ICT & Business Intelligence & CRM
Databases - Relational Model

Andrea Vandin
Scuola Superiore Sant'Anna
Paolo Ferragina
Scuola Superiore Sant'Anna
Anna Monreale
Università di Pisa

Relational Model

First presented in seminal work from the 70s:

> ***«A model based on n-ary relations, a normal form for data base relations, and the concept of a universal data sublanguage are introduced.»***
>
> **Edgar F. Codd, 1970**

Nice overview:

https://users.dimi.uniud.it/~massimo.franceschet/ds/syllabus/learn/database/RM.html

Relational Model

It is based on the concept of *relation*

- Whose representation is a table
  - relation-table can be used interchangeably
- comes from set theory
- Not to be confused with the conceptual relationship of the ER model

The relational model consists of two main components:

- the structures that allow to organize the data
  - The tables
- the integrity constraints that allow data to be kept consistent
  - The constraints on the values in specific columns of tables

Elements of the model

- **Database**
  - Set of **tables**
- **Table** (or relation)
  - Set of **records**
  - It has
    - **Columns**, aka attributes,
      - have a **name** and a **domain** (type of data)
      - One special column (or a subset of columns) contains the **primary key**
        - » a value that distinguishes every row (e.g. tax code)
    - **Rows**, aka records,
      - containing a value per attribute
- **Record**
  - Set of pairs **(attribute, value)**

Example: Students, Courses, Exams

- **Database: University**

Tables:

- **Students**
  - name, surname, **studentID**, date of birth
- **Courses**
  - **code**, title of course, teacher
- **Exams**
  - **course**, **student**, mark, laud

Example: Students, Courses, Exams

- **Students**
  - name: string
  - surname: string
  - **studentID**: integer
  - date of birth: date
- **Courses**
  - **code**: string
  - title: string
  - teacher: string
- **Exams**
  - **course**: “reference” to a course (its code)
  - **student:** “reference” to a student (its studentID)
  - mark: integer
  - laud: yes/no

Students

Table: relation **(instance)**

record

Attribute: property

value

Students

StudentID	Surname	Name	Date of Birth
276545	Rossi	Maria	25/11/1991
485745	Neri	Anna	23/04/1992
200768	Verdi	Fabio	12/02/1992
587614	Rossi	Luca	10/10/1991
937653	Bruni	Mario	01/12/1991

Relational Schema

```
TABLE Students(
    studentID integer,
    surname char(20),
    name char(20),
    DateofBirth date);
```

Domain
Type

Attributes

Students

StudentID	Surname	Name	Date of Birth
276545	Rossi	Maria	25/11/1991
485745	Neri	Anna	23/04/1992
200768	Verdi	Fabio	12/02/1992
587614	Rossi	Luca	10/10/1991
937653	Bruni	Mario	01/12/1991

```
TABLE Students(
    studentID integer,
    surname  char(20),
    name  char(20),
    DateofBirth date);
```

- Each attribute has a **domain** defining the **set of valid values** for the attribute Ex. dom(studentID) = *integer*

- We cannot have a repetition of attributes (column names are univoque)
- We can have a repetition of domains in the same relation (name and surname have same domain)!

Types

- **numbers**
  - N := c | V | N1 + N2 | N1 – N2 | CASE WHEN B THEN N1 ELSE N2 END
    - X + 2*Y – Z
    - CASE WHEN x < y THEN 1 WHEN x > y THEN -1 ELSE 0 END
- **String**
  - S := c | CONCAT(S, S) | CASE WHEN B THEN S1 ELSE S2 END
    - ‘buon’ + ‘giorno’ |
    - ‘buon’ || ‘giorno’
    - CONCAT(‘buon’, ‘giorno’)
    - CASE WHEN x < y THEN ‘minore’ ELSE ‘maggiore’ END
- **boolean**
  - B := c | N1 < N2 | N1 <= N2 | N1 <> N2 | N1 = N2 | N1 > N2 | N1 >= N2 | N1 BETWEEN c1 AND c2 | S1 < S2 | S1 <= S2 | S1 <> S2 | S1 =S2 | S1 > S2 | S1 >= S2 | S LIKE c | S NOT LIKE c | B1 AND B2 | B1 OR B2 | NOT B
    - 3 BETWEEN 2 AND 7 → 2 <= 3 <= 7
    - surname NOT LIKE ‘rug%’

Constraints on the order of the data

Students

StudentID	Surname	Name	Date of Birth
276545	Rossi	Maria	25/11/1991
485745	Neri	Anna	23/04/1992
200768	Verdi	Fabio	12/02/1992
587614	Rossi	Luca	10/10/1991
937653	Bruni	Mario	01/12/1991

- Order of records is not important
- Order of columns is important

Constraints on data

Students

StudentID	Surname	Name	Date of Birth
20/11/1991	Rossi	Maria	25/11/1991
485745	Neri	Anna	23/04/1992
200768	Verdi	Fabio	12/02/1992
587614	Rossi	Luca	10/10/1991
937653	Bruni	Mario	01/12/1991

- No equal **attributes** (**1**) (no equal column names)
- No **equal rows** (**2**)
- Data of **column** must be **homogeneous** (**3**)

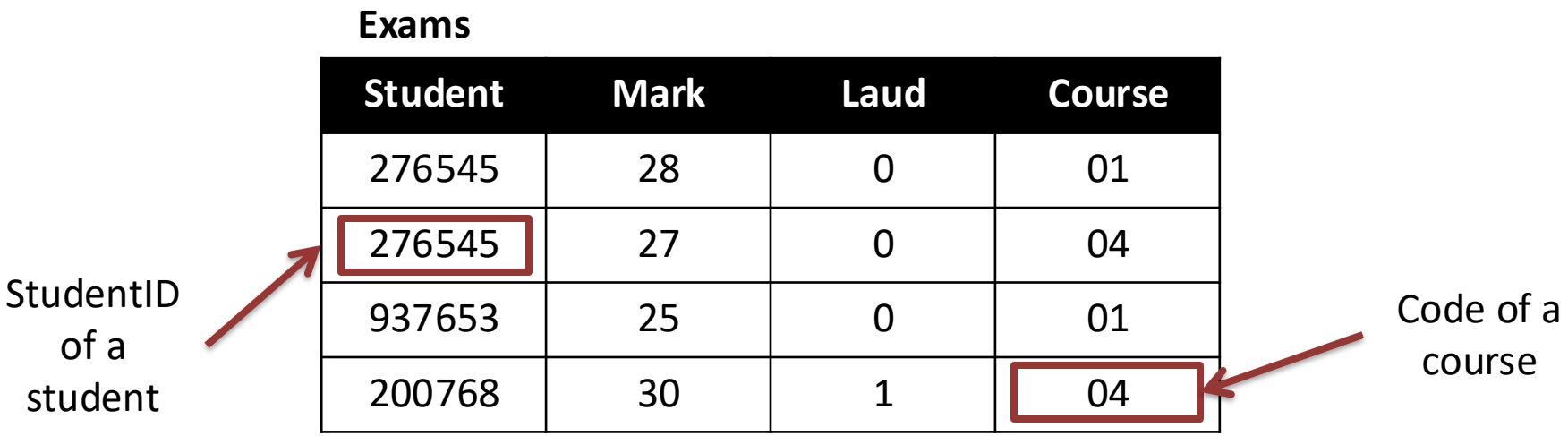
Courses

COURSES

Code	Title	Teacher
01	Analisi	Giani
03	Chimica	Melli
04	Chimica	Belli

```
TABLE Courses(
   code char(2),
   title char(50),
   teacher char(20));
```

Exams




```
TABLE Exams(
   student integer,
   mark integer,
   laud bool,
   course char(3));
```

Tables

Students

StudentID	Surname	Name	Date of Birth
276545	Rossi	Maria	25/11/1991
485745	Neri	Anna	23/04/1992
200768	Verdi	Fabio	12/02/1992
587614	Rossi	Luca	10/10/1991
937653	Bruni	Mario	01/12/1991

Exams

Student	Mark	Laud	Course
276545	28	0	01
276545	27	0	04
937653	25	0	01
200768	30	1	04

Courses

Code	Title	Teacher
01	Analisi	Giani
03	Chimica	Melli
04	Chimica	Belli

DB Schema

```
TABLE Students(
   studentID integer,
   surname char(20),
   name  char(20),
   DateofBirth date);
```

```
TABLE Exams(
   student integer,
   mark integer,
   laud bool,
   course char(3));
```

```
TABLE Courses(
   code char(3),
   title char(50),
   teacher char(20));
```

NULL Values

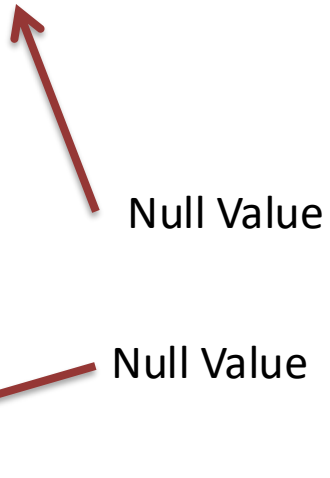
Are supported as a systematic representation of missing information

Students

StudentID	Surname	Name	Date of Birth
276545	Rossi	Maria	25/11/1991
485745	Neri	Anna	23/04/1992
200768	Verdi	Fabio	12/02/1992
587614	Rossi	Luca	10/10/1991
937653	Bruni	Mario	01/12/1991
993354	Gialli	Lucia	null

Courses

Code	Title	Teacher
01	Analisi	Giani
03	Chimica	Melli
04	Chimica	Belli
05	Basi Dati	null



Constraints on Data

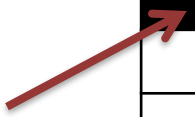
- **Rules of the scenario**
- **Unicity of identifiers (keys)**
  - Code of courses and studentID
- **Conditions on values of the records**
  - Students mark
    - Range: 18 - 30
    - Laud only if mark is 30
- **Correctness of references**

Primary Key

- **Key:** a minimal set of attributes **uniquely identifying**
  - tuples in a relation
  - I.E., rows in a table

Students

Key



StudentID	Surname	Name	Date of Birth
276545	Rossi	Maria	25/11/1991
485745	Neri	Anna	23/04/1992
200768	Verdi	Fabio	12/02/1992
587614	Rossi	Luca	10/10/1991
937653	Bruni	Mario	01/12/1991

- {*Surname,Name*}: is not a real Key!
- A (Primary) Key cannot have *null value*

Database with errors

Students

StudentID	Surname	Name	Date of Birth
276545	Rossi	Maria	25/11/1991
485745	Neri	Anna	23/04/1992
200768	Verdi	Fabio	12/02/1992
937653	Rossi	Luca	10/10/1991
937653	Bruni	Mario	01/12/1991

Unicity

Exams

Student	Mark	Laud	Course
276545	32	0	01
276545	27	1	04
937653	25	0	01
300300	30	1	04

incorrect mark

Reference

Integrity Constraints

- **Rules on data**
- Constraints on single tables
  - **Key Constraints**
  - **Constraints on tuples**
- Constraints between tables
  - **Constraints of referential integrity**

Integrity Constraints

- **Key Constraints**
- Key: identifier for tuples
  - ex: “studentID” is a Key for “Students”
- **Constraints on tuple**
  - Predicates on the values of tuples
  - ex: (mark>=18 and mark<=30)
- **Referential Constraints**
  - Absence of references that do not exist

Key Constraints

Students

StudentID	Surname	Name	Date of Birth
276545	Rossi	Maria	25/11/1991
485745	Neri	Anna	23/04/1992
200768	Verdi	Fabio	12/02/1992
587614	Rossi	Luca	10/10/1991
937653	Bruni	Mario	01/12/1991

```
TABLE Students(
    studentID integer PRIMARY KEY,
    TaxCode char(16),
    surname char(20),
    name  char(20),
    DateofBirth date,
    UNIQUE(TaxCode) );
```

Constraints on tuples

Exams

Student	Mark	Laud	Course
276545	28	0	01
276545	27	0	04
937653	25	0	01
200768	30	1	04

```
TABLE Exams (
   student integer,
   mark integer,
   course char(3),
   laud bool,
   CHECK (mark>=18 and mark<=30),
   CHECK (not laud or mark=30));
```

Referential Constraints

Exams

Student	Mark	Laud	Course
276545	28	0	01
276545	27	0	04
937653	25	0	01
200768	30	1	04

```
TABLE Exams (
    student integer,
    mark integer,
    course char(3),
    laud bool,
    CHECK (mark>=18 and mark<=30),
    CHECK (not laud or mark=30),
    FOREIGN KEY(student) REFERENCES Students(studentID),
    FOREIGN KEY(course)  REFERENCES Courses(code));
```

Schema with Integrity Constraints

```
TABLE Students(
    studentID integer PRIMARY KEY,
    surname char(20),
    name  char(20),
    DateofBirth date,
    UNIQUE(studentID));
```

```
TABLE Exams(
    student integer,
    mark integer,
    laud bool,
    course char(3),
    CHECK (mark>=18 and mark<=30),
    CHECK (not laud or mark=30),
    FOREIGN KEY(course)  REFERENCES Courses(code),
    FOREIGN KEY(student) REFERENCES Students(studentID),
    UNIQUE(studentID, course));
```

```
TABLE Courses(
    code char(3) PRIMARY KEY,
    title char(50),
    teacher char(20),
    UNIQUE(code));
```

Characteristics of the Model

- **Links based on values**
  - **Absence of pointers**
  - The DB does not have a real notion of pointer
    - The linking (the arrow from previous slides) is by **value**
- **Value must be simple**
  - **Atomic** values : number, chars, string, boolean, date ecc.
  - no ‘nesting’

Pointers

The red arrows are in reality just values:

- Exams:
  - Third row contain 937653, fourth row contain 04

Students

StudentID	Surname	Name	Date of Birth
276545	Rossi	Maria	25/11/1991
485745	Neri	Anna	23/04/1992
200768	Verdi	Fabio	12/02/1992
587614	Rossi	Luca	10/10/1991
937653	Bruni	Mario	01/12/1991

Exams

Student	Mark	Laud	Course
276545	28	0	01
276545	27	0	04
937653	25	0	01
200768	30	1	04

Courses

Code	Title	Teacher
01	Analisi	Giani
03	Chimica	Melli
04	Chimica	Belli

Nested Structure

not supported!

Dal Sudicio *Via Buia, Pisa*		
Ricevuta Fiscale 1235 *del* 12/10/2001		
3	Coperti	3,00
2	Antipasti	6,20
3	Primi	12,00
2	Bistecche	18,00
	Totale	39,20

Dal Sudicio *Via Buia, Pisa*		
Ricevuta Fiscale 1240 *del* 13/10/2001		
2	Coperti	2,00
2	Antipasti	7,00
2	Primi	8,00
2	Orate	20,00
2	Caffè	2,00
	Totale	39,00

Possible Representation

Ricevute

numero	data	totale
1235	12/10/2000	39,20
1240	13/10/2000	39,00

Dettaglio

numero	qta	portata	prezzo
1235	3	Coperti	3,00
1235	2	Antipasti	6,20
1235	3	Primi	12,00
1235	2	Bistecche	18,00
1240	2	Coperti	2,00
...	...	...	...