

Corso di Laurea in Ingegneria  
Informatica  
*Fondamenti di Informatica I*  
Modulo "*Basi di dati*"  
a.a. 2017-2018

Docente: Gigliola Vaglini  
Docente laboratorio: Francesco  
Pistolesi

1

Lezione 3

Significato e  
implementazione di una  
interrogazione

2

- Cosa si intende per interrogazione?
  - Operazione di lettura sul DB che può richiedere l'accesso a più di una tabella
- Cosa è necessario fare per realizzare una interrogazione?

3

## Semantica di un linguaggio di programmazione

- Operazionale
  - Si specificano le modalità di generazione del risultato
  - Nel caso di SQL si definisce questa semantica tramite la cosiddetta Algebra relazionale
- Dichiarativa
  - Si specificano le proprietà del risultato,
  - Nel caso di SQL si usa il Calcolo relazionale

4

## Due semantiche?

- Il metodo dichiarativo è l'effettiva semantica del linguaggio, infatti le interrogazioni sono espresse ad alto livello (ricordare il concetto di **indipendenza dei dati**)
  - Nessun concetto di costo (ma in realtà si può vedere perchè ↓)
- Il metodo operativo è il modo che usa il DBMS per eseguire un'istruzione SQL

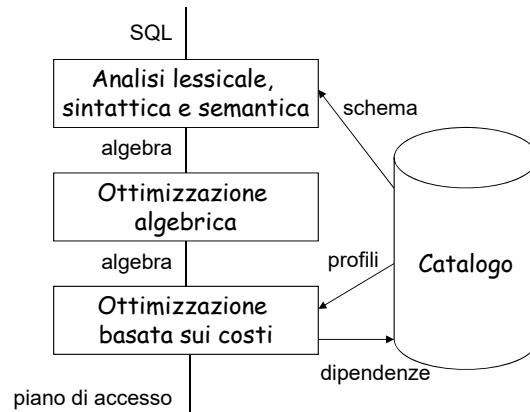
5

## Il DBMS

- Il DBMS contiene un modulo specifico detto **Query processor**, all'interno del quale è definito il processo di esecuzione delle interrogazioni
- Una parte del QP si occupa di *ottimizzare* (il QP è detto anche Ottimizzatore) la query prima dell'esecuzione (la query è scritta indipendentemente dal suo costo, importa solo il risultato).  
L'ottimizzatore sceglie la strategia (di solito fra diverse alternative) di esecuzione per ogni istruzione SQL.

6

# Il Query Processor



7

## "Profili" delle relazioni

- Informazioni quantitative:
  - cardinalità di ciascuna relazione
  - dimensioni delle tuple
  - dimensioni dei valori
  - numero di valori distinti degli attributi
  - valore minimo e massimo di ciascun attributo
- Sono memorizzate nel "catalogo" e possono essere aggiornate con comandi del tipo *update statistics*
- Utilizzate nella fase finale dell'ottimizzazione, per stimare le dimensioni dei risultati intermedi

8

## Ottimizzazione algebrica

- Il termine ottimizzazione è improprio perché il processo utilizza euristiche
- Si basa sulla nozione di equivalenza:
  - Due espressioni sono equivalenti se producono lo stesso risultato qualunque sia l'istanza attuale della base di dati

9

## Algebra relazionale

- Algebra = dati + operatori
- Algebra relazionale:
  - Dati: relazioni
  - Operatori:
    - su relazioni
    - che producono relazioni
    - e possono essere composti

10

## Operatori dell'algebra relazionale

- Operatori su insiemi
  - unione, intersezione, differenza
- Operatori su relazioni
  - ridenominazione
  - selezione
  - proiezione
  - join (join naturale, prodotto cartesiano, theta-join)

11

## Operatori su insiemi

- Le relazioni sono insiemi
- Però i risultati debbono essere relazioni
  - quindi è possibile applicare unione, intersezione, differenza solo a relazioni definite sugli stessi attributi, in modo che il risultato sia una relazione sugli stessi attributi

12

## Unione

- L'unione di due relazioni sullo stesso insieme di attributi X è una relazione su X che contiene le tuple sia dell'una che dell'altra relazione originaria

13

## Unione

**Laureati triennali**

Matricola	Nome	Età
7274	Rossi	32
7432	Neri	24
9824	Verdi	25

**Laureati magistrali**

Matricola	Nome	Età
9297	Neri	33
7432	Neri	24
9824	Verdi	25

**Laureati triennali  $\cup$  Laureati magistrali**

Matricola	Nome	Età
7274	Rossi	32
7432	Neri	24
9824	Verdi	25
9297	Neri	33

14

## Intersezione

- L'intersezione di due relazioni sullo stesso insieme di attributi X è una relazione su X che contiene le tuple appartenenti ad entrambe le relazioni

15

## Intersezione

**Laureati triennali**

Matricola	Nome	Età
7274	Rossi	32
7432	Neri	24
9824	Verdi	25

**Laureati magistrali**

Matricola	Nome	Età
9297	Neri	33
7432	Neri	24
9824	Verdi	25

**Laureati triennali  $\cap$  Laureati magistrali**

Matricola	Nome	Età
7432	Neri	24
9824	Verdi	25

16



## Differenza

- La differenza di due relazioni sullo stesso insieme di attributi X,  
–  $r_1(X) - r_2(X)$
- è una relazione su X che contiene le tuple di  $r_1$  che non appartengono anche ad  $r_2$ .

17

## Differenza

**Laureati triennali**

Matricola	Nome	Età
7274	Rossi	32
7432	Neri	24
9824	Verdi	25

**Laureati magistrali**

Matricola	Nome	Età
9297	Neri	33
7432	Neri	24
9824	Verdi	25

**Laureati triennali – Laureati magistrali**

Matricola	Nome	Età
7274	Rossi	32

18

È possibile l'unione delle due relazioni seguenti?

**Paternità**

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

**Maternità**

Madre	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

**Paternità  $\cup$  Maternità**  
??

19

## Ridenominazione

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

$$\rho_{B_1..B_n \leftarrow A_1..A_n}(r)$$

20

### Paternità

Padre	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

$\rho_{\text{Genitore} \leftarrow \text{Padre}}$  (Paternità)

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

21

$\rho_{\text{Genitore} \leftarrow \text{Padre}}$  (Paternità)

Genitore	Figlio
Adamo	Abele
Adamo	Caino
Abramo	Isacco

$\rho_{\text{Genitore} \leftarrow \text{Padre}}$  (Paternità)



$\rho_{\text{Genitore} \leftarrow \text{Madre}}$  (Maternità)

$\rho_{\text{Genitore} \leftarrow \text{Madre}}$  (Maternità)

Genitore	Figlio
Eva	Abele
Eva	Set
Sara	Isacco

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

22

## Selezione

- operatore monadico
- produce un risultato che
  - ha lo stesso schema dell'argomento e
  - contiene il sottoinsieme delle sue tuple che soddisfano una condizione fissata

23

## Sintassi e semantica

- data una relazione  $r(X)$ 
$$\sigma_F(r) = r'$$
  - $F$ : espressione booleana ottenuta componendo con and, or e not condizioni atomiche del tipo  $A \vartheta B$  oppure  $A \vartheta c$ , con  $A$  e  $B$  attributi in  $X$  con domini compatibili,  $\vartheta$  operatore di confronto ( $<, >, =, \dots$ ) e  $c$  costante compatibile con il dominio di  $A$ .
  - $r'$  contiene il sottoinsieme delle tuple di  $r$  per cui  $F$  e' vera

24

- impiegati che guadagnano più di 50000 euro

### Impiegati

Matricola	Cognome	Filiale	Stipendio
7309	Rossi	Roma	55000
5998	Neri	Milano	64000
5698	Neri	Napoli	64000

$\sigma_{\text{Stipendio} > 50000}$  (Impiegati)

25

- impiegati che guadagnano più di 50000  
e lavorano a Milano

### Impiegati

Matricola	Cognome	Filiale	Stipendio
5998	Neri	Milano	64000

$\sigma_{(\text{Stipendio} > 50000) \text{ AND } (\text{Filiale} = \text{'Milano'})}$  (Impiegati)

26

## Selezione con valori nulli

### Impiegati

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

$\sigma_{\text{Età} > 40}$  (Impiegati)

- la condizione è vera solo per valori non nulli

27

per riferirsi ai valori nulli esistono forme apposite di condizioni:

IS NULL

IS NOT NULL

### Impiegati

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

$\sigma_{(\text{Età} > 40) \vee (\text{Età IS NULL})}$  (Impiegati)

28

## Proiezione

- operatore monadico
- produce un risultato che
  - ha parte degli attributi dell'argomento e su tali attributi contiene tutte le possibili tuple di valori esistenti nella relazione argomento

29

## Sintassi e semantica

- Sintassi,  $Y \subseteq X$ 
$$\pi_Y (r(X)) = r'$$
- Semantica
  - $r'$  è una relazione su  $Y$  e contiene l'insieme delle tuple di  $r$  ristrette agli attributi in  $Y$

30

– matricola e cognome di tutti gli impiegati

Matricola	Cognome	
7309	Neri	
5998	Neri	
9553	Rossi	
5698	Rossi	

$\pi_{\text{Matricola, Cognome}}$  (Impiegati)

31

– cognome e filiale di tutti gli impiegati

	Cognome	Filiale	
	Neri	Napoli	
	Neri	Milano	
	Rossi	Roma	

$\pi_{\text{Cognome, Filiale}}$  (Impiegati)

Attenzione: perché c'è differenza nella dimensione dei due risultati?

32



## Cardinalità delle proiezioni

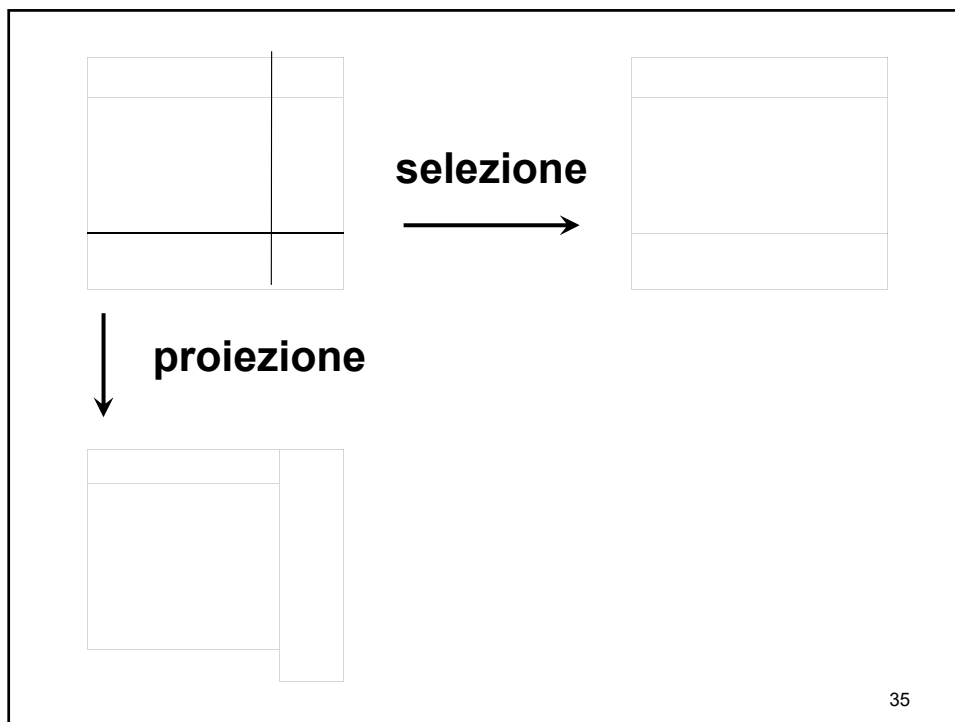
- una proiezione di  $r$ 
  - contiene al più tante tuple quante ne ha  $r$
  - può contenerne di meno
- se  $X$  è una superchiave di  $r$ , allora  $\pi_X(r)$  contiene esattamente tante tuple quante ne ha  $r$

33

## Proiezione e selezione

- Selezione  $\sigma$ 
  - decomposizione orizzontale
- Proiezione  $\pi$ 
  - decomposizione verticale

34



## Selezione e proiezione

- Combinando selezione e proiezione, possiamo estrarre interessanti informazioni da una relazione

36

- matricola e cognome degli impiegati che guadagnano più di 50

Matricola	Cognome		
7309	Rossi		
5998	Neri		
5698	Neri		

$\pi_{\text{Matricola, Cognome}} ( \sigma_{\text{Stipendio} > 50} (\text{Impiegati}) )$

Attenzione: ordine degli operatori

37

- Combinando selezione e proiezione, non possiamo però correlare informazioni presenti in relazioni diverse

38

## Una prima combinazione

- Il prodotto cartesiano di due relazioni, R e Q: operatore X

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
B	Mori
C	Bruni

R.Impiegato	R.Reparto	Q.Cap	Q.Reparto
Neri	B	Mori	B
Bianchi	B	Mori	B
Neri	B	Mori	C
Bianchi	B	Bruni	C
Rossi	A	Bruni	C
Rossi	A	Bruni	B

39

## Join

- il join è l'operatore più interessante dell'algebra relazionale
- permette appunto di correlare dati in relazioni diverse
- Non è un operatore primitivo

40

## Join naturale

- operatore binario (generalizzabile)
- produce come risultato una relazione tale che
  - Il suo schema ha l'unione degli attributi degli argomenti
  - L'insieme delle tuple è ottenuto componendo una tupla di ognuno degli operandi per valori uguali degli attributi comuni

41

## Sintassi e semantica

- $R_1(X_1), R_2(X_2)$
- $R_1 \bowtie R_2$  è una relazione su  $X_1 \cup X_2$  definita come
$$\{ t \mid \text{esistono } t_1 \in R_1 \text{ e } t_2 \in R_2 \\ \text{con } t[X_1] = t_1 \text{ e } t[X_2] = t_2 \}$$

42

## Join e proiezioni

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
B	Mori
C	Bruni

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

Impiegato	Reparto
Neri	B
Bianchi	B

Reparto	Capo
B	Mori

43

## Proiezioni e join

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

Impiegato	Reparto
Neri	B
Bianchi	B
Verdi	A

Reparto	Capo
B	Mori
B	Bruni
A	Bini

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

44

## In generale

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

$$\pi_{X_1}(R_1 \bowtie R_2) \subseteq R_1$$

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

$$(\pi_{X_1}(R)) \bowtie (\pi_{X_2}(R)) \supseteq R$$

45

## Relazioni senza attributi comuni

- La definizione di join funziona ugualmente
  - $R_1(X_1), R_2(X_2)$
  - $R_1 \bowtie R_2$  è una relazione su  $X_1 \cup X_2$  definita come

$$\{ t \mid \text{esistono } t_1 \in R_1 \text{ e } t_2 \in R_2$$

$$\text{con } t[X_1] = t_1 \text{ e } t[X_2] = t_2 \}$$

46

## Risultato

- La relazione risultato contiene sempre un numero di tuple pari al prodotto delle cardinalità degli operandi (le tuple sono tutte combinabili )
- Equivale al prodotto cartesiano su tuple

47

### Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

### Reparti

Codice	Capo
A	Mori
B	Bruni

### Impiegati $\bowtie$ 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

48



- Il **prodotto** cartesiano, può essere ridotto eseguendo una selezione

$$\sigma_F (R_1 \times R_2)$$

- L'operazione complessiva può venire eseguita tramite un operatore derivato chiamato theta-join e indicato con

$$R_1 \bowtie_F R_2$$

49

## Perché "theta-join"

- La condizione  $F$  è spesso una congiunzione (AND) di atomi di confronto  $A_1 \vartheta A_2$  dove  $\vartheta$  è uno degli operatori di confronto ( $=, >, <, \dots$ ) e  $A_1, A_2$  sono attributi di relazioni diverse
- se l'operatore è sempre l'uguaglianza ( $=$ ) allora si parla di equi-join

50

### Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

### Reparti

Codice	Capo
A	Mori
B	Bruni

Impiegati  $\triangleright \triangleleft$  Reparto=Codice Reparti

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

51

## Join naturale ed equi-join

### Impiegati

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

### Reparti

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

Impiegati  $\triangleright \triangleleft$  Reparti =

$\pi$  Impiegato,Reparto,Capo (

$\rho$  I.Reparto  $\leftarrow$  Reparto (Impiegati)  $\triangleright \triangleleft$  I.Reparto= Reparto Reparti ))

52

## Prodotto cartesiano e join naturale

- **Impiegati  $\bowtie$  Reparti =**

$$\pi_{\text{Impiegato, Reparto, Capo}}(\sigma_{\text{I.reparto}=\text{reparto}}(\rho_{\text{I.Reparto} \leftarrow \text{Reparto}}(\text{Impiegati})) \times \text{Reparti})$$

Il join non è un operatore primitivo

53

## Esempi

Impiegati	Matricola	Nome	Età	Stipendio
	7309	Rossi	34	45000
	5998	Bianchi	37	38000
	9553	Neri	42	35000
	5698	Bruni	43	42000
	4076	Mori	45	50000
	8123	Lupi	46	60000

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

54

- Trovare le matricole dei capi degli impiegati che guadagnano più di 40000 euro

$$\pi_{\text{Capo}}(\text{Supervisione} \triangleright \triangleleft \text{Impiegato} = \text{Matricola} (\sigma_{\text{Stipendio} > 40000}(\text{Impiegati})))$$

55

- Trovare le matricole dei capi i cui impiegati guadagnano tutti più di 40000 euro

$$\pi_{\text{Capo}}(\text{Supervisione}) - \pi_{\text{Capo}}(\text{Supervisione} \triangleright \triangleleft \text{Impiegato} = \text{Matricola} (\sigma_{\text{Stipendio} \leq 40000}(\text{Impiegati})))$$

56

## Esempi

Impiegati	Matricola	Nome	Età	Stipendio
	7309	Rossi	34	45000
	5998	Bianchi	37	38000
	9553	Neri	42	35000
	5698	Bruni	43	42000
	4076	Mori	45	50000
	8123	Lupi	46	60000

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

57

- Trovare nome e stipendio dei capi degli impiegati che guadagnano più di 40000 euro

$\pi_{\text{Nome, Stipendio}} ($   
 $\text{Impiegati} \triangleright \triangleleft_{\text{Matricola=Capo}}$   
 $\pi_{\text{Capo}}(\text{Supervisione}$   
 $\triangleright \triangleleft_{\text{Impiegato=Matricola}} (\sigma_{\text{Stipendio} > 40000}(\text{Impiegati}))))$

58

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

$$\pi_{\text{Matr, Nome, Stip, MatrC, NomeC, StipC}} \left( \sigma_{\text{Stipendio} > \text{StipC}} \left( \rho_{\text{MatrC, NomeC, StipC, EtàC} \leftarrow \text{Matr, Nome, Stip, Età}(\text{Impiegati}) \right. \right. \right. \\ \left. \left. \left. \triangleright \triangleleft \text{MatrC} = \text{Capo} \right. \right. \right. \\ \left. \left. \left. (\text{Supervisione} \triangleright \triangleleft \text{Impiegato} = \text{Matricola} \text{ Impiegati)) \right) \right)$$

59

## Limiti del formalismo

- l'insieme di interrogazioni esprimibili è significativo
- Ci sono però interrogazioni interessanti non esprimibili, ad es.
  - interrogazioni inerentemente ricorsive, come la chiusura transitiva

60

## Chiusura transitiva

Supervisione(Impiegato, Capo)

- Per ogni impiegato, trovare tutti i superiori (cioè il capo, il capo del capo, e così via)

Impiegato	Capo
Rossi	Lupi
Neri	Bruni
Lupi	Falchi

Impiegato	Superiore
Rossi	Lupi
Neri	Bruni
Lupi	Falchi
Rossi	Falchi

61

## Chiusura transitiva

- Nell'esempio, basterebbe il join della relazione con se stessa, previa opportuna ridenominazione
- Ma aggiungiamo una nuova ennupla

Impiegato	Capo
Rossi	Lupi
Neri	Bruni
Lupi	Falchi
Falchi	Leoni

Impiegato	Superiore
Rossi	Lupi
Neri	Bruni
Lupi	Falchi
Rossi	Falchi
Lupi	Leoni
Rossi	Leoni

62

## Chiusura transitiva

- Non esiste in algebra ( e anche nel calcolo)relazionale la possibilità di esprimere l'interrogazione che calcoli la chiusura transitiva di una relazione qualunque
- L'operazione si simula con un numero di join illimitato

63

## Equivalenza di espressioni

64



## 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"

65

## Equivalenze importanti (1)

- Pushing selections down (se  $A$  è attributo di  $R_2$ )  
$$\sigma_{A=10}(R_1 \bowtie R_2) = R_1 \bowtie \sigma_{A=10}(R_2)$$
- Riduce in modo significativo la dimensione del risultato intermedio (e quindi il costo dell'operazione)

66

## Equivalenze importanti (2)

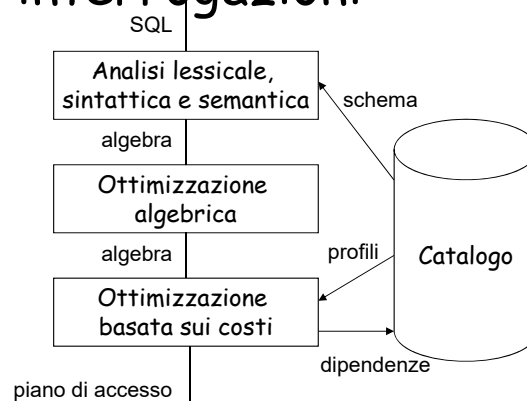
- Pushing projections down (siano dati  $R_1(X_1)$  e  $R_2(X_2)$  con  $Y_2 \subseteq X_2$  )

$$\pi_{X_1 Y_2}(R_1 \bowtie R_2) = R_1 \bowtie \pi_{Y_2}(R_2)$$

- Riduce in modo significativo la dimensione del risultato intermedio

67

## Il processo di esecuzione delle interrogazioni



68

## Ottimizzazione algebrica

- Il termine ottimizzazione è improprio perché il processo utilizza euristiche e si basa sulla
  - nozione di equivalenza:
    - Due espressioni sono equivalenti se producono lo stesso risultato qualunque sia l'istanza attuale della base di dati
- Euristiche fondamentali:
  - selezioni e proiezioni il più presto possibile (per ridurre le dimensioni dei risultati intermedi):
    - "push selections down"
    - "push projections down"

69

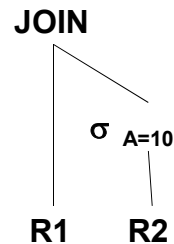
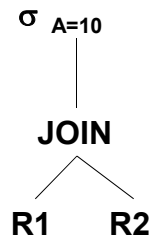
## Rappresentazione interna delle interrogazioni

- Alberi:
  - foglie: dati (relazioni, file)
  - nodi intermedi: operatori (operatori algebrici, poi effettivi operatori di accesso ai dati)

70

## Alberi per la rappresentazione di interrogazioni

- $\sigma_{A=10} (R_1 \text{ JOIN } R_2)$
- $R_1 \text{ JOIN } \sigma_{A=10} (R_2)$



71

## Procedura euristica dell'ottimizzatore

- Decomporre le selezioni congiuntive in successive selezioni atomiche
- Anticipare il più possibile le selezioni
- In una sequenza di selezioni, anticipare le più selettive
- Combinare prodotti cartesiani e selezioni per formare join
- Anticipare il più possibile le proiezioni (anche introducendone di nuove)

72

## Esempio

R1(ABC), R2(DEF), R3(GHI)

$$\pi_{AE} (\sigma_{B>100 \text{ AND } H=7 \text{ AND } I>2} ((R1 \text{ JOIN}_{C=D} R2) \text{ JOIN}_{F=G} R3))$$

73

## Esempio, continua

$$\pi_{AE} (\sigma_{B>100 \text{ AND } H=7 \text{ AND } I>2} ((R1 \text{ JOIN}_{C=D} R2) \text{ JOIN}_{F=G} R3))$$

- viene trasformata in

$$\pi_{AE} (\sigma_{B>100} (R1) \text{ JOIN}_{C=D} \pi_{AE} (R2) \text{ JOIN}_{F=G} \sigma_{I>2} (\sigma_{H=7} (R3))))$$

- oppure

$$\pi_{AE} ((\pi_{AC} (\sigma_{B>100} (R1))) \text{ JOIN}_{C=D} R2) \text{ JOIN}_{F=G} \pi_G (\sigma_{I>2} (\sigma_{H=7} (R3))))$$

74

## Relazioni derivate

75

## Relazioni derivate

- Relazioni di base: contenuto autonomo
- Relazioni derivate:
  - relazioni il cui contenuto è funzione del contenuto di altre relazioni (ed è definito per mezzo di interrogazioni)
  - Due tipi di relazioni derivate:
    - viste materializzate
    - viste virtuali (o viste)

76

## Viste materializzate

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

77

## Viste virtuali

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

78

## Esempio

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

- una vista

Supervisione =

$\pi_{\text{Impiegato, Capo}} (\text{Afferenza} \triangleright \triangleleft \text{Direzione})$

79

## Interrogazioni sulle viste

- Sono eseguite sostituendo alla vista la sua definizione:

$\sigma_{\text{Capo}='Leoni'} (\text{Supervisione})$

viene eseguita come

$\sigma_{\text{Capo}='Leoni'} (\pi_{\text{Impiegato, Capo}} (\text{Afferenza} \triangleright \triangleleft \text{Direzione}))$

80



## Viste, motivazioni

- Strumento di programmazione :
  - si può semplificare la scrittura di interrogazioni: espressioni complesse e sottoespressioni ripetute
- L'utilizzo di viste virtuali non influisce sull'efficienza delle interrogazioni

81

## Viste come strumento di programmazione

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

$$\pi_{\text{Impiegato,Capo}} (\text{Afferenza} \triangleright \triangleleft \text{Direzione}) \triangleright \triangleleft$$

$$\pi_{\text{Capo}} ($$

$$\sigma_{\text{Impiegato}='Rossi'} (\text{Afferenza} \triangleright \triangleleft \text{Direzione}))$$

- Con la vista:

$$\pi_{\text{Impiegato,Capo}} (\text{Supervisione}) \triangleright \triangleleft$$

$$\pi_{\text{Capo}} ($$

$$\sigma_{\text{Impiegato}='Rossi'} (\text{Supervisione}))$$

82

Con le viste posso evitare di usare  
l'operatore di ridenominazione, ad  
esempio, posso scrivere

**Capi := Impiegati**

Ed usare due istanze della relazione impiegati con  
nome diverso

83

Estensioni dell'algebra

84

## Algebra relazionale estesa

- Il modello relazionale può essere facilmente esteso a comprendere gli operatori SQL non direttamente riconducibili agli operatori algebrici introdotti
- Questa estensione non modifica il funzionamento del modello

85

## Estensioni (1)

- Join esterno (left, right, full)
- permette di generare valori null per mezzo delle espressioni dell'algebra relazionale per modellare le informazioni mancanti

86

## Estensioni (2)

- Proiezione generalizzata

- $\pi_{F1,F2,F3}(E)$

- F1,F2, F3 sono espressioni aritmetiche su attributi di E (che è una qualunque espressione dell'algebra) e costanti

87

Conto		
Cliente	Credito	Spese
Andrea	6000	1000
Andrea	4000	500
Maria	10000	2000
Anna	3000	1500
Filippo	3000	1000
Luigi	5000	1800
Franco	5000	2000
Maria	6000	2000
Andrea	10000	5000
Anna	5000	1000

88

## Esempio proiezione

- Si può scrivere ad esempio

$$- \pi_{\text{Cliente, Credito-Spesa}}(\text{Conto})$$

- ed ottenere il seguente risultato

89

	Cliente	Credito-Spesa
	Andrea	5000
	Andrea	3500

90

## Estensioni (3)

- Funzioni aggregate
  - Si possono usare nelle espressioni alcuni nomi di funzioni (operatori) che si applicano a (multi)insiemi e producono un valore come risultato

91

## Operatori aggregati

- sum, count, min, max
  - $\text{sum}_{\text{Spese}}(\text{Conto})$
  - $\text{count}_{\text{Cliente}}(\text{Conto})$
  - $\text{max}_{\text{Credito}}(\text{Conto})$
  - $\text{count-distinct}_{\text{Cliente}}(\text{Conto})$

92

## Raggruppamento

- Si possono raggruppare gli elementi di una relazione usando un'operatore apposito

$\text{cliente} G_{\text{sum}(\text{Credito})}(\text{Conto})$

- Cliente è l'attributo su cui si fa il raggruppamento, sum è la funzione aggregata che si applica all'attributo Credito, Conto è la relazione su cui si applica il tutto.
- Si possono avere più attributi a sinistra e più funzioni a destra di  $G$

93

## Un altro operatore derivato: divisione

- Vogliamo trovare i nomi dei clienti che hanno un conto corrente in tutte le filiali di banca di Pisa.
- Le relazioni sono
  - Branch(branch\_name, branch\_name, branch\_city)
  - Account(branch\_name, bank\_name, account\_number, branch\_city)
  - Depositor(account\_number, customer\_name)

$$\Pi_{\text{CN,BN}} (\text{depositor} \triangleright \triangleleft \text{account}) \div \Pi_{\text{BN}} (\sigma_{\text{BC}='Pisa'} (\text{branch}))$$

94

## Cosa si intende

$\Pi_{CN,BN} (\text{depositor} \bowtie \text{account}) \div$

$\Pi_{BN} (\sigma_{BC='Pisa'} (\text{branch}))$

• Equivale a

$\Pi_{CN} (\text{depositor} \bowtie \text{account}) -$

$\Pi_{CN} ((\Pi_{CN} (\text{depositor} \bowtie \text{account}) \bowtie$

$\Pi_{BN} (\sigma_{BC='Pisa'} (\text{branch})) -$

$\Pi_{CN,BN} (\text{depositor} \bowtie \text{account}))$

95

$\Pi_{CN} (\text{depositor} \bowtie \text{account}) \times$

$\Pi_{BN} (\sigma_{BC='Pisa'} (\text{branch}))$

Si combinano tutti i clienti presenti nel data base con le filiali di Pisa; togliendo da questo insieme le coppie (cliente, filiale) presenti nel data base, cioè

$\Pi_{CN,BN} (\text{depositor} \bowtie \text{account})$

Restano i clienti che hanno un conto in una filiale a Pisa, ma non in tutte.

Togliendo dai clienti del data base i clienti ottenuti, restano i clienti che hanno un conto in tutte le filiali di Pisa.

96



## Definizione

- Siano  $r(R)$  e  $s(S)$  relazioni con  $R \supseteq S$ ,  $r \div s$  è una relazione su  $R-S$ ; una tupla  $t \in r \div s$  iff
  - $t \in \Pi_{R-S}(r)$
  - $\forall t' \in s, \exists t'' \in r$  tale che
    - $t'[S] = t''[S]$  e
    - $t'[R-S] = t$

97

## Calcolo relazionale

- Una famiglia di linguaggi dichiarativi, basati sul calcolo dei predicati del primo ordine
- Diverse versioni:
  - calcolo relazionale su domini
  - calcolo su ennuple con dichiarazioni di range

98

## Calcolo su domini

$\{ A1: x1, \dots, An: xn \mid f \}$

- $A_i$  sono nomi di attributi
- $x_i$  sono nomi di variabili
- La lista di coppie  $A_i : x_i$  viene detta target list (descrive il risultato)
- $f$  è una formula
  - Formule atomiche sono  $R(A1: x1, \dots, An: xn)$ , che è vera sui valori di  $x1 \dots xn$  che formano una tpla di  $R$ , e  $x_i \vartheta x_j$ , che è vera sui valori di  $x_i$  e  $x_j$  che soddisfano  $\vartheta$

99

## Calcolo su tuple con dichiarazione di range

$\{ x1.Z1, \dots, xn.Zn \mid x_i(R1), \dots, x_j(Rm) \mid f \}$

- $x1.Z1, \dots, xn.Zn$  è la target list
- $x_i(R1), \dots, x_j(Rm)$  è la range list (dice il campo di variabilità delle variabili)
- $f$  è una formula, con formule atomiche del tipo  $x_i.Z_i \vartheta x_j.Z_j$ , ad esempio

100

## Base di dati per gli esempi

Impiegati(Matricola, Nome, Età,  
Stipendio)

Supervisione(Capo, Impiegato)

101

## Esempio 1a

- Trovare gli impiegati che guadagnano più di 40 milioni

{ Matricola: m, Nome: n, Età: e, Stipendio:  
s |  
Impiegati (Matricola: m, Nome: n, Età: e,  
Stipendio: s)  $\wedge$  s > 40 }

102

## Esempio 1b

- Trovare gli impiegati che guadagnano più di 40 milioni

$$\{ i.* \mid i(\text{Impiegati}) \mid i.\text{Stipendio} > 40 \}$$

103

## Esempio 2a

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

$$\begin{aligned} & \{ \text{Matricola: } m, \text{ Nome: } n \mid \\ & \text{Impiegati (Matricola: } m, \text{ Nome: } n, \text{ Et\`a: } e, \\ & \text{Stipendio: } s) \wedge s > 40 \} \\ & \text{oppure} \\ & \{ \text{Matricola: } m, \text{ Nome: } n \mid \\ & \exists e, s (\text{Impiegati (Matricola: } m, \text{ Nome: } n, \text{ Et\`a: } e, \\ & \text{Stipendio: } s) \wedge s > 40) \} \end{aligned}$$

104

## Esempio 2b

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

$\{ i.(\text{Matricola}, \text{Nome}) \mid i(\text{Impiegati}) \mid i.\text{Stipendio} > 40 \}$

105

## Quantificatori esistenziali e universali

- Per interrogazioni più complesse, che in algebra ad esempio richiedevano una differenza, servono altri strumenti
- $\exists, \forall$ 
  - Sono intercambiabili
  - $\exists x(f) = \neg(\forall x(\neg(f)))$
  - $\forall x(f) = \neg(\exists x(\neg(f)))$

106

## Quantificatori esistenziali e universali

- Trovare matricola e nome dei capi i cui impiegati guadagnano tutti più di 40 milioni.

$$\{ \text{Matricola: } c, \text{ Nome: } n \mid$$

$$\text{Impiegati}(\text{Matricola: } c, \text{ Nome: } n, \text{ Et\`a: } e, \text{ Stipendio: } s)$$

$$\wedge$$

$$\text{Supervisione}(\text{Capo: } c, \text{ Impiegato: } m) \wedge$$

$$\forall m' (\forall n' (\forall e' (\forall s' ($$

$$\neg (\text{Impiegati}(\text{Matricola: } m', \text{ Nome: } n', \text{ Et\`a: } e', \text{ Stipendio: } s'))$$

$$\wedge$$

$$\text{Supervisione}(\text{Capo: } c, \text{ Impiegato: } m')) \vee s' > 40))))\}$$

107

## Quantificatori esistenziali e universali

- Trovare matricola e nome dei capi i cui impiegati guadagnano tutti più di 40 milioni.

$$\{ \text{Matricola: } c, \text{ Nome: } n \mid$$

$$\text{Impiegati}(\text{Matricola: } c, \text{ Nome: } n, \text{ Et\`a: } e, \text{ Stipendio: } s)$$

$$\wedge$$

$$\text{Supervisione}(\text{Capo: } c, \text{ Impiegato: } m) \wedge$$

$$\neg \exists m' (\exists n' (\exists e' (\exists s' ($$

$$\text{Impiegati}(\text{Matricola: } m', \text{ Nome: } n', \text{ Et\`a: } e', \text{ Stipendio: } s'))$$

$$\wedge$$

$$\text{Supervisione}(\text{Capo: } c, \text{ Impiegato: } m') \wedge s' \leq 40 \wedge$$

$$m \neq m'))))\}$$

108

## Quantificatori esistenziali e universali

$$\begin{aligned}
 & \{ \text{Matricola: } c, \text{ Nome: } n \mid \\
 & \text{Impiegati}(\text{Matricola: } c, \text{ Nome: } n, \text{ Et\`a: } e, \text{ Stipendio: } s) \\
 & \quad \wedge \\
 & \quad \text{Supervisione}(\text{Capo: } c, \text{ Impiegato: } m) \wedge \\
 & \quad \neg \exists m' (\exists n' (\exists e' (\exists s' ( \\
 & \quad \text{Impiegati}(\text{Matr: } m', \text{ Nome: } n', \text{ Et\`a: } e', \text{ Stip: } s') \wedge \\
 & \quad \text{Supervisione}(\text{Capo: } c, \text{ Impiegato: } m') \wedge s' \leq 40)))))) \} \\
 & \{ i.(\text{Matricola}, \text{Nome}) \mid s(\text{Supervisione}), i(\text{Impiegati}) \mid \\
 & i.\text{Matricola} = s.\text{Capo} \wedge \neg (\exists i' (\text{Impiegati})(\exists s' (\text{Supervisione}) \\
 & (s.\text{Capo} = s'.\text{Capo} \wedge s'.\text{Impiegato} = i'.\text{Matricola} \wedge \\
 & i'.\text{Stipendio} \leq 40)))) \}
 \end{aligned}$$

109

## Calcolo su domini, discussione

- Pregi:
  - dichiaratività
- Difetti:
  - "verbosità": tante variabili!
    - Le variabili del calcolo dei domini rappresentano singoli valori
    - Nel calcolo su tple rappresentano tple
  - possibilità di scrivere espressioni senza senso (dipendenti dal dominio)
    - $\{ A:x, B:y \mid R(A:x) \wedge y=y \}$
  - nell'algebra tutte le espressioni hanno un senso (indipendenti dal dominio)

110

## Calcolo su tuple, discussione

- Il calcolo su tuple con dichiarazioni di range non permette di esprimere alcune interrogazioni importanti, in particolare le unioni:

$$R_1(AB) \cup R_2(AB)$$

- Ogni variabile ha un solo range nel risultato, mentre vorremmo tple sia della prima relazione che della seconda
- Nota: intersezione e differenza sono esprimibili
- Per questa ragione SQL (che è basato su questo calcolo) prevede un operatore esplicito di unione, ma non tutte le versioni prevedono intersezione e differenza

111

## Calcolo e algebra

- Calcolo e algebra sono "equivalenti"
  - per ogni espressione del calcolo relazionale che sia indipendente dal dominio esiste un'espressione dell'algebra relazionale equivalente a essa
  - per ogni espressione dell'algebra relazionale esiste un'espressione del calcolo relazionale equivalente a essa (e di conseguenza indipendente dal dominio)

112



## Calcolo e algebra: limiti

- l'insieme di interrogazioni esprimibili è significativo
- Ci sono però interrogazioni interessanti non esprimibili, ad es.
  - interrogazioni inerentemente ricorsive, come la chiusura transitiva

113

## Si consideri il seguente schema di base di dati

- Film( CodiceFilm, Titolo, CodiceRegista, Anno)
- Produzione ( CasaProduzione, Nazionalità, CodiceFilm, Costo, Incasso1annoSala)
- Artista ( CodiceAttore, Cognome, Nome, Sesso, DataDiNascita, Nazionalità)
- Interpretazione ( CodiceFilm, CodiceAttore, Personaggio, SessoPersonaggio)
- Regista ( CodiceRegista, Cognome, Nome, Sesso, DataDiNascita, Nazionalità)
- Noleggio ( CodiceFilm, Incasso1annoVideo, Incasso1annoDVD)

114

## Formulare in algebra relazionale le seguenti interrogazioni (1)

- nomi e cognomi dei registi che hanno diretto film che hanno incassato il primo anno di uscita meno nelle sale che per il noleggio di DVD

$$\pi_{N,C} \left( \pi_{N,C,CF} \left( \pi_{N,C,CR} (\text{Regista}) \bowtie \pi_{CF,CR} (\text{Film}) \right) \right. \\ \left. \bowtie \pi_{CF} \left( \sigma_{\text{Inc1sala} < \text{Inc1DVD}} \left( \pi_{\text{Inc1sala},CF} (\text{Produzione}) \right) \right. \right. \\ \left. \left. \bowtie \pi_{\text{Inc1DVD},CF} (\text{Noleggio}) \right) \right) \\ \left. \right)$$

115

## calcolo dei domini

- {Nome: n, Cognome: c |  
Regista(CodiceRegista: cr, Cognome:c,  
Nome:n,...)  $\wedge$  Film( CodiceFilm: cf,...  
CodiceRegista: cr,...)
- $\wedge$  Produzione(.. CodiceFilm:cf,  
...Incasso1annoSala:is)  $\wedge$  Noleggio(Codice  
Film:  
cf,...Incasso1annoDVD:idvd)  $\wedge$  (is < idvd)  
}

116

## Formulare in algebra relazionale le seguenti interrogazioni (2)

- i titoli dei film i cui attori sono tutti dello stesso sesso

$$1. \pi_{\text{Titolo}}(\text{Film}) - \pi_{\text{Titolo}}(\text{Film} \bowtie \sigma_{\text{Sesso} < > S'}($$

$$\pi_{CF, \text{Sesso}}(\text{Artista} \bowtie \text{Interpretazione})$$

$$\bowtie \rho_{S' \leftarrow \text{Sesso}}($$

$$\pi_{CF, \text{Sesso}}(\text{Artista} \bowtie \text{Interpretazione})))$$

$$2. (\pi_{\text{Titolo}}(\text{Film}) - \pi_{\text{Titolo}}(\text{Film} \bowtie ($$

$$\pi_{CF}(\sigma_{\text{Sesso} = 'M'}(\text{Artista}) \bowtie \text{Interpretazione}))) \cup$$

$$(\pi_{\text{Titolo}}(\text{Film}) - \pi_{\text{Titolo}}(\text{Film} \bowtie ($$

$$\pi_{CF}(\sigma_{\text{Sesso} = 'F'}(\text{Artista}) \bowtie \text{Interpretazione})))$$

117

## calcolo dei domini

- $\{ \text{Titolo: } t \mid \text{Film}(\text{CodiceFilm: } cf, \text{Titolo: } t, \dots) \wedge \neg \exists$   
 $ca1, c1, n1, s1, nz1, dn1, ca2, c2, n2, s2, nz2, dn2, p1, p2, sp1, sp2$   
 $(\text{Artista}(\text{CodiceAttore: } ca1, \text{Cognome: } c1, \text{Nome: } n1,$   
 $\text{Sesso: } s1, \text{DataDiNascita: } dn1,$   
 $\text{Nazionalità: } nz1) \wedge (\text{Artista}(\text{CodiceAttore: } ca2,$   
 $\text{Cognome: } c2, \text{Nome: } n2, \text{Sesso: } s2, \text{DataDiNascita: } dn2,$   
 $\text{Nazionalità: } nz2) \wedge \text{Interpretazione}(\text{CodiceFilm: } cf,$   
 $\text{CodiceAttore: } ca1, \text{Personaggio: } p1,$   
 $\text{SessoPersonaggio: } sp1) \wedge \text{Interpretazione}(\text{CodiceFilm: } c$   
 $f, \text{CodiceAttore: } ca2, \text{Personaggio: } p2,$   
 $\text{SessoPersonaggio: } sp2) \wedge s1 \neq s2) \}$

118

## Formulare in algebra relazionale le seguenti interrogazioni (3)

- i titoli di film con solamente attori donna che abbiano incassato in sala più del proprio costo

$$\pi_{\text{Titolo}} (\pi_{\text{CF}, \text{Titolo}}(\text{Film}) \bowtie \pi_{\text{CF}} (\sigma_{\text{Inc1S} > \text{Costo}} (\text{Produzione})) \bowtie \pi_{\text{CF}} (\text{Film}) - \pi_{\text{CF}} (\pi_{\text{CA}} (\sigma_{\text{Sesso} = 'M'} (\pi_{\text{CA}, \text{Sesso}}(\text{Artista}))) \bowtie \pi_{\text{CA}, \text{CF}}(\text{Interpretazione})))$$