escola d'enginyeria

UAB

ESCI upf. School of International Studies

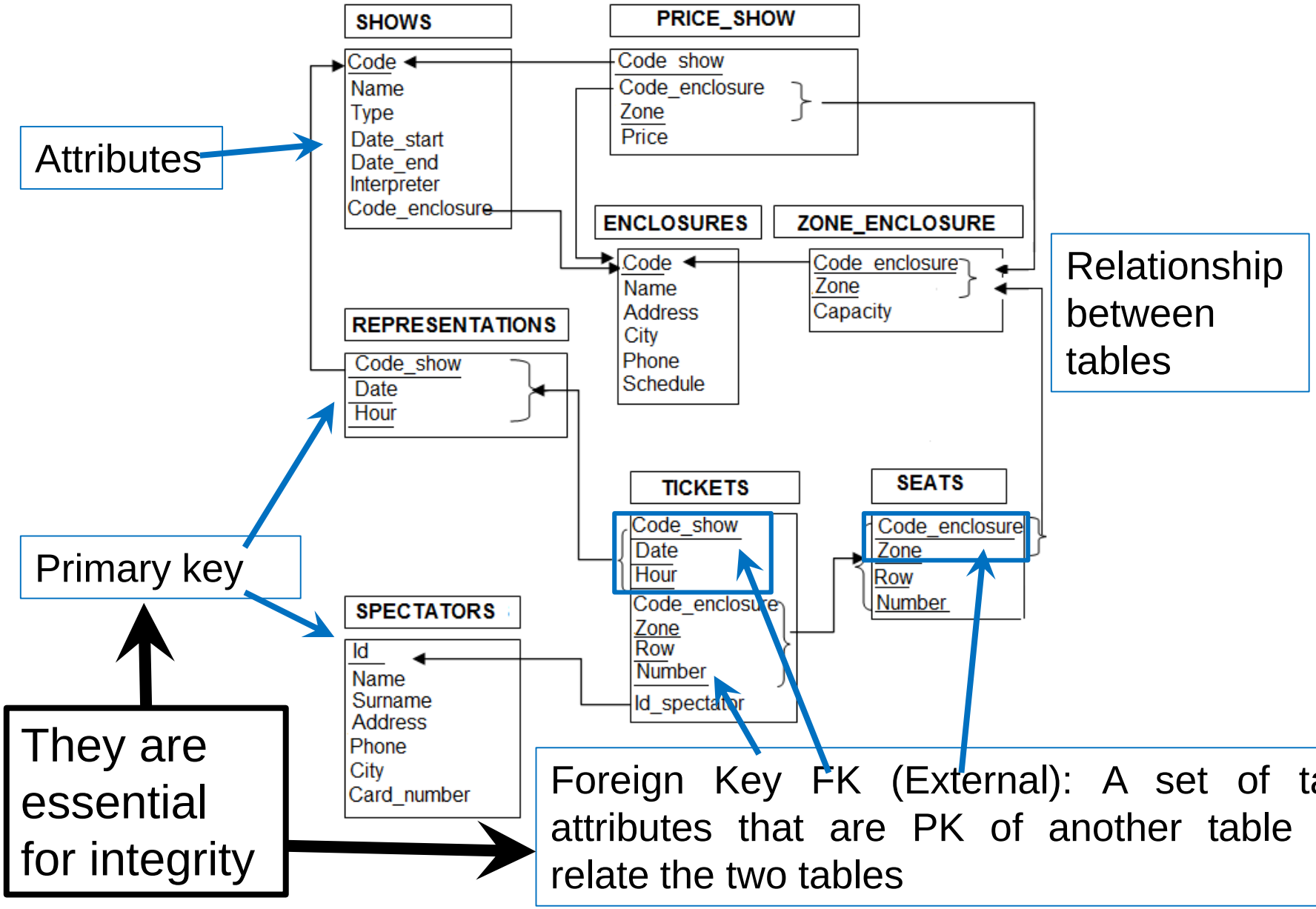
Block 3
DATA INTEGRITY

Debora Gil, Oriol Ramos, Carles Sanchez

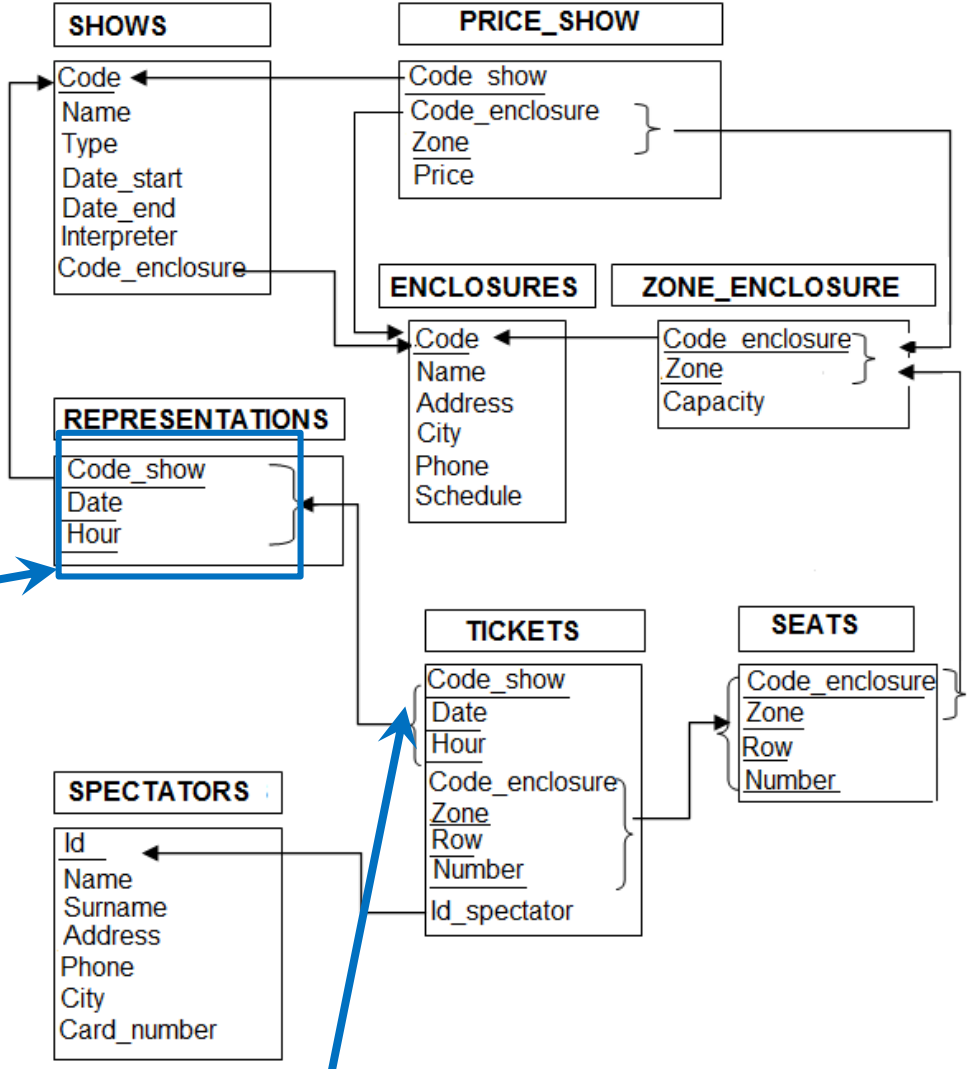
1 Relational Database

Database

Collection of related tables (referenced, linked)



Database Integrity



Relationship between tables (FK referencing to a PK)

FK contains all the PK attributes of the table referred (pointed to)
The arrow must always point to PK to ensure propagation of changes

2. Integrity Rules

2.1 Basic Concepts

2.2 Domains

2.3 Relationships

2.4 Referential

2.1 Basic Concepts

Definition

Rules (constraints) used to ensure validity and accuracy of a data set.

Most are incorporated into the DBMS

Two Levels:

- Specifics
- Generals

Specific Rules

- Specific to the type of data that each DB contains.

- They are defined through specific domains.

Examples:

- Weight is positive
- Boarding gates are identified by a letter
- Number of pieces per box are multiple of 100

General Rules

Rules applicable to any type of data and BD

Three rules for data integrity:

- **Domain** Integrity
- **Entity** Integrity
- **Reference** Integrity

2.1 Domain Integrity

Domain Integrity

Every value of an attribute must belong to the domain on which we defined it.

Ex: Data type (integer, character or decimal)

Allowed length of data

Null support

When is not possible validate the domain integrity

Application-level checking

2.3 Entity Integrity

2.3.1 Motivation

2.3.2 Primary Key

2.3.3 Entities integrity

2.3.4 SQL Syntax

2.3.1 Motivation

The tuples (Rows) in a table represent objects / individuals in the real world

The real-world objects are identifiable and distinguishable from others (uniqueness)

As a representant of a real object each tuple is unique (although some attribute values match those of another tuple)

2.3.2 Primary Key (PK)

Candidate Keys (CK): Minimum set (there are not redundancies) of attributes that uniquely identify each instance

Primary key (PK): CK chosen by the designer

2.3.2 Primary Key: Example1

Copy
ISBN
Copy #
Barcode
MaxDuration
Penalisation

REQUIREMENTS: The ISBN is a code that is assigned according to the title, author and book publisher.

The barcode is a library internal encoding that is assigned to each copy

Identify CKs and PK

2.3.3 Entity Integrity

No component of the primary key can be null

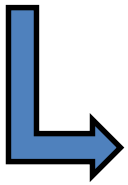
$$\text{NULL} \equiv \text{Missing Information}$$

NULL is equivalent to an existint object without identification or with an unknown identification

2.3.3 Entity Integrity: Remarks

DBMS checks integrity when the primary key is defined

If you do not specify primary key, some DBMS assign PK automatically. Enter an extra attribute to enumerate the tuples.



Automatic numbering is not recommended:

It depends on the insertion order (Independence)

It has no semantic meaning (Domain)

Enter redundancies and dependencies between data

2.3.4 SQL Sintax

CREATE TABLE <nameT> (

<nameA1> <domain1> <values>,

....

<nameAk1><domaink1> NOT NULL,

<nameAk2><domaink2> NOT NULL,

....

<nameAN> <domainN> <values>,

CONSTRAINT nameT_PK PRIMARY KEY (nameAk1,nameAk2)

);

2.3.4 Syntax SQL: Example1

Simple PK

```
CREATE TABLE PLACE(
 code NUMBER NOT NULL,
 name VARCHAR2 (50)
 Address VARCHAR2 (50)
 Telephone NUMBER
 Timetable VARCHAR2 (50)

 Constraint PLACE_PK PRIMARY KEY ( code ) )
```

2.3.4 Syntax SQL: Exemple2

Compound PK

```
CREATE TABLE PRICE_SHOW (
 Code_show NUMBER NOT NULL,
 Code_enclosure NUMBER NOT NULL,
 Zone VARCHAR2 (20) NOT NULL,
 Price NUMBER,
 Constraint Price_Show_PK
 PRIMARY KEY ( Code_show, Code_enclosure_zone) )
```

2.4 Reference Integrity

2.4.1 Motivation

2.4.2 Foreign Key (FK)

2.4.3 Reference Integrity

2.4.4 SQL Syntax

2.4.5 Referential Diagram

2.4.1 Motivation

Relational databases are a set of referenced tables

Student

NIA	Name	DNI	Tel
1	Pepe	46956514	657545454
2	Joan	56056512	666666666

Subject

PK_Sub	Name	Classroom
100	BD	1011
202	IA	1013
305	SO	2010

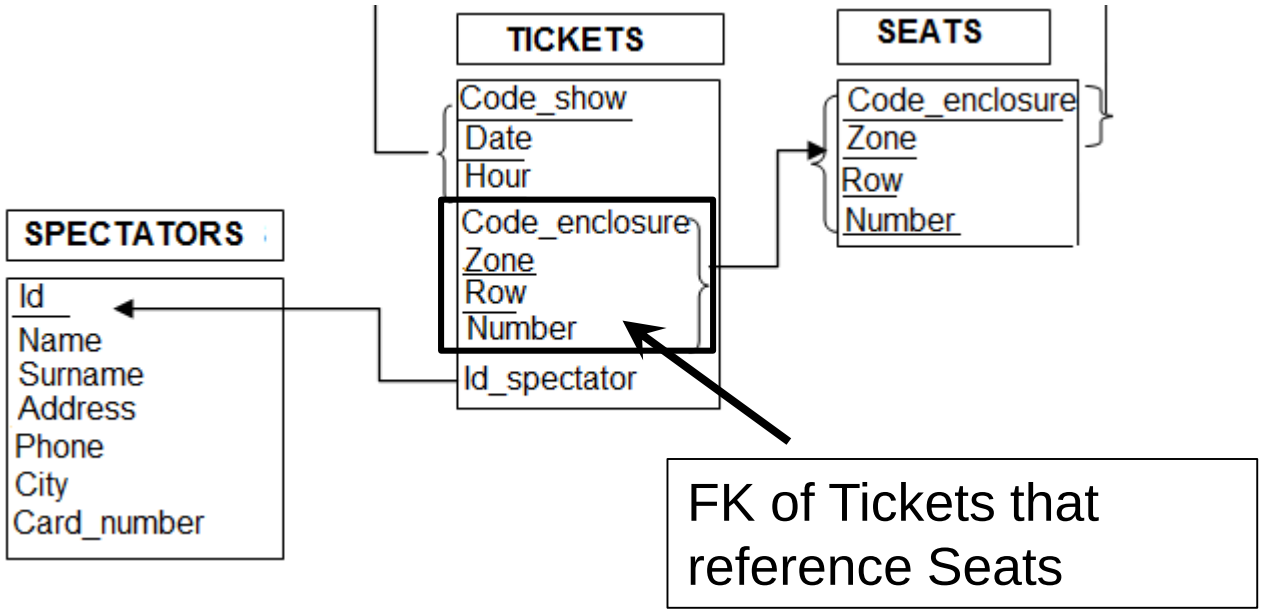
References to tables containing data

PK_St	PK_Sub
1	100
2	100
2	202
2	305

Enrollment

2.4.2 Foreign Key (FK)

Set of attributes of a relation R2 whose values are the same values as another PK takes in a relationship which refers to R1



OBS: PK and FK must be of the same domain

2.4.2 FK: Concepts

Referenced relation. Table containing PK

Referenced tuple. Tuple where exists PK value of FK

Student

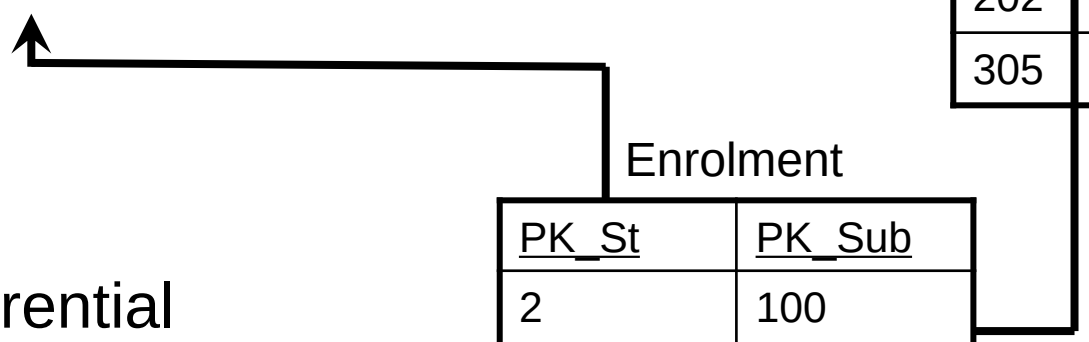
NIA	Name	DNI	Tel
1	Pepe	46956514	657545454
2	Joan	56056512	666666666

Subject

PK_Sub	Name	Classroom
100	BD	1011
202	IA	1013
305	SO	2010

Enrolment

PK_St	PK_Sub
2	100
1	100
2	202
2	305



Referential Relationship.

Value containing the FK

2.4.3 Reference Integrity:

Basic Concepts

1. There may be primary key value of R1 that are not referenced by any foreign key of R2

Examples:

- There can be subjects with no students enrolled
- Students can be NOT enrolled

2.4.3 Reference Integrity:

Basic Concepts

2. A value of FK in a R2 relation referencing to a R1 relation is only allowed if this value appears in the primary key of one of the tuples of R1

Example:

- It makes no sense that a student enrols for a subject that does not exist.
- We can not sell tickets for a flight to the Moon

2.4.3 Reference Integrity: Definition

The DB can not have values (NO_NULL) for foreign key

non-primary key of a tuple of the referenced relation

If R2 refers to R1 → R1 must exist

The DBMS makes checking whether foreign keys are specified

2.4.3 SQL Syntax

Constraint <nameFK> **FOREIGN KEY** (<AttributseFK>) REFERENCES <nameReferencedTable> (<AttributesPKReferencedTable>)

NULL [NOT] ALLOWED
DELETE OF < nameReferencedTable > <effect>
UPDATE OF < AttributesPKReferencedTable > < effect >

Where <effect> can be
RESTRICTED |
CASCADES |
NULLIFIES

2.4.3 SQL Syntax: Example

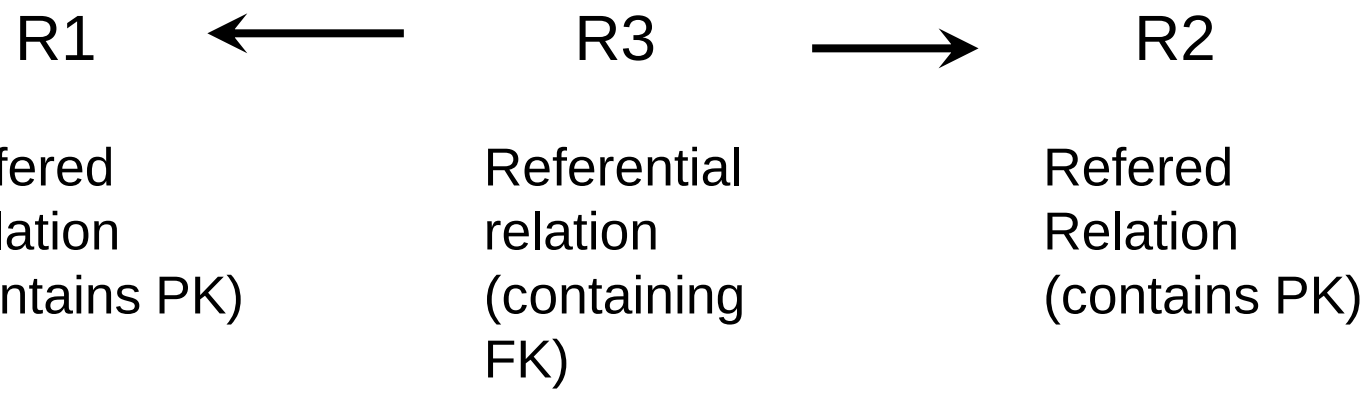
FKs are defined once all tables have been created using the following syntax

ALTER TABLE <nameTable> ADD Constraint

```
ALTER TABLE AUXILIAR ADD CONSTRAINT
AUXILIAR_PERSONAL_FK FOREIGN KEY ( SS ) REFERENCES
PERSONAL ( SS ) ;
```

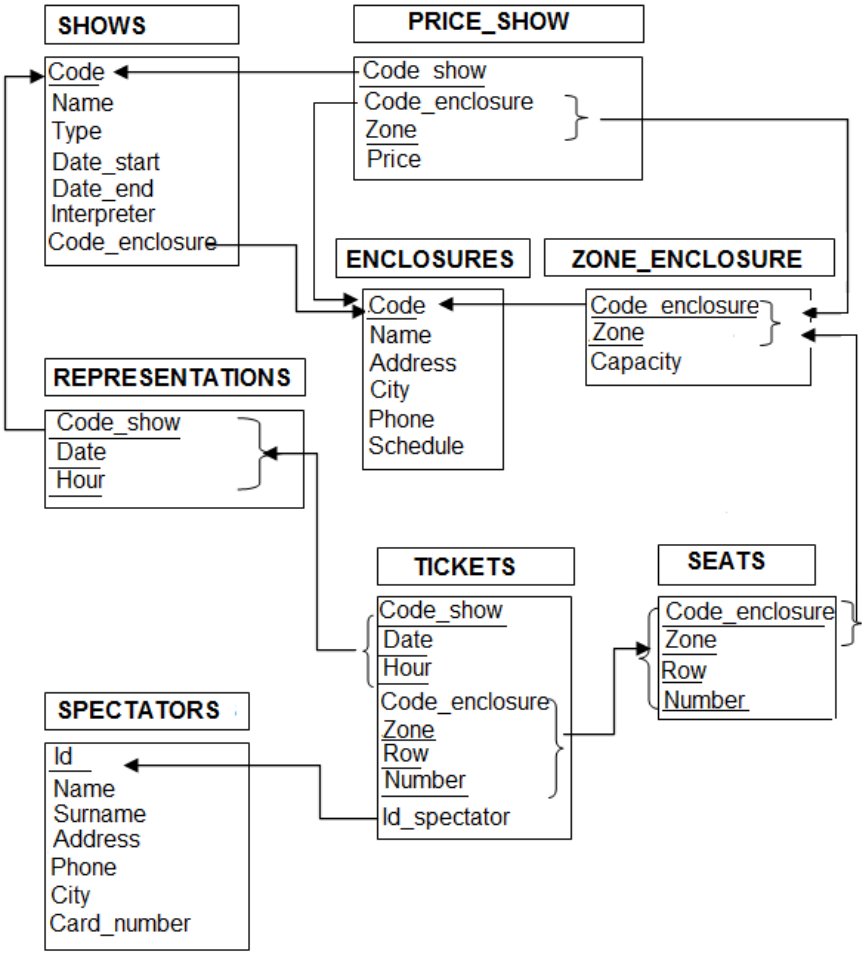
```
ALTER TABLE TICKETS ADD CONSTRAINT TICKETS_TRIP_FK
FOREIGN KEY ( Code_ticket, Passenger ) REFERENCES
TRIP( Code_ticket, NIF_Passenger ) ;
```

2.4.4 Referential Diagram



Relations can be labeled with foreign keys:

2.4.4 Referential Diagram



A relation can be referenced and referential

$$R3 \xrightarrow{b} R2 \xrightarrow{a} R1$$

Referential Path:

Referential chain constraints

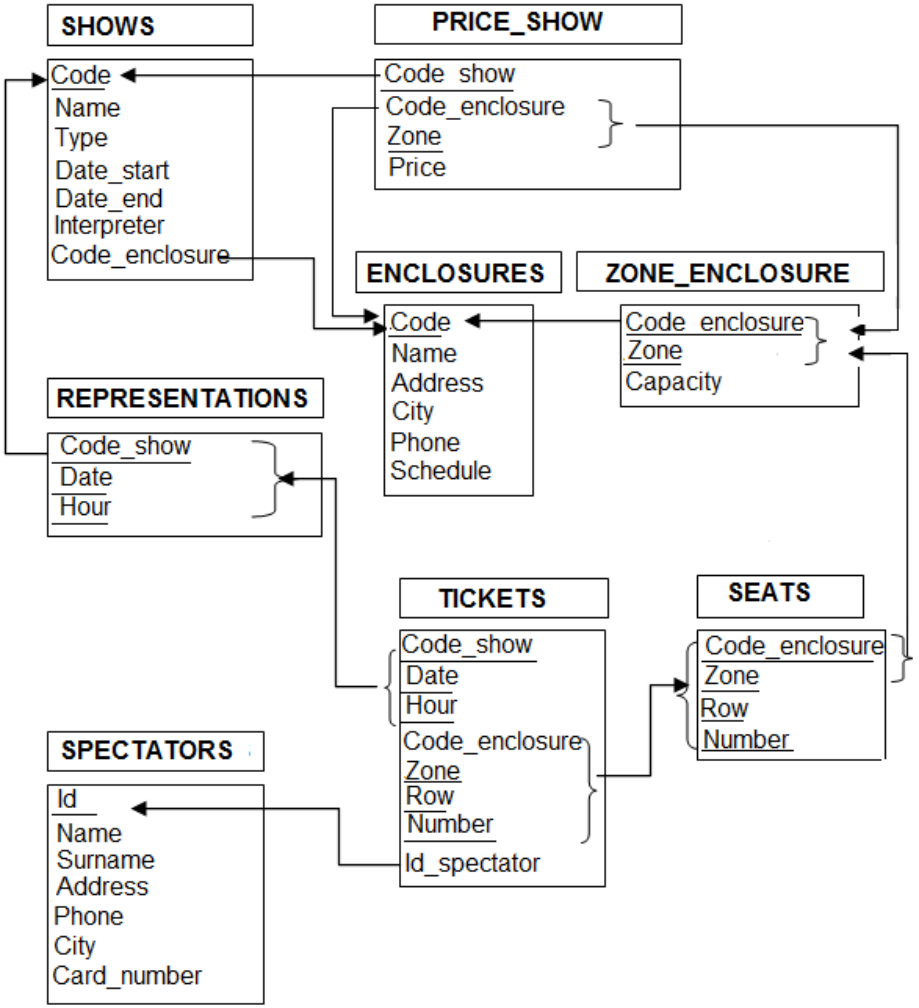
Determines the order to follow when it modifies (Inserts, Deletes) DB

$$R(n) \longrightarrow R(n-1) \longrightarrow \cdots \longrightarrow R2 \longrightarrow R1$$

2.4.4 Referential diagram: Example

Referential Diagram and insert / delete order:

1. A enclosure
2. A zone enclosure
3. A seat



3. Propagation Constrains

3.1 Motivation

3.2 Null allowed (Nullify)

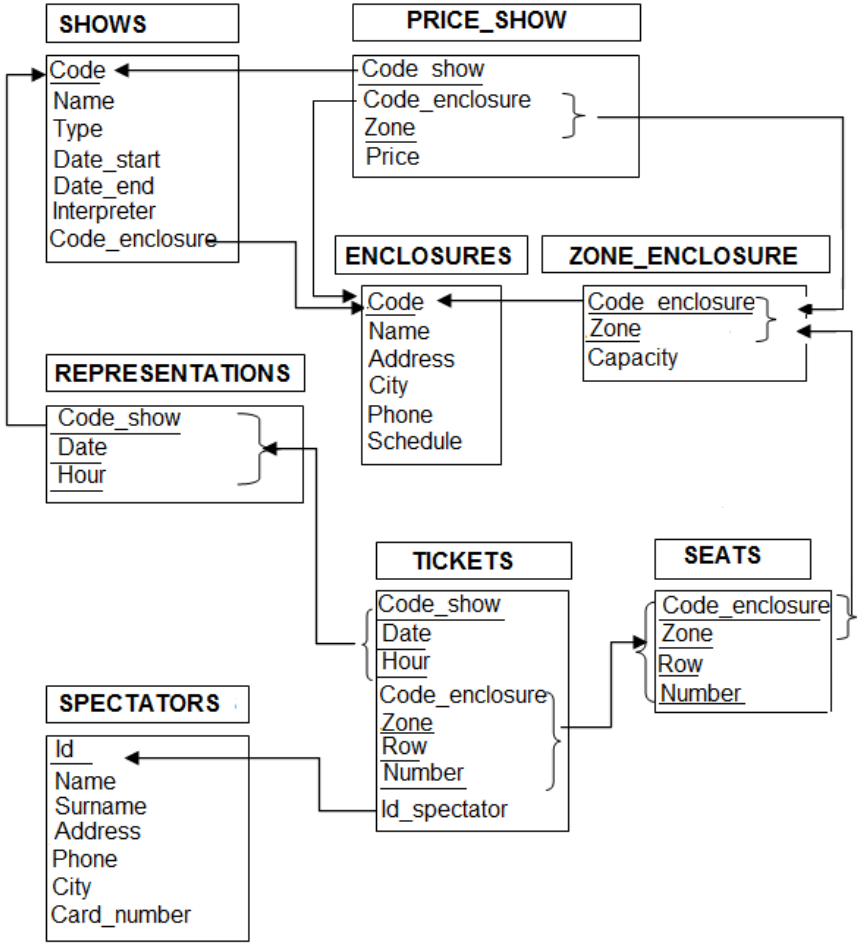
3.3 Delete

3.4 Updates (UpDate)

3.5 Examples

3.1 Motivation

What happens if we delete a referenced tuple?



3.1 Motivation

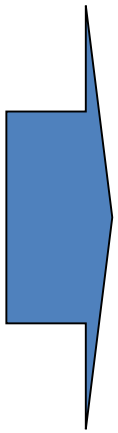
Foreign keys determine the link between tables

Modifying a referenced tuple affects ALL those referring to it

3.1 Motivation

Rules for foreign keys to guarantee reference integrity

- NULLS acceptation
- Tuples deletion (Delete)
- PK modification (Update)



Restrict (Restricted)

Propagate (Cascade)

Accept nulls (Nullifies)

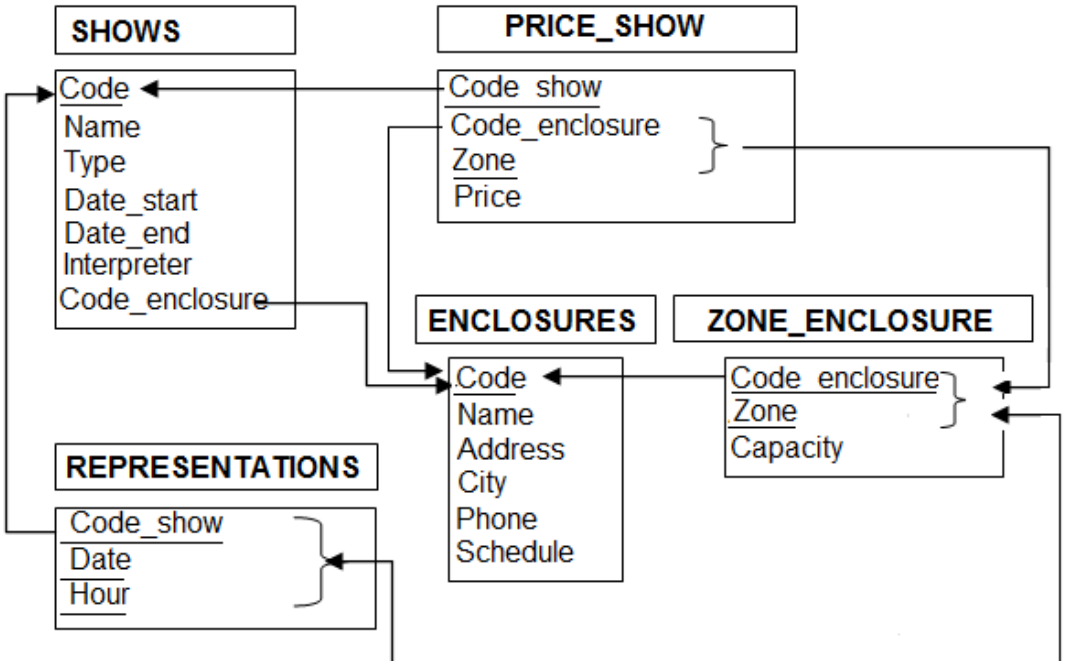
3.2 NULLS Acceptation

NULLS Acceptation

Admitting NULL in a FK not always make sense. It depends on the meaning of the BD and the data policy to be modelled

SQL: By default FK supports NULLs

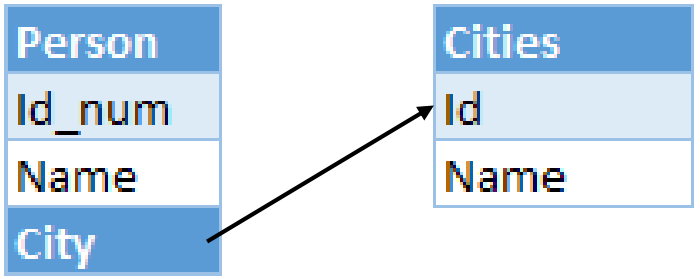
A show without a venue (price_show) could make sense in an event organization company.



NULL values meaning

FK can accept NULL values

→ What does it mean?



Id_num	Name	City
234234234	Lucia	002
453454534	Paul	~~003~~ NULL
585765837	Ernest	001

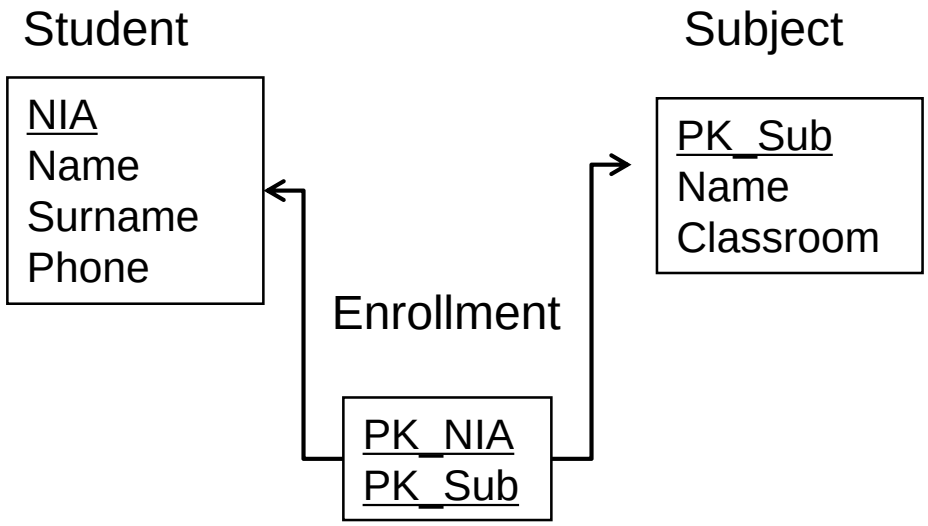
Id	Name
001	Nápoles
002	Barcelona
~~003~~	~~Paris~~

If we delete a city which is referenced in Person table. Their value is replace by NULL.

If FK consists of more than one attribute, they must be entirely NULLs or NOT NULLs

NULL values meaning

A FK may be part of the PK of the referential relation (should it?)



In this case the Enrollment table must not allow Null values in their FKs

3.3 Delation (Delete)

3.3.1 Restricted

3.3.2 Cascade

3.3.3 Nullify

3.3.1 Restricted

- Delete S2 supplier from S Relation

S

S#	Snom	Zona	Ciutat
S1	Salazar	20	Londres
S2	Jaime	10	Paris
S3	Bernal	30	Paris
S4	Corona	20	Londres
S5	Aldana	30	Atenas

SP

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

P

P#	Pnom	Color	Pes	Ciutat
P1	Femella	Vermell	12	Londres
P2	Pern	Verd	17	Paris
P3	coixinet	Blau	17	Paris

3.3.2 Cascade

- Delete S2 supplier from S Relation

⇨ **Deleted in S, SP**

S

S#	Snom	Zona	Ciutat
S1	Salazar	20	Londres
S2	Jaime	10	Paris
S3	Bernal	30	Paris
S4	Corona	20	Londres
S5	Aldana	30	Atenas

SP

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

P

P#	Pnom	Color	Pes	Ciutat
P1	Femella	Vermell	12	Londres
P2	Pern	Verd	17	Paris
P3	coixinet	Blau	17	Paris

3.3.2 Cascade

- Delete S2 supplier from S Relation

S

S#	Snom	Zona	Ciutat
S1	Salazar	20	Londres
S2	Jaime	10	Paris
S3	Bernal	30	Paris
S4	Corona	20	Londres
S5	Aldana	30	Atenas

SP

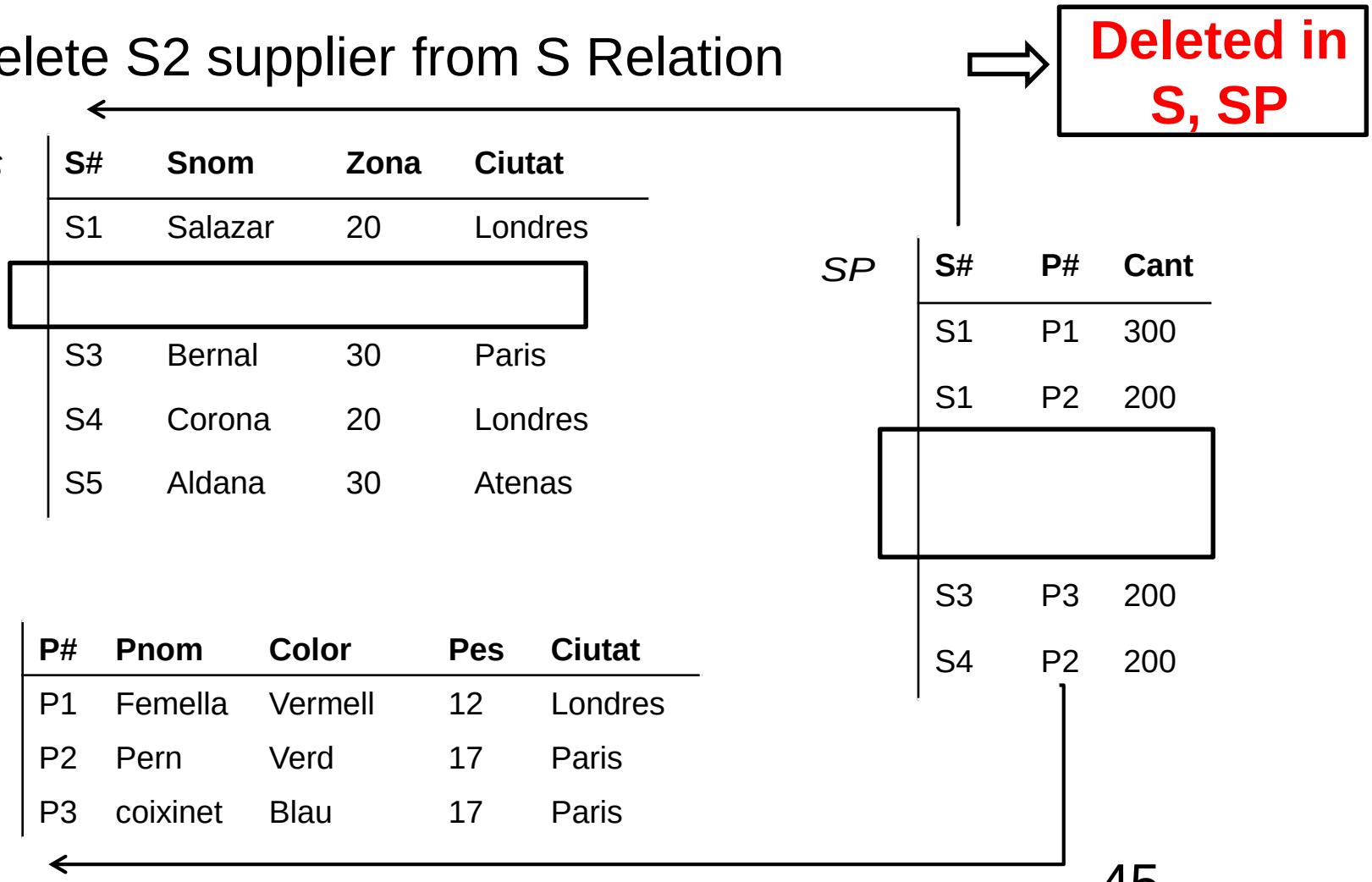
S#	P#	Cant
S1	P1	300
S1	P2	200
S3	P3	200
S4	P2	200

P

P#	Pnom	Color	Pes	Ciutat
P1	Femella	Vermell	12	Londres
P2	Pern	Verd	17	Paris
P3	coixinet	Blau	17	Paris

3.3.2 Cascade

- Delete S2 supplier from S Relation



3.3.3 Nullifies

- Delete S2 supplier from S Relation

⇨ **NULL values in SP**

S

S#	Snom	Zona	Ciutat
S1	Salazar	20	Londres
S2	Jaime	10	Paris
S3	Bernal	30	Paris
S4	Corona	20	Londres
S5	Aldana	30	Atenas

SP

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

P

P#	Pnom	Color	Pes	Ciutat
P1	Femella	Vermell	12	Londres
P2	Pern	Verd	17	Paris
P3	coixinet	Blau	17	Paris

3.3.3 Nullifies

- Delete S2 supplier from S Relation

⇨ **NULL values in SP**

S

S#	Snom	Zona	Ciutat
S1	Salazar	20	Londres
S2	Jaime	10	Paris
S3	Bernal	30	Paris
S4	Corona	20	Londres
S5	Aldana	30	Atenas

SP

S#	P#	Cant
S1	P1	300
S1	P2	200
NULL	P1	300
NULL	P2	400
S3	P3	200
S4	P2	200

P

P#	Pnom	Color	Pes	Ciutat
P1	Femella	Vermell	12	Londres
P2	Pern	Verd	17	Paris
P3	coixinet	Blau	17	Paris

3.3.3 Nullifies

- Delete S2 supplier from S Relation

NULL values in SP

S

S#	Snom	Zona	Ciutat
S1	Salazar	20	Londres
S3	Bernal	30	Paris
S4	Corona	20	Londres
S5	Aldana	30	Atenas

SP

S#	P#	Cant
S1	P1	300
S1	P2	200
NULL	P1	300
NULL	P2	400
S3	P3	200
S4	P2	200

P

P#	Pnom	Color	Pes	Ciutat
P1	Femella	Vermell	12	Londres
P2	Pern	Verd	17	Paris
P3	coixinet	Blau	17	Paris

3.4 UpDate

3.4.1 Restricted

3.4.2 Nullify

3.4.3 Cascade

3.4.1 Restricted

- Update S2 value from S Relation

S

S#	Snom	Zona	Ciutat
S1	Salazar	20	Londres
S2	Jaime	10	Paris
S3	Bernal	30	Paris
S4	Corona	20	Londres
S5	Aldana	30	Atenas

SP

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

P

P#	Pnom	Color	Pes	Ciutat
P1	Femella	Vermell	12	Londres
P2	Pern	Verd	17	Paris
P3	coixinet	Blau	17	Paris

3.4.2 Cascade

- Update P2 value from P Relation

we change the values in P, SP

S

S#	Snom	Zona	Ciutat
S1	Salazar	20	Londres
S2	Jaime	10	Paris
S3	Bernal	30	Paris
S4	Corona	20	Londres
S5	Aldana	30	Atenas

SP

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

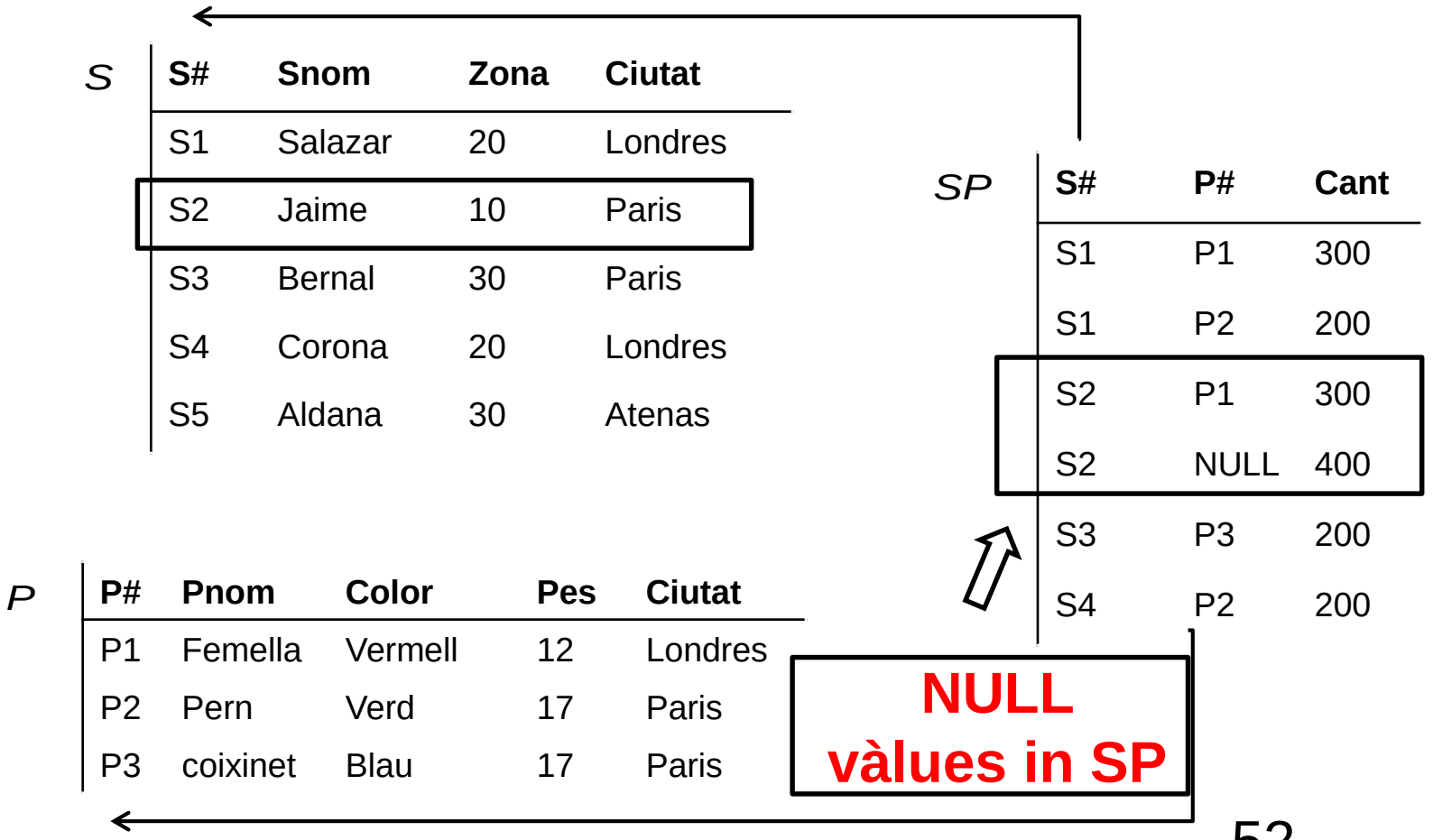
P

P#	Pnom	Color	Pes	Ciutat
P1	Femella	Vermell	12	Londres
P2	Pern	Verd	17	Paris
P3	coixinet	Blau	17	Paris

3.4.3 Nullifies

In what order should it be done?

- Update P2 by P5 value from P Relation



S

S#	Snom	Zona	Ciutat
S1	Salazar	20	Londres
S2	Jaime	10	Paris
S3	Bernal	30	Paris
S4	Corona	20	Londres
S5	Aldana	30	Atenas

SP

S#	P#	Cant
S1	P1	300
S1	P2	200
S2	P1	300
S2	NULL	400
S3	P3	200
S4	P2	200

P

P#	Pnom	Color	Pes	Ciutat
P1	Femella	Vermell	12	Londres
P2	Pern	Verd	17	Paris
P3	coixinet	Blau	17	Paris

NULL vàlues in SP

Observations

- CASCADES foreign key rule can lead to problems

$$R3 \xrightarrow{b} R2 \xrightarrow{a} R1$$

Reference: CASCADES

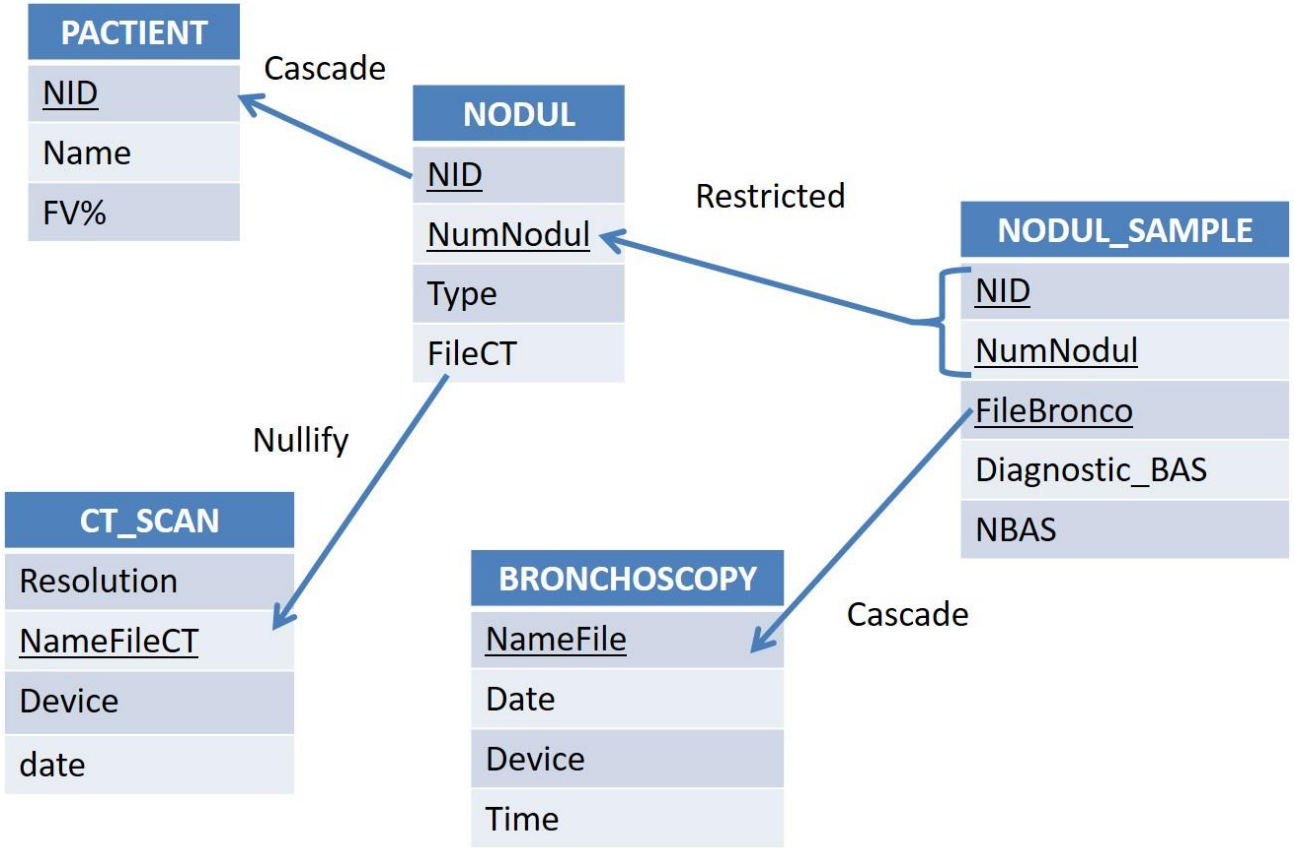
What if we delete tuples at R1?

If the rule a, b does not allow to delete tuples in R3 as a result of removing elements in R2 ⇨ R1 tupla **can not** be deleted

Observations

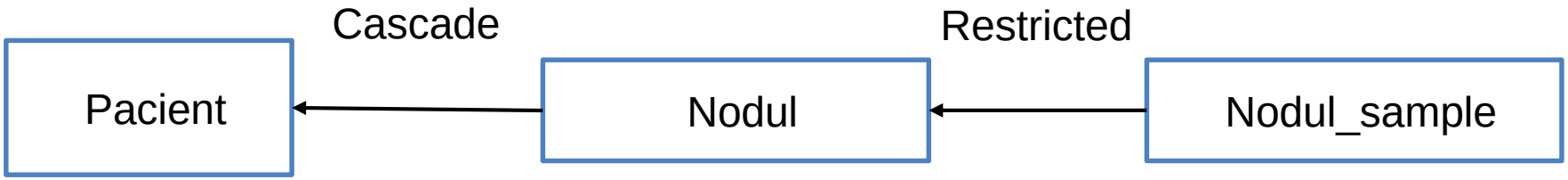
Restricted (default mode) is the rule that has fewer risks and it is easier to implement but less comfortable for the user (deleting with a certain order)

Example


PACTIENT
NID
Name
FV%
Cascade
NODUL
NID
NumNodul
Type
FileCT
Restricted
NODUL_SAMPLE
NID
NumNodul
FileBronco
Diagnostic_BAS
NBAS
Nullify
CT_SCAN
Resolution
NameFileCT
Device
date
BRONCHOSCOPY
NameFile
Date
Device
Time
Cascade

Referential diagram

What happens if we delete a pacient?



1) A patient (NID) could be automatically removed if It doesn't have related nodules.

2) If patient has nodules:

- If these nodules doesn't have related Nodul_sample, these nodules will be automatically removed too (cascade)

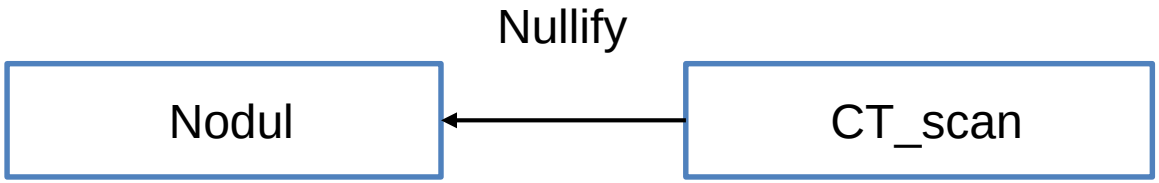
- If these related nodules have Nodul_sample they have to be removed first manually. (Restricted)

The order to delete will be : Nodul_sample

Pacient

What happens if we delete a CT_scan?

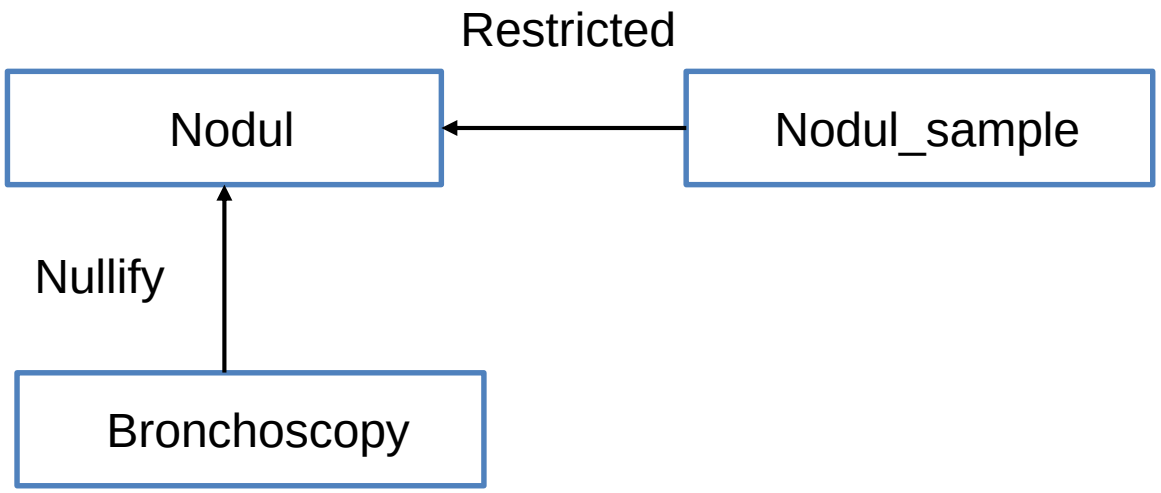
Referential diagram



The value of the field File_CT will be updated asNull in all records which contains the deleted CT.

Referential diagram

What happens if we delete a nodul?



1) *A nodul could be automatically deleted if Nodul_sample doesn´t have any reference of it.*

2) *If the Nodul_sample has the nodul, first we need to remove from it and then from Nodul table because the restricted ruler.*