SAPIENZA
UNIVERSITÀ DI ROMA

Data Management and Analysis
Unit 2

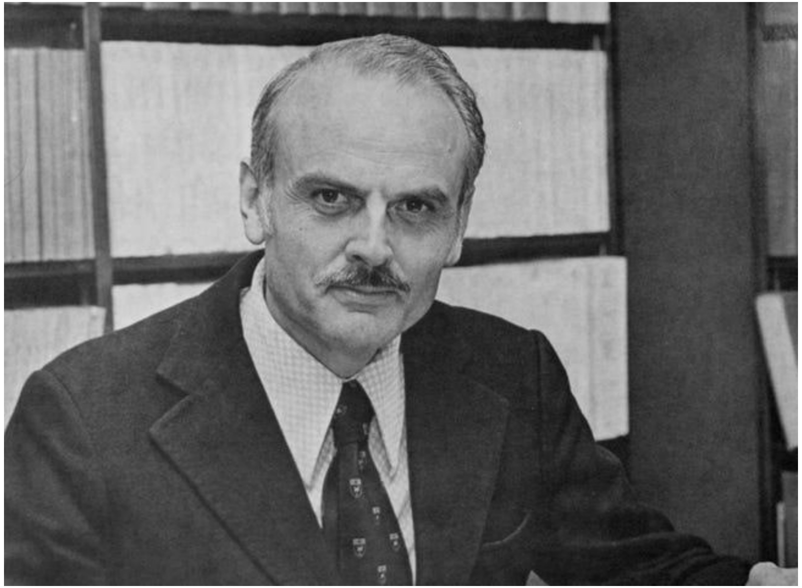
Relational Model

Dott. Franco Liberati
liberati@di.uniroma1.it

RELATIONAL MODEL

General

Proposed in 1970 by Edgard Frank Codd, a researcher at IBM in San Jose, CA.



Relation model ensures independence between levels External and Physical

It is intuitive, and based on basic algebra notions

Very popular (PostgreSQL, Oracle, MySQL, DB2, SQL Server, etc.)

RELATIONAL MODEL

Definitions

The relational model is based on two concepts: **relation** and **table**

FILM

Title	Date	Production	FirstActor
Rambo	1980	USA	Stallone
Batman	1990	USA	Keaton
Spectre	2000	UK	Craig
Flash	2023	USA	Miller
WW2	2019	USA	Gadot

ACTOR

Name	Surname	Nationality
Silvester	Stallone	USA
Robert	De Niro	USA
Daniel	Craig	UK

RELATIONAL MODEL

Definitions

In the Relational Model, **data** is organized into fixed-size **records** (tulples) and divided into **tables** (relationships)

COURSES		
Course	**Code**	**Teacher**
Databases	BD2	G. Bruni
Programming	PO1	D. Gray
Operating Systems	SO1	E. Ros

NB: Column/row sorting is irrelevant (it is not applied the classical definition of mathematical relationship such as cartesian product)

RELATIONAL MODEL

Definitions:attributes, schema, instances

Attributes: table columns 🟥

Relationship **schema**: table name + attribute names 🟩

Relationship **instances**: table row 🟨

COURSES		
Course	**Code**	**Teacher**
Databases	BD2	G. Bruni
Programming	PO1	D. Gray
Operating Systems	SO1	E. Ros

RELATIONAL MODEL

Sample

Attributes

Course, Code, Teacher

Relationship schema

COURSES (Course, Code, Teacher)

Relationship instances

<Databases, BD2, G. Bruni> <Programming, PO1, D. Gray> …

COURSES		
Course	**Code**	**Teacher**
Databases	BD2	G. Bruni
Programming	PO1	D. Gray
Operating Systems	SO1	E. Ros

RELATIONAL MODEL

Definition: domain

Each attribute has a **DOMAIN** that defines the set of valid values for that attribute

Example: DOMAIN(“Course”) = String

DOMAIN(“AccademicYear”) = Integer

COURSES			
Course	**Code**	**Teacher**	**AccademicYear**
Databases	BD2	G. Bruni	2
Programming	PO1	D. Gray	1
Operating Systems	SO1	E. Ros	3

RELATIONAL MODEL

Constraints on the relation

There can be no equal attributes 🟩

No equal rows can exist 🟥

The data in a column must be homogeneous (same domain) 🟨

COURSES			
Course	**Code**	**Teacher**	**Teacher**
Databases	BD2	G. Bruni	G. Bruni
345,56	BD1	M. DeMarsico	M. De Marisco
Programming	PO1	D. Gray	D. Gray
Operating Systems	SO1	E. Ros	E. Ros
Operating Systems	SO1	E. Ros	E. Ros

RELATIONAL MODEL

Constraints on the relation

It is possible to have a **relationship schema** without instances (e.g. during the creation of the DB)

COURSES		
Course	Code	Teacher

It is NOT possible to have instance without schema!

Databases	BD2	G. Bruni
Programming	PO1	D. Gray
Operating Systems	SO1	E. Ros

RELATIONAL MODEL

Definition: relations

A database is generally made up of several **relations**, whose records contain common values where this is necessary in order to establish correspondences

COURSES		
Course	**Code**	**Teacher**
Databases	BD2	G. Bruni
Programming	PO1	D. Gray
Operating Systems	SO1	E. Ros

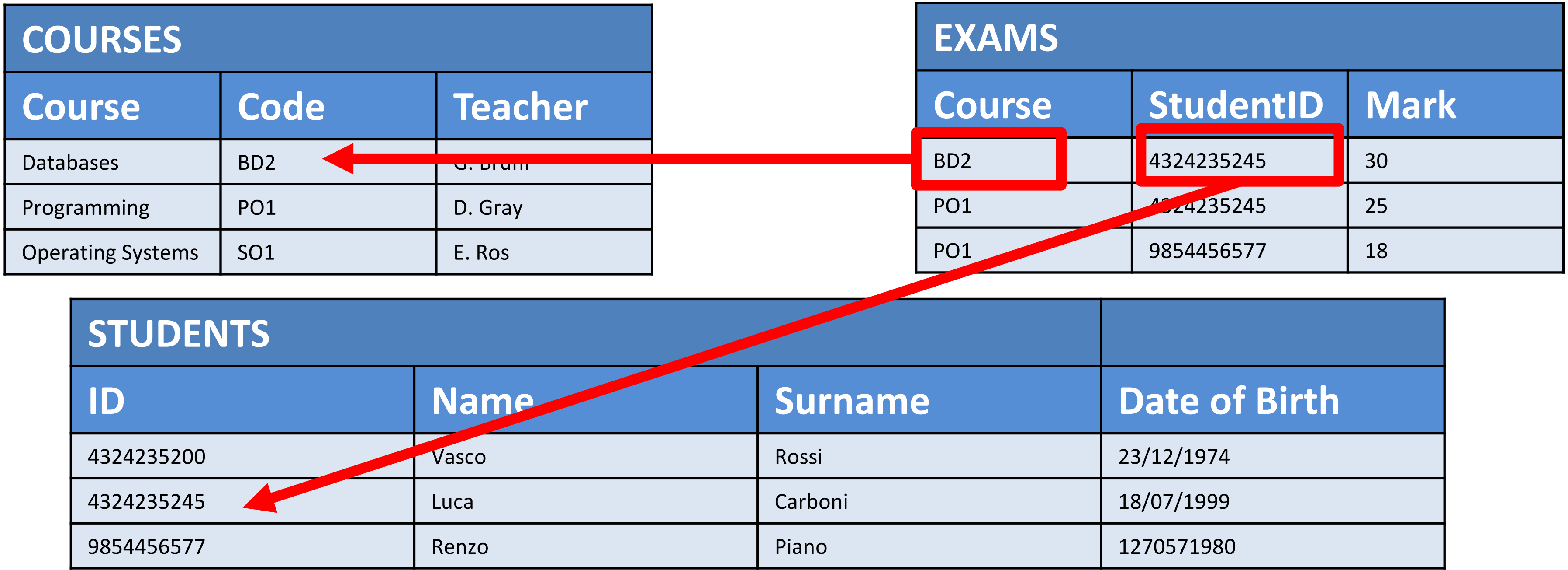
EXAMS		
CourseCode	**StudentID**	**Mark**
BD2	4324235245	30
PO1	4324235245	25
PO1	9854456577	18

STUDENTS			
ID	**Name**	**Surname**	**Date of Birth**
4324235200	Vasco	Rossi	23/12/1974
4324235245	Luca	Carboni	18/07/1999
9854456577	Renzo	Piano	1270571980

RELATIONAL MODEL

Definition: relations

In the relational model, **references** between data in different relationships are **expressed by values**



COURSES		
Course	Code	Teacher
Databases	BD2	G. Bruni
Programming	PO1	D. Gray
Operating Systems	SO1	E. Ros

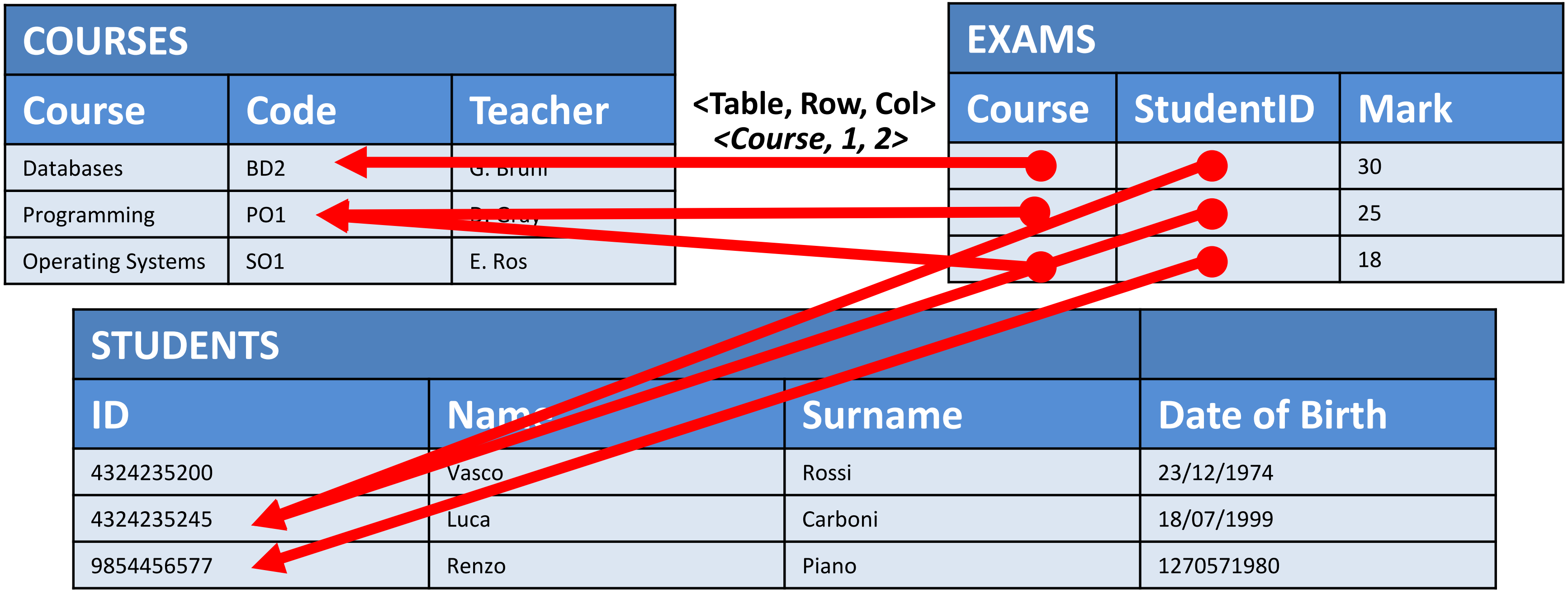
EXAMS		
Course	StudentID	Mark
BD2	4324235245	30
PO1	4324235245	25
PO1	9854456577	18

STUDENTS			
ID	Name	Surname	Date of Birth
4324235200	Vasco	Rossi	23/12/1974
4324235245	Luca	Carboni	18/07/1999
9854456577	Renzo	Piano	1270571980

RELATIONAL MODEL

NB: hierarchical models

The network and hierarchical models, which were defined before the relational model, represent references explicitly by means of pointers and for this reason are called ‘pointer-based’ models



COURSES		
Course	Code	Teacher
Databases	BD2	G. Bruni
Programming	PO1	D. Gray
Operating Systems	SO1	E. Ros

<Table, Row, Col>
<Course, 1, 2>

EXAMS		
Course	StudentID	Mark
●	●	30
●	●	25
●	●	18

STUDENTS			
ID	Name	Surname	Date of Birth
4324235200	Vasco	Rossi	23/12/1974
4324235245	Luca	Carboni	18/07/1999
9854456577	Renzo	Piano	1270571980

RELATIONAL MODEL

Relation Schema

A relation schema consists of a symbol, called **name of the relation R**, and a set of (names of) attributes

A domain is associated with each attribute

COURSES		
Course	**Code**	**Teacher**
Databases	BD2	G. Bruni
Programming	PO1	D. Gray
Operating Systems	SO1	E. Ros

RELATION SCHEMA

COURSES(Course, Code, Teacher)

Name *attributes*

DOMAIN

COURSES ⊆stringxstringxstring

RELATION SCHEMA *(Formal representation)*

COURSES(Course:string, Code:string, Teacher:string)

RELATIONAL MODEL

Database Schema

A **database schema** consists of a set of relation schemas with different names

DATABASE SCHEMA

```
{
        COURSES(Course, Code, Teacher),
        EXAMS(Course, StudentID, Mark),
        STUDENTS(ID,Name,Surname,Date of Birth)
}
```

RELATIONAL MODEL

Relation instance

- An ennuple on a set of attributes X is a function **t** that associates each attribute A in X with a domain value of A
- t[A] denotes the value of the ennuple t on attribute A
- **Relation instance** on an R(X) schema: set r of ennuple on X

COURSES		
Course	**Code**	**Teacher**
Databases	BD2	G. Bruni
Programming	PO1	D. Gray
Operating Systems	SO1	E. Ros

RELATION INSTANCE FUNCTION SAMPLE

t1[Course]= “Databases”

t1[Code]= “BD2”

t3[Teacher]= “E.Ros”

RELATION INSTANCE (OF COURSE)

COURSES(Course, Code, Teacher)= {t1,t2,t3}

RELATIONAL MODEL

Relation instance: generally

Generally, a **database instance** (or simply **database**) on a schema $R = \{R_1(X_1),R_2(X_2),\ldots(R_n(X_n)\}$is a set of relations $r = \{r_1, r_2,\ldots,r_n\}$, where every r_i , for $1 \leq i \leq n$, is a relation on the schema $R_i(X_i)$

```
DATABASE{
            COURSES(Course, Code, Teacher)={c1,c2,c3},
            EXAMS(Course, StudentID, Mark) ={e1,e2,e3},
            STUDENTS(ID,Name,Surname,Date of Birth)={s1,s2,s3,s4}
}
```

RELATIONAL MODEL:
Incomplete information and null values

RELATIONAL MODEL

Incomplete information and null values: problem

In a relationship, the data n-ples must be **homogeneous** (same domain)

But sometimes:

The attribute value does not exist

It exists but is not known at the time of creation of the n-ple

COURSES		
Course	**Code**	**Teacher**
Databases	BD2	G. Bruni
Programming	PO1	D. Gray
Operating Systems	SO1	E. Ros
Database managment and Analisys	DMA2	???????????

RELATIONAL MODEL

Incomplete information and null values: solution 1

A possible solution could be to fill in the missing information using **special values**

COURSES		
Course	**Code**	**Teacher**
Databases	BD2	G. Bruni
Programming	PO1	D. Gray
Operating Systems	SO1	E. Ros
Database managment and Analisys	DMA2	XXXXXXXXXX

RELATIONAL MODEL

Incomplete information and null values: solution 2

Alternatively: the missing information is labeled with the **NULL** value

Through NULL values, it is possible to manage three cases (unknown, non-existent or informationless values), without the need to distinguish between them

COURSES

Course	Code	Teacher
Databases	BD2	G. Bruni
Programming	PO1	D. Gray
Operating Systems	SO1	E. Ros
Digital image	DI1	**NULL**

REGIONALEGOVERNANCE

City	GovernmentAdress
Roma	Viale IV Novembre
Tivoli	**NULL**
Rieti	**NULL**
Ciampino	**NULL**

RELATIONAL MODEL

Incomplete information and null values (remark)

These are distinct rows (these are not duplicates!)

STUDENTS			
ID	Name	Surname	Date of Birth
4324235200	Vasco	NULL	23/12/1974
4324235200	Vasco	NULL	23/12/1974
4324235245	Luca	Carboni	18/07/1999
NULL	Renzo	Piano	12/05/1980
NULL	Renzo	Piano	12/05/1980

RELATIONAL MODEL: Integrity constraints

RELATIONAL MODEL

Integrity constraints

Not all instances of a relationship (or database) can be considered correct

COURSES		
Course	**Code**	**Teacher**
Databases	BD2	G. Bruni
Programming	PO1	D. Gray
Operating Systems	SO1	E. Ros

EXAMS		
Course	**StudentID**	**Mark**
BD2	4324235245	30
PO1	4324235245	56
PO1	9854456577	18

STUDENTS			
ID	**Name**	**Surname**	**Date of Birth**
4324235200	Vasco	Rossi	23/12/1974
4324235245	Luca	Carboni	18/07/1999
9854456577	Renzo	Piano	1270571980

RELATIONAL MODEL

Integrity constraints

It is possible to classify the constraints according to the elements of the database that are involved in it.

There are two categories, the first of which has some particular subcases.

- A **constraint is intra-relational** if its satisfaction is defined with regard to a single relation of the database; in some cases, the definition of the constraint considers the record (or even the values) separately from each other

    A record constraint is a constraint that can be evaluated on each record independently from the others (**record/tuple constraint**)

    As a still more specific case, a constraint defined with reference to single values is called a **value constraint** (or domain constraint), given that it imposes a restriction on the domain of the attribute.

- A **constraint is inter-relational** if it involves more that one relation

RELATIONAL MODEL: Intra-Relation

RELATIONAL MODEL

Integrity constraints: tuple constraint

The **record constraints** express conditions on each n-ple, considered individually

A record constraint can be expressed by algebraic expressions or Boolean expressions

Example: $\mathbf{18} \leq \mathrm{MARK} \leq \mathbf{30}$

EXAMS		
Course	**StudentID**	**Mark**
BD2	4324235245	30
PO1	4324235245	**56**
PO1	9854456577	18

RELATIONAL MODEL

Integrity constraints: value constraint

The **value constraints** is defined with reference to single values

Example: Course: **Character varying (3)**

EXAMS		
Course	**StudentID**	**Mark**
BD2	4324235245	30
PO1	4324235245	**24**
PO1	9854456577	18

RELATIONAL MODEL

Integrity constraints: n-ple constraint

They express conditions on each n-ple, considered individually

They can be expressed as function between attributes:

Example: BALANCE=**REVENUE-COST**

MARKET			
Date	**Revenue**	**Cost**	**Balance**
09/05/2023	500	0	500
10/05/2023	10	180	170
11/05/2023	200	200	0

Key constract

RELATIONAL MODEL

Key: definition

A **key** is a set (one or more) of attributes that allows you to uniquely identify the records of a relation

STUDENTS			
ID	**Name**	**Surname**	**Date of Birth**
4324235200	Vasco	Rossi	23/12/1974
4324235245	Luca	Carboni	18/07/1999
9854456577	Renzo	Piano	1270571980

ID is a key because students have different ID/freshmen
Given the ID of a student, it is possible to trace uniquely all his data (surname / name / date of birth)

RELATIONAL MODEL

SuperKey: definition

A subset K of attributes of a relation is a **superkey** if it does not contain two distinct records t1 and t2 with t1[K]=t2[K].

STUDENTS			
ID	**Name**	**Surname**	**Date of Birth**
4324235200	Vasco	Rossi	23/12/1974
4322113403	Valentino	Rossi	21/02/1999
4324235245	Luca	Carboni	18/07/1999
9854456577	Renzo	Piano	1270571980

With this istances:

{ID} is a SuperKey

{Surname} is not a Superkey

{ID, Surname} is a SuperKey

{Name, Surname} is a SuperKey

RELATIONAL MODEL

minimal SuperKey: definition

A key to a relation r is a **minimal superkey** of r (i.e. there is no other superkey K' that is contained in K)

STUDENTS			
ID	**Name**	**Surname**	**Date of Birth**
4324235200	Vasco	Rossi	23/12/1974
4322113403	Valentino	Rossi	21/02/1999
4324235245	Luca	Carboni	18/07/1999
9854456577	Renzo	Piano	1270571980

{ID} is a Key **and it is a** minimal SuperKey

{Surname} is not a Key

{ID, Surname} is a SuperKey but is not minimal

{ID, Name, Surname} is a SuperKey but is not minimal

RELATIONAL MODEL

SuperKey: frequency in a database

There is always at least one superkey for each relationship

There may be multiple superkeys for the same relationship

STUDENTS				
ID	**Name**	**Surname**	**Date of Birth**	**SSN**
4324235200	Vasco	Rossi	23/12/1974	80001234
4322113403	Valentino	Rossi	21/02/1999	80001235
4324235245	Luca	Carboni	18/07/1999	80001236
9854456577	Renzo	Piano	1270571980	80001237

RELATIONAL MODEL

Keys and null values

When a **Key** has a NULL values, the unique addressing of the records and the correlations between different tables may not be guaranteed

STUDENTS			
ID	**Name**	**Surname**	**Date of Birth**
4324235200	Vasco	Rossi	23/12/1974
NULL	Valentino	Rossi	21/02/1999
4324235245	Valentino	Rossi	NULL
9854456577	Renzo	Piano	1270571980

RELATIONAL MODEL

Key: time definition

At design time, keys should be defined at the schema level; not at the instance level

{Name} it is a key in this instance, but it is only a coincidence

HOSPITALIZATION				
Name	**Surname**	**Sector**	**Date of hospitalization**	**Date of discharge**
Vasco	Rossi	cardiology	23/12/2022	23/01/2023
Luca	Carboni	orthopedy	18/07/2022	28/07/2022
Marco	Carboni	orthopedy	12/04/2023	NULL

RELATIONAL MODEL

Keys and null values

Primary key: Key of a relationship on which NULL values are not allowed

STUDENTS			
ID	**Name**	**Surname**	**Date of Birth**
4324235200	Vasco	Rossi	23/12/1974
4324235245	Luca	Carboni	18/07/1999
9854456577	Renzo	Piano	1270571980

The attributes of a primary key are usually indicated with an underline

STUDENTS			
ID	**Name**	**Surname**	**Date of Birth**

STUDENTS(**ID**, Surname, Name, DateOfBirth)

RELATIONAL MODEL

Primary Key

Each relationship must have a primary key

What should I do if the contidion is not verified or all keys have NULL values?

Add a new identifiers or a progressive codes

	HOSPITALIZATION				
ID	**Name**	**Surname**	**Sector**	**Date of hospitalization**	**Date of discharge**
1	Vasco	NULL	cardiology	23/12/2022	23/01/2023
2	Luca	Carboni	orthopedy	18/07/2022	28/07/2022
3	Luca	Piano	orthopedy	12/04/2023	NULL
4					

RELATIONAL MODEL

Primary Key (remark)

A key can consist of multiple attributes

MATCH

Team1	Team2	Date	Point1	Point2
Virtus	Ciampino FC	23/04/2023	35	60
Romulea	Lupa Frascati	18/03/2023	12	70
Polisportiva Rieti	MarinoLand	09/05/2023	23	22

But you MUST pay attention

MATCH

Team1	Team2	Date	Point1	Point2
Virtus	Ciampino FC	NULL	35	35
Virtus	Ciampino FC	18/03/2023	12	70

MATCH

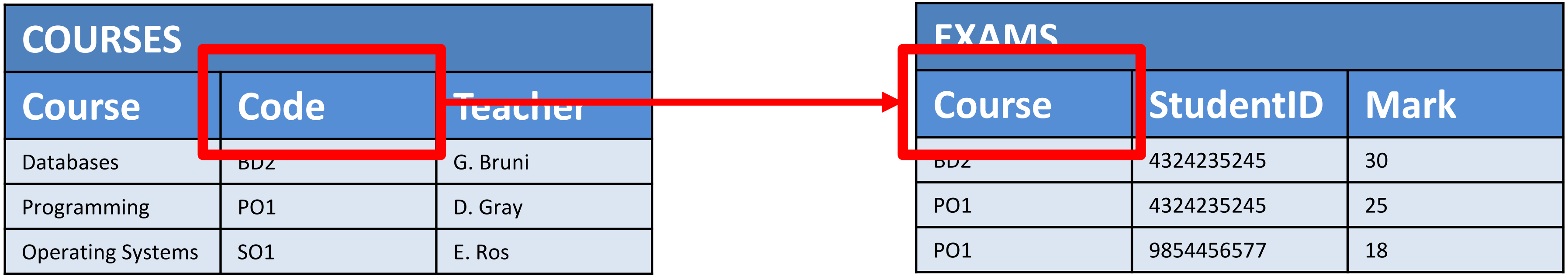
Team1	Team2	Date	Point1	Point2
Virtus	Romulea	23/04/23	35	35
Virtus	Romulea	18/03/23	12	70

RELATIONAL MODEL

Key: purpose

Access each database ennuple, uniquely

Correlating data between different relationships

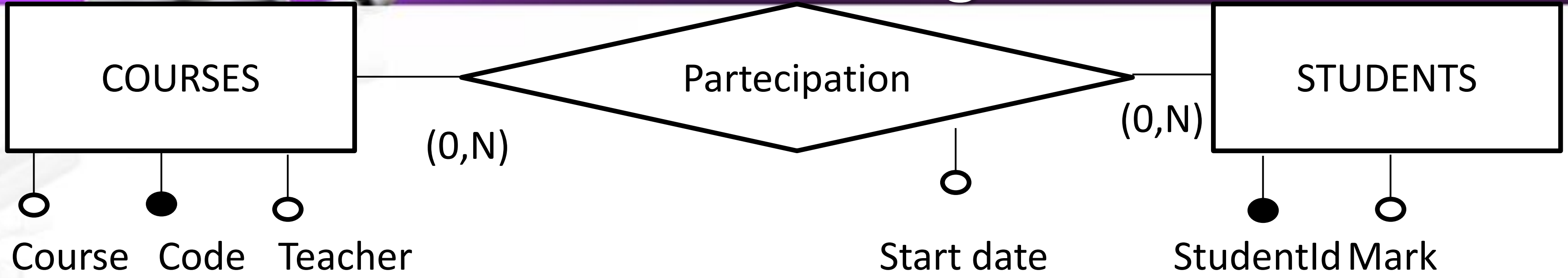


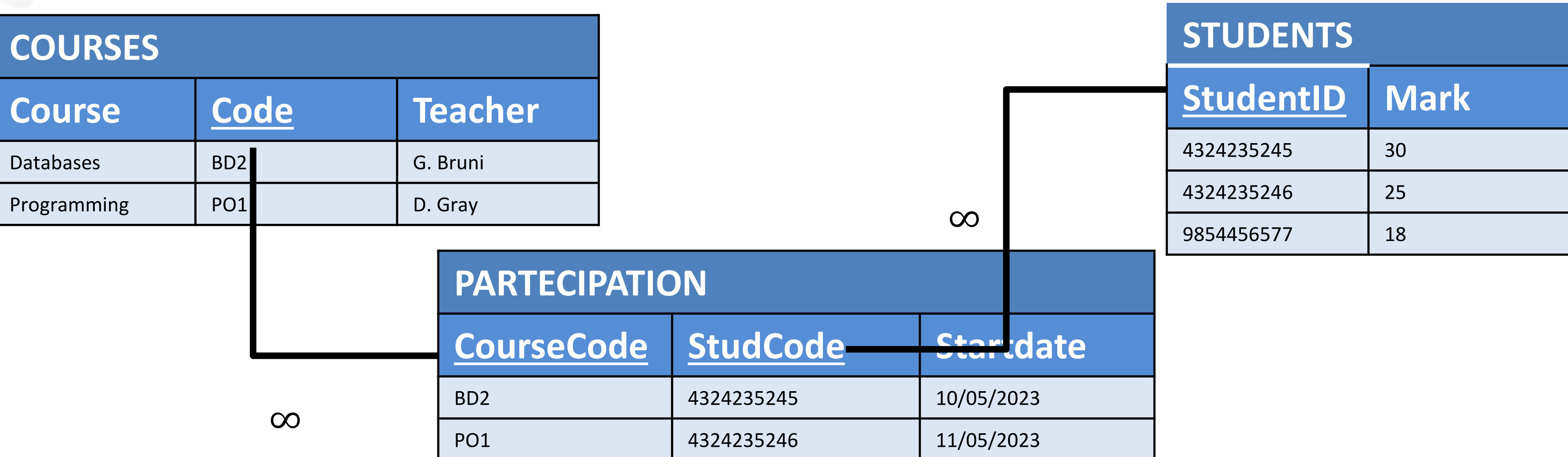
COURSES		
Course	**Code**	**Teacher**
Databases	BD2	G. Bruni
Programming	PO1	D. Gray
Operating Systems	SO1	E. Ros

EXAMS		
Course	**StudentID**	**Mark**
BD2	4324235245	30
PO1	4324235245	25
PO1	9854456577	18

INTER NOS

A little magic





COURSES		
Course	Code	Teacher
Databases	BD2	G. Bruni
Programming	PO1	D. Gray

STUDENTS	
StudentID	Mark
4324235245	30
4324235246	25
9854456577	18

PARTECIPATION		
CourseCode	StudCode	Startdate
BD2	4324235245	10/05/2023
PO1	4324235246	11/05/2023

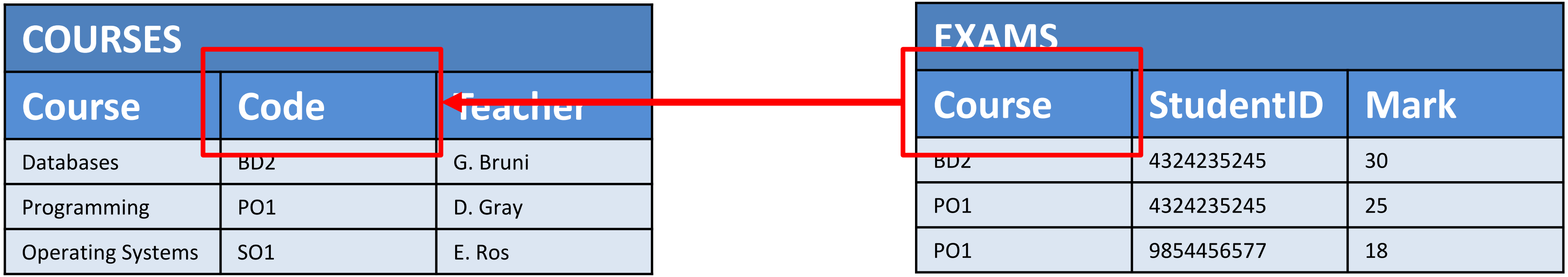
RELATIONAL MODEL: Inter-Relation

RELATIONAL MODEL

Internal relation

In the relational model, a database can be composed of many related relations

Links between different relations are expressed by common values in replicated attributes



COURSES

Course	Code	Teacher
Databases	BD2	G. Bruni
Programming	PO1	D. Gray
Operating Systems	SO1	E. Ros

EXAMS

Course	StudentID	Mark
BD2	4324235245	30
PO1	4324235245	25
PO1	9854456577	18

RELATIONAL MODEL

Foreign key: definition

A **foreign key** (or foreign key constraint), between a set of attributes X of a relation R1 e another relation R2, forces the values on X of each tuple of the instance of R1 to appear as primary key values of the instance of R2

In other words: **a foreign key(s) links information between different tables through common values**

SURGERIES

hospital ward	IdUser	Date
A	M000065	10/05/2022
B	M000065	08/04/2023
B	M000067	09/03/2023

referencing relationship

USERS

Code	Name	Surname	Birth
M000065	Maria	Toscano	13/05/1947
M000067	Beatrice	Mesti	16/06/1945
E022067	Enza	Brambilla	09/07/1956
F055067	Silvio	Bernardi	22/08/1955

referenced relationship

Foreign key: SURGERIES(IdUser) $\subseteq$USERS(Code)

RELATIONAL MODEL

Foreign key: use

VIGILANTS

ID	Name	Surname
123	John	Rambo
345	Salvo	Montalbano
678	Vish	Puri

CARS

State	Plate	Owner
NY	3DIE87H	Bruce Willis
IND	3789807	Priyanka Chopra
ITA	RS768DR	Cristiano Malgioglio
IND	3789811	Salman Khan

Foreign keys of referencing relationship must have same attributes/columns and data types of primary key of referenced relationship

INFRINGMENTS

VerbalNum	Date	Agent	State	PlateNum
C100	05/02/2023	123	NY	3DIE87H
C202	05/02/2023	678	IND	3789807
C503	12/06/2023	678	IND	3789811

Foreign key: INFRINGMENTS(Agent) $\subseteq$VIGILANTS(ID)

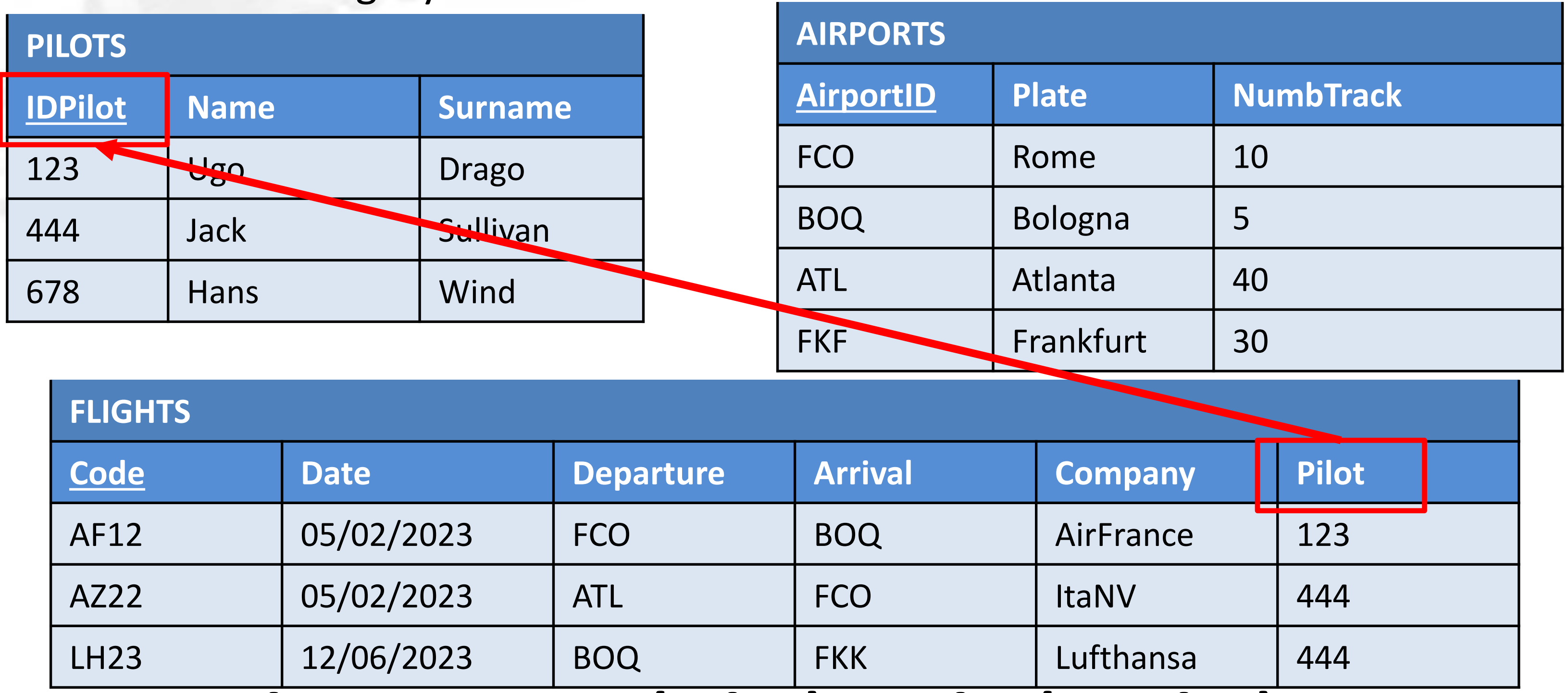
referencing relationship

referenced relationship

RELATIONAL MODEL

Foreign key: sample I

Referential integrity constraint between Pilot attribute of the FLIGHTS relation and the



PILOTS		
IDPilot	Name	Surname
123	Ugo	Drago
444	Jack	Sullivan
678	Hans	Wind

AIRPORTS		
AirportID	Plate	NumbTrack
FCO	Rome	10
BOQ	Bologna	5
ATL	Atlanta	40
FKF	Frankfurt	30

FLIGHTS					
Code	Date	Departure	Arrival	Company	Pilot
AF12	05/02/2023	FCO	BOQ	AirFrance	123
AZ22	05/02/2023	ATL	FCO	ItaNV	444
LH23	12/06/2023	BOQ	FKK	Lufthansa	444

Foreign key: FLIGHTS(Pilot) ⊆Pilot(IDPilot)

RELATIONAL MODEL

Foreign key: sample II

Referential integrity constraint between Departure attribute of the FLIGHTS relation and

PILOTS

IDPilot	Name	Surname
123	Ugo	Drago
444	Jack	Sullivan
678	Hans	Wind

AIRPORTS

AirportID	Plate	NumbTrack
FCO	Rome	10
BOQ	Bologna	5
ATL	Atlanta	40
FKF	Frankfurt	30

FLIGHTS

Code	Date	Departure	Arrival	Company	Pilot
AF12	05/02/2023	FCO	BOQ	AirFrance	123
AZ22	05/02/2023	ATL	FCO	ItaNV	444
LH23	12/06/2023	BOQ	FKK	Lufthansa	444

Foreign key: FLIGHTS(Departure) ⊆AIRPORTS(AirportID)

RELATIONAL MODEL

Foreign key: sample III

Referential integrity constraint between Arrival attribute of the FLIGHTS relation and the AIRPORTS relation

PILOTS

IDPilot	Name	Surname
123	Ugo	Drago
444	Jack	Sullivan
678	Hans	Wind

AIRPORTS

AirportID	Plate	NumbTrack
FCO	Rome	10
BOQ	Bologna	5
ATL	Atlanta	40
FKF	Frankfurt	30

FLIGHTS

Code	Date	Departure	Arrival	Company	Pilot
AF12	05/02/2023	FCO	BOQ	AirFrance	123
AZ22	05/02/2023	ATL	FCO	ItaNV	444
LH23	12/06/2023	BOQ	FKK	Lufthansa	444

Foreign key: FLIGHTS(Arrival) $\subseteq$AIRPORTS(AirportID)

RELATIONAL MODEL

Foreign key: VIOLATION

PILOTS

IDPilot	Name	Surname
123	Ugo	Drago
444	Jack	Sullivan
678	Hans	Wind

AIRPORTS

AirportID	Plate	NumbTrack
FCO	Rome	10
BOQ	Bologna	5
ATL	Atlanta	40
FKF	Frankfurt	30

FLIGHTS

Code	Date	Departure	Arrival	Company	Pilot
AF12	05/02/2023	FCO	BOQ	AirFrance	123
AZ22	05/02/2023	ATL	CMP	ItaNV	444
LH23	12/06/2023	BOQ	FKK	Lufthansa	234

Is it the only violation ?

Foreign key: FLIGHTS(Arrival) ⊄AIRPORTS(IDAirport)

RELATIONAL MODEL

Foreign key: MORE FK

Integrity constraint between RESERVATION (FlightCode, Flight Date) and FLIGHTS (constraint on multiple attributes).

In these case there are two Foreign Keys

FLIGHTS

Code	Date	Departure	Arrival	Company	Pilot
AF12	05/02/2023	FCO	BOQ	AirFrance	123
AZ22	05/02/2023	ATL	FCO	ItaNV	444
LH23	12/06/2023	BOQ	FKK	Lufthansa	444

FLIGHTS

NrTick	CodeFlight	DateFlight	Name	Surname
123	AZ22	05/02/2023	Vasco	Rossi
124	LH23	12/06/2023	Achille	Lauro

Why is NOT possibile have only one FK?

Foreign keys: RESERVATION(CodeFlight, DateFlight) $\subseteq$ FLIGHTS(Code,Date)

RELATIONAL MODEL

Referential integrity constraints

An update or delete operation on a relation can cause inconsistencies

How to react?

Do not allow the operation
Cascade elimination
Inserting NULL Values

RELATIONAL MODEL

Referential integrity constraints

PILOTS

IDPilot	Name	Surname
123	Ugo	Drago
444	Jack	Sullivan
678	Hans	Wind

AIRPORTS

AirportID	Plate	NumbTrack
FCO	Rome	10
BOQ	Bologna	5
ATL	Atlanta	40
FKF	Frankfurt	30

FLIGHTS

Code	Date	Departure	Arrival	Company	Pilot	
AF12	05/02/2023	FCO	BOQ	AirFrance	123	?
AZ22	05/02/2023	ATL	FCO	ItaNV	444	?
LH23	12/06/2023	BOQ	FKK	Lufthansa	444	

RELATIONAL MODEL

Referential integrity constraints

CASCADE DELETION

PILOTS

IDPilot	Name	Surname
123	Ugo	Drago
444	Jack	Sullivan
678	Hans	Wind

AIRPORTS

AirportID	Plate	NumbTrack
FCO	Rome	10
BOQ	Bologna	5
ATL	Atlanta	40
FKF	Frankfurt	30

FLIGHTS

Code	Date	Departure	Arrival	Company	Pilot
AF12	05/02/2023	FCO	BOQ	AirFrance	123
AZ22	05/02/2023	ATL	FCO	ItaNV	444
LH23	12/06/2023	BOQ	FKK	Lufthansa	444

RELATIONAL MODEL

Referential integrity constraints

INSERT NULL VALUES

PILOTS		
IDPilot	Name	Surname
123	Ugo	Drago
444	Jack	Sullivan
678	Hans	Wind

AIRPORTS		
AirportID	Plate	NumbTrack
FCO	Rome	10
BOQ	Bologna	5
ATL	Atlanta	40
FKF	Frankfurt	30

FLIGHTS					
Code	Date	Departure	Arrival	Company	Pilot
AF12	05/02/2023	FCO NULL	BOQ	AirFrance	123
AZ22	05/02/2023	ATL	FCO NULL	ItaNV	444
LH23	12/06/2023	BOQ	FKK	Lufthansa	444

End