

# **IL MODELLO RELAZIONALE**

# Il modello relazionale: concetti generali

- Il modello fu proposto da E.F. Codd (IBM Research) nel 1970 in un famosissimo articolo:
  - "A Relational Model for Large Shared Data Banks," Communications of the ACM, June 1970
- Il modello si basa sul concetto matematico di relazione, basato sulla teoria degli insiemi
- Questo lavoro fu alla base di una profonda rivoluzione nel campo delle basi di dati, che portò Codd a vincere il prestigioso ACM Turing Award

# Cos'è una relazione?

- *Informalmente*, una **relazione** può essere vista come una **tabella** con un insieme di valori su ogni riga
- Ci sono due livelli che definiscono una relazione:
  - Lo **schema** della relazione (livello *intensionale*)
  - **Istanze** della relazione (livello *estensionale*)

Field names

FIELDS (ATTRIBUTES, COLUMNS)

<i>sid</i>	<i>name</i>	<i>login</i>	<i>age</i>	<i>gpa</i>
50000	Dave	dave@cs	19	3.3
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

TUPLES  
(RECORDS,  
ROWS)

# Relazioni: definizione intensionale

- Lo schema di una relazione definisce:
  - Il nome della relazione (per esempio, STUDENTI)
  - Il nome di ogni attributo (per esempio, SID, NOME, LOGIN, ...)
  - Il dominio di ogni attributo (per esempio, INTEGER, STRING, ...), ovvero l'insieme dei valori che quell'attributo può assumere
- L'ordine degli attributi nello schema non ha alcun significato specifico

# Relazioni: definizione intensionale

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

- Un modo standard di rappresentare lo schema di una relazione è il seguente:

`Students(sid:string, name:string, login:string, age:integer, gpa:real)`

- Il numero di attributi definisce il **grado** (o *arietà*) **della relazione** (5 nell'esempio sopra)

# Relazioni: definizione estensionale

- Un'istanza di uno schema di relazione è un insieme di **tuple** (o **record**), ognuna delle quali ha lo stesso numero di campi dello schema della relazione
  - Essendo un insieme, non possono esserci duplicati!
  - L'ordine delle tuple non ha alcun valore specifico
- *Informalmente*, un'istanza di una relazione può essere pensata come una riga di una tabella, dove ovviamente tutte le righe hanno lo stesso numero di campi
- Il numero di istanze della definisce la **cardinalità** della relazione (6 nell'esempio sulla slide #3)

# Stato di una relazione

Lo **stato di una relazione** è un sottoinsieme del prodotto cartesiano dei domini dei suoi attributi:

- Data una relazione  $R(A_1, A_2, \dots, A_n)$ , uno specifico stato di  $R$  è definito come segue:

$$r(R) \subset \text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n)$$

- ovvero:

- $r(R) = \{t_1, t_2, \dots, t_n\}$  dove ogni  $t_i$  è una tupla
- $t_i = \langle v_1, v_2, \dots, v_n \rangle$  dove ogni  $v_j$  è un elemento di  $\text{dom}(A_j)$

- Nello stato di una relazione, l'ordine delle tuple non conta (è un insieme!)

# Esempio

- Sia  $R(A_1, A_2)$  lo schema di una relazione con:
  - $\text{dom}(A_1) = \{0,1\}$
  - $\text{dom}(A_2) = \{a,b,c\}$
- $\text{dom}(A_1) \times \text{dom}(A_2)$  è l'insieme di tutte le possibili combinazioni dei valori di  $A_1$  e  $A_2$  (in quest'ordine)  
 $\{ \langle 0,a \rangle, \langle 0,b \rangle, \langle 0,c \rangle, \langle 1,a \rangle, \langle 1,b \rangle, \langle 1,c \rangle \}$
- Un possibile valore di  $r(R)$  è  $\{ \langle 0,a \rangle, \langle 0,b \rangle, \langle 1,c \rangle \}$ 
  - Si dice che questo è un possibile stato (o «popolazione» o «estensione»)  $r$  della relazione  $R$ , definita su  $A_1$  and  $A_2$
  - Questo stato ha cardinalità 3 (ci sono 3 tuple)



# Chiave di una relazione

- Ogni riga di una relazione ha un campo (o un insieme di campi) il cui valore (o i cui valori) identificano unicamente quella riga in quella tabella
  - Si parla di **chiave** (*key*) della relazione
    - Nella relazione STUDENTS, l'attributo SSN ha valori che non possono ripetersi mai uguali in due righe diverse e quindi identifica univocamente uno studente
- Talvolta si usano valori convenzionali per identificare una riga in una tabella
  - Si parla di *chiavi artificiali* (*artificial key*) o *chiavi surrogate* (*surrogate key*)

# Terminologia

<u>Informale</u>	<u>Formale</u>
Tabella	Relazione
Intestazione colonna	Attributo
Tutti i possibili valori di una colonna	Dominio
Riga della tabella	Tupla
Definizione della tabella	Schema della relazione
Tabella popolata	Stato della Relazione

# VINCOLI

I vincoli determinano quali stati di una relazione in una base di dati relazionale sono ammissibili e quali non lo sono

I principali vincoli sono di tre tipi:

1. **Vincoli impliciti:** dipendono dal *data model* stesso (per es. il modello relazionale non ammette liste come valore di alcun attributo)
2. **Vincoli basati sullo schema (o espliciti):** sono definiti nello schema usando gli strumenti forniti dal modello (per es. un vincolo di partecipazione totale nel modello ER)
3. **Vincoli applicativi o semantici:** si tratta di vincoli che vanno al di là del potere espressivo del modello e devono essere imposti a livello di programma applicativo (per es. che un libro deve essere restituito alla biblioteca entro 30 giorni dal prestito)

# Vincoli di integrità relazionale

- Un vincolo è definito come una **condizione** che DEVE valere affinché lo **stato** di una relazione sia **valido**
- I principali tipi di vincoli espliciti che possono essere espressi nel modello relazionale sono:
  - Vincolo di **dominio** (già discusso)
  - Vincolo di **chiave**
  - Vincolo di **integrità** delle entità
  - Vincolo di **integrità referenziale**

# Vincolo di chiave - Definizioni

- **Superchiave** di una relazione R:
  - È un insieme di attributi  $S_K$  di R tali che:
    - Non esistono due tuple di  $r(R)$  in cui gli attributi in  $S_K$  hanno lo stesso valore (ovvero, se  $t_1$  e  $t_2$  sono tuple distinte di  $r(R)$ ,  $t_1[S_K] \neq t_2[S_K]$ )
    - Questa condizione deve essere rispettata in *ogni stato valido* di R
- **Chiave** di una relazione R:
  - Una chiave è una superchiave **minimale**, ovvero una superchiave tale che la rimozione di qualsiasi attributo da  $S_K$  produrrebbe un insieme di attributi che non è più una superchiave di R
  - Ogni chiave minimale è detta anche una **chiave candidata**
- **Una chiave è sempre una superchiave, ma non viceversa**

# Esempio

- Consideriamo la relazione STUDENT qui sotto:
  - STUDENT ha due chiavi (*candidate*):
    - Chiave 1 = {sid}
    - Chiave 2 = {login}
  - Entrambe sono chiavi (e quindi superchiavi) di STUDENT
  - {sid, name} è una superchiave ma *non* una chiave (perché?)
  - Ogni insieme di attributi che contiene una chiave è ovviamente una superchiave

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

# Vincolo di chiave

- Se una relazione ha più di una **chiave candidata**, una viene scelta (tipicamente dal DBA) come **chiave primaria**.
  - In generale, viene scelta la chiave candidata più piccola (come numero di attributi che la compongono)
  - Quando non è possibile, possono intervenire fattori soggettivi o addirittura arbitrari
- I valori della chiave primaria sono usati per *identificare in modo univoco* ogni tupla della relazione
  - In un certo senso forniscono l'«identità» della tupla
- Come vedremo, può essere usata per fare riferimento a quella tupla da tuple di un'altra relazione

# Vincolo di integrità di un'Entità (*entity integrity*)

- Nessuno degli attributi che compongono la chiave primaria  $P_K$  di una relazione  $R$  può avere valore NULL in alcuna tupla di  $r(R)$ 
  - $t[P_K] \neq \text{null}$  per ogni tupla  $t$  in  $r(R)$
  - Se  $P_K$  ha più di un attributo, il valore NULL non è ammesso per nessuno degli attributi che la compongono
- Nota bene: è possibile imporre che anche altri attributi non possano assumere il valore NULL, anche se non sono (parte della) chiave primaria.



# Vincoli di integrità referenziale

- A differenza degli altri vincoli, un vincolo di integrità referenziale coinvolge due relazioni:
  - una **relazione referenziante**  $R_1$
  - una **relazione referenziata**  $R_2$
- In  $R_1$  c'è un insieme di attributi FK (chiamati **chiave esterna** / *Foreign Key*) che fanno riferimento agli attributi della chiave primaria PK (*Primary Key*) di  $R_2$
- Una tupla  $t_1$  in  $R_1$  si dice che **referenzia** una tupla  $t_2$  in  $R_2$  se  $t_1[\text{FK}] = t_2[\text{PK}]$

# Vincoli di integrità referenziale: esempio

- Enrolled è la **relazione** referenziante
- Students è la **relazione** referenziata
- Enrolled.sid è la **chiave esterna** che fa riferimento alla **chiave primaria** di Students (Students.sid)
  - La prima, la seconda e la quarta tupla di Enrolled **referenziano** la prima di Students

Enrolled

sid	cid	grade
53666	Carnatic101	C
53666	Reggae203	B
53650	Topology112	A
53666	History105	B

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

# Vincoli di integrità referenziale

- I valori degli attributi della chiave esterna FK della **relazione referenziante**  $R_1$  possono essere:
  1. Uno dei valori del corrispondente attributo della chiave primaria PK in  $R_2$ , oppure
  2. Assumere il valore NULL
- Se NULL, ovviamente la FK in  $R_1$  non deve far parte degli attributi della propria chiave primaria!

# Vincoli di integrità referenziale

- **Osservazione 1:** la FK può fare riferimento alla stessa relazione di appartenenza della PK!
- **Osservazione 2:** gli attributi appartenenti alla FK non necessariamente hanno lo stesso nome dei corrispondenti attributi appartenenti alla PK

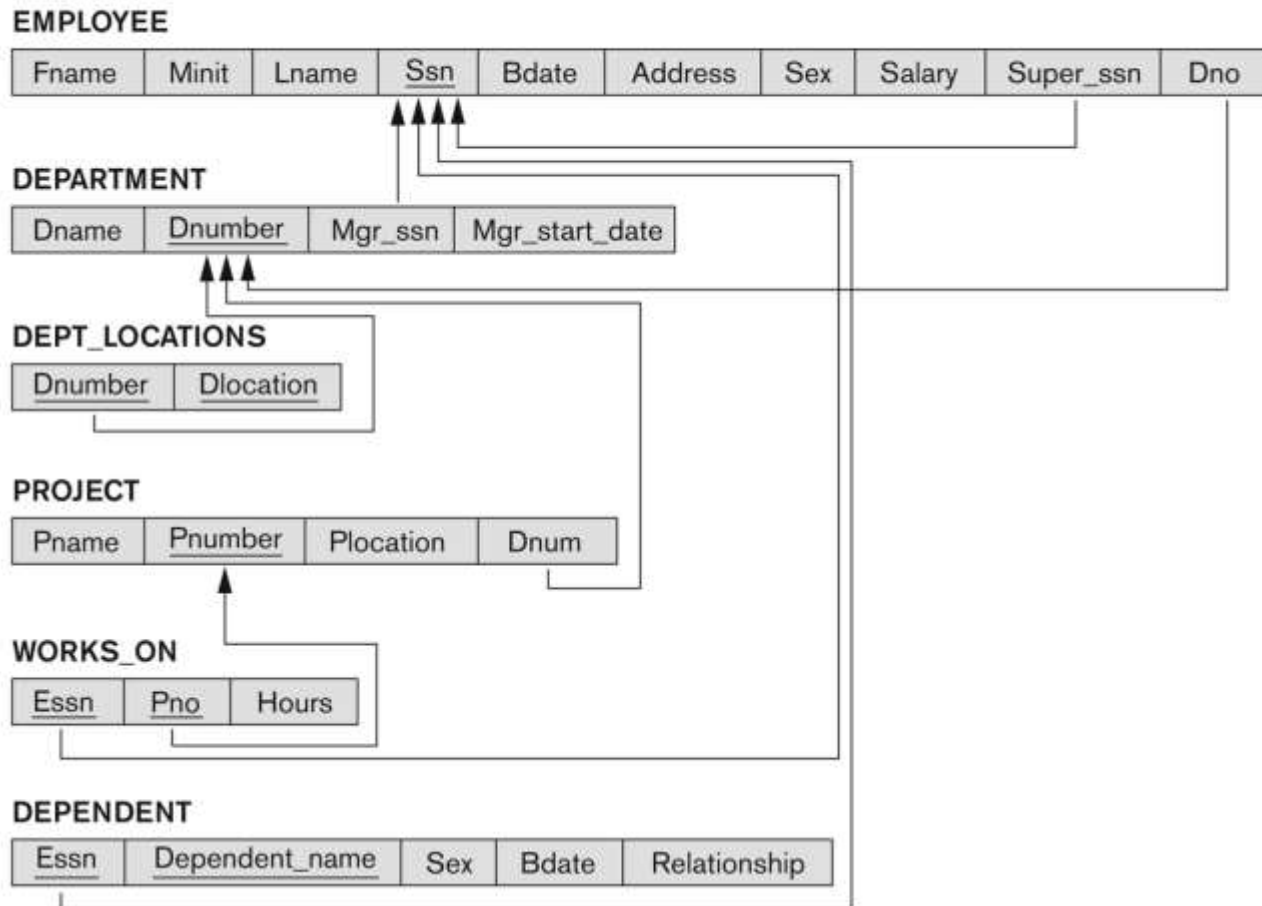
EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

# Referential Integrity Constraints for COMPANY database

**Figure 5.7**

Referential integrity constraints displayed on the COMPANY relational database schema.



# Schema di un database relazionale

- A differenza degli altri vincoli, i vincoli di integrità referenziale coinvolgono più di una relazione, per cui non basta considerare lo schema di una singola relazione
- Uno **schema di un database relazionale** è un insieme  $S$  di schemi di relazione che appartengono alla stessa base di dati
  - $S = \langle \{R_1, R_2, \dots, R_n\}, IC \rangle$  dove:
    - $R_1, R_2, \dots, R_n$  sono i nomi dei singoli schemi di relazione che appartengono alla base di dati  $S$
    - $IC$  è un insieme di vincoli di integrità

# Altri tipi di vincolo

- **Vincoli di Integrità Semantica:**
  - Sono riferiti al significato dell'applicazione e non possono essere espressi nel modello in sé
  - Esempio: “Il numero massimo di ore che un dipendente può lavorare su tutti i progetti è di 46 ore per settimana”
- **Diventa necessario un linguaggio per la **specifica di vincoli****
  - SQL-99 introduce i concetti di «trigger» e «asserzioni» per definire condizioni, ma non tutti i DBMS li implementano e spesso in modi diversi. In MYSQL si può usare il comando CHECK per esprimere vincoli sui valori degli attributi
  - Esempio:

```
CREATE TABLE Person (...,  
Age int CHECK (Age>=18) );
```

# Stato di una base di dati relazionale

- Ogni *relazione* è tipicamente popolata da un certo numero di tuple (*stato della relazione*)
- Uno **stato di una base di dati relazionale** con schema  $S$  è un insieme di stati delle relazioni  $\{r_1, r_2, \dots, r_m\}$  tali che ogni  $r_i$  è uno stato di  $R_i$  e tale che  $r_i$  soddisfa i vincoli di integrità relazionale in IC.
  - Uno **stato di una base di dati relazionale** viene talvolta chiamato un'*istantanea* (*snapshot*) o *istanza* (*instance*) della base di dati
- Uno stato di base di dati che non rispetta i vincoli in IC è uno **stato non valido**



# Stato di una base di dati popolata

- Lo *stato della base di dati* è l'unione di tutti i singoli stati delle relazioni che compongono la base di dati
- Ogni volta che la base di dati è modificata, si passa in un nuovo stato della base di dati
- Le principali operazioni per cambiare lo stato di una base di dati sono:
  - INSERT: inserimento di una nuova tupla in una relazione
  - DELETE: cancellazione di una tupla da una relazione
  - MODIFY: modifica di un attributo di una tupla
- **Importante: queste operazioni non devono portare alla violazione di alcun vincolo di integrità!!**
- Garantire questa condizione può richiedere di **propagare automaticamente** gli aggiornamenti

# Operazioni di aggiornamento su relazioni

- Se si verifica una violazione di vincoli di integrità, ci sono diverse possibili opzioni:
  - Opzione RESTRICT (NO ACTION, REJECT): impedire l'esecuzione dell'operazione che causa la violazione
  - Opzioni CASCADE, SET NULL, SET DEFAULT: avviare altri aggiornamenti per eliminare la violazione
  - Eseguire *routine* specifiche definite dall'utente
  - [Eseguire l'operazione informando l'utente della violazione dell'integrità]

# Violazioni per operazione INSERT

- L'operazione INSERT può violare tutti i vincoli:
  - **Vincoli di dominio:**
    - Il valore di uno o più attributi della/e nuova/e tupla/e non appartiene al dominio specificato nel modello
  - **Vincolo di chiave:**
    - Inserimento di tupla/e in cui il valore della chiave già esiste
  - **Integrità referenziale:**
    - Valore della chiave esterna (*foreign key*) che fa riferimento a valori della chiave primaria della relazione referenziata che non esistono
  - **Integrità dell'entità:**
    - Il valore della chiave primaria della/e nuova/e tupla/e è NULL

# Esempi di violazione dei vincoli

Enrolled

sid	cid	grade
53666	Carnatic101	C
53666	Reggae203	B
53650	Topology112	A
53666	History105	B

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

- Inserimento tupla in STUDENTS <53666, Red, red@math,19, 3.1>
  - Vincolo di chiave
- Inserimento tupla in STUDENTS <53666, Red, red@math,19, A>
  - Vincolo di dominio
- Inserimento in ENROLLED di nuova tupla <53501, Reggae03, 3.1>
  - Integrità referenziale
- Inserimento tupla in STUDENTS <NULL, Dantoni, dan@math, 18, 3.6>
  - Integrità entità

# Violazioni per operazione DELETE

- L'operazione DELETE può violare vincoli di integrità referenziale rimuovendo una o più tuple la cui chiave primaria è referenziata da altre tuple della base di dati

Enrolled

sid	cid	grade
53666	Carnatic101	C
53666	Reggae203	B
53650	Topology112	A
53666	History105	B

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

- Cancellazione di <53666, Jones, jones@cs, 18, 3,4> in STUDENTS
  - Vincolo di integrità referenziale
- Cancellazione di <53666, Reggae03, A> in ENROLLED
  - No violazione

# Violazioni per operazione UPDATE

- L'operazione di aggiornamento (UPDATE) può violare tutti i vincoli:
  - UPDATE della chiave primaria (PK):
    - Integrità referenziale
  - UPDATE di una chiave esterna (FK):
    - Integrità referenziale
  - UPDATE di un attributo ordinario (non PK né FK):
    - Vincoli di dominio o vincoli UNIQUE e NOT NULL

# Esempi di violazione dei vincoli

Enrolled

sid	cid	grade
53666	Carnatic101	C
53666	Reggae203	B
53650	Topology112	A
53666	History105	B

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

- Aggiornamento in STUDENTS: da  $\langle 53666, \text{Red}, \text{red@math}, 19, 3.1 \rangle$  a  $\langle 53777, \text{Red}, \text{red@math}, 19, 3.1 \rangle$ 
  - Vincolo di integrità referenziale
- Aggiornamento in ENROLLED: da  $\langle 53666, \text{Reggae03}, A \rangle$  a  $\langle 535011, \text{Reggae03}, A \rangle$ 
  - Vincolo di integrità referenziale
- Aggiornamento in STUDENTS: da  $\langle 53666, \text{Red}, \text{red@math}, 19, 3.1 \rangle$  a  $\langle 53777, \text{Red}, \text{red@math}, A, 3.1 \rangle$ 
  - Vincolo di dominio

# Come preservare l'integrità referenziale

- **RESTRICT (NO ACTION):** rifiutare l'operazione
- **CASCADE:** cancellare tutte le tuple che referenziavano la chiave primaria della tupla cancellata o modificata
- **SET NULL:** assegnare il valore NULL alla chiave esterna delle tuple che referenziavano la chiave primaria della tupla cancellata o modificata
- **SET DEFAULT:** assegnare un valore di default alle chiavi esterne che referenziavano la chiave primaria della tupla cancellata o modificata



# Come preservare l'integrità referenziale

NB: le opzioni SET NULL e SET DEFAULT **non sono disponibili** se la chiave esterna include attributi che sono anche chiave primaria della tabella referenziante

Enrolled

sid	cid	grade
53666	Carnatic101	C
53666	Reggae203	B
53650	Topology112	A
53666	History105	B

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

La chiave esterna di ENROLLED contiene gli attributi *sid* e *cid* che sono anche chiave primaria. Quindi se cancello la tupla con *sid* 53666 da STUDENTS e fisso un default (o NULL) per la chiave esterna di ENROLLED, introduco una violazione del vincolo di integrità della relazione

# Specifiche delle opzioni in SQL (esempio)

```
CREATE TABLE Enrolled ( studid CHAR(20),  
                          cid    CHAR(20),  
                          grade CHAR(10),  
                          PRIMARY KEY (studid, cid),  
                          FOREIGN KEY (studid) REFERENCES Students  
                              ON DELETE CASCADE  
                              ON UPDATE NO ACTION )
```

- Propagazione della cancellazione quando si viola un vincolo di integrità referenziale
- Impedire un UPDATE se viola un qualsiasi vincolo