escola d'enginyeria

UAB

ESCI upf. School of International Studies

Block 3
SIMPLE QUERIES (PART 2)

Debora Gil, Oriol Ramos, Carles Sanchez

Contents: Simple Queries

Contents: Simple Queries

3. Basic Codd Operators

3.3 Cartesian product

3.4 Join

3.5 Examples

3.6 Exercises

3.3 Cartesian Product

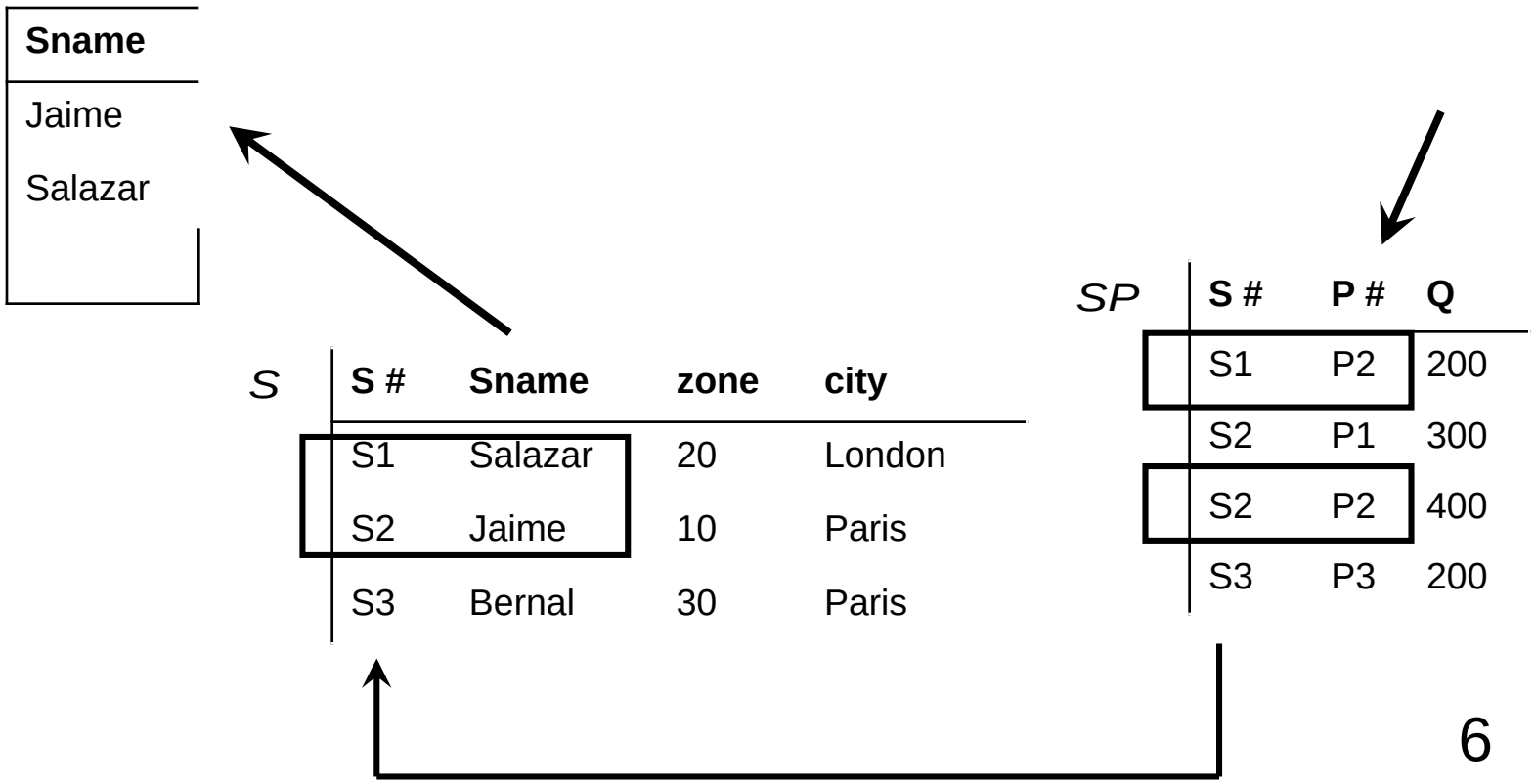
Tables Merging

	Access to content	**Fusion Tables**
Set theory	Union Difference Intersection	**Cartesian product**
RA specific	Division Restriction Projection	**Join**

RA Queries

Most questions need to merge information from more than one table

Names of suppliers who supplied the piece P2



Sname
Jaime
Salazar

S

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

SP

S #	P #	Q
S1	P2	200
S2	P1	300
S2	P2	400
S3	P3	200

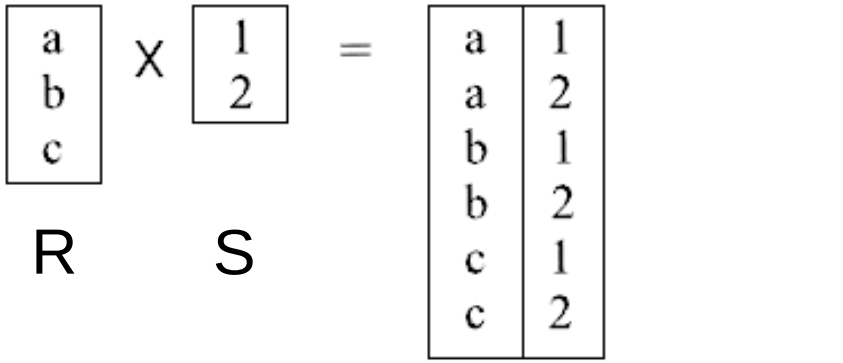
Fusion Tables

Set Operator (brute force, without relational structure): **Cartesian product**

Algebraic Operator (bearing in mind the relational structure): **natural join**, general join

Cartesian product

Pairing Set of combinations of one element of S and all of R



The cartesian product generates all possible combinations among all the elements of the tables involved.

T = Prod (R, S)
T = R x S

Cartesian product: Definition

Two relations R and S with some headers. The Cartesian product R and S is a T relation formed:

- Primary Key: Union of primary keys,

  PK (T) = (PK(R),PK (S))

- Body: For all tuples R, their combination between a tuple of R to everyone of S
- Header: Union of headers

Notation: T = Prod(R, S)

T = R x S

Example1

S

S #	Sname
S1	Salazar
S4	crown
S2	Jaime

Cartesian producto SxP ?

P

P #
P1
P2
P3

Example1

S

S #	Sname
S1	Salazar
S4	crown
S2	Jaime

P

P #
P1
P2
P3

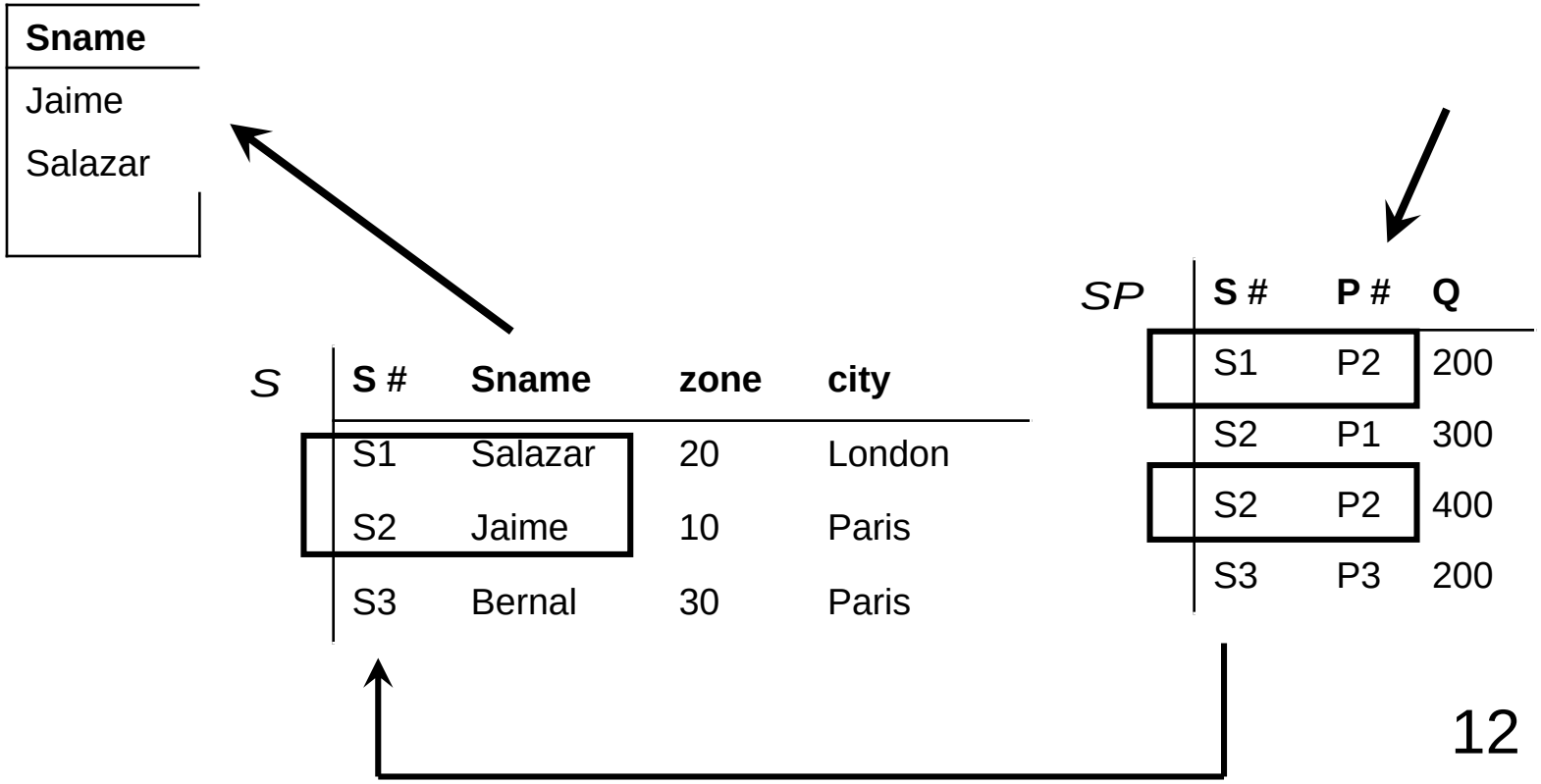
$S \times P$

S #	Sname	P #
S1	Salazar	P1
S4	crown	P1
S2	Jaime	P1
S1	Salazar	P2
S4	crown	P2
S2	Jaime	P2
S1	Salazar	P3
S4	crown	P3
S2	Jaime	P3

Example2

Q: names of suppliers who supplied the piece P2

This query requires merging information of SP and S.
We can solve it using **Cartesian product**?



Example2

The products Cartesian between S and SP →

SxSP

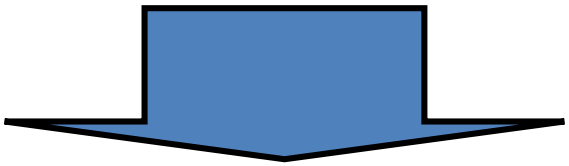
S #	Sname	zone	city	S #	P #	Q
S1	**Salazar**	**20**	**London**	**S1**	**P2**	**200**
S1	Salazar	20	London	S1	P4	200
S1	Salazar	20	London	S2	P1	300
S1	**Salazar**	**20**	**London**	**S2**	**P2**	**400**
S1	Salazar	20	London	S3	P3	200
S2	**Jaime**	**10**	**Paris**	**S1**	**P2**	**200**
S2	Jaime	10	Paris	S1	P4	200
S2	Jaime	10	Paris	S2	P1	300
S2	**Jaime**	**10**	**Paris**	**S2**	**P2**	**400**
S2	Jaime	10	Paris	S3	P3	200
S3	**Bernal**	**30**	**Paris**	**S1**	**P2**	**200**
S3	Bernal	30	Paris	S1	P4	200
S3	Bernal	30	Paris	S2	P1	300
S3	**Bernal**	**30**	**Paris**	**S2**	**P2**	**400**

T=Projection(Restriction (SP x S | P # = 'P2') | Sname)

Does it return the expected results? → NO!

If R, S have NTr, NTs tuples and NAr and NAs atributes, the RxS table has **NTr*NTs** tuples and **NAr + NAs** attributes.

Then: Cartesian product multiplies all R cells by all S cells generating a matrix with all the posible combinations.



CARTESIAN PRODUCT IS NOT APPROPRIATE TO JOIN RELATIONS.

CAN JOIN TUPLES NOT RELATED BY FK

Cartesian product utilities:

- Cartesian producto between two tables R and S, can be useful in thecase we want create a matrix table with all the posible combinations between R and S.

- Can be useful in the case of linking two tables R and S that don’t shared attributes and, then, there is not FK link between them.

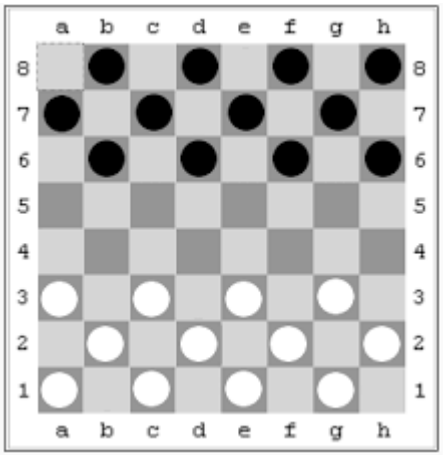
Cartesian product utilities:

In “Draughts” game there are **64 squares** and **12 pieces per colour**; not all squares from board have a piece all time, But they exist.

Then, if we want to keep all the row and columna piece positions all time, is not a solution to have only into account squares with pieces in one moment.

We need to generate board with all `possibles combinations between rows and columns

→ We have to do **Cartesian product** between **row** and **columns** RxC



3.4 Join

Natural Join

Only considering tuples of artesian product that has some attributes (common to R and S) with the same value

Cartesian product

S #	Sname	zone	city	S #	P #	Q
S1	Salazar	20	London	**S1**	P2	200
S1	Salazar	20	London	**S1**	P4	200
S1	Salazar	20	London	S2	P1	300
S1	Salazar	20	London	S2	P2	400
S1	Salazar	20	London	S3	P3	200
S2	Jaime	10	Paris	S1	P2	200
S2	Jaime	10	Paris	S1	P4	200
S2	Jaime	10	Paris	**S2**	P1	300
S2	Jaime	10	Paris	**S2**	P2	400
S2	Jaime	10	Paris	S3	P3	200
S3	Bernal	30	Paris	S1	P2	200
S3	Bernal	30	Paris	S1	P4	200
S3	Bernal	30	Paris	S2	P1	300
S3	Bernal	30	Paris	S2	P2	400

Natural join

S #	Sname	zone	city	S #	P #	Q
S1	Salazar	20	London	**S1**	P2	200
S1	Salazar	20	London	**S1**	P4	200
S2	Jaime	10	Paris	**S2**	P1	300
S2	Jaime	10	Paris	**S2**	P2	400

Natural Join: Definition

$$R(X_1,....,X_m,\underline{Y_1,....,Y_n})$$
$$S(\underline{Y_1,....,Y_n},Z_1,...,Z_p)$$

Let R and S be relations with attributes Y_j in common:

- Primary Key: PK (T) = PK (R) U PK (S) (no duplicate attributes)
- Body: Tuples (x, y, z) such that:
  - There is a tuple with values (x, y) in R
  - There is a tuple with values (y, z) in S
- Header: Attributes X,Y and Z

Notation:

T = Join(S, R | R.A_1=SA_1 and ... R.A_p=SA_p)

(R x S where R.A_1=SA_1 and ... R.A_p=SA_p)

where A_1...,A_p are the join attributes

Natural join: properties

Commutative:

Join (R, S) = Join (S, R)

Associative:

Join (R, S, T) = Join (R, Join (S, T)) = Join (Join (R, S), T)

Observations:

Join (R, S) = Projection(Restriction (Prod (R, S) | R.Y= S.Y), R.X, R.Y, S.Z)

If you try a join between two tables what they have no common attributes (Or not specified) = Cartesian Product

Joins in SQL.

ALERT: SQL does a Cartesian product of all the tables in the FROM clausule unless you specify the attributes to do the the joins. You must to match attributes in common, usually PK and FK.

3.5 Examples

Example1

S

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

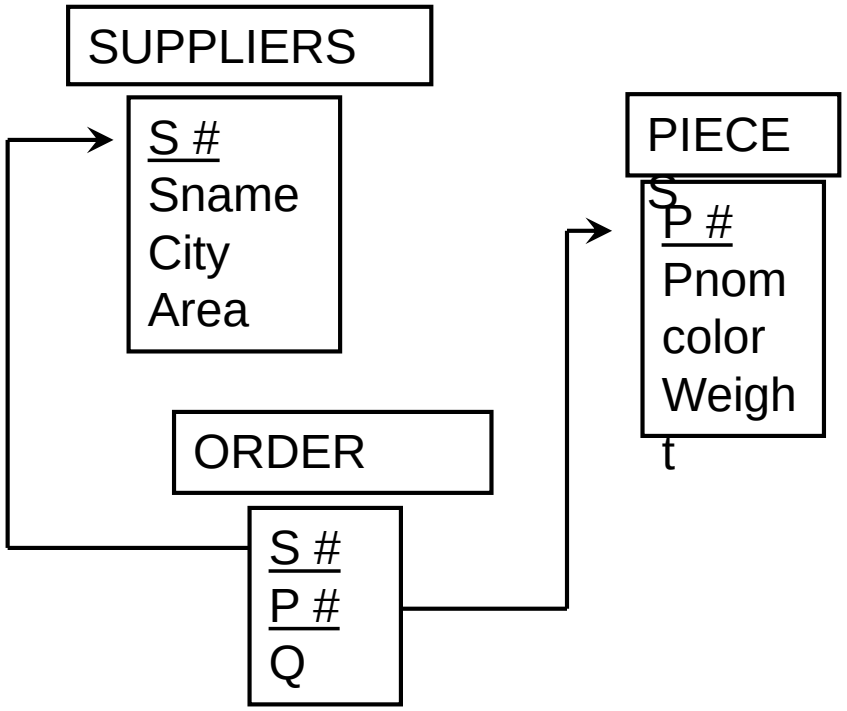
Join (S,SP)

SP

S #	P #	Q
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 #	Sname	zone	city	P #	Q
S1	Salazar	20	London	P1	300
S1	Salazar	20	London	P2	200
S1	Salazar	20	London	P3	400
S1	Salazar	20	London	P4	200
S1	Salazar	20	London	P5	100
S1	Salazar	20	London	P6	100
S2	Jaime	10	Paris	P1	300
S2	Jaime	10	Paris	P2	400
S3	Bernal	30	Paris	P3	200

DB Suppliers:



1. Names of suppliers who supplied the piece P2
2. Cities who received orders above 350 units.

Example2

Names of suppliers who supplied the piece P2

Suppliers

S

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

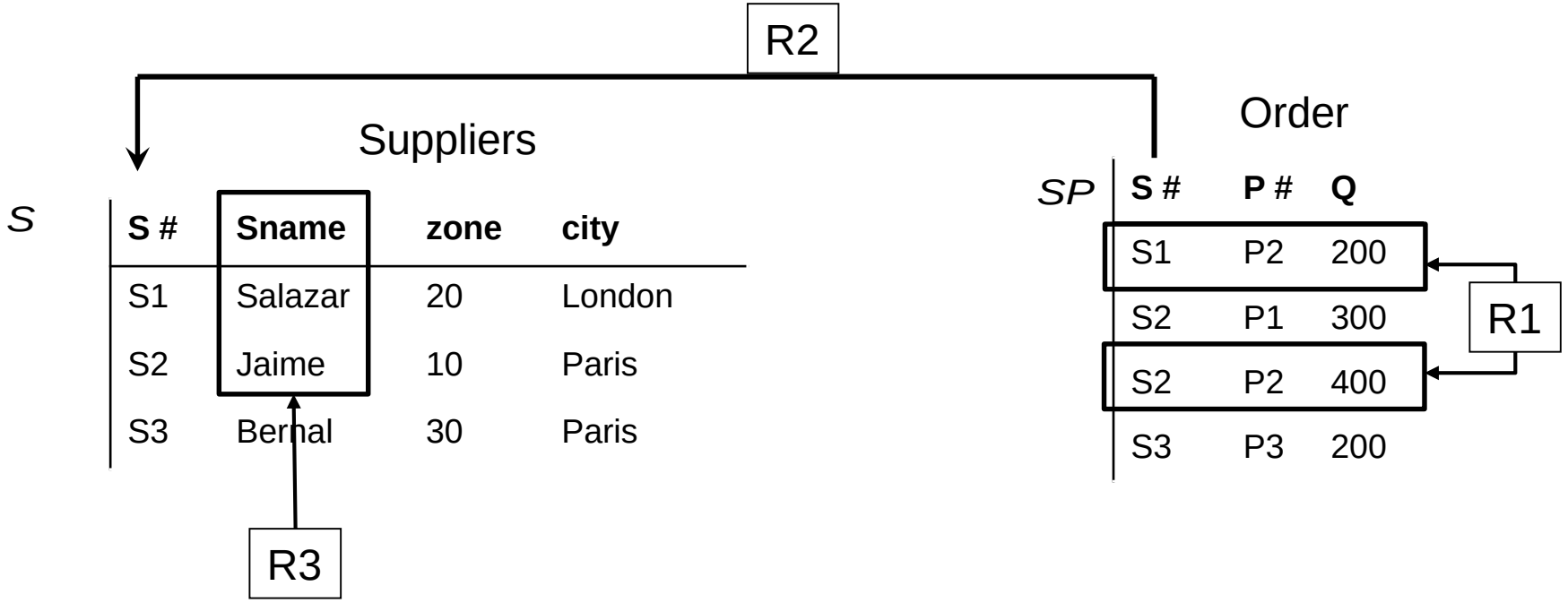
Order

SP

S #	**P #**	**Q**
S1	P2	200
S2	P1	300
S2	P2	400
S3	P3	200

Example2

Names of suppliers who supplied the piece P2



R1 =Restriction(SP | P# = 'P2')
R2 = Join(S, R1 | S.S# = R1.S#)
R3 = Projection(R2 | Sname)

Example2

Names of suppliers who supplied the piece P2

Suppliers

S

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

Order

SP

S #	P #	Q
S1	P2	200
S2	P1	300
S2	P2	400
S3	P3	200

RA solutions:

R1 = Restriction(SP | P # = 'P2')
R2 = Join(S, R1 | S.S # = R1.S#)
R3 = Projection(R2 | Sname)

SQL Code:

```
SELECT S.Sname
FROM suppliers S, Order SP
WHERE SP.P# = 'P2' and SP.S# = S.S#
```

Example3

Cities who received orders above 350 units.

Suppliers

S

S #	**Sname**	**zone**	**city**
S1	Salazar	20	London
S2	Jaime	10	Paris
S3	Bernal	30	Paris
S4	Pitter	18	Rome

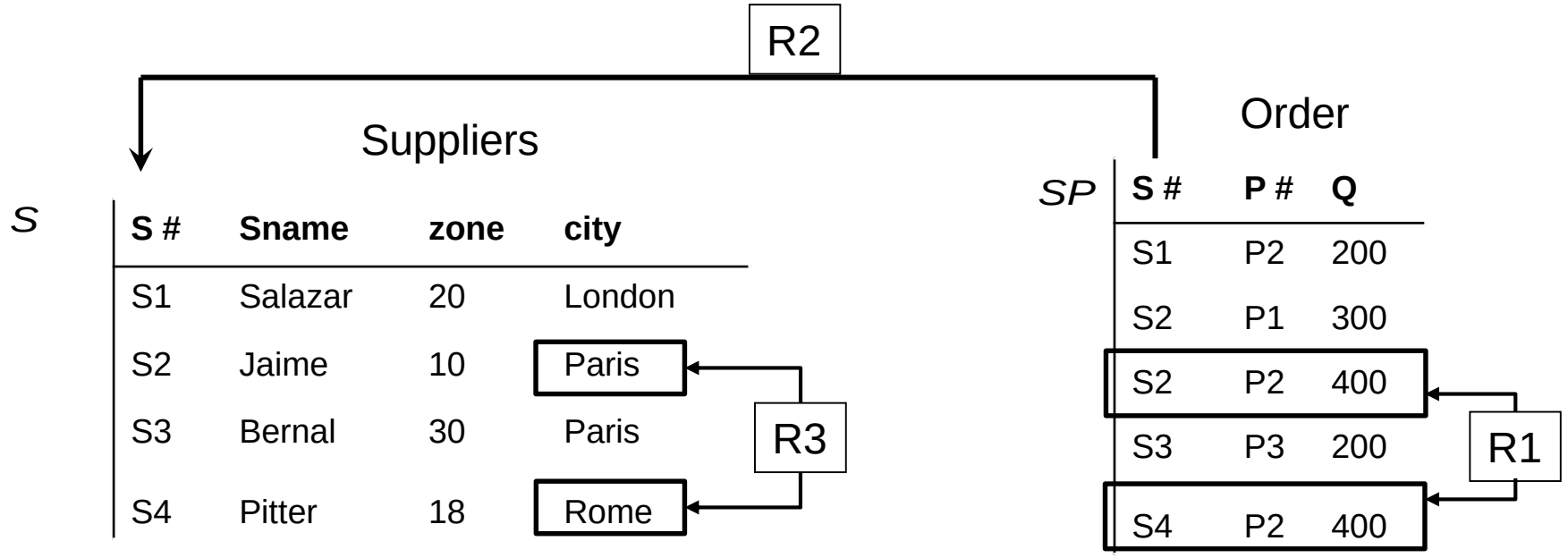
Order

SP

S #	**P #**	**Q**
S1	P2	200
S2	P1	300
S2	P2	400
S3	P3	200
S4	P2	400

Example3

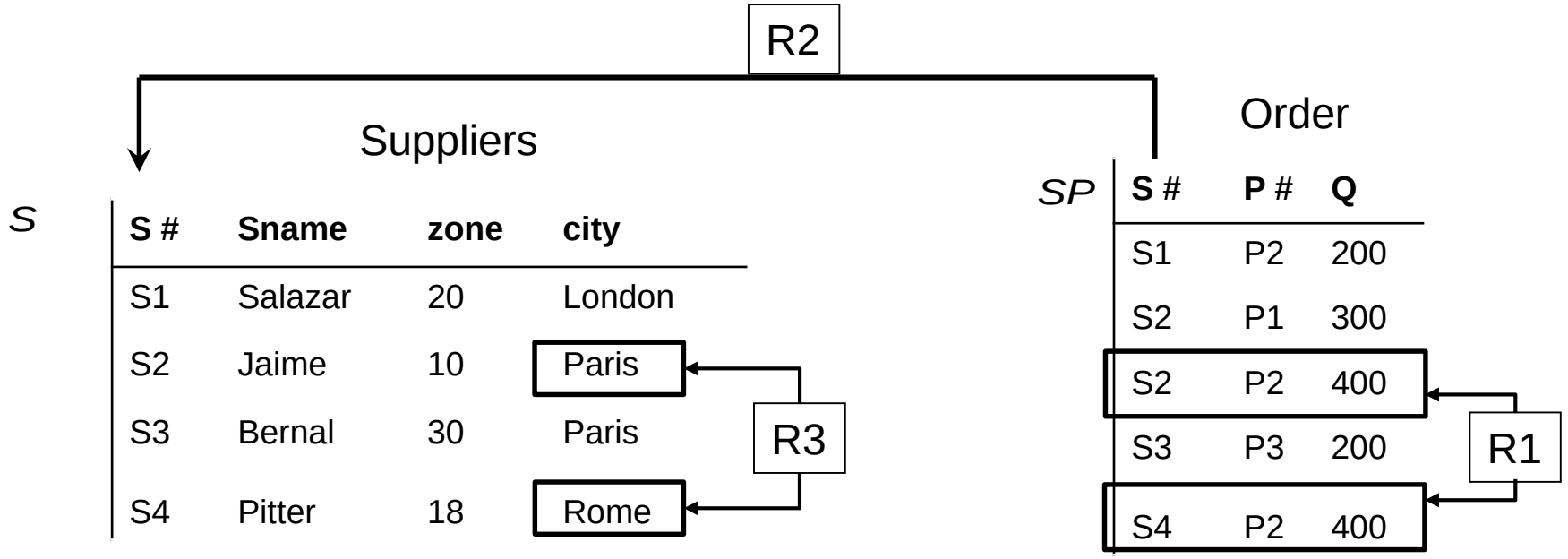
Cities who received orders above 350 units.



R1 = Restriction(SP | Q> 350)
R2 = Join(S, R1 | S.S # =R1.S#)
R3 = Projection(R2 | city)

Example3

Cities who received orders above 350 units.



RA:
R1 = Restriction(SP | Q> 350)
R2 = Join(S, R1 | S.S # = R1.S#)
R3 = Projection(R2 | city)

SQL:
SELECT city
FROM S, SP
WHERE AND S.S# =SP.S# AND SP.Q > 350

3.6 Exercises

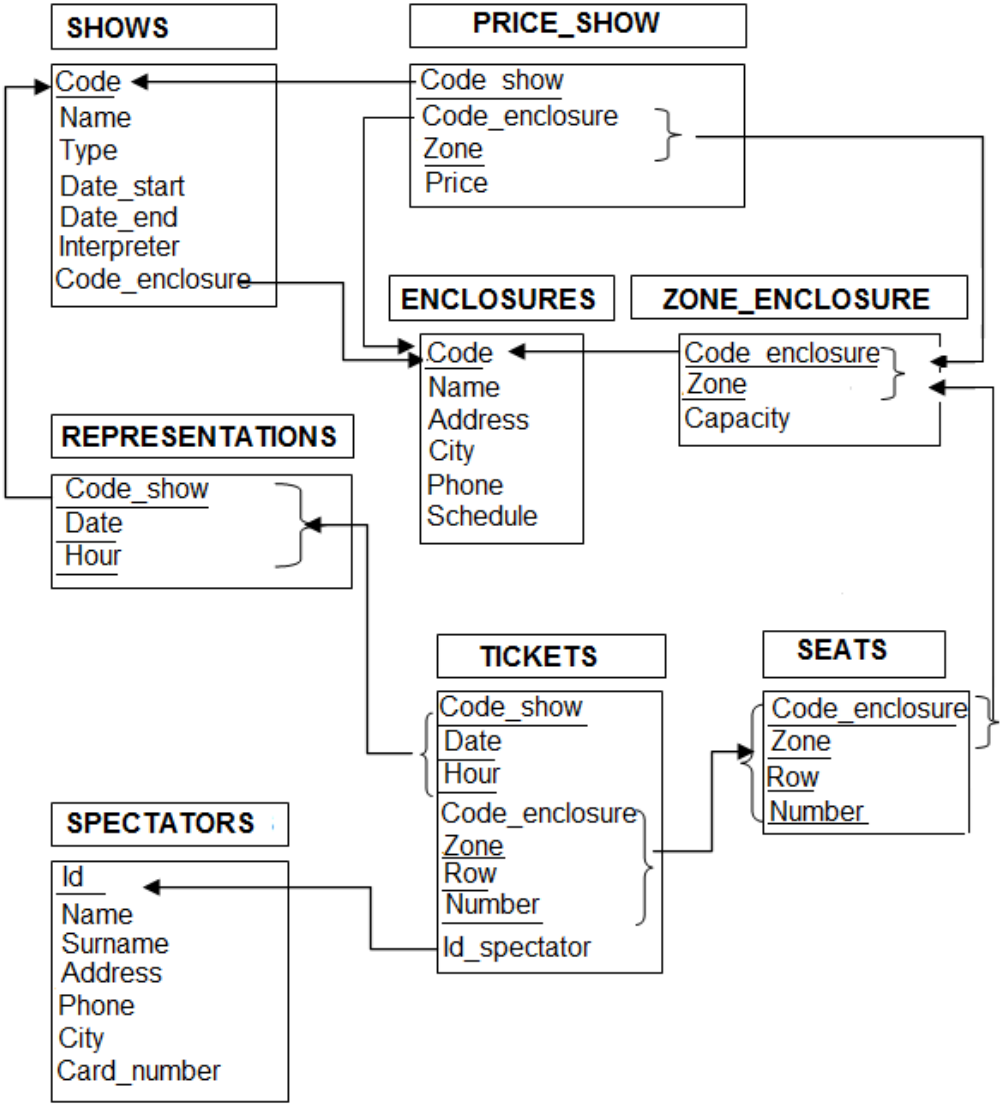
Example 1: DB Shows

1. Name of the shows that take place at “Romea theater” enclosure.

2. Name of the enclosures with zone Lateral or Floor.

3. Type of shows which start in January 2013.

4.Name of the enclosures in the city of Sabadell with capacity of 120 seats.



1. Name of the shows that take place at Romea theater.

```
R1 = Restriction(Enclosures | name = Romea theater
R2 = Join(Shows, R1 | S.Code_enclosure = R1.Code)
R3 = Projection(R2 | Name)
```

```
Select s.Name
from shows s, enclosures e
where  s.code_enclosure = e.Code
and e.Name = 'Romea theater'
```

2. Name of the enclosures with zone Lateral or Floor.

```
R1 = Restriction(Zone_enclosure| zone = 'Lateral' or zone= 'Floor')
R2 = Join(Enclosures, R1 | E.Code = R1.Code_enclosure)
R3 = Projection(R2 | Name)
```

```
Select e.Name
from enclusures e, zone_enclosure z
where e.Code = z.Code_enclosure
and (z.Zone = 'Lateral' or z.Zone = 'Floor')
```

3. Type of shows which start in January 2013

```
R1 = Restriction(Shows | dateStart >=01/01/2013
                         and dateStart <=31/01/2013)
R2 = Projection(R1 | Type)
```

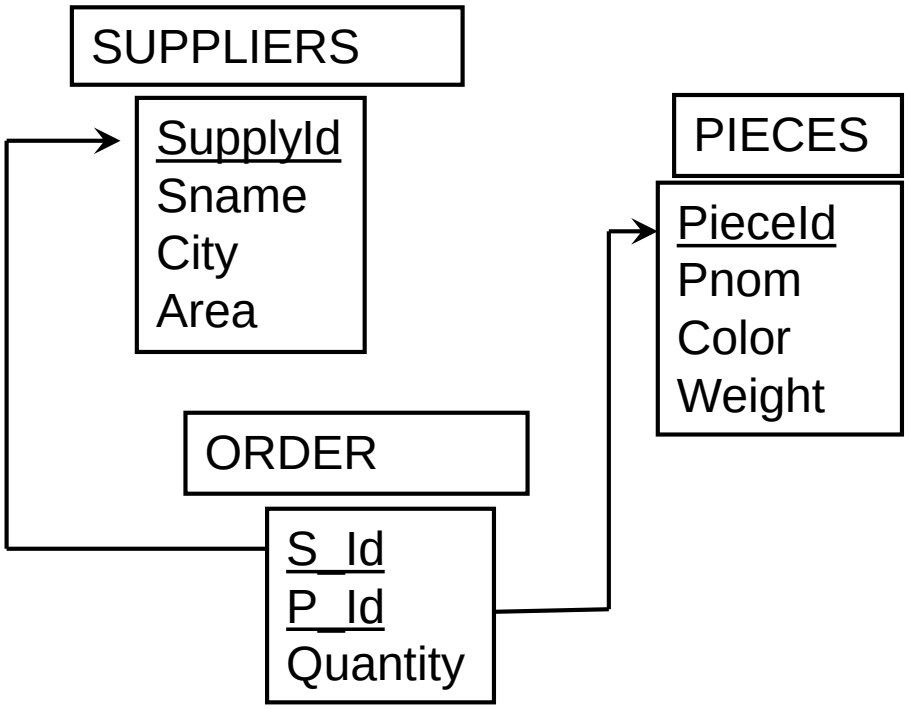

```
Select distinct type
from Shows
where dateStart between to_date('01/01/2013','DD/MM/YYYY')
and to_date('31/01/2013','DD/MM/YYYY')
```

4. Name of the enclosures in the city of Sabadell with capacity of 120 seats.

```
R1 = Restriction(Enclosures | city= Sabadell)
R2 = Restriction(Zone_enclosure  | capacity >= 120)
R3 = Join(R1, R2 | R1.Code = R2.Code_enclosure)
R4 = Projection(R3 | Name)
```

```
Select Name
from Enclosure e, Zone_enclosure z
where e.Code = z.Code_enclosure
and e.city= 'Sabadell' and z.capacity >= 120
```

Example 2:
BD Suppliers



1. Name of suppliers who supply at least one red piece
2. Names of suppliers supplied more than 200 red pieces
3. Suppliers that are not from London
4. What pieces are not supplied by S1?
5. Suppliers that not supply red pieces.
6. Suppliers that supply all blue pieces

1.
```
Select Sname
from Supplier s, Order o, Pieces p
where s.SupplyId = o.S_id
and o.P_id = p.PieceId
and p.Color = 'Red'
```

2.
```
Select Sname
from Supplier s, Order o, Pieces p
where s.SupplyId = o.S_id
and o.P_id = p.PieceId
and p.Color = 'Red'
and o.Quantity >200
```

3.
```
Select Sname
from Suppliers s
where City <> 'London'
```

4.

```
Select p.Pnom
from Order o, Piece p
where
o.P_id = p.PieceId
and o.S_id <> 'S1'
```

5.

```
Select s.Pnom
from Piece p, Order o, Suppliers s
where
o.P_id = p.PieceId
and o.S_id = s.SupplyId
and p.Color <> ‘red‘
```

6.

```
Select s.Pnom
from Piece p, Order o, Suppliers s
where
o.P_id = p.PieceId
and o.S_id = s.SupplyId
and p.Color =‘blue‘
```