

Capitolo 3

Algebra relazionale

Copyright © 2004 – The McGraw-Hill Companies srl

Linguaggi di interrogazione relazionale

- Linguaggi di interrogazione: permettono la manipolazione e il reperimento di dati da una base di dati
- Il modello relazionale supporta LI semplici e potenti:
 - Forte base formale basata sulla logica
 - Ottimizzazione
- Linguaggi di interrogazione $\neq$ linguaggi di programmazione!
 - I LI non sono necessariamente "Turing completi"
 - I LI non sono fatti per essere usati in calcoli complessi
 - I LI supportano un accesso semplice ed efficiente a grandi insiemi di dati

Copyright © 2004 – The McGraw-Hill Companies srl

Linguaggi formali di interrogazione relazionale

- Due linguaggi di interrogazione matematici formano la base per i linguaggi "reali" (es. SQL) e per l'implementazione:
 - Algebra relazionale: più operativa, utilissima per rappresentare i piani di esecuzione
 - Calcolo relazionale: permette agli utenti di descrivere ciò che vogliono, piuttosto che il modo in cui calcolarlo (non operativa, dichiarativo)

☛ Capire l'algebra e il calcolo è la chiave per la comprensione dell'SQL e dell'elaborazione delle interrogazioni!

Copyright © 2004 – The McGraw-Hill Companies srl

Nozioni Preliminari

- Una interrogazione si applica alle istanze di relazione, e il risultato di una interrogazione è anch'esso una istanza di relazione
 - Gli schemi delle relazioni in ingresso sono fissi (ma l'interrogazione viene eseguita indipendentemente dall'istanza!)
 - Anche lo schema per il risultato di una data interrogazione è fisso! Determinato dalla definizione dei costrutti del linguaggio di interrogazione.
- Notazione posizionale verso notazione nominale dei campi:
 - La notazione posizionale è più semplice per definizioni formali, la notazione nominale è più leggibile
- In SQL sono usate entrambe

Copyright © 2004 – The McGraw-Hill Companies srl

Istanze di esempio

- Relazioni "Velisti" e "Prenota" per i nostri esempi
- Useremo la notazione posizionale o a campi nominati, assumeremo che i nomi dei campi nei risultati delle interrogazioni siano "ereditati" dai nomi dei campi delle relazioni in ingresso dell'interrogazione

P1

<u>vid</u>	<u>bid</u>	<u>giorno</u>
22	101	10/10/96
58	103	11/12/96

V1

<u>vid</u>	<u>vnome</u>	<u>esperienza</u>	<u>età</u>
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

V2

<u>vid</u>	<u>vnome</u>	<u>esperienza</u>	<u>età</u>
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

Copyright © 2004 – The McGraw-Hill Companies srl

L'Algebra relazionale in una slide

- Operazioni di base
 - Selezione (σ) Seleziona un sottoinsieme di righe della relazione, secondo una data condizione
 - Proiezione (π) Cancella colonne non desiderate dalla relazione
 - Prodotto cartesiano ($\times$) Consente di combinare due relazioni
 - Differenza insiemistica ($-$) Tuple presenti nella relazione 1, ma non nella relazione 2
 - Unione ($\cup$) Tuple presenti nella relazione 1 e nella relazione 2
- Altre operazioni:
 - intersezione, **join**, divisione, ridenominazione: non essenziali ma (molto!) utili
 - Poiché ogni operazione restituisce una relazione, le operazioni possono essere composte (l'algebra è "chiusa")

Copyright © 2004 – The McGraw-Hill Companies srl

Selezione

- Seleziona le righe che soddisfano una condizione di selezione
- Niente duplicati nel risultato! (Perché?)
- Lo schema del risultato è identico allo schema della (unica) relazione in ingresso
- La relazione risultato può essere l'ingresso per un'altra operazione di algebra relazionale! (Composizione di operatori)

vid	vnome	espe- rienza	età
28	yuppy	9	35.0
58	rusty	10	35.0

$\sigma_{esperienza > 8}(V2)$

vnome	espe- rienza
yuppy	9
rusty	10

$\pi_{vnome, esperienza}(\sigma_{esperienza > 8}(V2))$

Copyright © 2004 – The McGraw-Hill Companies srl

Selezione, sintassi e semantica

- sintassi

$\sigma_{Condizione}(Operando)$

– **Condizione**: espressione booleana (come quelle dei vincoli di ennuola)

- semantica

– il risultato contiene le ennuole dell'operando che soddisfano la condizione

Copyright © 2004 – The McGraw-Hill Companies srl

- impiegati che guadagnano più di 50

Impiegati

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

$\sigma_{Stipendio > 50}(\text{Impiegati})$

Copyright © 2004 – The McGraw-Hill Companies srl

- impiegati che guadagnano più di 50 e lavorano a Milano

Impiegati

Matricola	Cognome	Filiale	Stipendio
5998	Neri	Milano	64

$\sigma_{Stipendio > 50 \text{ AND Filiale} = \text{'Milano'}}(\text{Impiegati})$

$\sigma_{Stipendio > 50}(\sigma_{Filiale = \text{'Milano'}}(\text{Impiegati}))$

Copyright © 2004 – The McGraw-Hill Companies srl

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

Impiegati

Matricola	Cognome	Filiale	Stipendio
9553	Milano	Milano	44

$\sigma_{Cognome = Filiale}(\text{Impiegati})$

Copyright © 2004 – The McGraw-Hill Companies srl

Proiezione

- Cancella gli attributi che non sono nella lista di proiezione
- Lo schema del risultato contiene esattamente i campi nella lista di proiezione, con gli stessi nomi che avevano nella (unica) relazione in ingresso
- L'operatore di proiezione deve eliminare i duplicati (perché??)
 - Nota: i sistemi reali tipicamente non eliminano i duplicati a meno che l'utente non lo richieda esplicitamente (perché no?)

vnome	espe- rienza
yuppy	9
lubber	8
guppy	5
rusty	10

$\pi_{vnome, esperienza}(V2)$

età
35.0
55.5

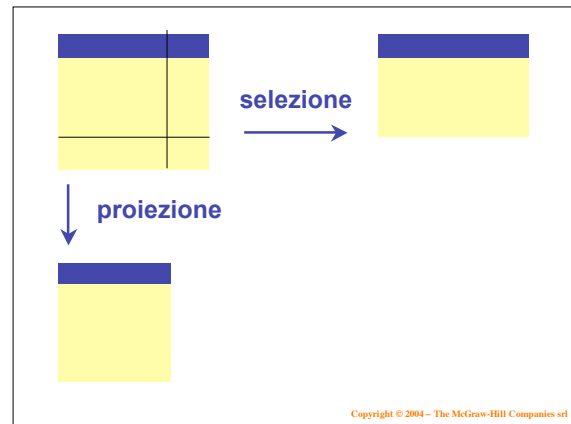
$\pi_{età}(V2)$

Copyright © 2004 – The McGraw-Hill Companies srl

Selezione e proiezione

- operatori "ortogonali"
- selezione**:
 - decomposizione orizzontale
- proiezione**:
 - decomposizione verticale

Copyright © 2004 – The McGraw-Hill Companies srl



Copyright © 2004 – The McGraw-Hill Companies srl

Proiezione

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

Copyright © 2004 – The McGraw-Hill Companies srl

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

Copyright © 2004 – The McGraw-Hill Companies srl

Proiezione, sintassi e semantica

- Sintassi: π , oppure PROJ
- π ListaAttributi (**Operando**)
- semantica
 - il risultato contiene le ennuple ottenute da tutte le ennuple dell'operando ristrette agli attributi nella lista

Copyright © 2004 – The McGraw-Hill Companies srl

- matricola e cognome di tutti gli impiegati

Matricola	Cognome
7309	Neri
5998	Neri
9553	Rossi
5698	Rossi

PROJ Matricola, Cognome (**Impiegati**)

Copyright © 2004 – The McGraw-Hill Companies srl

- cognome e filiale di tutti gli impiegati

Cognome	Filiale
Neri	Napoli
Neri	Milano
Rossi	Roma

PROJ Cognome, Filiale (Impiegati)

Copyright © 2004 – The McGraw-Hill Companies srl

Unione, intersezione, differenza insiemistica

- Tutte queste operazioni prendono in ingresso due relazioni che devono essere compatibili rispetto all'unione:
 - stesso numero di campi
 - campi "corrispondenti" sono dello stesso tipo

vid	vnome	espe- rienza	età
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0
44	guppy	5	35.0
28	yuppy	9	35.0

$V1 \cup V2$

vid	vnome	espe- rienza	età
22	dustin	7	45.0

$V1 - V2$

vid	vnome	espe- rienza	età
31	lubber	8	55.5
58	rusty	10	35.0

$V1 \cap V2$

Copyright © 2004 – The McGraw-Hill Companies srl

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

Copyright © 2004 – The McGraw-Hill Companies srl

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

Copyright © 2004 – The McGraw-Hill Companies srl

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

Copyright © 2004 – The McGraw-Hill Companies srl

Prodotto scalare

- Ciascuna riga di V1 è accoppiata con ciascuna riga di P1
- Lo schema del risultato ha un campo per ogni campo di V1 e P1, i cui nomi sono, se possibile, "ereditati"
 - Conflitto: sia V1 che P1 hanno un campo chiamato vid

(vid)	vnome	espe- rienza	età	(vid)	bid	giorno
22	dustin	7	45.0	22	101	10/10/96
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	22	101	10/10/96
31	lubber	8	55.5	58	103	11/12/96
58	rusty	10	35.0	22	101	10/10/96
58	rusty	10	35.0	58	103	11/12/96

- Operatore per ridenominare i campi $\rho(C(1 \rightarrow vid1, 5 \rightarrow vid2), V1 \times P1)$

Copyright © 2004 – The McGraw-Hill Companies srl

Join

- Join condizionale: $P \bowtie_{\theta} V = \sigma_{\theta}(P \times V)$

(vid)	vnome	espe- rienza	età	(vid)	bid	giorno
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	58	103	11/12/96

$$V1 \bowtie_{V1.vid < P1.vid} P1$$

- Lo schema del risultato è lo stesso del prodotto scalare
- Meno tuple del prodotto scalare, potrebbe essere calcolato più efficientemente
- A volte chiamato *theta-join*

Copyright © 2004 – The McGraw-Hill Companies srl

Join

- Equi-join: un caso speciale di join condizionale dove la condizione c contiene solo uguaglianze

vid	vnome	espe- rienza	età	bid	giorno
22	dustin	7	45.0	101	10/10/96
58	rusty	10	35.0	103	11/12/96

$$V1 \bowtie_{vid} P1$$

- Lo schema del risultato è simile al prodotto scalare, ma c'è solo una copia dei campi per i quali è specificata l'uguaglianza
- Join naturale: equijoin su tutti i campi in comune

Copyright © 2004 – The McGraw-Hill Companies srl

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

Copyright © 2004 – The McGraw-Hill Companies srl

Un join vuoto

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

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

Copyright © 2004 – The McGraw-Hill Companies srl

Un join completo, con $n \times m$ 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
Neri	B	Bruni

Copyright © 2004 – The McGraw-Hill Companies srl

Join, una difficoltà

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"

Copyright © 2004 – The McGraw-Hill Companies srl

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 ...

Copyright © 2004 – The McGraw-Hill Companies srl

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

Impiegati JOIN _{LEFT} Reparti		
Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
Rossi	A	NULL

Copyright © 2004 – The McGraw-Hill Companies srl

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

Impiegati JOIN _{RIGHT} Reparti		
Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
NULL	C	Bruni

Copyright © 2004 – The McGraw-Hill Companies srl

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

Impiegati JOIN _{FULL} Reparti		
Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
Rossi	A	NULL
NULL	C	Bruni

Copyright © 2004 – The McGraw-Hill Companies srl

Esempio: Trovare i nomi dei velisti che hanno prenotato la barca #103

- Soluzione 1: $\pi_{vnome}(\sigma_{bid=103}(Prenota) \bowtie Velisti)$
- ❖ Soluzione 2: $\rho(Temp1, \sigma_{bid=103}(Prenota))$
 $\rho(Temp2, Temp1 \bowtie Velisti)$
 $\pi_{vnome}(Temp2)$
- ❖ Soluzione 3: $\pi_{vnome}(\sigma_{bid=103}(Prenota \bowtie Velisti))$

Copyright © 2004 – The McGraw-Hill Companies srl

Trovare i nomi dei velisti che hanno prenotato una barca rossa

- Informazioni sul colore sono disponibili solo in Barche; quindi abbiamo bisogno di un ulteriore join:

$$\pi_{vnome}(\sigma_{colore='rosso'}(Barche) \bowtie Prenota \bowtie Velisti)$$

- ❖ Una soluzione più efficiente:

$$\pi_{vnome}(\pi_{vid}(\pi_{bid}(\sigma_{colore='rosso'}(Barche)) \bowtie Prenota) \bowtie Velisti)$$

➡ Un ottimizzatore di interrogazioni può trovare la seconda soluzione data la prima!

Copyright © 2004 – The McGraw-Hill Companies srl

Trovare i nomi dei velisti che hanno prenotato una barca rossa o una verde

- Possiamo identificare tutte le barche rosse o verdi, poi trovare i velisti che hanno prenotato una di esse :

$\rho(Tempbarche, (\sigma_{colore='rosso' \text{ o } colore='verde'} Barche))$

$\pi_{vnome}(Tempbarche \bowtie Prenota \bowtie Velisti)$

- ❖ Possiamo anche definire TempBarche usando l'unione (come?)
- ❖ Che succede se $\cup$ è sostituito da $\cap$ in questa interrogazione?

Copyright © 2004 – The McGraw-Hill Companies srl

Trovare i nomi dei velisti che hanno prenotato una barca rossa e una verde

- L'approccio precedente non funziona! Dobbiamo identificare i velisti che hanno prenotato barche rosse, i velisti che hanno prenotato barche verdi, poi trovare l'intersezione (notate che *vid* è una chiave per Velisti)

$\rho(Temprosse, \pi_{vid}(\sigma_{colore='rosso'}(Barche)) \bowtie Prenota)$

$\rho(Tempverdi, \pi_{vid}(\sigma_{colore='verde'}(Barche)) \bowtie Prenota)$

$\pi_{vnome}(Temprosse \cap Tempverdi) \bowtie Velisti$

Copyright © 2004 – The McGraw-Hill Companies srl

Trovare i nomi dei velisti che hanno prenotato tutte le barche

- Usa la divisione; gli schemi delle relazioni in ingresso alla divisione devono essere scelti con cura:

$\rho(Tempvids, (\pi_{vid,bid} Prenota) / (\pi_{bid} Barche))$

$\pi_{vnome}(Tempvids \bowtie Velisti)$

- ❖ Per trovare i velisti che hanno riservato tutte le barche "Interlago":

$..... / \pi_{bid}(\sigma_{bnome='Interlago'} Barche)$

Copyright © 2004 – The McGraw-Hill Companies srl

Esempi sugli impiegati

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

Copyright © 2004 – The McGraw-Hill Companies srl

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

$SEL_{Stipendio > 40}(Impiegati)$

Copyright © 2004 – The McGraw-Hill Companies srl

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

$SEL_{Stipendio > 40}(Impiegati)$

Copyright © 2004 – The McGraw-Hill Companies srl

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

PROJ_{Matricola, Nome, Età} (SEL_{Stipendio>40}(Impiegati))

Copyright © 2004 – The McGraw-Hill Companies srl

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

**PROJ_{Matricola, Nome, Età}
(SEL_{Stipendio>40}(Impiegati))**

Copyright © 2004 – The McGraw-Hill Companies srl

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

**PROJ_{Capo} (Supervisione
JOIN_{Impiegato=Matricola}
(SEL_{Stipendio>40}(Impiegati)))**

Copyright © 2004 – The McGraw-Hill Companies srl

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

**PROJ_{Nome,Stipendio} (Impiegati JOIN_{Matricola=Capo}
PROJ_{Capo}(Supervisione
JOIN_{Impiegato=Matricola} (SEL_{Stipendio>40}(Impiegati))))**

Copyright © 2004 – The McGraw-Hill Companies srl

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

**PROJ_{Matr, Nome, Stip, MatrC, NomeC, StipC}
(SEL_{Stipendio>StipC}
REN_{MatrC, NomeC, StipC, EtàC ← Matr, Nome, Stip, Età} (Impiegati)
JOIN_{MatrC=Capo}
(Supervisione JOIN_{Impiegato=Matricola} Impiegati)))**

Copyright © 2004 – The McGraw-Hill Companies srl

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

**PROJ_{Capo} (Supervisione) –
PROJ_{Capo} (Supervisione
JOIN_{Impiegato=Matricola}
(SEL_{Stipendio ≤ 40}(Impiegati)))**

Copyright © 2004 – The McGraw-Hill Companies srl

Equivalenza di espressioni

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

Copyright © 2004 – The McGraw-Hill Companies srl

Un 'equivalenza importante

- Push selections (se C coinvolge solo attributi di R_2)

$$SEL_C (R_1 \text{ JOIN } R_2) = R_1 \text{ JOIN } SEL_C (R_2)$$
- Riduce in modo significativo la dimensione del risultato intermedio (e quindi il costo in tempo e memoria dell'operazione)

Copyright © 2004 – The McGraw-Hill Companies srl

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

Copyright © 2004 – The McGraw-Hill Companies srl

Un risultato non desiderabile

$SEL_{Età > 30} (\text{Persone}) \cup SEL_{Età \leq 30} (\text{Persone}) \neq \text{Persone}$

- Perché? Perché le selezioni vengono valutate separatamente!
- Ma anche

$$SEL_{Età > 30 \vee Età \leq 30} (\text{Persone}) \neq \text{Persone}$$
- Perché? Perché anche le condizioni atomiche vengono valutate separatamente!

Copyright © 2004 – The McGraw-Hill Companies srl

Selezione con valori nulli: soluzione

$SEL_{Età > 40} (\text{Impiegati})$

- 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)

Copyright © 2004 – The McGraw-Hill Companies srl

- Quindi:

$$SEL_{Età > 30} (\text{Persone}) \cup SEL_{Età \leq 30} (\text{Persone}) \cup SEL_{Età\ IS\ NULL} (\text{Persone})$$

$$=$$

$$SEL_{Età > 30 \vee Età \leq 30 \vee Età\ IS\ NULL} (\text{Persone})$$

$$=$$

$$\text{Persone}$$

Copyright © 2004 – The McGraw-Hill Companies srl

Impiegati

Matricola	Cognome	Filiale	Età
5998	Neri	Milano	45
9553	Bruni	Milano	NULL

SEL (Età > 40) OR (Età IS NULL) (Impiegati)

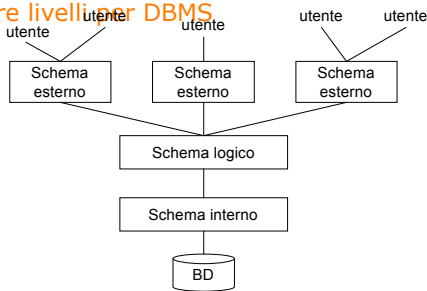
Copyright © 2004 – The McGraw-Hill Companies srl

Viste (relazioni derivate)

- Rappresentazioni diverse per gli stessi dati ([schema esterno](#))
- **Relazioni derivate:**
 - relazioni il cui contenuto è funzione del contenuto di altre relazioni (definito per mezzo di interrogazioni)
- **Relazioni di base:** contenuto autonomo
- Le relazioni derivate possono essere definite su altre derivate, ma ...

Copyright © 2004 – The McGraw-Hill Companies srl

Architettura standard (ANSI/SPARC) a tre livelli per DBMS



Copyright © 2004 – The McGraw-Hill Companies srl

Viste virtuali e materializzate

- Due tipi di relazioni derivate:
 - [viste materializzate](#)
 - [relazioni virtuali](#) (o viste)

Copyright © 2004 – The McGraw-Hill Companies srl

Viste materializzate

- relazioni **derivate** memorizzate nella base di dati
 - vantaggi:
 - immediatamente disponibili per le interrogazioni
 - svantaggi:
 - ridondanti
 - appesantiscono gli aggiornamenti
 - sono raramente supportate dai DBMS

Copyright © 2004 – The McGraw-Hill Companies srl

Viste virtuali

- **relazioni virtuali** (o viste):
 - sono supportate dai DBMS (tutti)
 - una interrogazione su una vista viene eseguita "ricalcolando" la vista (o quasi)

Copyright © 2004 – The McGraw-Hill Companies srl

Viste, esempio

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

- una vista:
Supervisione =
PROJ_{Impiegato, Capo} (Afferenza JOIN Direzione)

Copyright © 2004 – The McGraw-Hill Companies srl

Interrogazioni sulle viste

- Sono eseguite sostituendo alla vista la sua definizione:

SEL_{Capo='Leoni'} (Supervisione)
viene eseguita come
SEL_{Capo='Leoni'} (
PROJ_{Impiegato, Capo} (Afferenza JOIN Direzione))

Copyright © 2004 – The McGraw-Hill Companies srl

Viste, motivazioni

- Schema esterno: ogni utente vede solo
 - ciò che gli interessa e nel modo in cui gli interessa, senza essere distratto dal resto
 - ciò che e' autorizzato a vedere (autorizzazioni)
 - Strumento di programmazione:
 - si può semplificare la scrittura di interrogazioni: espressioni complesse e sottoespressioni ripetute
 - Utilizzo di programmi esistenti su schemi ristrutturati
- Invece:
- L'utilizzo di viste non influisce sull'efficienza delle interrogazioni



Copyright © 2004 – The McGraw-Hill Companies srl

Viste come strumento di programmazione

- Trovare gli impiegati che hanno lo stesso capo di Rossi
- Senza vista:

PROJ_{Impiegato} (Afferenza JOIN Direzione) JOIN
REN ImpR,RepR ← Imp,Reparto (
SEL_{Impiegato='Rossi'} (Afferenza JOIN Direzione))

- Con la vista:

PROJ_{Impiegato} (Supervisione) JOIN
REN ImpR,RepR ← Imp,Reparto (
SEL_{Impiegato='Rossi'} (Supervisione))



Copyright © 2004 – The McGraw-Hill Companies srl

Viste e aggiornamenti, attenzione

Afferenza		Direzione	
Impiegato	Reparto	Reparto	Capo
Rossi	A	A	Mori
Neri	B	B	Bruni
Verdi	A	C	Bruni

Supervisione	
Impiegato	Capo
Rossi	Mori
Neri	Bruni
Verdi	Mori

- Vogliamo inserire, nella vista, il fatto che Lupi ha come capo Bruni; oppure che Belli ha come capo Falchi; come facciamo?

Copyright © 2004 – The McGraw-Hill Companies srl

Viste e aggiornamenti

- "Aggiornare una vista":
 - modificare le relazioni di base in modo che la vista, "ricalcolata" rispecchi l'aggiornamento
- L'aggiornamento sulle relazioni di base corrispondente a quello specificato sulla vista deve essere univoco
- In generale però non è univoco!
- Ben pochi aggiornamenti sono ammissibili sulle viste

Copyright © 2004 – The McGraw-Hill Companies srl

Una convenzione e notazione alternativa per i join

- Nota: è sostanzialmente l'approccio usato in SQL
- Ignoriamo il join naturale (cioè non consideriamo implicitamente condizioni su attributi con nomi uguali)
- Per "riconoscere" attributi con lo stesso nome gli premettiamo il nome della relazione
- Usiamo "assegnazioni" (viste) per ridenominare le relazioni (e gli attributi solo quando serve per l'unione)

Copyright © 2004 – The McGraw-Hill Companies srl

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

```

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

Copyright © 2004 – The McGraw-Hill Companies srl

```

PROJMatr,Nome,Stip,MatrC,NomeC,StipC
  (SELStip>StipC(
RENMatrC,NomeC,StipC,EtàC ← Matr,Nome,Stip,Età(Imp)
  JOINMatrC=Capo
  (Sup JOINImp=Matr Imp)))

Capi := Imp

PROJImp.Matr, Imp.Nome, Imp.Stip,Capi.Matr,Capi.Nome, Capi.Stip
  (SELImp.Stip>Capi.Stip(
Capi JOINCapi.Matr=Capo (Sup JOINImp=Imp.Matr Imp)))
  
```

Copyright © 2004 – The McGraw-Hill Companies srl