Databases

Unit 2
Relational data model and relational algebra

Relational model and relational algebra

- Relational data model
- Relational algebra

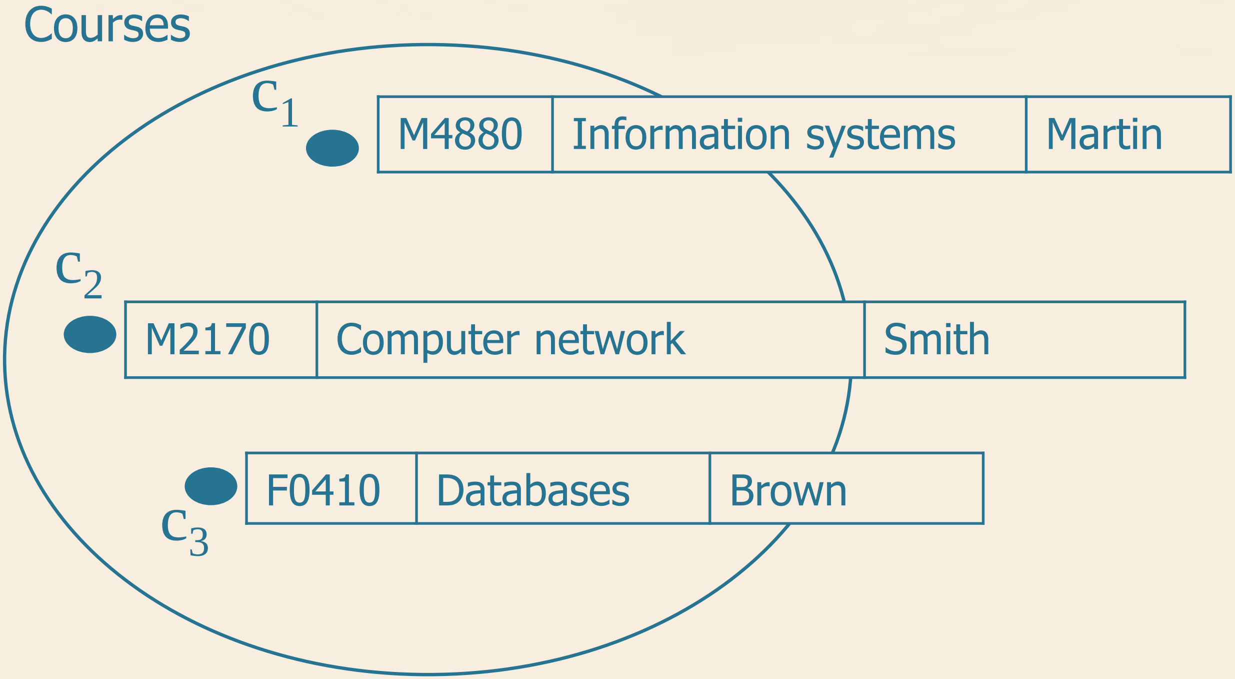
Relational data model

- Introduction
- Definitions
- References between relations
- Incomplete information
- Integrity constraints
- Primary key
- Tuple constraint and domain constraint
- Referential integrity constraint

Relational data model

Introduction

Intuition



M4880	Information systems	Martin
M2170	Computer network	Smith
F0410	Databases	Brown

Relational model

- Proposed by E. F. Codd in 1970 to support higher abstract levels compared to the previous models
  - Data independence
- Made available in commercial DBMSs in 1981,
  - Today it is the main model exploited in commercial DBMSs
- Based on (a variant of) *relation* mathematical concept
  - Each relation is represented in the informal way by means of a table

Example

Courses

Code	Name	TeacherID
M2170	Information systems	D101
M4880	Computer Networks	D102
F0410	Databases	D321

Teachers

ID	Name	Department	Phone#
D101	Green	Computer Engeneering	123456
D102	White	Telecommunications	636363
D321	Black	Computer Engeneering	414243

Realtional model

Definitions

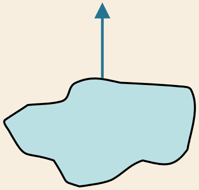
Definitions

attributes

Courses

n-tuples

Code	Name	TeacherID
M2170	Information systems	D101
M4880	Computer Networks	D102
F0410	Databases	D321



Domain: identifier set of the Politecnico courses

Definitions

- Attribute
  - Column name of a table
- Domain
  - Value set that can be assumed by an attribute
- Tuple (or record)
  - A row in a table
- Cardinality
  - Number of tuples in a relation
- Degree
  - Number of attributes in a relation

Properties

- Tuples (rows) *are not* ordered
- Tuples are *distinct* among them (there are not duplicated rows)
- Attributes are not ordered
  - It is not possible to identify an attribute by means of its position

Relational model

References between relations

References between relations

- The relational model is *value-based*
  - References between data in different relations are represented by means of values of the domains

Value-based reference: Example

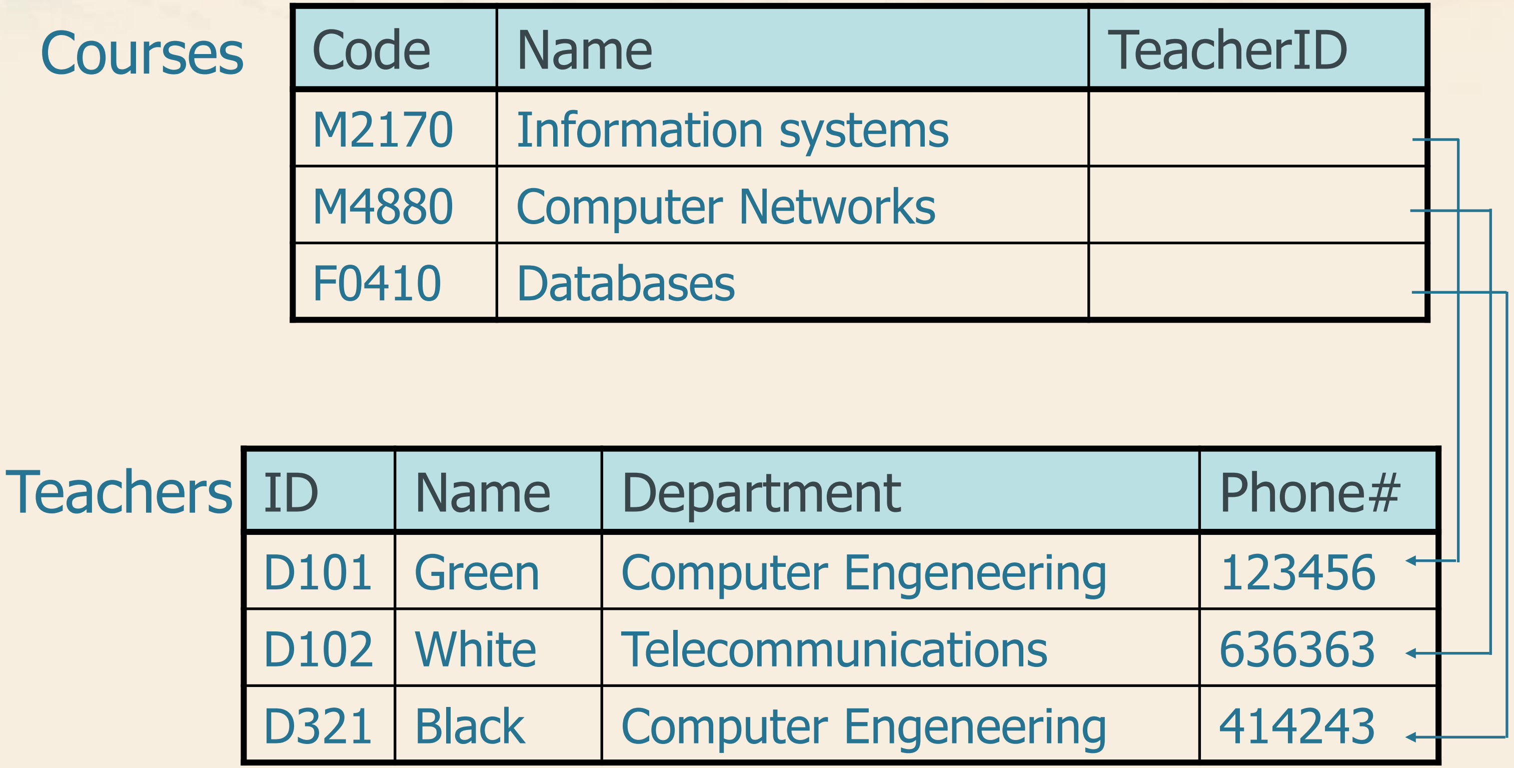
Courses

Code	Name	TeacherID
M2170	Information systems	D101
M4880	Computer Networks	D102
F0410	Databases	D321

Teachers

ID	Name	Department	Phone#
D101	Green	Computer Engeneering	123456
D102	White	Telecommunications	636363
D321	Black	Computer Engeneering	414243

Pointer-based reference: Example



Courses

Code	Name	TeacherID
M2170	Information systems	
M4880	Computer Networks	
F0410	Databases	

Teachers

ID	Name	Department	Phone#
D101	Green	Computer Engeneering	123456
D102	White	Telecommunications	636363
D321	Black	Computer Engeneering	414243

References between relations

- The relational model is *value-based*
  - References between data in different relations are represented by means of values of the domains
- Advantages
  - Independence of physical structures
  - Only information that is relevant from the application point of view is stored
  - Easy transferrability of data between different systems
  - Differently from pointers, the link is not oriented

Relational model

Null values

Incomplete information

- Some information could be not available for any tuples in the relation
- Example
  Student (StudentID, Surname,BirthDate, Phone#,
  DegreeYear)
  - The phone number could be (temporarily?) unknown
  - for students not yet graduated, year degree is not defined
  - for students just graduated, degree year is not yet defined or unknown

Null values

- To represent lack of information we should use a special value belonging to the domain (0, empty string, 999, …)
  - A value not used is required (example: DegreeYear=0, Phone#=?)
  - “unused” values could become meaningful (Phone#= 999999)
  - it is necessary to deal separately with "special" values in different application
- The representation is not adequate

Null value

- Definition of a spelcial value named *null value* (NULL)
  - It is not a value of the domain
  - It denotes both the absence of a domain value and value not defined
  - It must be used with caution (example: StudentID=NULL?)

Relational model

Integrity constraints

Integrity constraints

Courses

Code	Name	TeacherID
M2170	Information systems	D101
M4880	Computer Networks	D102
F0410	Databases	D321

Teachers

ID	Name	Department	Phone#
D101	Green	Computer Engeneering	123456
D102	White	Telecommunications	636363
D321	Black	Computer Engeneering	414243

Integrity constraints

Courses

Code	Name	TeacherID
M2170	Information systems	D101
F0410	Computer Networks	D102
F0410	Databases	D321

Teachers

ID	Name	Department	Phone#
D101	Green	Computer Engeneering	123456
D102	White	Telecommunications	636363
D321	Black	Computer Engeneering	414243

Integrity constraints

Courses

Code	Name	TeacherID
M2170	Information systems	D101
M4880	Computer Networks	D102
F0410	Databases	~~D342~~

Teachers

ID	Name	Department	Phone#
D101	Green	Computer Engeneering	123456
D102	White	Telecommunications	636363
D321	Black	Computer Engeneering	414243

Integrity constraints

Courses

Code	Name	TeacherID
M2170	Information systems	D101
M4880	Computer Networks	D102
F0410	Databases	D321

Teachers

ID	Name	Department	Phone#
D101	Green	Computer Engeneering	123456
D102	White	Telecommunications	636363
D321	Black	Computer Engeneering	~~000001~~

Integrity constraint

- Integrity constraint
  - a property that must be satisfied by all meaningful database instances
- Types of constraint
  - Intra-relational constraints, defined on the attributes of a single relation (examples: unique constraint, domain constraints, tuple constraints)
  - Inter-relational constraints, defined on many relations at the same time (example: referential constraint)

Relational model

Primary key

Unique identification for tuples

Students

StudentID	Name	Surname	BirthDate	EnrollementYear
64655	Mike	Red	4/8/1978	1998
81999	Paul	White	4/8/1978	1999
75222	Marco	Red	8/3/1979	1998

- There is no pair of students with the same value for the StudentID
  - The StudentID uniquely identifies students

Unique identification for tuples

Students

StudentID	Name	Surname	BirthDate	EnrollementYear
64655	Mike	Red	4/8/1978	1998
81999	Paul	White	4/8/1978	1999
75222	Marco	Red	8/3/1979	1998

- There is no pair of students with the same value for the personal data
  - name, surname and birth date uniquely identify students

Key

- A *key* is an attribute set that uniquely identifies tuples in a relation
  - It is a property of the relational schema
- Formal definition: a set K of attributes is a key in a relation r if
  - The relation r does not contain a pair of distinct tuples with the same values for K (univocity)
  - K is minimal (there exists no other superkey K’ of r that is contained in K as proper subset)

Example

- The attribute

  {StudentID}

  is unique and minimal, thus it is a key

- The attribute set

  {Name, Surname, BirthDate}

  is unique and minimal (none of its subsets is unique), thus it is a key

Superkey

- A set K of attributes is a key in a relation r if
  - The relation r does not contain a pair of distinct tuples with the same values for K (univocity)
  - K is minimal (there are not proper subsets of K still unique)
- If only the first property is satisfied, K is a *superkey* of r

Examples

- The attribute set

  {StudentID,Name}

  is unique, but no minimal (the StudentID is unique), thus the attribute set is a superkey, but it is *not* a key

- The attribute set

  {BirthDate, EnrollementYear}

  is unique and minimal: is it a general property?

Primary key

- If a key can assume the NULL value, it cannot be a key (the univocity property is lost)
  - It is mandatory to avoid the NULL values in the keys
- Solution
  - a reference key, which does not allow null values, is defined. It is called *primary key*
  - The other keys (candidate keys) can assume null values
  - References between data in different relations are defined by means of the primary key

Relational model

Tuple constraint and domain constraint

Domain constraint

- Domain constraint
  - expresses conditions on the value assumed by a single attribute of a tuple
    - It can be a Boolean expression (and, or, not) of simple predicates
  - example: Score > 0 and Score ≤ 30

Tuple constraint

- Tuple constraint
  - expresses conditions on the values of each tuple, indipendently of other tuples
    - It can correlate many attributes
    - It can be a Boolean expression (and, or, not) of simple predicates (e.g., comparison between attributes, between an attribute and a constant)
  - example: Price = Cost + TaxPerc*Cost

Relational model

Referential integrity constraint

Referential integrity constraint

- Information in different relations are correlated by common values of one or more attributes

Courses

Code	Name	TeacherID
M2170	Information systems	D101
M4880	Computer Networks	D102
F0410	Databases	D321

Teachers

ID	Name	Department	Phone#
D101	Green	Computer Engeneering	123456
D102	White	Telecommunications	636363
D321	Black	Computer Engeneering	414243

Referential integrity constraint

- Information in different relations are correlated by common values of one or more attributes
  - The TeacherID attribute in the COURSES relation refers the ID attribute in TEACHERS
- The values of an attribute in the referencing/internal relation must exist as values of an attribute in the instance of the referenced/external relation
  - The values of TeacherID in the COURSES relation must exist as values of the ID attribute in TEACHERS

Referential integrity constraint

- Referential constraint
  - Given two relations
    - R (referenced/external relation)
    - S, that refers R through a set X of attributes (referencing/internal relation)
  - values on a set X of attributes in a relation S can be exclusively values for the primary key of the relation R
- The set X of attributes in S represents its foreign key

Referential integrity constraint

- Referential integrity constraints are imposed in order to guarantee that the values refer to actual values in the referenced relation (the relational model is value-based)

Example

Flight

F-ID	Date
AZ111	10/16/1996
AZ234	12/4/1998
AZ543	3/9/2000

Ticket

F-ID	Date	Seat#	Passenger
AZ111	10/16/1996	23	Luis Red
AZ111	10/16/1996	56	John White
AZ234	12/4/1998	9	Mark Black
AZ234	12/4/1998	11	Joe Green
AZ234	12/4/1998	21	Paul Red

Example

Flight

F-ID	*Date*
AZ111	10/16/1996
AZ234	12/4/1998
AZ543	3/9/2000

Ticket

F-ID	Date	Seat#	Passenger
AZ111	10/16/1996	23	Luis Red
AZ111	10/16/1996	56	John White
AZ234	12/4/1998	9	Mark Black
AZ234	12/4/1998	11	Joe Green
AZ234	12/4/1998	21	Paul Red

Example

Flight

F-ID	*Date*
AZ111	10/16/1996
AZ234	12/4/1998
AZ543	3/9/2000

Ticket

F-ID	Date	Seat#	Passenger
AZ111	10/16/1996	23	Luis Red
AZ111	~~*11/16/1996*~~	56	John White
AZ234	12/4/1998	9	Mark Black
AZ234	12/4/1998	11	Joe Green
AZ234	12/4/1998	21	Paul Red

Example

Flight

F-ID	*Date*
AZ111	10/16/1996
AZ234	12/4/1998
AZ543	3/9/2000

Ticket

F-ID	Date	Seat#	Passenger
AZ111	10/16/1996	23	Luis Red
AZ111	10/16/1996	56	John White
AZ234	12/4/1998	9	Mark Black
AZ234	12/4/1998	11	Joe Green
~~*AZ534*~~	~~*12/4/1998*~~	21	Paul Red