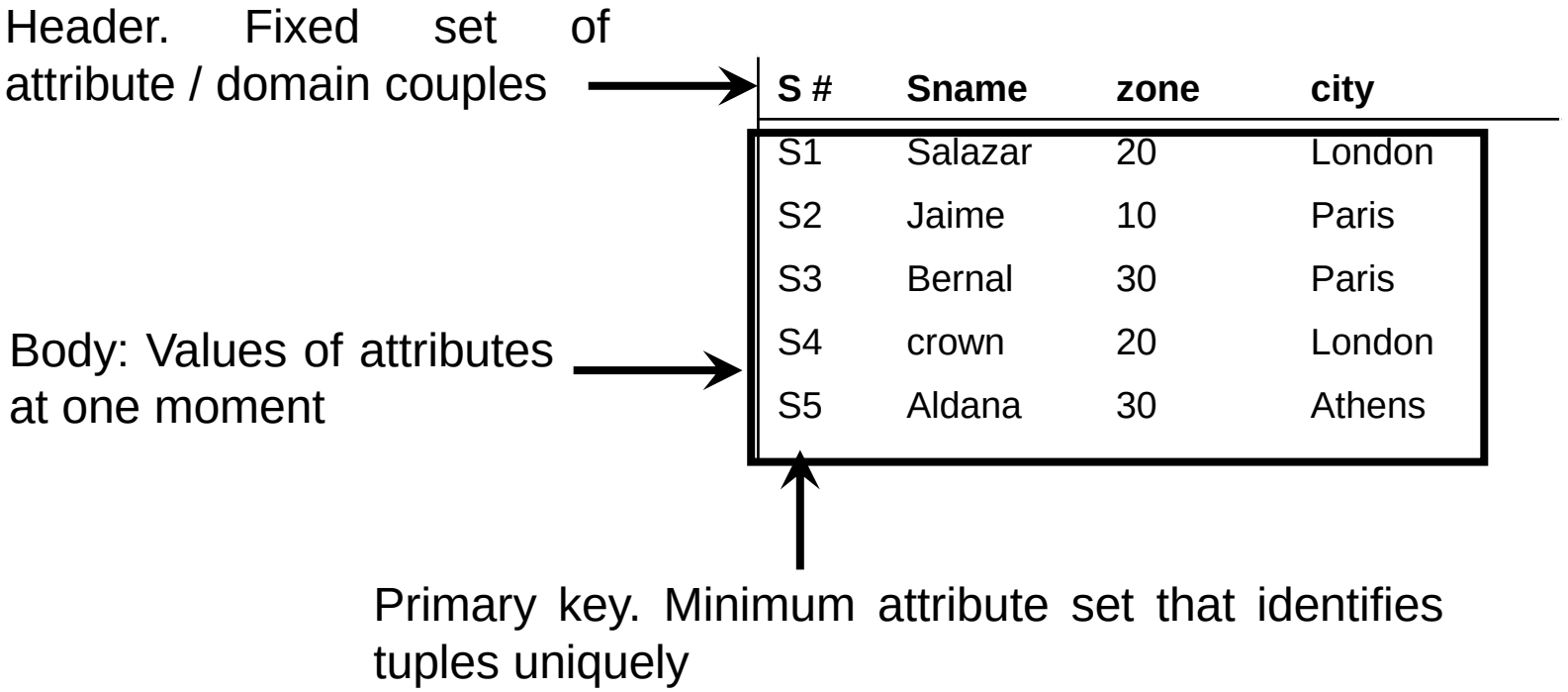


Sname	**zone**	**city**	**S #**	**P #**	**Q**
Salazar	20	London	S1	P2	200
Jaime	10	Paris	S2	P2	400

Sname	**S #**
Salazar	S1
Jaime	S2

Definition of Relational Algebra Operators

They are applied to tuples of the existing tables to create new tables.



S #	Sname	zone	city
S1	Salazar	20	London
S2	Jaime	10	Paris
S3	Bernal	30	Paris
S4	crown	20	London
S5	Aldana	30	Athens

They are defined specifying PK, header and body:

TableName(PK, attribute list)

R (S#, Sname, Area, City)

Notations

OPERATOR	**NOTATION 1**	**NOTATION 2**
Union	Union T = (R, S)	$R \cup S$
Intersection	T = Inter (R, S)	$R \cap S$
Difference	T = Dif(R, S)	r \ S
Cartesian product	T = Prod(R, S)	R x S
Restriction	T = Restriction(R | Q) Q condition on attributes R	R where Q, Q condition on attributes R
Projection	T =Projection(R | A_1.., A_p) A_1.., A_p attributes are requested	R [A_1... A_p] A_1.., A_p requested attributes
Join by common attributes A_1..., A_p	T = Join(S, R | R A_1SA =$_1$ and ... R. A_p=SA_p)	(R x S where R. A_1SA =$_1$ and ... R. A_p=SA_p)
Division	T =div(R (X, Y), S (Y))	R (X, Y) / S (Y)
Group by X_1...,X_p	T = Group (R by X_1...,X_p | Op (X_j) as NewAtribut)	summarize R group by X_1...,X_p add Op (X_j) as NewAtribut

3. Codd basic operators

3.1 Restriction

3.2 Projection

3.3 Examples

3.1 Restriction

Definition

Selection of the tuples (Rows) of the relation R that fulfil a condition Q:

- Primary Key: Same as R,

  PK (T) = PK (R)
- Body: tuples R fulfilling the condition logic Q
- Header: Same that R

Notation: T = Restriction(R | Q)

R where Q

Restriction Condition

Q condition is given by simple comparisons on R atributes:

X and Y can be: Attributes names

Constants

Results of queries (subqueries)

It is possible to combine simple comparisons with logical operators: AND, OR (composite conditional expressions)

Q: X1 θ1 Y1 AND/OR X2 θ2 Y2 …. AND/OR Xn θn Yn

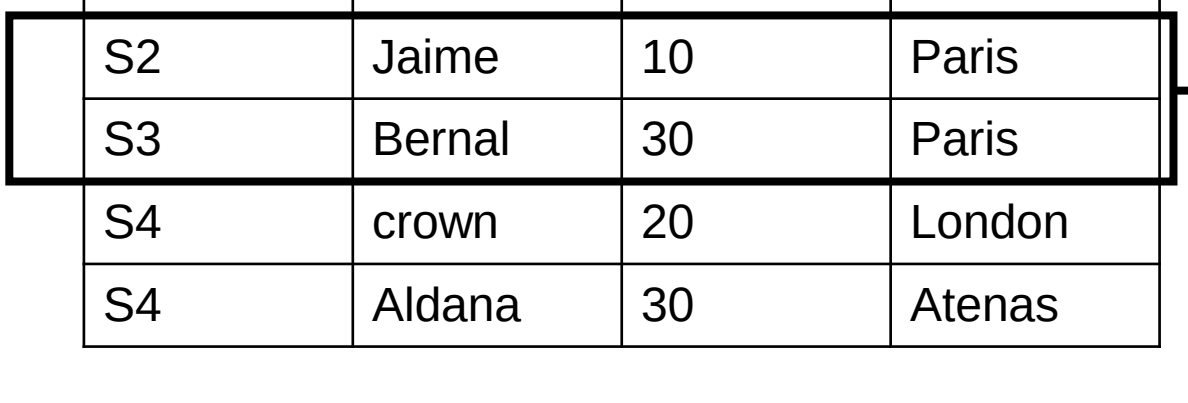
Example1

Q1: Suppliers from Paris

T = Restriction(S | City = "Paris")

S

S #	Sname	zone	city
S1	Salazar	20	London
S2	Jaime	10	Paris
S3	Bernal	30	Paris
S4	crown	20	London
S4	Aldana	30	Atenas



S #	Sname	zone	city
S2	Jaime	10	Paris
S3	Bernal	30	Paris

Example2

Q2: Suppliers distributing to areas bellow 20

S

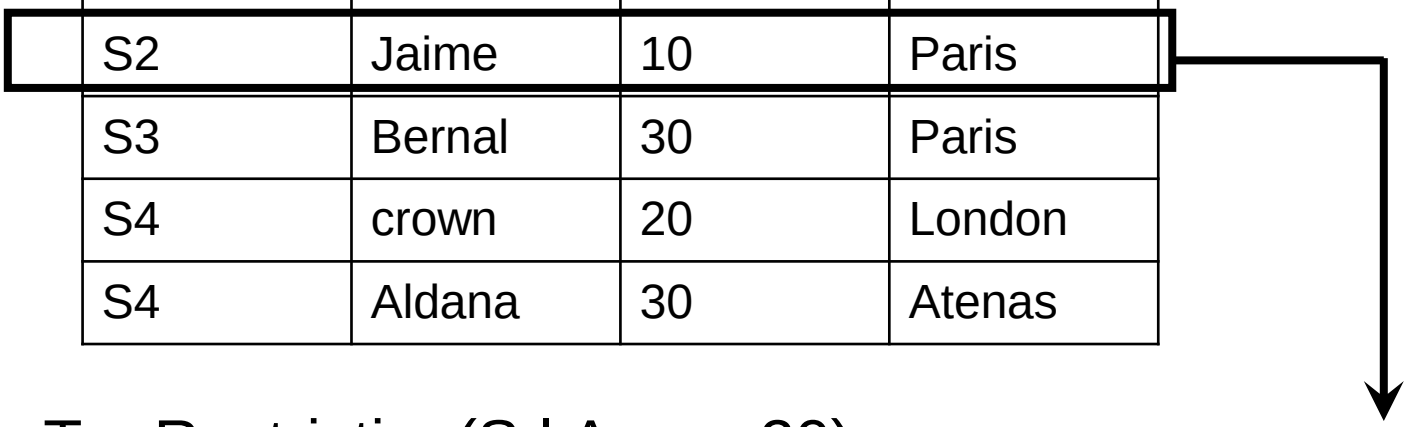
S #	Sname	zone	city
S1	Salazar	20	London
S2	Jaime	10	Paris
S3	Bernal	30	Paris
S4	crown	20	London
S4	Aldana	30	Atenas

Example2

Q2: Suppliers distributing to areas bellow 20

S

S #	Sname	zone	city
S1	Salazar	20	London
S2	Jaime	10	Paris
S3	Bernal	30	Paris
S4	crown	20	London
S4	Aldana	30	Atenas



T = Restriction(S | Area <20)

S #	**Sname**	**zone**	**city**
S2	Jaime	10	Paris

Example3

Q3: Suppliers distributing less than 400 pieces of type P2

SP:

S#	P#	CANT
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P3	200
S4	P2	200
S4	P4	300
S4	P5	400

Example3

T = Restriction(SP | P # = "P2" AND Q <400)

SP:

S#	P#	CANT
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P3	200
S4	P2	200
S4	P4	300
S4	P5	400

S #	P #	Q
S1	P2	200
S4	P2	200

3.1 Projection

Definition

Let R (A_1, A_2...,A_n). The Projection R on the attributes A_{i1}...,A_{ip} p<n is a relation T with:

- Primary Key: projected PK(R) attributes
- Header: T (A_{i1}..., A_{ip})
- Body: values corresponding to the attributes selected without repetitions (DISTINCT in SQL)

Notation: T =Projection(R | A_{i1}...,A_{ip})
R [A_{i1}...,A_{ip}]

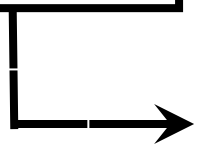
Example1

Q1: Cities of suppliers

S

S #	Sname	zone	city
S1	Salazar	20	London
S2	Jaime	10	Paris
S3	Bernal	30	Paris
S4	crown	20	London
S4	Aldana	30	Atenas

T = Projection(S | City)



city
London
Paris
Atenas

Example2

Q2: Suppliers Name from Paris

S

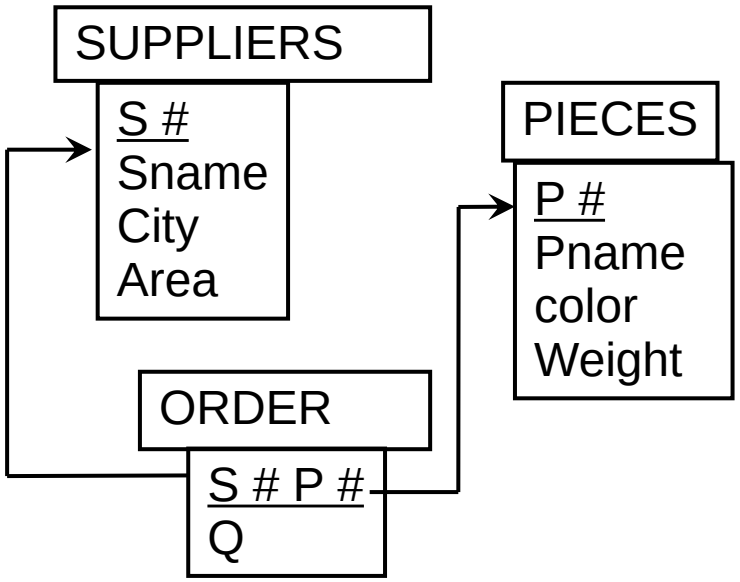
S #	Sname	zone	city
S1	Salazar	20	London
S2	Jaime	10	Paris
S3	Bernal	30	Paris
S4	crown	20	London
S4	Aldana	30	Atenas

Sname
Jaime
Bernal

T = Projection(Restriction(S | city = 'Paris') | Sname)

3.1 Examples

DB Suppliers



1. Weight of red pieces
2. Suppliers that are not from London
3. Name of suppliers who have a order of more than 400 pieces

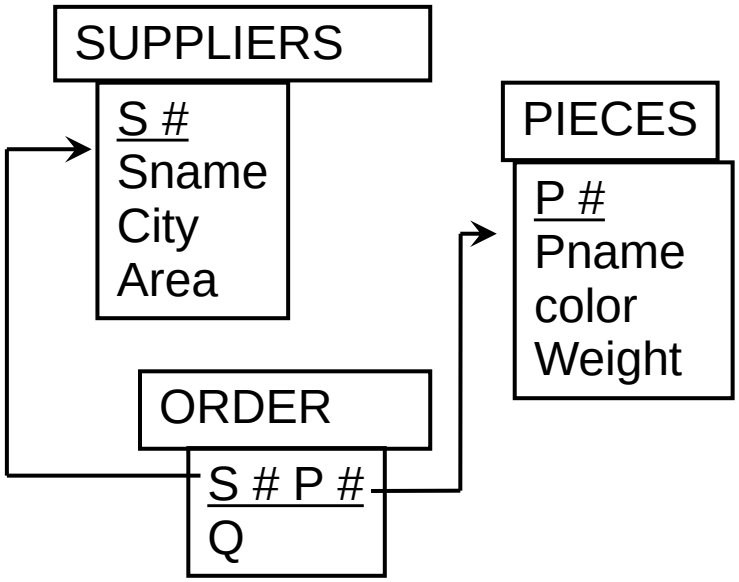
DB Suppliers

1. Weight of red pieces

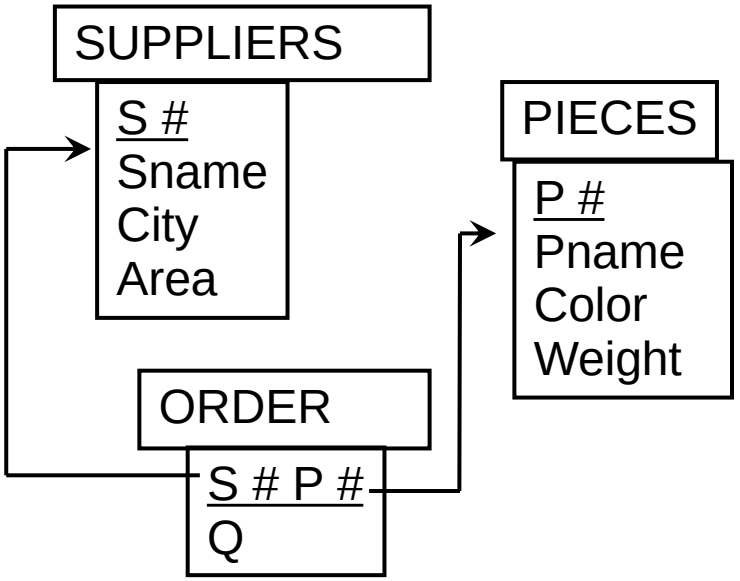
Tables: Pieces, P

RA:

T1 =Restriction(P | = Color ='red')
T2 =Projection(T1 | Weight)



DB Suppliers



1. Weight of red pieces

Tables: Pieces, P

AR:

T1 =Restriction(P | = Color= 'red')
T2 =Projection(T1 | Weight)

SQL:

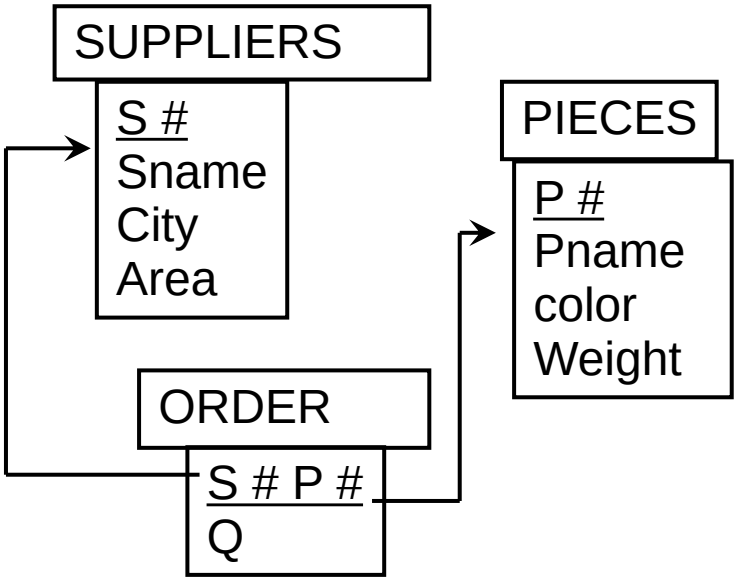
```
select P.Weight
from Pieces P
where P.Color= 'red';
```

Projection (select)

Tables (from)

Restriction (where)

DB Suppliers



2. Name of the suppliers that are not from London

Tables: Suppliers, S

RA:

T1 =Restriction(S | city= 'London')
T2 =Projection(T1 | Sname)

SQL: not to have repeated tuples

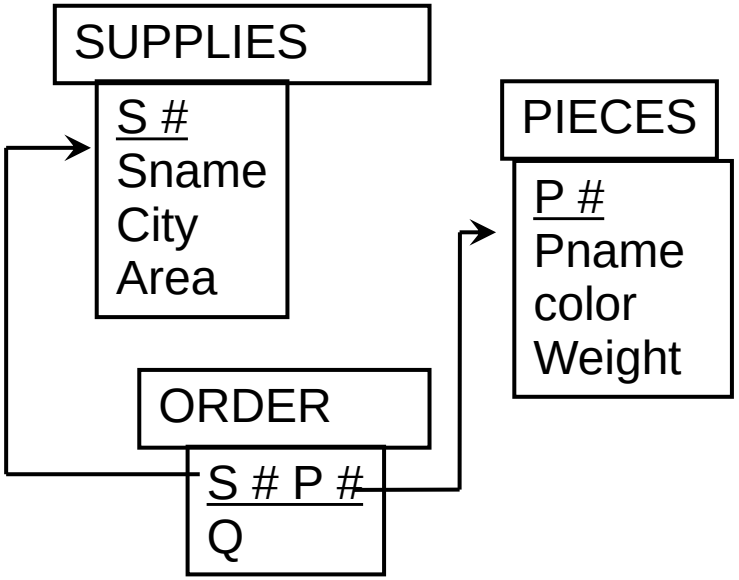
SQL:

```
Select DISTINCT S.Sname
from Suppliers S
where S.City ! = London '
order by 1;
```

SQL: to order tuples

DB Suppliers

3. Name of the suppliers who have a order of more than 400 pieces

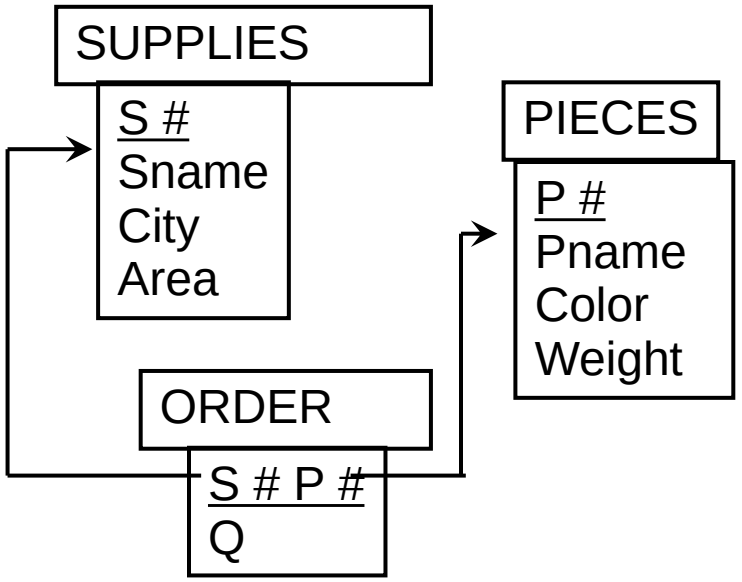


This query requires 2 different tables: Order for the restriction and Suppliers for the Projection

DB Suppliers

3. Name of the suppliers who have a order of more than 400 pieces

How do you think should combine information of the two tables to have a consistent result?



S

S#	Snom	Zona	Ciutat
S1	Salazar	20	Londres
S2	Jaime	10	Paris
S3	Bernal	30	Paris
S4	Corona	20	Londres
S4	Aldana	30	Atenas

SP:

S#	P#	CANT
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P3	200
S4	P2	200
S4	P4	300
S4	P5	400