Data Management and Analysis

Giuseppe Perelli
perelli@di.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

Slide credits: Prof. Maria de Marsico, Prof. Marina Moscarini, and Prof. Maurizio Mancini, Sapienza University of Rome

Relational model

- proposed by E. F. Codd in 1970 to promote data independence
- available in real DBMS in 1981 (<u>not easy to implement independence with efficiency and reliability</u>!).

Relational model

- based on the mathematical notion of <u>relation</u>
- relations translate naturally into <u>tables</u>
- data and <u>relationships/associations</u> between data of different sets (relations/tables) are represented as <u>values</u>
- careful with the different meanings of relation and relationship...

Definitions 1

- **domain:** a (possibly infinite) set of values
- examples:
  - the set of integers is a domain
  - the set of decimal numbers is a domain
  - the set of character strings of length = 20 is a domain
  - {0,1} is a domain

Definitions 2

let $D_1, D_2, \ldots\ldots D_k$ be domains, not necessarily distinct;
the Cartesian product of these domains, denoted by:

$$D_1 \times D_2 \times \ldots\ldots \times D_k$$

is the set

$$\{(v_1, v_2, \ldots\ldots, v_k) \mid v_1 \in D_1, v_2 \in D_2, \ldots\ldots V_k \in D_k\}$$

ordered list of values such that belongs to

Definitions 3

- mathematical relation is any subset of the Cartesian product of one or more domains
- a relation that is a subset of the Cartesian product of k domains is said to be of degree k
- the elements of a relation are called tuples (or n-uples or ennuples): the number of tuples of a relation is its cardinality
- each tuple of a relation of degree k has k ordered components (the i-th value comes from the i-th domain) but there is no ordering among tuples
- the tuples of a relation are all distinct (at least for a value)… as a relation is a set!

Definitions - example

- suppose k = 2
- D_1 = {white, black}, D_2 = {0, 1, 2}

  D1 X D2 = {(white, 0), (white, 1), (white, 2), (black, 0), (black, 1), (black, 2)}

  so, {(white, 0), (black, 0), (black, 2)} is a relation of degree 2, cardinality 3 and with tuples {(white, 0), (black, 0), (black, 2)}

  {(black, 0), (black, 2)} is a relation of degree 2, cardinality 2 and with tuples {(black, 0), (black, 2)}

Report - example

- $D_1=\{a,b\}$
- $D_2=\{x,y,z\}$

- Cartesian product $D_1 \times D_2$ →

a	x
a	y
a	z
a	x
b	y
b	z

- relation $r \subseteq D_1 \times D_2$ →

a	x
a	z
a	y

Report - example

- almost always we use well-known domains also used in programming languages

- the following is a relation with two 4-ple on String, String, Integer, Real

Paolo	Rossi	2	26.5
Mario	Bianchi	10	28.7

Notation

- given *r*, a relation of degree *k*
- given *t*, a tuple in *r*
- if *i* is an integer in {1,...,k}

  t[*i*] (or t.i) indicates the *i-th* component of *t*

- example:
  - *r* = {(0,a), (0,c),(1,b)}
  - *t* = (0,a) is a tuple of *r*
  - *t*[2] = a
  - t[1] = 0
  - *t*[1,2] = (0, a)

0	a
0	c
1	b

Relations and Tables

- how do we interpret the data in the table?

Paolo	Rossi	2	26.5
Mario	Bianchi	10	28.7

Relations and Tables

- solution:
  - assign <u>names</u> to the table and the columns

Student

Name	**Surname**	**Exams**	**Avg**
Paolo	Rossi	2	26.5
Mario	Bianchi	10	28.7

from data to information!

Relations and Tables

- an ***attribute*** is defined by name A and its domain which we denote by *dom*(*A*)

- let *R* be a set of attributes: an *ennuple* (tuple) *on R* is a function defined on *R* that associates with each attribute *A* in *R* an element of *dom*(*A*)

- if *t* is an ennuple in *R* and *A* is an attribute in *R*, then by *t*(*A*) we will denote the value taken by the function *t* at the attribute *A*

Reports and Tables

relation schema

relation name

attributes

Student

Name	Surname	Exams	Avg

Reports and Tables

Student

Name	Surname	Exams	Avg
Paolo	Rossi	2	26.5
Mario	Bianchi	10	28.7

tuples

relation instance

Relations and tables: summary

- a relation can be regarded as a table in which **each row is a distinct tuple** and **each column corresponds to a component** (with **homogeneous** values, i.e., coming from the same domain)

- the **columns correspond to the domains** $D_1, D_2, \ldots D_k$ *having* **unique self-explanatory** names within the table

- the pair (attribute name, domain) is called an attribute; the set of attributes of a relation is called **schema**

- if R is a relation and its attributes are
  $A_1, A_2, \ldots, A_k$, the schema is often indicated as:
  $$\boldsymbol{R}(A_1, A_2, \ldots, A_k)$$

Schemas and instances

- **relation schema**: a relation name R with a set of attribute names:

$$\boldsymbol{R}(A_1, A_2, \ldots, A_k)$$

- the schema of a relation is invariant over time, and describes its **structure** (intentional aspect)

- the **instance** of a relation with schema R(X): a set r of tuples on X

- the **instance** contains the current values, which can also change very quickly (extensional aspect)

Relations and databases

- **database schema**: a set of relation schemas with different names
- **relational database schema**: set $R_1, R_2, ..., R_n$ of schemas
- **relational database** with schema $R_1, R_2, ..., R_n$: set $r_1, r_2, ..., r_n$ where r_i is a relation instance with schema R_i

Example

- Schema: Info_City(City,Region,Population)
- Info_City instance:

City	Region	Population
Rome	Lazio	3000000
Milan	Lombardy	1500000
Genoa	Liguria	800000
Pisa	Tuscany	150000

Relations and Tables

- in the definition of a relational model, the components of a relation are indicated by the names of attributes, rather than by their position

- $t[A_i]$ indicates the value of the attribute with name A_i of the tuple t

- if t is the second tuple in the previous example, then t[Region] = Lombardy

- if Y is a subset of attributes of the schema X of a relation ($Y \subseteq X$) then t[Y] is the subset of values in the tuple t that correspond to attributes in Y (also called **restriction** of t)

To recap

- Object = tuple (implemented as a record)
- Fields = Information of interest -> schema of the relation

- Subject = "Staff Member"
- Information of interest = Code, Surname, First name, Role, Hiring year

CODE	SURNAME	NAME	ROLE	HIRING
COD1	Rossi	Mario	Analyst	1995

To recap

- Table = Set of tuples of homogeneous type a particular **INSTANCE** of the relation

- STAFF table = Set of tuples of type "Staff Member".

CODE	SURNAME	NAME	ROLE	HIRING
COD1	Rossi	Mario	Analyst	1995
COD2	Bianchi	Peter	Analyst	1990
COD3	Neri	Paolo	Administrator	1985

The model is based on values

- in the relational model, references between data in different relations are represented by means of domain **values** that appear in the ennuples

students

Matric	Surname	Name	Date of birth
6554	Rossi	Mario	05/12/1978
8765	Neri	Paolo	03/11/1976
9283	Greens	Luisa	12/11/1979
3456	Rossi	Maria	01/02/1978

exams

Student	Grade	Course
3456	30	04
3456	24	02
9283	28	01

courses

Code	Title	Lecture
01	Chemistry	Mario
02	Math	Bruni
04	Chemistry	Verdi

Null values

- NULL values represent lack of information or the fact that the information is not applicable
- e.g. phone number:
  - the person doesn't have a phone
  - I don't know if the person has a phone
  - the person has a phone, but I don't know the number
- we can't avoid to enter a value (the tuple must adhere to the schema)! we can define a default value...
- warning: some attributes should never assume null values
  - student ID/matriculation
  - examination grades

Null value

- bad habit: "unused" domain values
  - they could be used later
  - they might skew the calculations (we know an employee's salary, but a 0 weighs in an average value calculation like any other number)
- special value: **NULL**
- NULL: polymorphic value = does not belong to any domain but can replace values in any domain
- two NULL values, even on the same domain, are considered different
- **Warning:** NULL is not 0 (integer)

Too many null values

students

Matric	Surname	Name	Date of birth
6554	Rossi	Mario	05/12/1978
9283	Greens	Luisa	12/11/1979
NULL	Rossi	Maria	01/02/1978

exams

Student	Vote	Course
NULL	30	NULL
NULL	24	02
9283	28	01

courses

Course	Title	Lecturer
01	NULL Math	NULL
02	Chemis try	02
04		01

An "incorrect" database

EMPLOYEE

CODE	SURNAME	NAME	ROLE	HIRING	DEPT
COD1	Rossi	Mario	Analyst	1795	01
COD2	Bianchi	Luigi	Analyst	1990	05
COD2	Neri	Paolo	Admin	1985	01

DEPARTMENT

NUMBER	NAME
01	Management
02	Administration

what's wrong? syntactically, it is correct...

An "incorrect" database

EMPLOYEE

CODE	SURNAME	NAME	ROLE	HIRING	DEPT
COD1	Rossi	Mario	Analyst	1795	01
COD2	Bianchi	Luigi	Analyst	1990	05
COD2	Neri	Paolo	Administrator	1985	01

DEPARTMENT

NUMBER	NAME
01	Management
02	Administration

An "incorrect" database

Exams

Student	Grade	Honor	Course
276545	32		01
276545	30	Yes	02
787643	27	Yes	03
739430	24		04

Students

Matric	Surname	Name
276545	Rossi	Mario
787643	Neri	Piero
787643	Bianchi	Luca

Integrity constraints

- **integrity constraint**: property that must be satisfied **by every instance of** the database
- constraints describe specific properties of the scope, and therefore of the information related to it modeled through the database
- a database instance is **correct** if it satisfies **all** integrity constraints associated with its schema

How to avoid "violations"

EMPLOYEE

CODE	SURNAME	NAME	ROLE	HIRING	DEPT
COD1	Rossi	Mario	Analyst	1795	01
COD2	Bianchi	Peter	Analyst	1990	05
COD2	Neri	Paolo	Administrator	1985	01

DEPARTMENT

NUMBER	NAME
01	Management
02	Administration

(HIRING > 1980)

CODE UNIQUE

DEPT REFERENCES DEPARTMENT.NUMBER

How to avoid "violations"

Exams

Student	Grade	Honor	Course
276545	32		01
276545	30	Yes	02
787643	27	Yes	03
739430	24		04

(Grade>=18) AND (Grade<=31)

(Vote=30) OR NOT (Honor="yes")

intra-relational constraints

Student references Students.Matric

inter-relational constraints

Students

Matric	Surname	Name
276545	Rossi	Mario
787643	Neri	Piero
787643	Bianchi	Luca

Unique Matric

Integrity constraints

- intra-relational constraints: defined on single attribute values (domain) or between values of the same tuple or between tuples of the same relation

- interrelational constraints: defined between multiple relations

Integrity constraints

- Domain Constraints
  - HIRING > 1980
  - (Grade>=18) AND (Grade<=31)
- Tuple Constraints
  - (Grade=31) OR NOT (Honor = "yes")
- Uniqueness constraints
  - Matric UNIQUE
- Constraints between values in tuples of different relations
  - DEPT REFERENCES DEPARTMENT.NUMBER
- Primary key constraints
  - unique
  - not null
- Value existence constraints
  - not null

Keys 1

- we need to uniquely identify the tuples of an instance
- a relation key (not necessarily unique) is an attribute or set of attributes that uniquely identifies a tuple

Keys 2

- a set X of attributes of a relation R is a key of R if it satisfies the following conditions:
- 1) for each instance of R, there do not exist two distinct tuples t_1 and t_2 having the same values for all attributes in X, i.e., such that $t_1[X] = t_2[X]$
- 2) there does not exist a proper subset of X satisfying condition 1)

Example

- instance of Staff:

ID	SURNAME	NAME	ROLE	HIRING
COD1	Rossi	Mario	Analyst	1995
COD2	Bianchi	Peter	Analyst	1990
COD3	Neri	Paolo	Admin	1985

Key(s)? based on this instance, any attribute except Role could be a key.... but it is correct?

Example

- Info_City report request

City	Region	Population
Rome	Lazio	3000000
Milan	Lombardy	1500000
Genoa	Liguria	800000
Pisa	Tuscany	150000

Key? does it depend... on the "size" of the city?

Keys 3

- a relation could have several alternative keys
- we choose the most used one or the one consisting of a smaller number of attributes = **primary** key
- the primary key does not allow null values
- there is always at least one key. Why? there can't be identical tuples!
- the keys allow us to refer to data in different tables

Example

- instance of Staff

TAXCODE	CODE	SURNAME	NAME	ROLE	HIRING
CSR...	COD1	Rossi	Mario	Analyst	1995
BA...	COD2	Bianchi	Peter	Analyst	1990
NRI...	COD3	Neri	Paolo	Admin	1985

- keys?

- according to the definition, is it possible that (Surname, Role) is a key?

Interrelational Constraints

- referential integrity constraint (<u>foreign key</u>): portions of information in different relations are associated by (the same) key values
- the first relation refers to the value of an attribute or set of attributes that should appear in the second relation
- a referential integrity constraint between the X attributes of a relation R_1 and another relation R_2 forces the values on X in R_1 to appear as values of the primary key of R_2

Road violations

Code	Date	Officer	Prov	Plate
34321	1/2/95	3987	MI	39548K
53524	4/3/95	3295	TO	E39548
64521	5/4/96	3295	PR	839548
73321	5/2/98	9345	PR	839548

Officers

ID	Surname	Name
3987	Bianchi	Luca
3295	Gialli	Piero
9345	Rossi	Mario
7543	Verdi	Gino

Road violations

Code	Date	Officer	Prov	Plate
34321	1/2/95	3987	MI	39548K
53524	4/3/95	3295	TO	E39548
64521	5/4/96	3295	PR	839548
73321	5/2/98	9345	PR	839548

Cars

Prov	Plate	Surname	Name
MI	E39548	Rossi	Mario
TO	F34268	Rossi	Mario
PR	839548	Neri	Luigi

Interrelational Constraints

- referential integrity constraints between:
  - the Officer attribute of the relation Road violations and the attribute ID (key) of the relation Officers
  - the attributes Prov and Plate of Road violations and the attributes Prov and Plate (key) of the relation Cars

Breach of referential integrity constraint

Road violations

Code	Date	Officer	Prov	Plate
34321	1/2/95	3987	MI	39548K
53524	4/3/95	3295	TO	E39548
64521	5/4/96	3295	PR	839548
73321				

Cars

Prov	Plate	Surname	Name
MI	E39548	Rossi	Mario
TO	F34268	Rossi	Mario
PR	839548	Neri	Luigi

BEWARE OF CONSTRAINTS ON MULTIPLE ATTRIBUTES

Referential integrity constraints: comments

- they play a key role in the concept of a "values-based model" (the relational model)

- in the presence of null values the constraints can be made less restrictive

- it is possible to define compensatory "actions" following violations

- beware of constraints on multiple attributes

Referential integrity and null values

Employees

ID	Project	Name
34321	Rossi	IDEA
53524	Neri	XYZ
64521	Greens	NULL
73032	Bianchi	IDEA

The referential integrity constraint is not violated by the NULL value

Projects

Name	Start	Duration	Cost
IDEA	01/2000	36	200
XYZ	07/2001	24	120
BOH	09/2001	24	150

Multiple Constraints on Multiple Attributes

- not all properties of interest can be represented by explicit constraints in the logic model ...
- ...but that will be covered in unit 2
- how do we formalize the constraints?
- intra-relational constraints ⊇ relations between attribute values of the same tuple -> **functional dependencies defined on the same schema**

Functional dependencies

- a functional dependency establishes a semantic link between two non-empty sets of attributes X and Y belonging to a schema R
- this constraint is written $X \rightarrow Y$ and is read:
- "*X determines Y*"

Functional dependencies: example

- suppose we have a relation scheme Flights (Code, Day, Pilot, Time)
- with the constraints, dictated by common sense:
  - a flight with a certain code always leaves at the same time
  - there is only one flight with a given pilot, on a given day at a given time
  - there is only one pilot of a given flight on a given day
- constraints give raise to the functional dependencies:
  - Code $\rightarrow$ Time
  - {Day, Pilot, Time} $\rightarrow$ Code
  - {Code, Day} $\rightarrow$ Pilot

Functional dependencies: are satisfied if...

- we say that a relation r with schema R satisfies the functional dependence $X \rightarrow Y$ if:
  - (i) the functional dependence $X \rightarrow Y$ is applicable to R, in the sense that both X and Y are subsets of R;
  - (ii) ennuples in r that are identical on X are also identical on Y, i.e., for each pair of ennuples t_1 and t_2 in r:

$$t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y].$$

important note: right arrow means that if tuples are equal on X, then they must also be equal on Y