Basi di Dati e
Sistemi Informativi I

Relational databases:
the relational model

(2\2)

A meaningless database instance

Exams

RegNum	Name	Course	Grade	Honours
6554	Rossi	B01	K	
8765	Neri	B03	C	
3456	Bruni	B04	B	honours
3456	Verdi	B03	A	honours

Courses

Code	Title
B01	Physics
B02	Calculus
B03	Chemistry

- grades are between A and F
- honours can be awarded only if grade is A
- different students must have different registration numbers
- exames must refer to existing courses

Integrity constraints

- **integrity constraint**: a property that must be satisfied by all meaningful database instances;
- it can be seen as a **predicate**, which associates the value *true* or *false* with each instance
- In general, we associate a collection of constraints with a database schema and we consider *correct* (or *legal*) the instance that satisfy all the constraints.

*a database instance is **legal** if it satisfies all integrity constraints*

Types of constraints

- *Intra-relational* constraints, when its satisfaction is defined with regard to a single relation of the database (the first three cases in the previous example)
  - special cases:
    - *domain constraints* is a constraint defined with reference to single values (first case in the example). Domain because it imposes a restriction on the domain for the attribute.
    - *tuple constraints* is a constraint that can be evaluated on each tuple independently from the others (first two cases in the example)
- *Inter-relational* constraints, when it involves more than one relation (fourth case in the example)

Integrity constraints, motivations

- Useful to describe the application in greater detail
- A contribution to “data quality”
- An element in the design process (we will discuss “normal forms”)
- Used by the system in choosing the strategy for query processing

Note:

- it is not the case that all properties can be described by means of integrity constraints

Tuple constraints

- express conditions on the values of each tuple, independently of other tuples
- a possible syntax: boolean expressions (that is, with connectives AND, OR, and NOT) with atoms that compare attributes, constants or expressions over them
- domain constraint: a tuple constraint that involve a single attribute
- a domain constraint

  (Grade ≥ “A”) AND (Grade ≤ “F”)
- a tuple constraint

  (NOT (Honours = “honours”)) OR (Grade = “A”)

- More complex expressions, for example

  PAYMENTS(Date,Amount,Deductions,Net)

  Net = Amount - Deductions

Unique identification of tuples

RegNum	Surname	FirstName	BirthDate	DegreeProg
284328	Smith	Luigi	29/04/59	Computing
296328	Smith	John	29/04/59	Computing
587614	Smith	Lucy	01/05/61	Engineering
934856	Black	Lucy	01/05/61	Fine Art
965536	Black	Lucy	05/03/58	Fine Art

- the registration number identifies students:
  - there is no pair of tuples with the same value for RegNum
- personal data identifies students:
  - there is no pair of tuples with the same values on each of Surname, FirstName, BirthDate

Keys

- **Key** :
  - a set of attributes that uniquely identifies tuples in a relation
- more precisely: à
  - a set of attributes K is a **superkey** for a relation r if r does not contain two distinct tuples t_1 and t_2 with $t_1[K]=t_2[K]$;
  - K is a **key** for r if K is a minimal superkey (that is, there exists no other superkey K' of r that is contained in K as proper subset)

RegNum	Surname	FirstName	BirthDate	DegreeProg
284328	Smith	Luigi	29/04/59	Computing
296328	Smith	John	29/04/59	Computing
587614	Smith	Lucy	01/05/61	Engineering
934856	Black	Lucy	01/05/61	Fine Art
965536	Black	Lucy	05/03/58	Fine Art

- RegNum is a key:
  - RegNum is a superkey
  - it contains a sole attribute, so it is minimal
- Surname, Firstname, BirthDate is another key:
  - Surname, Firstname, BirthDate form a superkey
  - no proper subset is also a superkey

RegNum	Surname	FirstName	BirthDate	DegreeProg
284328	Smith	Luigi	29/04/59	Computing
296328	Smith	John	29/04/59	Computing
587614	Smith	Lucy	01/05/61	Engineering
934856	Black	Lucy	01/05/61	Fine Art
965536	Black	Lucy	05/03/58	Fine Art

- RegNum, DegreeProg is a superkey, but it is not a minimal superkey, because on of its proper subsets, RegNum, is itself a minimal superkey, and thus RegNum, DegreeProg is not a key
- Firstname, DegreeProg is not a supekey, because there are two tuples in the raltion, the last two equal, on both FirstName and DegreeProg

another relation:

RegNum	Surname	FirstName	BirthDate	DegreeProg
296328	Smith	John	29/04/59	Computing
587614	Smith	Lucy	01/05/61	Engineering
934856	Black	Lucy	01/05/61	Fine Art
965536	Black	Lucy	05/03/58	Engineering

- there is no pair of tuples with the same values on both Surname and DegreeProg:
  - in each programme students have different surnames;
  - Surname and DegreeProg form a key for this relation
- is this a general property?
  - No! There could be students with the same surname in the same programme

Keys, schemas, and instances

- Constraints correspond to properties in the real world to be modelled by our database
- therefore, they are relevant at the schema level (wrt the whole set of instances)
  - we associate with a schema a set of constraints, and we consider as correct (legal, valid, …) the instances that satisfy all the constraints
  - individual instances could satisfy (“by chance”) other constraints

Existence of keys

- Relations are sets; therefore each relation is composed of distinct tuples: the whole set of attributes is a superkey;
- so each relation has a superkey; since the set of attributes is finite, each relation schema has at least a key:
  - the whole set is either a key
  - or it contains a (smaller superkey), and for it we can repeat the argument, over a smaller set

Importance of keys

- The existence of keys guarantees that each piece of data in the database can be accessed
- Keys are the major feature that allows us to say that the relational model is “value-based”
- The values of the key attributes are used to refer to the content of each of the relations from outside (that is from other relations)

Students

RegNum	Surname	FirstName	BirthDate
6554	Rossi	Mario	5/12/1978
8765	Neri	Paolo	3/11/1976
9283	Verdi	Luisa	12/11/1979
3456	Rossi	Maria	1/2/1978

Courses

Code	Title	Tutor
01	Analisi	Neri
02	Chimica	Bruni
04	Chimica	Verdi

Exams

Student	Grade	Course
3456	30	04
3456	24	02
9283	28	01
6554	26	01

Keys and null values

- If there are nulls, keys do not work that well
  - they do not guarantee unique identification
  - they do not allow to establish correspondences between data in different relations

RegNum	Surname	FirstName	BirthDate	DegreeProg
NULL	Smith	John	NULL	Computing
587614	Smith	Lucy	01/05/61	Engineering
934856	Black	Lucy	NULL	NULL
NULL	Black	Lucy	05/03/58	Engineering

  - How do we access the first tuple?
  - Are the third and fourth tuple the same?

Primary keys

- The presence of nulls in keys has to be limited
- Practical solution: for each relation we select a **primary key** on which nulls are not allowed
  - notation: the attributes in the primary key are underlined
- References between relations are realized through primary keys

RegNum	Surname	FirstName	BirthDate	DegreeProg
643976	Smith	John	NULL	Computing
587614	Smith	Lucy	01/05/61	Engineering
934856	Black	Lucy	NULL	NULL
735591	Black	Lucy	05/03/58	Engineering

Primary keys: do we always have them?

- In most cases we do have reasonable primary keys
- In other cases we don't:
  - we need to introduced new attributes (identifying "codes")
- Note that most of the "obvious" codes we have now (social security number, student number, area code, …) were introduced (possibly before the adoption of databases) with the same goal: unambiguous identification of objects

Referential constraints (“foreign keys”)

- Pieces of data in different relations are correlated by means of values of (primary) keys
- Referential integrity constraints are imposed in order to guarantee that the values refer to actual values in the referenced relation

A database with referential constraints

Offences

Code	Date	Officer	Dept	Registartion
143256	25/10/1992	567	75	5694 FR
987554	26/10/1992	456	75	5694 FR
987557	26/10/1992	456	75	6544 XY
630876	15/10/1992	456	47	6544 XY
539856	12/10/1992	567	47	6544 XY

Officers

RegNum	Surname	FirstName
567	Brun	Jean
456	Larue	Henri
638	Larue	Jacques

Cars

Registration	Dept	Owner	…
6544 XY	75	Cordon Edouard	…
7122 HT	75	Cordon Edouard	…
5694 FR	75	Latour Hortense	…
6544 XY	47	Mimault Bernard	…

Referential constraints

- A **referential constraint** imposes to the values on a set X of attributes of a relation R_1 to appear as values for the primary key of another relation R_2
- If the key of R_2 is unique and consists of a sole attribute B (and therefore the set X is in its turn made up of a sole attribute A), then the referencial contraint between the attribute A of R_1 and the relation R_2 is satisfied if, for every tuple t_1 in R_1 such that $t_1[A]$ is not null, there exists a tuple t_2 in R_2 such that $t_1[A]= t_2[B]$

- In general, each of the attributes in X must correspond to a precise attribute of the primary key of R_2. For this, it is necessary to specify an order both in the set X and in K.
- Indicating the attributes in order $X= A_1 A_2 ... A_p$, $K= B_1 B_2 ... B_p$ the constarint is satisfied if for every tuple t_1 in R_1 with no nulls in X, there exists a tuple t_2 in R_2 with $t_1[A_i]= t_2[B_i]$, for $1 \leq i \leq p$

Referential constraints

- In the example, we have referential constraints between
  - the attribute Officer of Offences and relation Officers
  - the attributes Registration and Department of Offences and relation Cars

  The database of the example satisfies both constarints.

Database that violates referential constraints

Offences

Code	Date	Officer	Dept	Registartion
987554	26/10/1992	456	75	6544 XY
630876	15/10/1992	456	47	6544 XY

Officers

RegNum	Surname	FirstName
567	Brun	Jean
638	Larue	Jacques

Cars

Registration	Dept	Owner	…
7122 HT	75	Cordon Edouard	…
5694 FR	93	Latour Hortense	…
6544 XY	47	Mimault Bernard	…

- Officers does not contain a tuple on RegNum equal to 456
- Cars contains no tuple with the value 75 for Dept and 6544XY for Registration (note that there is a tuple with the value 75 for Dept and another with the value 6544XY for Registration, but this is not sufficient, because there is a need for a tuple with both values; only in this way can the two values make reference to a tuple of the relation Cars).

Referential constraints: comments

- Referential constraints play an essential role in the issue “the relational model is value-based.”
- It is possible to have features that support the management of referential constraints (“actions” activated by violations)
- In presence of null values definitions have to be adapted
- Care is needed in case of constraints that involve two or more attributes

Integrity constraints can get intricated

Accidents

Code	Dept1	Registration1	Dept2	Registration2
6207	75	6544 XY	93	9775 GF
6974	93	5694 FR	93	9775 GF

Cars

Registration	Dept	Owner	…
7122 HT	75	Cordon Edouard	…
5694 FR	93	Latour Hortense	…
9775 GF	93	LeBlanc Pierre	
6544 XY	75	Mimault Bernard	…

- we have two referential constraints
  - from Registration1, Dept1 to Cars
  - from Registration2, Dept2 to Cars

  Note that ordering in the set of attributes is essential!

  The key of cars is Registration, Dept and not Dept, Registration

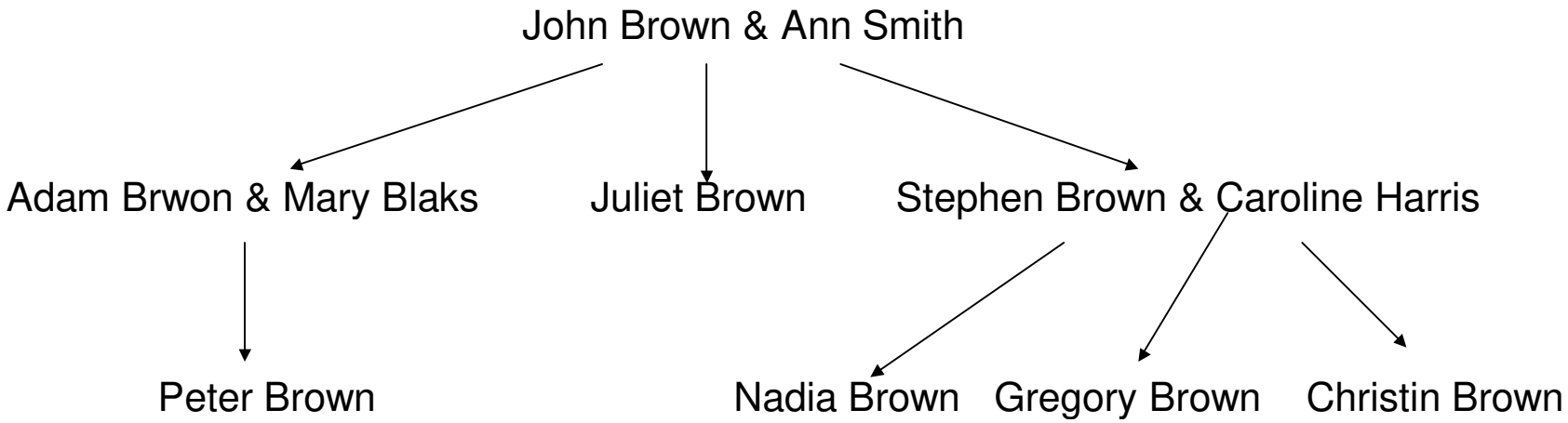
A final observation

- In the case we need to regard relations with more than one key, one of the keys should be indicated as the primary key, and the references should be direct towards it.
- Not all DBMSs on market allow the explicit indication of the primary key. In these cases, the referencial constraint must indicate explicitly the attributes that make up the key to which it refers.

Exercise

A family tree represents the structure of a family. Show hoe the information of a family tree can be represented by means of a relational database.

A tipical family tree may be like this:



Exercise: solution

MARRIAGE(Husband, Wife)
PATERNITY(Father,Child)

MARRIGE

Husband	**Wife**
John Brown	Ann Smith
Adam Brown	Mary Blacks
Stephen Brown	Caroline Harris

PATERNITY

Father	**Child**
John Brown	Adam Brown
John Brown	Juliet Brown
John Brown	Stephen Brown
Adam Brown	Peter Brown
Stephen Brown	Nadia Harris
Stephen Brown	Cregory Brown
Stephen Brown	Christine Brown