

Elementi di Informatica LB

Basi di Dati

Anno accademico 2007/2008

Prof. Stefano Contadini

Elementi di Informatica LB

Il Modello Relazionale

Introduzione:

1. Basi di dati relazionali

Il modello Relazionale

- ❑ Proposto da E. F.Codd nel 1970 per favorire l'indipendenza dei dati
- ❑ Disponibile come modello logico in DBMS reali nel 1981 (non è facile realizzare l'indipendenza con efficienza e affidabilità!)
- ❑ Si basa sul concetto matematico di **relazione**(con una variante)
- ❑ Le relazioni hanno una rappresentazione naturale per mezzo di tabelle
- ❑ Il modello è "**basato su valori**": anche i riferimenti fra dati in strutture (relazioni) diverse sono rappresentati per mezzo dei valori stessi

Basi di dati relazionali

Relazione: tre accezioni

- **relazione matematica**: come nella teoria degli insiemi
- relazione (dall'inglese **relationship**) che rappresenta una classe di fatti — una relazione matematica fra due entità, nel modello **Entity-Relationship**; talvolta tradotto con associazione o correlazione
- **relazione** secondo il modello relazionale dei dati: tabella

Basi di dati relazionali

Relazione matematica

- D_1, D_2, \dots, D_n (n insiemi anche non distinti)
- il **prodotto cartesiano** $D_1 \times D_2 \times \dots \times D_n$, è l'insieme di tutte le n -uple ordinate (d_1, d_2, \dots, d_n) tali che $d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$
- una relazione matematica su D_1, D_2, \dots, D_n è un **sottoinsieme del prodotto cartesiano** $D_1 \times D_2 \times \dots \times D_n$
- D_1, D_2, \dots, D_n sono i **domini** della relazione. Una relazione su n domini ha **grado** (o **arietà**) n
- il numero di n -uple è la **cardinalità** della relazione

Basi di dati relazionali

Relazione matematica, esempio

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

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

- una relazione $r \subseteq D_1 \times D_2$

a	x
a	z
b	y
b	z

Basi di dati relazionali

Relazione matematica, proprietà

Una relazione matematica è un insieme di n-uple ordinate:
(d_1, \dots, d_n) tali che $d_1 \in D_1, \dots, d_n \in D_n$

- la relazione è un **insieme**; quindi:
 - non c'è ordinamento fra le n-uple
 - le n-uple sono distinte
- ciascuna n-upla è **ordinata**; quindi:
 - l' i-esimo valore proviene dall' i-esimo dominio

Basi di dati relazionali

Relazione matematica, esempio

Partite \subseteq string \times string \times integer \times integer

Juve	Lazio	3	1
Lazio	Milan	2	0
Juve	Roma	1	2
Roma	Milan	0	1

- Ciascuno dei domini ha due **ruoli** distinti, distinguibili attraverso la posizione: il primo e il terzo dominio si riferiscono a nome e reti della squadra ospitante; il secondo e il quarto a nome e reti della squadra ospitata
- La struttura è **posizionale**

Basi di dati relazionali

Relazioni nel modello relazionale dei dati

- Ogni relazione è sostanzialmente una **tabella**
- A ciascun dominio associamo un nome (**attributo**), unico nella relazione, che “descrive” il ruolo del dominio
- Nella rappresentazione tabellare, gli attributi sono usati come intestazioni delle colonne

Casa	Fuori	RetiCasa	RetiFuori
Juve	Lazio	3	1
Lazio	Milan	2	0
Juve	Roma	1	2
Roma	Milan	0	1

- L'ordinamento fra gli attributi è irrilevante: la struttura è **non posizionale**

Basi di dati relazionali

Notazioni

- Se t è una ennupla su X e $A \in X$, allora $t[A]$ (oppure $t.A$) indica il **valore di t su A**
- Nell'esempio, se t è la prima ennupla della tabella
 $t[\text{Fuori}] = \text{Lazio}$
- La stessa notazione è estesa anche ad insiemi di attributi, nel qual caso denota ennuple:
 $t[\text{Fuori}, \text{RetiFuori}]$ è una ennupla su due attributi.
- Nell'esempio, se t è la prima ennupla della tabella
 $t[\text{Fuori}, \text{RetiFuori}] = \langle \text{Lazio}, 1 \rangle$

Basi di dati relazionali

Tabelle e relazioni

- Una tabella rappresenta una relazione se
 - i valori di ciascuna colonna sono fra loro omogenei (appartengono allo stesso dominio)
 - le righe sono diverse fra loro
 - le intestazioni delle colonne (attributi) sono diverse tra loro
- Inoltre, in una tabella che rappresenta una relazione
 - l'ordinamento tra le righe è irrilevante
 - l'ordinamento tra le colonne è irrilevante
- **Il modello relazionale è basato su valori:** i riferimenti fra dati in relazioni diverse sono rappresentati per mezzo di valori dei domini che compaiono nelle ennuple.

Basi di dati relazionali

studenti

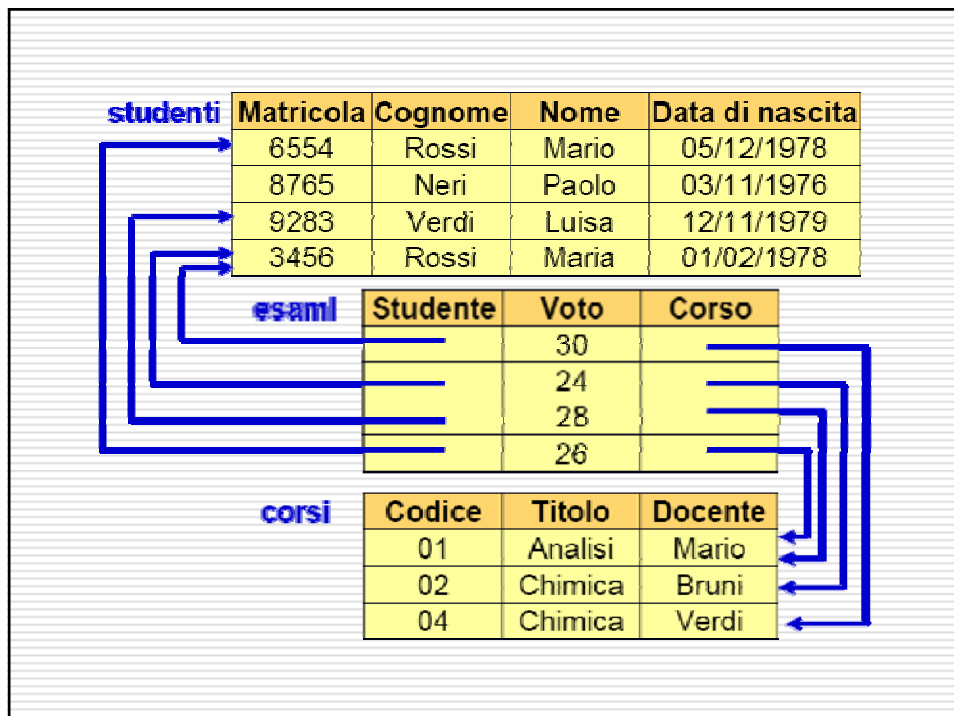
Matricola	Cognome	Nome	Data di nascita
6554	Rossi	Mario	05/12/1978
8765	Neri	Paolo	03/11/1976
9283	Verdi	Luisa	12/11/1979
3456	Rossi	Maria	01/02/1978

esami

Studente	Voto	Corso
3456	30	04
3456	24	02
9283	28	01
6554	26	01

corsi

Codice	Titolo	Docente
01	Analisi	Mario
02	Chimica	Bruni
04	Chimica	Verdi



Vantaggi della struttura basata su valori

- **indipendenza dalle strutture fisiche**, che possono cambiare anche dinamicamente
- si rappresenta solo ciò che è **rilevante dal punto di vista dell'applicazione** (dell'utente); i puntatori sono meno comprensibili per l'utente finale (senza, l'utente finale vede gli stessi dati dei programmatori)
- i dati sono **portabili** più facilmente da un sistema ad un altro
- i valori consentono **bi-direzionalità**, i puntatori sono direzionali

Nota: i puntatori possono essere usati a livello fisico

Definizioni

Schema di relazione:

un **nome di relazione** R con un insieme di **attributi** A_1, \dots, A_n

$$R(A_1, \dots, A_n)$$

Schema di base di dati:

insieme di schemi di relazione con nomi diversi:

$$R = \{R_1(X_1), \dots, R_n(X_n)\}$$

(Istanza di) relazione su uno schema $R(X)$:

insieme r di ennuple su X

(Istanza di) base di dati su uno schema $R = \{R_1(X_1), \dots, R_n(X_n)\}$:

insieme di relazioni $r = \{r_1, \dots, r_n\}$ (con r_i relazione su R_i)

Esempio

studenti

Matricola	Cognome	Nome	Data di nascita
6554	Rossi	Mario	05/12/1978
8765	Neri	Paolo	03/11/1976
9283	Verdi	Luisa	12/11/1979
3456	Rossi	Maria	01/02/1978

studenti lavoratori

Matricola
6554
3456

Informazione incompleta

- Il modello relazionale impone ai dati una struttura rigida:
 - le informazioni sono rappresentate per mezzo di enuple
 - le enuple ammesse sono dettate dagli schemi di relazione
- I dati disponibili possono non corrispondere esattamente al formato previsto, per varie ragioni.

Esempio:

- di Firenze non conosciamo l'indirizzo della prefettura
- Tivoli non è provincia: non ha prefettura
- Prato è "nuova" provincia: ha la prefettura?

Città	Prefettura
Roma	Via IV Novembre
Firenze	
Tivoli	
Prato	

Basi di dati relazionali

Informazione incompleta: soluzioni?

Non conviene (anche se spesso si fa) utilizzare valori ordinari del dominio (0, stringa nulla, "99", etc), per vari motivi:

- potrebbero non esistere valori "non utilizzati"
- valori "non utilizzati" potrebbero diventare significativi
- in fase di utilizzo (ad esempio, nei programmi) sarebbe necessario ogni volta tener conto del "significato" di questi valori

Basi di dati relazionali

Informazione incompleta nel modello relazionale

- Si adotta una tecnica rudimentale ma efficace:
 - **valore nullo**: denota l'assenza di un valore del dominio (e non è un valore del dominio)
- Formalmente, è sufficiente estendere il concetto di ennupla:
t [A], per ogni attributo A, è un valore del dominio $\text{dom}(A)$
oppure il valore nullo **NULL**
- Si possono (e debbono) imporre restrizioni sulla presenza di valori nulli

Basi di dati relazionali

Tipi di interpretazione del valore nullo

- tre casi differenti
 - valore **sconosciuto**: esiste un valore del dominio, ma non è noto (Firenze)
 - valore **inesistente**: non esiste un valore del dominio (Tivoli)
 - valore **senza informazione**: non è noto se esista o meno un valore del dominio (Prato)
- I DBMS non distinguono i tipi di valore nullo (e quindi implicitamente adottano l'interpretazione "**senza informazione**")

Basi di dati relazionali

2. Modello relazionale - 20

Vincoli di integrità

Esistono istanze di basi di dati che, pur sintatticamente corrette, non rappresentano informazioni possibili per l'applicazione di interesse.

Studenti	Matricola	Cognome	Nome	Nascita
	276545	Rossi	Maria	23/04/1968
	276545	Neri	Anna	23/04/1972
	788854	Verdi	Fabio	12/02/1972

Esami	Studente	Voto	Lode	Corso
	276545	28	e lode	01
	276545	32		02
	788854	23		03
	200768	30	e lode	03

Corsi	Codice	Titolo	Docente
	01	Analisi	Giani
	03	NULL	NULL
	02	Chimica	Belli

Basi di dati relazionali

Vincolo di integrità

Definizione

- proprietà che deve essere soddisfatta dalle istanze che rappresentano informazioni corrette per l'applicazione
- ogni vincolo può essere visto come una funzione booleana (o un predicato) che associa ad ogni istanza il valore **vero** o **falso**
- Ad uno schema associamo un insieme di vincoli e consideriamo **corrette** (lecite, valide, ammissibili) solo le istanze che soddisfano tutti i vincoli
- Tipi di vincoli:
 - intrarelazionali
 - interrelazionali

Basi di dati relazionali

Vincoli di integrità: motivazioni

- risultano utili al fine di descrivere la realtà di interesse in modo più accurato di quanto le strutture permettano;
- forniscono un contributo verso la "qualità dei dati"
- costituiscono uno strumento di ausilio alla progettazione
- sono utilizzati dal sistema nella scelta della strategia di esecuzione delle interrogazioni

Nota:

- non tutte le proprietà di interesse sono rappresentabili per mezzo di vincoli esprimibili direttamente

Basi di dati relazionali

Vincoli intrarelazionali: vincoli di ennupla

- Esprimono **condizioni sui valori di ciascuna ennupla**, indipendentemente dalle altre ennuple.
- Una possibile sintassi: espressione booleana (con AND, OR e NOT) di atomi che confrontano valori di attributo o espressioni aritmetiche su di essi.
- Un vincolo di ennupla è un **vincolo di dominio** se coinvolge un solo attributo

Esempi:

$(Voto \geq 18) \text{ AND } (Voto \leq 30)$

$(Voto = 30) \text{ OR NOT } (Lode = \text{"e lode"})$

$Lordo = (Ritenute + Netto)$

Basi di dati relazionali

Vincoli intrarelazionali: vincoli di chiave

Matricola	Cognome	Nome	Corso	Nascita
27655	Rossi	Mario	Ing Inf	5/12/78
78763	Rossi	Mario	Ing Inf	3/11/76
65432	Neri	Piero	Ing Mecc	10/7/79
87654	Neri	Mario	Ing Inf	3/11/76
67653	Rossi	Piero	Ing Mecc	5/12/78

- il numero di matricola identifica gli studenti:
non ci sono due ennuple con lo stesso valore sull'attributo Matricola
- i dati anagrafici identificano gli studenti:
non ci sono due ennuple uguali su tutti e tre gli attributi Cognome, Nome e Nascita

Basi di dati relazionali

Vincoli di chiave

Chiave:

insieme di attributi che identificano univocamente le ennuple di una relazione

Più precisamente:

- un insieme K di attributi è **superchiave** per una relazione r se r non contiene due ennuple distinte t_1 e t_2 tali che $t_1[K] = t_2[K]$
- K è **chiave** per r se è una superchiave minimale (cioè non contiene un'altra superchiave) per r

Basi di dati relazionali

Vincoli di chiave: esempi

Matricola	Cognome	Nome	Corso	Nascita
27655	Rossi	Mario	Ing Inf	5/12/78
78763	Rossi	Mario	Ing Inf	3/11/76
65432	Neri	Piero	Ing Mecc	10/7/79
87654	Neri	Mario	Ing Inf	3/11/76
87653	Rossi	Piero	Ing Mecc	5/12/78

- **Matricola** è una chiave:
 - Matricola è superchiave
 - contiene un solo attributo e quindi è minimale
- **Cognome, Nome, Nascita** è un'altra chiave:
 - l'insieme Cognome, Nome, Nascita è superchiave
 - nessuno dei suoi sottoinsiemi è superchiave
- **Cognome, Nome, Nascita, Corso** è superchiave (non chiave)

Basi di dati relazionali

Un'altra chiave?

Matricola	Cognome	Nome	Corso	Nascita
27655	Rossi	Mario	Ing Inf	5/12/78
78763	Rossi	Mario	Ing Civile	3/11/76
65432	Neri	Piero	Ing Mecc	10/7/79
87654	Neri	Mario	Ing Inf	3/11/76
67653	Rossi	Piero	Ing Mecc	5/12/78

- non ci sono ennuple uguali su Cognome e Corso:
Cognome e Corso formano una chiave
- ma è sempre vero?

Basi di dati relazionali

Vincoli, schemi e istanze

- i vincoli corrispondono a proprietà del mondo reale modellato dalla base di dati
- interessano a livello di schema (con riferimento cioè a tutte le istanze): ad uno schema associamo un insieme di vincoli che vogliamo siano soddisfatti da **tutte** le sue istanze corrette
- consideriamo quindi **corrette** (valide, ammissibili) le istanze che soddisfano tutti i vincoli

N.B.: un'istanza può soddisfare altri vincoli ("per caso")

Basi di dati relazionali

Individuazione delle chiavi

Individuiamo le chiavi

- considerando le proprietà che i dati soddisfano nell'applicazione (il "frammento di mondo reale di interesse")
- notando quali insiemi di attributi permettono di identificare univocamente le ennuple;
- e individuando i sottoinsiemi minimali di tali insiemi che conservano la capacità di identificare le ennuple.

Esempio:

Studenti (Matricola, Cognome, Nome, Corso, Nascita) ha 2 chiavi:

Matricola
Cognome, Nome, Nascita

Basi di dati relazionali

Esistenza delle chiavi

- poiché le relazioni sono insiemi, una relazione non può contenere ennuple uguali fra loro:
 - ogni relazione ha come superchiave l'insieme degli attributi su cui è definita
- poiché l'insieme di tutti gli attributi è una superchiave per ogni relazione, ogni schema di relazione ha almeno una superchiave
- ne segue che **ogni schema di relazione ha (almeno) una chiave**

Basi di dati relazionali

Importanza delle chiavi

- l'esistenza delle chiavi garantisce l'accessibilità a ciascun dato della base di dati
- ogni singolo valore è univocamente accessibile tramite:
 - nome della relazione
 - valore della chiave
 - nome dell'attributo
- le chiavi sono lo strumento principale attraverso il quale vengono correlati i dati in relazioni diverse ("il modello relazionale è basato su valori")

Basi di dati relazionali

Chiavi e valori nulli

- In presenza di valori nulli, i valori degli attributi che formano la chiave:
 - non permettono di identificare le entuple come desiderato
 - né permettono di realizzare facilmente i riferimenti da altre relazioni

Matricola	Cognome	Nome	Corso	Nascita
NULL	NULL	Mario	Ing Inf	5/12/78
78763	Rossi	Mario	Ing Civile	3/11/76
65432	Neri	Piero	Ing Mecc	10/7/79
87654	Neri	Mario	Ing Inf	NULL
NULL	Neri	Mario	NULL	5/12/78

Basi di dati relazionali

Chiave primaria

- La presenza di valori nulli nelle chiavi deve essere limitata.
- Soluzione pratica: per ogni relazione scegliamo una chiave (la **chiave primaria**) su cui non ammettiamo valori nulli.
- Notazione per la chiave primaria: gli attributi che la compongono sono sottolineati

<u>Matricola</u>	Cognome	Nome	Corso	Nascita
27655	Rossi	Mario	Ing Inf	5/12/78
78763	Rossi	Mario	Ing Civile	3/11/76
65432	Neri	Piero	Ing Mecc	10/7/79
87654	Neri	Mario	Ing Inf	NULL
67653	Rossi	Piero	NULL	5/12/78

Basi di dati relazionali

Vincoli interrelazionali: integrità referenziale

- Informazioni in relazioni diverse sono correlate attraverso valori comuni, in particolare, attraverso valori delle chiavi (primarie, di solito).
- Un **vincolo di integrità referenziale** (detto anche vincolo di “**foreign key**”) fra un insieme di attributi X di una relazione R_1 e un'altra relazione R_2 impone ai valori su X di ciascuna ennupla dell'istanza di R_1 di comparire come valori della chiave (primaria) dell'istanza di R_2 .
- Giocano un ruolo fondamentale nel concetto di “modello basato su valori”.

Basi di dati relazionali

Vincoli di integrità referenziale: esempio

Infrazioni

<u>Codice</u>	Data	Vigile	Prov	Numero
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

Vigili

<u>Matricola</u>	Cognome	Nome
3987	Rossi	Luca
3295	Neri	Piero
9345	Neri	Mario
7543	Mori	Gino

Basi di dati relazionali

Esempio (cont.)

Infrazioni

<u>Codice</u>	Data	Vigile	Prov	Numero
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

Auto

<u>Prov</u>	<u>Numero</u>	Cognome	Nome
MI	39548K	Rossi	Mario
TO	E39548	Rossi	Mario
PR	839548	Neri	Luca

Basi di dati relazionali

Vincoli di integrità referenziale: esempio

- Nell'esempio, i vincoli di integrità referenziale sussistono fra:
 - l'attributo **Vigile** della relazione **Infrazioni** e la relazione **Vigili**
 - gli attributi **Prov** e **Numero** di **Infrazioni** e la relazione **Auto**

Basi di dati relazionali

Violazione di vincolo di integrità referenziale

Infrazioni

<u>Codice</u>	Data	Vigile	Prov	Numero
34321	1/2/95	3987	MI	39548K
53524	4/3/95	3295	TO	E39548
64521	5/4/96	2468	PR	839548
73321	5/2/98	9345	PR	839548

Vigili

<u>Matricola</u>	Cognome	Nome
3987	Rossi	Luca
3295	Neri	Piero
9345	Neri	Mario
7543	Mori	Gino

Basi di dati relazionali

Violazione di vincolo di integrità ref. (cont.)

Infrazioni

<u>Codice</u>	Data	Vigile	Prov	Numero
34321	1/2/95	3987	MI	39548K
53524	4/3/95	3295	TO	39548K
64521	5/4/96	3295	PR	839548
73321	5/2/98	9345	PR	839548

Auto

<u>Prov</u>	<u>Numero</u>	Cognome	Nome
MI	39548K	Rossi	Mario
TO	E39548	Rossi	Mario
PR	839548	Neri	Luca

Basi di dati relazionali

Integrità referenziale e valori nulli

In presenza di valori nulli i vincoli possono essere resi meno restrittivi.

Impiegati	<u>Matricola</u>	Cognome	Progetto
	34321	Rossi	IDEA
	53524	Neri	XYZ
	64521	Verdi	NULL
	73032	Bianchi	IDEA

Progetti	<u>Codice</u>	Inizio	Durata	Costo
	IDEA	01/2000	36	200
	XYZ	07/2001	24	120
	BOH	09/2001	24	150

Basi di dati relazionali

Integrità referenziale: azioni compensative

Sono possibili meccanismi per il supporto alla gestione dei vincoli di integrità ("azioni" compensative a seguito di violazioni)

Ad esempio, se viene eliminata una ennupla causando una violazione:

- comportamento "standard": rifiuto dell'operazione
- azioni compensative:
 - eliminazione in cascata
 - introduzione di valori nulli

Basi di dati relazionali

Eliminazione in cascata

Impiegati

<u>Matricola</u>	Cognome	Progetto
34321	Rossi	IDEA
53524	Neri	XYZ
64521	Verdi	NULL
73032	Bianchi	IDEA

Progetti

<u>Codice</u>	Inizio	Durata	Costo
IDEA	01/2000	36	200
XYZ	07/2001	24	120
BOH	09/2001	24	150

Basi di dati relazionali

Introduzione di valori nulli

Impiegati

<u>Matricola</u>	Cognome	Progetto
34321	Rossi	IDEA
53524	Neri	NULL
64521	Verdi	NULL
73032	Bianchi	IDEA

Progetti

<u>Codice</u>	Inizio	Durata	Costo
IDEA	01/2000	36	200
XYZ	07/2001	24	120
BOH	09/2001	24	150

Basi di dati relazionali

Vincoli multipli su più attributi

Incidenti

<u>Codice</u>	Data	ProvA	NumeroA	ProvB	NumeroB
34321	1/2/95	TO	E39548	MI	39548K
64521	5/4/96	PR	839548	TO	E39548

Auto

<u>Prov</u>	<u>Numero</u>	Cognome	Nome
MI	39548K	Rossi	Mario
TO	E39548	Rossi	Mario
PR	839548	Neri	Luca

Basi di dati relazionali

Elementi di Informatica LB

Il Modello Relazionale

Introduzione:

1. Algebra Relazionale

Linguaggi per basi di dati

- operazioni sullo schema:
DDL: data definition language
- operazioni sui dati:
DML: data manipulation language
 - interrogazione ("query")
 - aggiornamento

Basi di dati relazionali

Linguaggi di interrogazione per basi di dati relazionali

Tipologia:

- **Dichiarativi**: specificano le proprietà del risultato ("che cosa")
- **Procedurali**: specificano le modalità di generazione del risultato ("come")

Rappresentanti più significativi:

- **Algebra relazionale**: procedurale
- **Calcolo relazionale**: dichiarativo (teorico)
- **SQL** (Structured Query Language): parzialmente dichiarativo (reale)
- **QBE** (Query by Example): dichiarativo (reale)

Basi di dati relazionali

Algebra relazionale

Costituita da un insieme di operatori

- definiti su relazioni
- che producono relazioni
- e possono essere composti

Operatori dell'algebra relazionale:

- unione, intersezione, differenza
- ridenominazione
- selezione
- proiezione
- join (join naturale, prodotto cartesiano, theta-join)

Basi di dati relazionali

Operatori insiemistici

- le relazioni sono insiemi
- i risultati debbono essere relazioni
- è possibile applicare **unione**, **intersezione**, **differenza** solo a relazioni definite sugli stessi attributi

Basi di dati relazionali

Unione

Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Quadri

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Laureati \cup Quadri

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45
9297	Neri	33

Basi di dati relazionali

Intersezione

Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Quadri

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Laureati \cap Quadri

Matricola	Nome	Età
7432	Neri	54
9824	Verdi	45

Basi di dati relazionali

Differenza

Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Quadri

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Laureati - Quadri

Matricola	Nome	Età
7274	Rossi	42

Basi di dati relazionali

Un'unione sensata ma impossibile

Paternità

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

Maternità

Madre	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

Paternità \cup Maternità

??

Basi di dati relazionali

Ridenominazione

- operatore monadico (con un argomento)
- "modifica lo schema" lasciando inalterata l'istanza dell'operando

Paternità

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

REN Genitore ← Padre (Paternità)

Genitore	
Adamo	Abele
Adamo	Caino
Abramo	Isacco

Basi di dati relazionali

Ridenominazione

Paternità

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

RENGenitore ← Padre (Paternità)

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

Maternità

Madre	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

RENGenitore ← Madre (Maternità)

Genitore	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

Basi di dati relazionali

Ridenominazione

REN Genitore ← Padre (Paternità)

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

REN Genitore ← Padre (Paternità)

U

REN Genitore ← Madre (Maternità)

REN Genitore ← Madre (Maternità)

Genitore	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco
Eva	Abele
Eva	Set
Sara	Isacco

Basi di dati relazionali

Impiegati

Cognome	Ufficio	Stipendio
Rossi	Roma	55
Neri	Milano	64

Operai

Cognome	Fabbrica	Salario
Bruni	Monza	45
Verdi	Latina	55

REN Sede, Retribuzione ← Ufficio, Stipendio (Impiegati)

U

REN Sede, Retribuzione ← Fabbrica, Salario (Operai)

Cognome	Sede	Retribuzione
Rossi	Roma	55
Neri	Milano	64
Bruni	Monza	45
Verdi	Latina	55

Selezione

- operatore monadico
- produce un risultato che
 - ha lo stesso schema dell'operando
 - contiene un sottoinsieme delle ennuple dell'operando,
 - quelle che soddisfano una condizione

Basi di dati relazionali

Selezione

Impiegati

Matricola	Cognome	Filiale	Stipendio
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Neri	Napoli	64

- impiegati che
 - guadagnano più di 50
 - guadagnano più di 50 e lavorano a Milano
 - hanno lo stesso nome della filiale presso cui lavorano

Basi di dati relazionali

Selezione, sintassi e semantica

Sintassi:

`SEL Condizione (Operando)`

Condizione: espressione booleana (come quelle dei vincoli di enupla)

Semantica:

il risultato contiene le enuple dell'operando che soddisfano la condizione

Basi di dati relazionali

Selezione

- impiegati che guadagnano più di 50

Impiegati

Matricola	Cognome	Filiale	Stipendio
7309	Rossi	Roma	55
5998	Neri	Milano	64
5698	Neri	Napoli	64

`SEL Stipendio > 50 (Impiegati)`

Basi di dati relazionali

Selezione

- impiegati che guadagnano più di 50 e lavorano a Milano

Impiegati

Matricola	Cognome	Filiale	Stipendio
5998	Neri	Milano	64

SEL `Stipendio > 50 AND Filiale = 'Milano' (Impiegati)`

Basi di dati relazionali

Selezione

- impiegati che hanno lo stesso nome della filiale presso cui lavorano

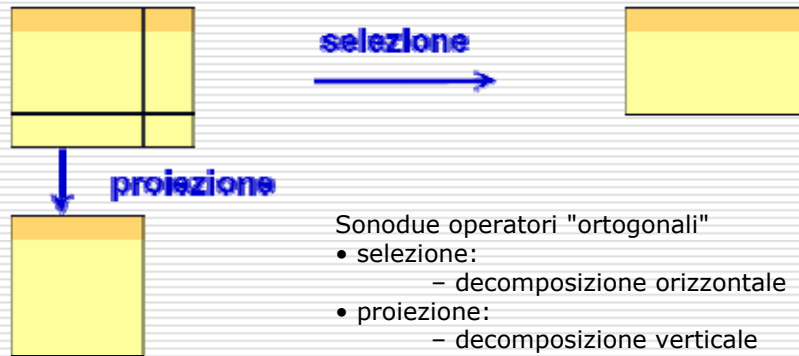
Impiegati

Matricola	Cognome	Filiale	Stipendio
9553	Milano	Milano	44

SEL `Cognome = Filiale (Impiegati)`

Basi di dati relazionali

Selezione e proiezione



Sono due operatori "ortogonali"

- selezione:
 - decomposizione orizzontale
- proiezione:
 - decomposizione verticale

Basi di dati relazionali

Proiezione

- operatore monadico
- produce un risultato che
 - ha parte degli attributi dell'operando
 - contiene ennuple cui contribuiscono tutte le ennuple dell'operando

Basi di dati relazionali

Proiezione

Impiegati

Matricola	Cognome	Filiale	Stipendio
7309	Neri	Napoli	55
5998	Neri	Milano	64
9553	Rossi	Roma	44
5698	Rossi	Roma	64

- per tutti gli impiegati:
- matricola e cognome
 - cognome e filiale

Basi di dati relazionali

Proiezione, sintassi e semantica

Sintassi

`PROJ ListaAttributi (Operando)`

Semantica

- il risultato contiene le ennuple ottenute da tutte le ennuple dell'operando ristrette agli attributi nella lista

Basi di dati relazionali

Proiezione

- matricola e cognome di tutti gli impiegati

Matricola	Cognome
7309	Neri
5998	Neri
9553	Rossi
5698	Rossi

PROJ Matricola, Cognome (Impiegati)

Basi di dati relazionali

Proiezione

- cognome e filiale di tutti gli impiegati

Cognome	Filiale
Neri	Napoli
Neri	Milano
Rossi	Roma

PROJ Cognome, Filiale (Impiegati)

Basi di dati relazionali

Cardinalità delle proiezioni

- una proiezione
 - contiene al più tante ennuple quante l'operando
 - può contenerne di meno
- se X è una superchiave di R , allora $PROJ_X(R)$ contiene esattamente tante ennuple quante R

Basi di dati relazionali

Selezione e proiezione

- Combinando selezione e proiezione, possiamo estrarre interessanti informazioni da una relazione
matricola e cognome degli impiegati che guadagnano più di 50

Matricola	Cognome
7309	Rossi
5998	Neri
5698	Neri

$PROJ_{Matricola,Cognome} (SEL_{Stipendio > 50}(Impiegati))$

Basi di dati relazionali

Join

- combinando selezione e proiezione, possiamo estrarre informazioni da **una** relazione
- non possiamo però correlare informazioni presenti in relazioni diverse
- il **join** è l'operatore più interessante dell'algebra relazionale
- permette di correlare dati in relazioni diverse

Basi di dati relazionali

Prove scritte in un concorso pubblico

- I compiti sono anonimi e ad ognuno è associata una busta chiusa con il nome del candidato
- Ciascun compito e la relativa busta vengono contrassegnati con uno stesso numero

1	25	1	Mario Rossi
2	13	2	Nicola Russo
3	27	3	Mario Bianchi
4	28	4	Remo Neri

Mario Rossi	25
Nicola Russo	13
Mario Bianchi	27
Remo Neri	28

Basi di dati relazionali

Prove scritte in un concorso pubblico

Numero	Voto	Numero	Candidato
1	25	1	Mario Rossi
2	13	2	Nicola Russo
3	27	3	Mario Bianchi
4	28	4	Remo Neri

Numero	Candidato	Voto
1	Mario Rossi	25
2	Nicola Russo	13
3	Mario Bianchi	27
4	Remo Neri	28

Basi di dati relazionali

Join naturale

- operatore binario (generalizzabile)
- produce un risultato
 - sull'unione degli attributi degli operandi
 - con ennuple costruite ciascuna a partire da una ennupla di ognuno degli operandi

Basi di dati relazionali

Join, sintassi e semantica

- $R_1(X_1), R_2(X_2)$
- $R_1 \text{ JOIN } R_2$ è una relazione su X_1X_2

$\{ t \text{ su } X_1X_2 \mid \text{esistono } t_1 \in R_1 \text{ e } t_2 \in R_2$
con $t[X_1] = t_1$ e $t[X_2] = t_2 \}$

Basi di dati relazionali

Un join completo

Impiegato	Reparto	Reparto	Capo
Rossi	A	A	Mori
Neri	B	B	Bruni
Bianchi	B		

Impiegato	Reparto	Capo
Rossi	A	Mori
Neri	B	Bruni
Bianchi	B	Bruni

In un **join completo** ogni ennupla contribuisce al risultato.

Basi di dati relazionali

Un join non completo

Impiegato	Reparto	Reparto	Capo
Rossi	A	B	Mori
Neri	B	C	Bruni
Bianchi	B		

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

Basi di dati relazionali

Un join vuoto

Impiegato	Reparto	Reparto	Capo
Rossi	A	D	Mori
Neri	B	C	Bruni
Bianchi	B		

Impiegato	Reparto	Capo
-----------	---------	------

Basi di dati relazionali

Un join completo, con n x m ennuple

Impiegato	Reparto	Reparto	Capo
Rossi	B	B	Mori
Neri	B	B	Bruni

Impiegato	Reparto	Capo
Rossi	B	Mori
Rossi	B	Bruni
Neri	B	Mori
Bianchi	B	Bruni

Basi di dati relazionali

Cardinalità del join

- $R_1(A,B)$, $R_2(B,C)$
- In generale, il join di R_1 e R_2 contiene un numero di ennuple compreso fra zero e il prodotto di $|R_1|$ e $|R_2|$:
$$0 \leq |R_1 \text{ JOIN } R_2| \leq |R_1| \times |R_2|$$
- se il join coinvolge una chiave di R_2 (ovvero, B è chiave in R_2), allora il numero di ennuple è compreso fra zero e $|R_1|$:
$$0 \leq |R_1 \text{ JOIN } R_2| \leq |R_1|$$
- se il join coinvolge una chiave di R_2 (ovvero, B è chiave in R_2) ed esiste un vincolo di integrità referenziale fra B (in R_1) e R_2 , allora il numero di ennuple è pari a $|R_1|$:
$$|R_1 \text{ JOIN } R_2| = |R_1|$$

Basi di dati relazionali

Join, un'osservazione

Impiegato	Reparto	Reparto	Capo
Rossi	A	B	Mori
Neri	B	C	Bruni
Bianchi	B		

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

Alcune ennuple non contribuiscono al risultato: vengono "tagliate fuori"

Basi di dati relazionali

Join esterno

- Il **join esterno** estende, con valori nulli, le ennuple che verrebbero tagliate fuori da un join (interno)
- esiste in tre versioni:
 - **sinistro**: mantiene tutte le ennuple del primo operando, estendendole con valori nulli, se necessario
 - **destra**: ... del secondo operando ...
 - **completo**: ... di entrambi gli operandi ...

Basi di dati relazionali

Join esterno sinistro

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati JOIN_{LEFT} Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
Rossi	A	NULL

Basi di dati relazionali

Join esterno destro

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati JOIN_{RIGHT} Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
NULL	C	Bruni

Basi di dati relazionali

Join esterno completo

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati $\text{JOIN}_{\text{FULL}}$ Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
Rossi	A	NULL
NULL	C	Bruni

Basi di dati relazionali

Join e proiezioni

Impiegato	Reparto	Reparto	Capo
Rossi	A	B	Mori
Neri	B	C	Bruni
Bianchi	B		

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

Impiegato	Reparto	Reparto	Capo
Neri	B	B	Mori
Bianchi	B		

Basi di dati relazionali

Proiezioni e join

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Bruni
Verdi	A	Bini

Impiegato	Reparto	Reparto	Capo
Neri	B	B	Mori
Bianchi	B	B	Bruni
Verdi	A	A	Bini

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Bruni
Neri	B	Bruni
Bianchi	B	Mori
Verdi	A	Bini

Basi di dati relazionali

Join e proiezioni

- $R_1(X_1), R_2(X_2)$

$$\text{PROJ}_{X_1}(R_1 \text{ JOIN } R_2) \subseteq R_1$$

- $R(X), X = X_1 \cup X_2$

$$(\text{PROJ}_{X_1}(R)) \text{ JOIN } (\text{PROJ}_{X_2}(R)) \supseteq R$$

Basi di dati relazionali

Prodotto cartesiano

- un join naturale su relazioni senza attributi in comune
- contiene sempre un numero di ennuple pari al prodotto delle cardinalità degli operandi (le ennuple sono tutte combinabili)

Basi di dati relazionali

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Codice	Capo
A	Mori
B	Bruni

Impiegati JOIN Reparti

Impiegato	Reparto	Codice	Capo
Rossi	A	A	Mori
Rossi	A	B	Bruni
Neri	B	A	Mori
Neri	B	B	Bruni
Bianchi	B	A	Mori
Bianchi	B	B	Bruni

Theta-join

- Il prodotto cartesiano, in pratica, ha senso (quasi) solo se seguito da selezione:

$SEL_{Condizione} (R_1 JOIN R_2)$

- L'operazione viene chiamata **theta-join** e indicata con

$R_1 JOIN_{Condizione} R_2$

- La condizione di selezione è spesso una congiunzione (**AND**) di atomi di confronto $A_1 \vartheta A_2$ dove ϑ è uno degli operatori di confronto ($=, >, <, \dots$). Da questo deriva il nome "theta-join".
- Se l'operatore è sempre l'uguaglianza ($=$) allora si parla di **equi-join**.

Basi di dati relazionali

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Codice	Capo
A	Mori
B	Bruni

Impiegati JOIN_{Reparto=Codice} Reparti

Impiegato	Reparto	Codice	Capo
Rossi	A	A	Mori
Neri	B	B	Bruni
Bianchi	B	B	Bruni

Impiegati JOIN Reparti

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
A	Mori
B	Bruni

Basi di dati relazionali

Join naturale ed equi-join

Possiamo riesprimere un join naturale usando un equi-join.

Impiegati

Impiegato	Reparto
-----------	---------

Reparti

Reparto	Capo
---------	------

Impiegati JOIN Reparti

```
PROJImpiegato,Reparto,Capo ( SELReparto=Codice  
( Impiegati JOIN RENCodice <- Reparto (Reparti) ))
```

Basi di dati relazionali

Esempi

Impiegati	Matricola	Nome	Età	Stipendio
	7309	Rossi	34	45
	5998	Bianchi	37	38
	9553	Neri	42	35
	5698	Bruni	43	42
	4076	Mori	45	50
	8123	Lupi	46	60

Supervisione	Impiegato	Capo
	7309	5698
	5998	5698
	9553	4076
	5698	4076
	4076	8123

Basi di dati relazionali

Esempi

Trovare matricola, nome, età e stipendio degli impiegati che guadagnano più di 40 milioni.

SELECT Stipendio > 40 **(Impiegati)**

Matricola	Nome	Età	Stipendio
7309	Rossi	34	45
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

Basi di dati relazionali

Esempi

Trovare matricola, nome ed età degli impiegati che guadagnano più di 40 milioni.

PROJ **Matricola, Nome, Età (SEL** **Stipendio > 40 (Impiegati))**

Matricola	Nome	Età
7309	Rossi	34
5698	Bruni	43
4076	Mori	45
8123	Lupi	46

Basi di dati relazionali

Esempi

Trovare le matricole dei capi degli impiegati che guadagnano più di 40 milioni.

Impiegati **Matricola** **Nome** **Età** **Stipendio**

Supervisione **Impiegato** **Capo**

PROJ **Capo (Supervisione**
JOIN **Impiegato=Matricola**
(SEL **Stipendio > 40 (Impiegati))**

Basi di dati relazionali

Esempi

Trovare nome e stipendio dei capi degli impiegati che guadagnano più di 40 milioni.

Impiegati

Matricola	Nome	Età	Stipendio
-----------	------	-----	-----------

Supervisione

Impiegato	Capo
-----------	------

```
PROJNome,Stipendio (  
Impiegati JOINMatricola=Capo  
PROJCapo (Supervisione  
JOINImpiegato=Matricola (SELStipendio>40(Impiegati))))
```

Basi di dati relazionali

Esempi

Trovare gli impiegati che guadagnano più del proprio capo, restituendo matricola, nome e stipendio dell'impiegato e del capo.

Impiegati

Matricola	Nome	Età	Stipendio
-----------	------	-----	-----------

Supervisione

Impiegato	Capo
-----------	------

```
PROJMatr, Nome, Stip, MatrC, NomeC, StipC  
(SELStipendio>StipC  
RENMatrC, NomeC, StipC, EtàC ← Matr, Nome, Stip, Età(Impiegati)  
JOINMatrC=Capo  
(Supervisione JOINImpiegato=Matricola Impiegati)))
```

Basi di dati relazionali

Esempi

Trovare le matricole dei capi i cui impiegati guadagnano **tutti** più di 40 milioni.

Impiegati Matricola Nome Et  Stipendio

Supervisione Impiegato Capo

```
PROJ_Capo (Supervisione) -  
  PROJ_Capo (Supervisione  
    JOIN Impiegato=Matricola  
      (SEL_stipendio > 40 (Impiegati)))
```

Basi di dati relazionali

Equivalenza di espressioni

- Due espressioni sono **equivalenti** se producono lo stesso risultato qualunque sia l'istanza attuale della base di dati
- L'equivalenza   importante in pratica perch  i DBMS cercano di eseguire espressioni equivalenti a quelle date, ma meno "costose"

Basi di dati relazionali

Un'equivalenza importante

- Effettuare le selezioni il prima possibile (**push selections**)

Esempio: se A è attributo di R₁

$SEL_{A=10}(R_1 JOIN R_2) = R_1 JOIN SEL_{A=10}(R_2)$

- Le selezioni tipicamente riducono in modo significativo la dimensione del risultato intermedio (e quindi il costo dell'operazione).

Basi di dati relazionali

Selezione con valori nulli

Impiegati

Matricola	Cognome	Filiale	Età
7309	Rossi	Roma	32
5998	Neri	Milano	45
9553	Bruni	Milano	NULL

$SEL_{Età > 40}(\text{Impiegati})$

La condizione atomica è vera **solo per valori non nulli**.

Basi di dati relazionali

Un risultato non desiderabile

$SEL_{Et > 50}(Persone) \cup SEL_{Et < 30}(Persone) \neq Persone$

Perché? Perché le selezioni vengono valutate separatamente!

Ma anche

$SEL_{Et > 50 \vee Et < 30}(Persone) \neq Persone$

Perché? Perché anche le condizioni atomiche vengono valutate separatamente!

Basi di dati relazionali

Selezione con valori nulli: soluzione

$SEL_{Et > 40}(Impiegati)$

- la condizione atomica è vera solo per valori non nulli
 - per riferirsi ai valori nulli esistono forme apposite di condizioni:
IS NULL
IS NOT NULL
 - si potrebbe usare (ma non serve) una "logica a tre valori" (vero, falso, sconosciuto)
-

Basi di dati relazionali

