Basi Dati

Algebra Relazionale

a.a. 2021/2022

Prof.ssa G. Tortora

L'Algebra Relazionale

- Abbiamo visto i concetti per definire la struttura ed i vincoli di un database nel modello relazionale.
- Realizzato lo scheletro di un db, abbiamo bisogno di un'insieme di operazione per manipolare i dati.

- **Algebra relazionale: collezione di operazioni usate per manipolare intere relazioni.**

Perché “Algebra”

- Proprietà di un’algebra, in senso matematico:
  - È basata su operatori e domini dei valori.
  - Gli operatori mappano gli argomenti da un dominio ad un altro.
  - Quindi un’espressione che coinvolge operatori ed argomenti genera un nuovo valore nel dominio.

L'algebra "relazionale"

- Applichiamo la definizione di algebra al modello relazionale:
  - Dominio: le relazioni;
  - Operatori:
    - Operazioni su insiemi, ereditate dalla teoria matematica degli insiemi: *Unione*, *Intersezione*, *Differenza* e *Prodotto Cartesiano*.
    - Operazioni specificatamente disegnate per database relazionali: *Select*, *Project* e *Join*.
  - Il risultato dell'applicazione di un operatore su una relazione è ancora una relazione.

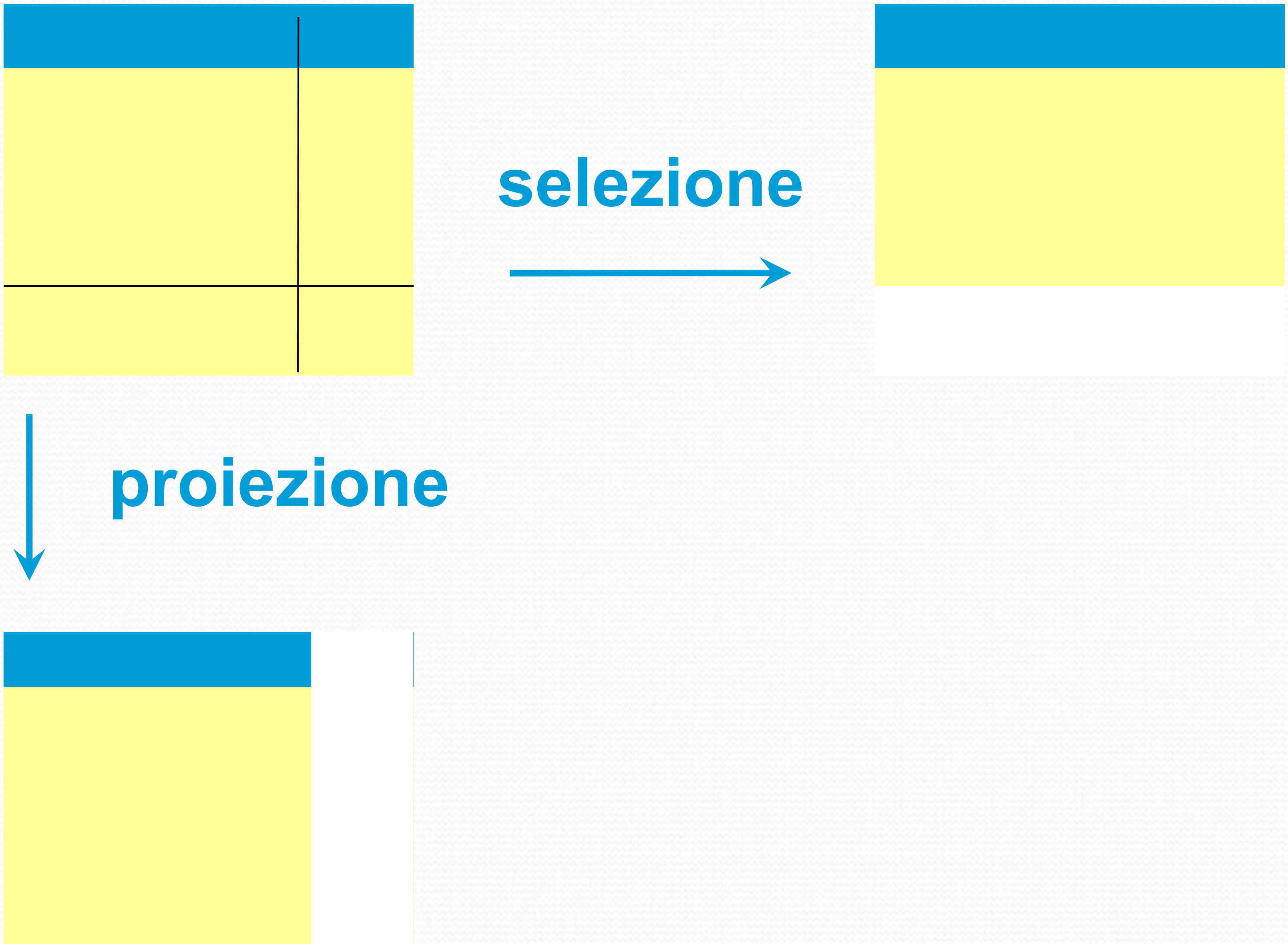
Selezione e proiezione

- operatori "ortogonali"

- selezione:
  - decomposizione orizzontale

- proiezione:
  - decomposizione verticale

(animazione)



L'operatore Select: σ

- Usato per selezionare un sottoinsieme di tuple in una relazione che soddisfa una condizione di selezione.
- Si indica con σ.
- ***Esempi:***
  - $\sigma_{dno=4}$ (Employee) restituisce il sottoinsieme degli impiegati che lavora nel dipartimento numero 4.
  - $\sigma_{salary>30000}$(Employee) restituisce il sottoinsieme degli impiegati con salario > 30000 $.

L'operatore Select (2)

- Sintassi:

  $\sigma_{\text{<selection condition>}}$ (<relazione>)

- La condizione di selezione è un'espressione booleana formata da clausole della forma:
  - <nome_attributo> ***op_confronto*** <valore costante>

  oppure
  - <nome_attributo> ***op_confronto*** <nome_attributo>

  eventualmente concatenate con operatori logici.
- ***op_confronto*** è uno degli operatori $\{=, \neq, <, >, \leq, \geq\}$

- ***Esempio:***
  - $\sigma_{(\text{dno=4 and Salary>25000 or dno=5 and Salary>30000 })}$(Employee)

L'operatore Select (3)

- La condizione di selezione è valutata per ogni tupla individualmente: se è vera, la tupla è inserita nella relazione risultante.
- Il grado della relazione risultante dopo un'operazione di select è uguale a quello della relazione di partenza.
- Il numero di tuple risultanti t_r è minore o uguale di quelle di partenza t_p:

$$t_r \leq t_p$$

- Il rapporto t_r/t_p è detto selettività della condizione.

Proprietà della Select

- L'operatore Select è unario.

- L'operatore di Select è commutativo:

$$\sigma_{<cond_1>}(\sigma_{<cond_2>}(R)) = \sigma_{<cond_2>}(\sigma_{<cond_1>}(R))$$

L’operatore Project: π

- Usato per selezionare un sottoinsieme delle colonne di una relazione.
- Sintassi:

$$\pi_{<attribute\_list>} (<relazione>)$$

- La relazione risultante ha gli attributi specificati nella <attribute_list>, nello stesso ordine in cui appaiono nella lista.
- Il grado della relazione risultante da un’operazione di PROJECT è uguale al numero di attributi specificati nella <attribute_list>.

L'operatore Project (2)

- Se nella <attribute_list> non è presente una chiave candidata, si potrebbero avere delle tuple duplicate: la PROJECT le rimuove implicitamente.
- Il numero t_r di tuple risultanti è minore o uguale del numero t_p di tuple di partenza:
  - Se la lista di attributi include una chiave candidata della relazione, sarà $t_r = t_p$.
- $\pi_{<list1>}(\pi_{<list2>}(R)) = \pi_{<list1>}(R)$ se <list2> contiene gli attributi presenti in <list1>; altrimenti la parte sinistra non è corretta

- La commutatività non vale per la PROJECT.

Selezione e proiezione

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

Matricola	Cognome
7309	Rossi
5998	Neri
5698	Neri

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

Sequenze di Operazioni

- Per applicare più operazioni una dopo l'altra si può scrivere un'unica espressione dell'algebra relazionale.
- ***Esempio:***
  - Trovare *nome*, *cognome* e *salario* dei dipendenti che lavorano nel dipartimento n° 5:
    $\pi_{< FNAME,\ LNAME,\ SALARY>}(\sigma_{dno=5}(EMPLOYEE))$

- Alternativamente si possono creare risultati intermedi:
  - DEPS_EMPS= $\sigma_{dno=5}(EMPLOYEE)$
  - RESULT=$\pi_{< FNAME,\ LNAME,\ SALARY>}(DEPS\_EMPS)$

L’operazione RENAME

- Per rinominare gli attributi in una relazione che risulta dall’algebra relazionale, semplicemente listiamo i nuovi nomi di attributi in parametri:
  - TEMP= $\sigma_{dno=5}$(EMPLOYEE)
  - R(FirstName, LastName, Salary) =

    $= \pi_{< FNAME,\ LNAME,\ SALARY>}(\ TEMP)$

- È un operatore unario.

Operazioni Insiemistiche

- Una relazione è un insieme di tuple, quindi possiamo applicare le classiche operazioni insiemistiche.
- Il risultato nella combinazione di due relazioni per mezzo di un operazione su insieme è una nuova relazione.
- Per poter applicare un’operazione insiemistica a due relazioni, queste devono avere la stessa struttura, ovvero essere union compatibili…

UNION COMPATIBILITY

- Due relazioni $R(A_1, A_2, \ldots, A_n)$ e $S(B_1, B_2, \ldots, B_n)$ sono union compatibili se hanno lo stesso grado e $dom(A_i) = dom(B_i)$ per $1 \leq i \leq n$.

Operazioni ammissibili

- Su due relazioni R ed S *union compatibili*, è possibile effettuare :
  - Unione
  - Intersezione
  - Differenza

Esempio di Unione

- Trovare il SSN di tutti gli impiegati che lavorano o nel dipartimento n° 5 o supervedono direttamente un impiegato che lavora nel dipartimento n° 5:
  - DEPS_EMPS = $\sigma_{dno=5}$(EMPLOYEE)
  - Result1 = π_{SSN}(DEPS_EMPS)
  - Result2=$\pi_{SUPERSSN}$(DEPS_EMPS)
  - RESULT = Result1 $\cup$ Result2

Result1
123456789
333444555
666888444
453453453

Result2
333444555
888666555

Result
123456789
333444555
666888444
453453453
888666555

Operazioni

- Unione:
  - Il risultato di questa operazione, $R \cup S$, è la relazione che include tutte le tuple che sono in R o in S, oppure in R ed S. Le tuple duplicate sono eliminate.
- Intersezione:
  - Il risultato di questa operazione, $R \cap S$, è la relazione che include tutte le tuple che sono sia in R che in S.
- Differenza:
  - Il risultato di questa operazione, R-S, è la relazione che include tutte le tuple che sono in R ma non in S.

- Si adotta la convenzione che il risultato ha gli stessi nomi di attributi della prima relazione.

Operazioni: *Esempi*

Date due relazioni S ed I

STUDENT	FN	LN
	Susan	Yao
	Ramesh	Shah
	Johnny	Kohler
	Barbara	Jones
	Amy	Ford
	Jimmy	Wang
	Ernest	Gilbert

INSTRUCTOR	FNANE	LNAME
	John	Smith
	Ricardo	Browne
	Susan	Yao
	Francis	Johnson
	Ramesh	Shah

FN	LN
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert
John	Smith
Ricardo	Browne
Francis	Johnson

$S \cup I$

FN	LN
Susan	Yao
Ramesh	Shah

$S \cap I$

FN	LN
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

$S - I$

FN	LN
John	Smith
Ricardo	Browne
Francis	Johnson

$I - S$

Proprietà delle operazioni

- Unione ed intersezione sono commutative, associative e possono essere applicate ad un numero qualsiasi di relazioni
  - $R \cup S = S \cup R$
  - $R \cap S = S \cap R$
  - $R \cup (S \cup T) = (R \cup S) \cup T$
  - $R \cap (S \cap T) = (R \cap S) \cap T$
- La differenza non è commutativa.
  - In generale $R - S \neq S - R$.

Prodotto cartesiano (Cross Product o Cross Join)

- Non è necessario che le relazioni siano union compatibili
  - $R(A_1, A_2, \ldots, A_n) \times S(B_1, B_2, \ldots, B_m) = Q(A_1, A_2, \ldots, A_n, B_1, B_2, \ldots, B_m)$
  - In Q si ha una tupla per ogni combinazione di una da R ed una da S.
  - Se R contiene $\mathbf{n_r}$ tuple ed S contiene $\mathbf{n_s}$ tuple, allora $R \times S$ contiene $\mathbf{n_r} \cdot \mathbf{n_s}$ tuple.

- In caso di attributi con lo stesso nome nelle due relazioni, si deve effettuare il rename di uno dei due.

Prodotto cartesiano: *Esempio*

- Vogliamo trovare per ogni impiegato di sesso femminile, una lista dei suoi familiari a carico:
  - FEMALE_EMPS= $\sigma_{Sex=\text{"F"}}$(EMPLOYEE)
  - EMPNAMES = $\pi_{< FNAME, LNAME, SSN>}$(FEMALE_EMPS)
  - EMP_DEPENDENTS = EMPNAMES x DEPENDENTS
  - ACTUAL_DEPENDENTS = $\sigma_{SSN=ESSN}$(EMP_DEPENDENTS)
  - RESULT = $\pi_{< FNAME, LNAME, DEPENDENT\_NAME>}$ (ACTUAL_DEPENDENTS)

Prodotto cartesiano: *Esempio* (2)

FEMALE_EMPS	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	Alicia	J	Zelaya	999887777	1968-07-19	3321 Castle,Spring,TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	1941-06-20	291 Berry,Bellaire,TX	F	43000	888665555	4
	Joyce	A	English	453453453	1972-07-31	5631 Rice,Houston,TX	F	25000	333445555	5

EMPNAMES	FNAME	LNAME	SSN
	Alicia	Zelaya	999887777
	Jennifer	Wallace	987654321
	Joyce	English	453453453

EMP_DEPENDENTS	FNAME	LNAME	SSN	ESSN	DEPENDENT_NAME	SEX	BDATE	• • •
	Alicia	Zelaya	999887777	333445555	Alice	F	1986-04-05	• • •
	Alicia	Zelaya	999887777	333445555	Theodore	M	1983-10-25	• • •
	Alicia	Zelaya	999887777	333445555	Joy	F	1958-05-03	• • •
	Alicia	Zelaya	999887777	987654321	Abner	M	1942-02-28	• • •
	Alicia	Zelaya	999887777	123456789	Michael	M	1988-01-04	• • •
	Alicia	Zelaya	999887777	123456789	Alice	F	1988-12-30	• • •
	Alicia	Zelaya	999887777	123456789	Elizabeth	F	1967-05-05	• • •
	Jennifer	Wallace	987654321	333445555	Alice	F	1986-04-05	• • •
	Jennifer	Wallace	987654321	333445555	Theodore	M	1983-10-25	• • •
	Jennifer	Wallace	987654321	333445555	Joy	F	1958-05-03	• • •
	Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	• • •
	Jennifer	Wallace	987654321	123456789	Michael	M	1988-01-04	• • •
	Jennifer	Wallace	987654321	123456789	Alice	F	1988-12-30	• • •
	Jennifer	Wallace	987654321	123456789	Elizabeth	F	1967-05-05	• • •
	Joyce	English	453453453	333445555	Alice	F	1986-04-05	• • •
	Joyce	English	453453453	333445555	Theodore	M	1983-10-25	• • •
	Joyce	English	453453453	333445555	Joy	F	1958-05-03	• • •
	Joyce	English	453453453	987654321	Abner	M	1942-02-28	• • •
	Joyce	English	453453453	123456789	Michael	M	1988-01-04	• • •
	Joyce	English	453453453	123456789	Alice	F	1988-12-30	• • •
	Joyce	English	453453453	123456789	Elizabeth	F	1967-05-05	• • •

ACTUAL_DEPENDENTS	FNAME	LNAME	SSN	ESSN	DEPENDENT_NAME	SEX	BDATE
	Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28

RESULT	FNAME	LNAME	DEPENDENT_NAME
	Jennifer	Wallace	Abner

Prodotto cartesiano

- Operazione binaria

- Contiene sempre un numero di *n*-ple pari al prodotto delle cardinalità degli operandi (le *n*-ple sono tutte combinabili).

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Codice	Capo
A	Mori
B	Bruni

Impiegati × 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

Le operazioni di Join

Join

- La sequenza di operazioni appena vista, riassumibile in $\sigma_{join\text{-}condition}\,(R \times S)$ è abbastanza frequente: per questo è stata creata un'operazione speciale, chiamata JOIN.

- La JOIN, denotata con $\bowtie$, è usata per combinare tuple relate in una sola tupla.
  - $\sigma_{join\text{-}condition}\,(R \times S)$ diventa $R \bowtie_{join\text{-}condition} S$

Join: *Esempio*

- Supponiamo di voler trovare il nome del manager di ciascun dipartimento:
  - Combiniamo ogni tupla dipartimento con la tupla impiegato il cui SSN fa match col valore MGRSSN nella tupla dipartimento.
- DEPT_MGR= department $\bowtie_{MGRSSN=SSN}$ EMPLOYEE
- RESULT= $\pi_{DNAME,\ LNAME,\ FNAME}$(DEPT_MGR)

DEPT_MGR	DNAME	DNUMBER	MGRSSN	• • •	FNAME	MINIT	LNAME	SSN	• • •
	Research	5	333445555	• • •	Franklin	T	Wong	333445555	• • •
	Administration	4	987654321	• • •	Jennifer	S	Wallace	987654321	• • •
	Headquarters	1	888665555	• • •	James	E	Borg	888665555	• • •

Join: *Esempio su prodotto cartesiano*

- L’esempio precedente sul prodotto cartesiano può essere risolto rimpiazzando le operazioni:

  - EMP_DEPENDENTS = EMPNAMES x DEPENDENTS
  - ACTUAL_DEPENDENTS = $\sigma_{SSN=ESSN}$(EMP_DEPENDENTS)

  Con:

  - ACTUAL_DEPENDENTS = EMPNAMES $\bowtie_{SSN=ESSN}$ DEPENDENTS

Nozioni sulla Join

- Sintassi :
  Date le relazioni $R(A_1, A_2, \ldots, A_n)$ e $S(B_1, B_2, \ldots, B_m)$
  Scriveremo $R \bowtie_{\text{<join\_condition>}} S$

- Il risultato della join ha n+m attributi:
  $Q(A_1, A_2, \ldots, A_n, B_1, B_2, \ldots, B_m)$

- La condizione della join è della forma:
  <condition> AND <condition> AND … AND <condition>
  dove ogni condizione è della forma $A_i\ \theta\ B_j$,
  con $A_i \in R$, $B_j \in S$, $dom(A_i) = dom(B_j)$ e $\theta \in \{=,,\neq,<,>,\leq,\geq\}$

- Tuple con attributi di join **null** non appaiono nel risultato.

Tipi di Join

- Una join con θ generica è detta THETA JOIN
- Il join più comune ha come θ l'operatore di uguaglianza ed è detto EQUIJOIN:
  - Nei risultati dell'EQUIJOIN abbiamo sempre una coppia di valori di attributi identici.

- Poiché tale ripetizione è superflua, esiste una nuova operazione, chiamata NATURAL JOIN e denotata da *, che elimina il secondo attributo superfluo.
  - È in sostanza una EQUIJOIN con la rimozione degli attributi con valori uguali superflui

(animazione)

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Codice	Capo
A	Mori
B	Bruni

Impiegati $\bowtie_{Reparto=Codice}$ Reparti

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

Natural Join

- La definizione standard di Natural join richiede che i due attributi di join (per ogni coppia) abbiano lo stesso nome. Se ciò non vale, occorre prima un'operazione di renaming.
  - DEPT = $\sigma_{(\text{DNAME, DNUM, MGRSSN, MGRSTARTDATE})}$ (DEPARTMENT)
  - PROJ_DEPT = PROJECT * DEPT
    - L'attributo DNUM è detto "attributo di join"

PROJ_DEPT	PNAME	PNUMBER	PLOCATION	DNUM	DNAME	MGRSSN	MGRSTARTDATE
	ProductX	1	Bellaire	5	Research	333445555	1988-05-22
	ProductY	2	Sugarland	5	Research	333445555	1988-05-22
	ProductZ	3	Houston	5	Research	333445555	1988-05-22
	Computerization	10	Stafford	4	Administration	987654321	1995-01-01
	Reorganization	20	Houston	1	Headquarters	888665555	1981-06-19
	Newbenefits	30	Stafford	4	Administration	987654321	1995-01-01

Natural Join (2)

- Qualora gli attributi di join abbiano lo stesso nome, non è necessario il renaming.
- ***Esempio:***
  - DEPT_LOCS = DEPARTMENT * DEPT_LOCATIONS

DEPT_LOCS	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE	LOCATION
	Headquarters	1	888665555	1981-06-19	Houston
	Administration	4	987654321	1995-01-01	Stafford
	Research	5	333445555	1988-05-22	Bellaire
	Research	5	333445555	1988-05-22	Sugarland
	Research	5	333445555	1988-05-22	Houston

- In generale il Natural Join verifica l'uguaglianza di tutte le coppie di attributi.

Natural Join (3)

Definizione più generale:

- $Q = R^{*}(<list_1>)(<list_2>)S$
  - $<list_1>$=lista di attributi da R
  - $<list_2>$=lista di attributi da S

- Le liste sono usate per le uguaglianze tra coppie di attributi corrispondenti;

  le condizioni sono combinate in AND.

  Solo la lista corrispondente ad attributi della prima relazione, $<list_1>$ è mantenuta nel risultato.

Natural Join (4)

- Se nessuna combinazione di tuple soddisfa la condizione di join, la relazione risultante avrà zero tuple:
  - Date le relazioni R(con $\mathbf{n_r}$ tuple) ed S (con $\mathbf{n_s}$ tuple),
    - $\mathbf{n_q}$ $Q = R \bowtie_{\text{<joincond>}} S$ avrà da **0** a $\mathbf{n_r} \cdot \mathbf{n_s}$ tuple.
    - $\mathbf{n_q}/(\mathbf{n_r} \cdot \mathbf{n_s})$ è detta join selectivity.

- Se non esiste alcuna <joincondition>, tutte le tuple compariranno nella relazione risultante, ed il join diventa un prodotto cartesiano, detto CROSS JOIN.

(animazione)

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
A	Mori
B	Bruni

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

- Ogni *n*-pla contribuisce al risultato:
  - join completo.

Un join non completo

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
B	Mori
C	Bruni

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

Un join vuoto

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
D	Mori
C	Bruni

Impiegato	Reparto	Capo

Un join completo, con n · m *n*-ple

Impiegato	Reparto
Rossi	B
Neri	B

Reparto	Capo
B	Mori
B	Bruni

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

Cardinalità del join

- Il join di R_1 e R_2 contiene un numero di *n*-ple compreso fra zero e il prodotto di $|R_1|$ e $|R_2|$.

- Se il join coinvolge una chiave di R_2, allora il numero di *n*-ple è compreso fra zero e $|R_1|$.

- Se il join coinvolge una chiave di R_2 e un vincolo di integrità referenziale, allora il numero di *n*-ple è pari a $|R_1|$.

(animazione)

Join, una difficoltà

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparto	Capo
B	Mori
C	Bruni

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

- Alcune *n*-ple non contribuiscono al risultato: vengono "*tagliate fuori*".

Join esterno

- Il join esterno estende, con valori nulli, le *n*-ple che verrebbero tagliate fuori da un join (interno).

- Esiste in tre versioni:
  - sinistro (⟕ o $\rhd\!\lhd_{\text{LEFT}}$),
  - destro (⟖ o $\rhd\!\lhd_{\text{RIGHT}}$),
  - completo (⟗ o $\rhd\!\lhd_{\text{FULL}}$).

Join esterno (2)

- Sinistro: mantiene tutte le *n*-ple *del primo operando*, estendendole con valori nulli, se necessario.
- Destro: … *del secondo operando* …
- Completo: … *di entrambi gli operandi* …

(animazione)

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati $\triangleright\triangleleft_{LEFT}$ Reparti

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

(animazione)

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati $\triangleright\triangleleft_{\text{RIGHT}}$ Reparti

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

(animazione)

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati $\triangleright\triangleleft_{\text{FULL}}$ Reparti

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

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 perché i DBMS cercano di eseguire espressioni equivalenti a quelle date, ma meno "*costose*".

- Trasformazioni di equivalenza.

L’operazione di DIVISIONE

- L’operazione di divisione è indicata con ÷ è un tipo particolare di interrogazione che talvolta si presenta nelle applicazioni di basi di dati.
- Un esempio è “*trova i nomi degli impiegati che lavorano a* ***tutti*** *i progetti su cui lavora ‘John Smith’*”.

- La divisione ÷, di r1 per r2, con r1 su $R_1(X_1X_2)$ e r2 su $R_2(X_2)$, è (il più grande) insieme di tuple con schema X_1 tale che, facendo il prodotto cartesiano con r2, ciò che si ottiene è una relazione contenuta in r1.

Esempio

Voli

Codice	Data
AZ427	21/07/2001
AZ427	23/07/2001
AZ427	24/07/2001
TW056	21/07/2001
TW056	24/07/2001
TW056	25/07/2001

Linee

Codice
AZ427
TW056

Voli ÷ Linee

Data
21/07/2001
24/07/2001

(Voli ÷ Linee) ⋈ Linee

Codice	Data
AZ427	21/07/2001
AZ427	24/07/2001
TW056	21/07/2001
TW056	24/07/2001

La divisione trova le date con voli per tutte le linee

Un insieme completo di operazioni

- Si può provare che $\{\sigma, \pi, \cup, -, \times\}$ è un insieme completo, cioè tutte le altre operazioni possono essere espresse come combinazioni di queste.
- ***Esempi:***
  - Intersezione:
    - $R \cap S = (R \cup S) - ((R-S) \cup (S-R))$
  - Join:
    - $R \bowtie_{<condition>} S = \sigma_{<condition>}(R \times S)$
  - Divisione:
    - $R \div S = T1 \leftarrow \pi_{<attribute\ of\ R\ -\ attribute\ of\ S>}(R)$

      $T2 \leftarrow \pi_{<attribute\ of\ R\ -\ attribute\ of\ S>}((S \times T1) - R)$

      $T \leftarrow T1 - T2$

Algebra relazionale: limiti

- Ci sono però interrogazioni interessanti non esprimibili:
  - Calcolo di valori derivati: possiamo solo estrarre valori, non calcolarne di nuovi; calcoli di interesse:
    - a livello di *n*-pla o di singolo valore (conversioni somme, differenze, etc.);
    - su insiemi di *n*-ple (somme, medie, etc.).
  - Interrogazioni inerentemente ricorsive, come la chiusura transitiva.

Funzioni aggregate e di raggruppamento

- *Non sono presenti nell'algebra relazionale base.*
- Operano su un insieme di dati per restituire come risultato una relazione con un solo valore.
- Sono funzioni applicate a collezioni di valori numerici:
  - SUM, AVERAGE, MAXIMUM, MINIMUM, COUNT.
- *Notazione:*

  $_{\text{<attributi di raggruppamento>}}\mathcal{F}_{\text{<lista funzioni>}}(R)$

  - dove <attributi di raggruppamento> è una lista di attributi della relazione R e raggruppa le tuple presenti nella relazione sulla base dei loro valori,
  - e <lista funzioni> è una lista di coppie <funzione><attributo>:
    - <funzione> è una delle funzioni SUM,…
    - <attributo> è un attributo della relazione R.

Esempio

- $T(N\_D, N\_IMP, STIP\_MEDIO) \leftarrow {}_{N\_D}\mathscr{F}_{COUNT\ SSN,\ AVERAGE\ STIPENDIO}(IMPIEGATO)$
- $R \leftarrow \sigma_{N\_D,\ N\_IMP, STIP\_MEDIO}(T)$

R	N_D	N_IMP	STIP_MEDIO
	5	4	33250
	4	3	31000
	1	1	55000

- $S \leftarrow \mathscr{F}_{COUNT\ SSN,\ AVERAGE\ STIPENDIO}(IMPIEGATO)$

S	COUNT_SSN	AVERAGE_STIPENDIO
	8	35125