

Databases

Relational Model

Anna Monreale
Università di Pisa

Elements of the model

- **Database**
 - Set of **tables**
- **Table**
 - Set of **records**
- **Record**
 - Set of pairs (**attribute, value**)

Example: Students, Courses, Exams

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

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
- **student** “reference” to a student
- **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

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

Relational Schema

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. $\text{dom}(\text{studentID}) = \textit{integer}$
- We can have a repetition of domains in the same relation!

Types

- **numbers**

- $N := c \mid V \mid N1 + N2 \mid N1 - N2 \mid \text{CASE WHEN } B \text{ THEN } N1 \text{ ELSE } N2 \text{ END}$
 - $X + 2 * Y - Z$
 - $\text{CASE WHEN } x < y \text{ THEN } 1 \text{ WHEN } x > y \text{ THEN } -1 \text{ ELSE } 0 \text{ END}$

- **String**

- $S := c \mid \text{CONCAT}(S, S) \mid \text{CASE WHEN } B \text{ THEN } S1 \text{ ELSE } S2 \text{ END}$
 - 'buon' + 'giorno' |
 - 'buon' || 'giorno'
 - $\text{CONCAT}(\text{'buon'}, \text{'giorno'})$
 - $\text{CASE WHEN } x < y \text{ THEN 'minore' ELSE 'maggiore' END}$

- **boolean**

- $B := c \mid N1 < N2 \mid N1 \leq N2 \mid N1 \neq N2 \mid N1 = N2 \mid N1 > N2 \mid N1 \geq N2 \mid N1 \text{ BETWEEN } c1 \text{ AND } c2 \mid S1 < S2 \mid S1 \leq S2 \mid S1 \neq S2 \mid S1 = S2 \mid S1 > S2 \mid S1 \geq S2 \mid S \text{ LIKE } c \mid S \text{ NOT LIKE } c \mid B1 \text{ AND } B2 \mid B1 \text{ OR } B2 \mid \text{NOT } B$
 - $3 \text{ BETWEEN } 2 \text{ AND } 7 \rightarrow 2 \leq 3 \leq 7$
 - $\text{surname NOT LIKE 'rug%'}$

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 not 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 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(3),  
  title char(50),  
  teacher char(20));
```

Exams


Exams

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

StudentID
of a
student



Code of a
course



```
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	Corso
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

DM 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

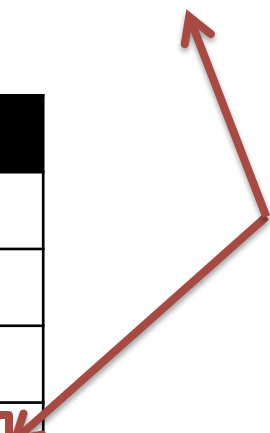
Students

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

Null Value



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

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

Key →

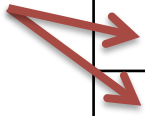
- $\{Surname, Name\}$: is not a real Key!
- A (Primary) Key cannot have *null value*

Database with errors

Students

StudentID	Surname	Name	Data di nascita
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



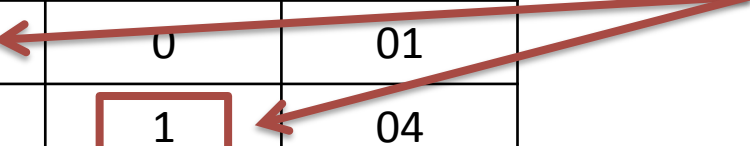
Exams

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

Reference



uncorrect
mark



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 \geq 18 and mark \leq 30)
- **Referential Constraints**
 - Absence of references that do not exist
 - ex: exist Exams only for Students of BD

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,  
  surname char(20),  
  name char(20),  
  DateofBirth date,  
  UNIQUE(studentID) );
```

Constraints on tuples

Exams

Student	Mark	Laud	Corso
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	Corso
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,  
    surname char(20),  
    name char(20),  
    DateofBirth date,  
    UNIQUE(studentID));
```

```
TABLE Exams(  
    student integer,  
    mark integer,  
    laud bool,  
    course char(3),  
    FOREIGN KEY(course) REFERENCES Courses(code),  
    FOREIGN KEY(student) REFERENCES Students(studentID),  
    UNIQUE(studentID, course));
```

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

Characteristics of the Model

- **Links based on values**
 - **Absence of pointers**
- **Value must be simple**
 - **Atomic** values : number, chars, string, boolean, date ecc.
 - no “nesting

Pointers

Students

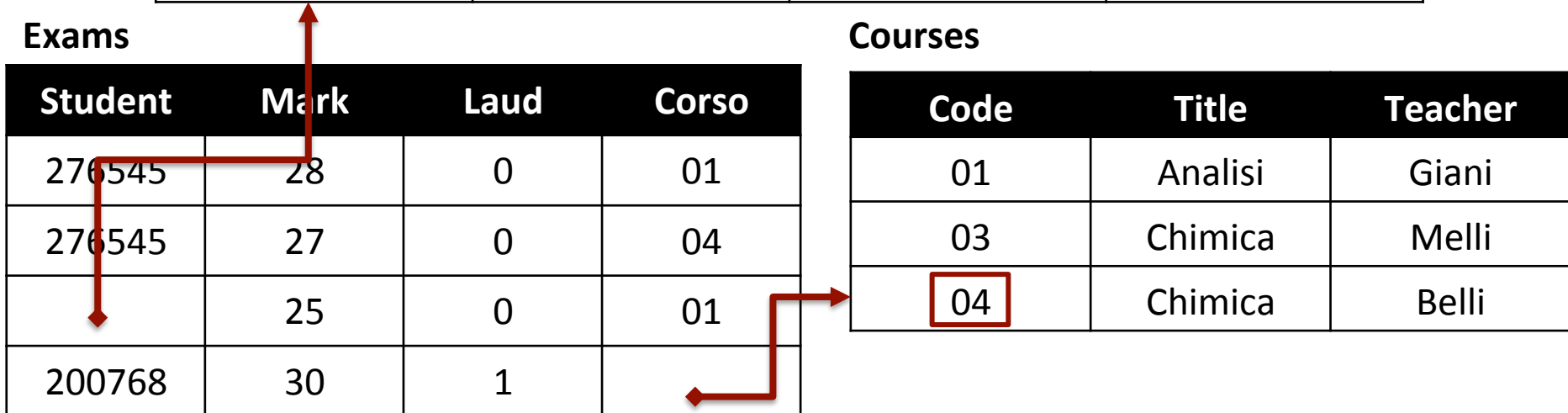
StudentID	Surname	Name	Date of Birth
276545	Rossi	Maria	25/11/1991
485745	Neri	Anna	23/04/1992
200768	Verdi	Fabio	12/02/1992
587614	Rossi	Luca	10/10/1991
937653	Bruni	Mario	01/12/1991

Exams

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

Courses

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



Nested Structure

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

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