Algebra Relazionale

L'algebra relazionale è un linguaggio procedurale, mentre l'SQL è un linguaggio dichiarativo.
L'algebra relazionale è composta da 5 operatori di base, dai quali si possono definire altri operatori, detti derivati.

Studente

Matr	*Nome*	*Città*	*ACorso*
M1	Marco Quaranta	SA	1
M2	Giacomo Tedesco	PA	2
M3	Maria Mei	BO	1
M4	Ugo Rossi	MO	2
M5	Sara Neri	MO	2
M6	Agata Verdi	MI	1

Corso

Codice	*Nome*	*CodDoc*
C1	Fisica 1	D1
C2	Analisi Matematica 1	D2
C3	Fisica 2	D1
C4	Analisi Matematica 2	D2
C5	Meccanica Razionale	D4

Docente

Codice	*Cf*	*Città*
D1	Cf1	MO
D2	Cf2	BO
D3	Cf3	MO
D4	Cf4	FI

Frequenza

Matr	*CodCor*
M1	C1
M1	C3
M2	C1
M2	C2
M3	C1
M3	C2
M4	C3
M4	C4

NB: gli esempi variano leggermente da quelli del libro per via della tabella "Frequenza"

Selezione: seleziona tutte le tuple che hanno una determinata caratteristica.

dati R(X) e r, si definisce $\sigma_F(r) = \{t \mid t \in r, F(t) = true\}$

ES: studenti al secondo anno di corso --> $\sigma_{ACorso=2}(Studente)$

Matr	*Nome*	*Città*	*ACorso*
M2	Giacomo Tedesco	PA	2
M4	Ugo Rossi	MO	2
M5	Sara Neri	MO	2

Proiezione: prende solamente gli attributi richiesti all'interno di ogni tupla

dati R(X) e r, si definisce $\pi_Y(r) = \{t[Y] \mid t \in r\}$

ES: Città e anno di corso degli studenti --> $\pi_{Città,\ ACorso}(Studente)$

Città	*ACorso*
SA	1
PA	2
BO	1
MO	2
MI	1

Unione: unisce gli attributi comuni a varie entità

date 2 relazioni ***con lo stesso schema***, si definisce $r \cup s = \{t / t \in r \vee t \in s\}$

ES: Città di Studenti o Docenti --> $\pi_{Città}(Studente) \cup \pi_{Città}(Docente)$

Città
SA
PA
BO
MO
MI
FI

Differenza: sottrae gli attributi comuni alle varie entità

date 2 relazioni ***con lo stesso schema***, si definisce $r - s = \{t / t \in r \wedge t \notin s\}$

ES: Città di Studenti ma non di Docenti --> $\pi_{Città}(Studente) - \pi_{Città}(Docente)$

Città
SA
PA
MI

Intersezione: crea un insieme di attributi comuni ad entrambe le entità

date 2 relazioni ***con lo stesso schema***, si definisce $r \cap s = r - (r - s) = \{t / t \in r \wedge t \notin (r - s)\}$

$= \{t / t \in r \wedge t \in s\}$

ES: Città di studenti e docenti --> $\pi_{Città}(Studente) \cap \pi_{Città}(Docente)$

Città
BO
MO

Prodotto Cartesiano : è l'ampliamento di una di una tabella in un'altra contenente più informazioni

date 2 relazioni, si definisca $r \times s = \{t / t = t_R t_S / t_R \in r \wedge t_S \in s\}$

ES: Studente x Frequenza

Matr	*Nome*	*Città*	*ACorso*	*Matr*	*CodCor*
M1	Marco Quaranta	SA	1	M1	C1
M1	Marco Quaranta	SA	1	M1	C3
M2	Giacomo Tedesco	PA	2	M2	C1
M2	Giacomo Tedesco	PA	2	M2	C2

Matr	*Nome*	*Città*	*ACorso*	*Matr*	*CodCor*
M3	Maria Mei	BO	1	M3	C1
M3	Maria Mei	BO	1	M3	C2
M4	Ugo Rossi	MO	2	M4	C3
M4	Maria Mei	MO	2	M4	C4
M5	Sara Neri	MO	2	M5	NULL
M6	Agata Verdi	MI	1	M6	NULL

<u>Theta-Join</u>: è un prodotto cartesiano al quale viene applicata un'operazione di selezione F

date 2 relazioni e un predicato F costituito da espressioni booleane, si definisce

$r \; x_F \; s = \sigma_F \; (r \; x \; s)$

ES: $\sigma_{CodCor = C1}$ (Studenti x Frequenza)

Matr	*Nome*	*Città*	*ACorso*	*Matr*	*CodCor*
M1	Marco Quaranta	SA	1	M1	C1
M2	Giacomo Tedesco	PA	2	M2	C1
M3	Maria Mei	BO	1	M3	C1

<u>Equi-Join</u>: è un Theta-Join dove la funzione di confronto è un'uguaglianza

ES: studenti e docenti della stessa città

Studente $x_{CittàStudenti=CittàDocenti}$ Docente

Matr	*Nome*	*Città*	*ACorso*	*Codice*	*Cf*	*Città*
M3	Maria Mei	BO	1	D2	Cf2	BO
M4	Ugo Rossi	MO	2	D1	Cf1	MO
M4	Ugo Rossi	MO	2	D3	Cf3	MO
M5	Sara Neri	MO	2	D1	Cf1	MO
M5	Sara Neri	MO	2	D3	Cf3	MO

<u>Natural-Join</u>: è un Equi-Join, dove però differisce in quanto nell'Equi-Join l'attributo si ripete, mentre qui no.

Studente x Docente

Matr	*Nome*	*Città*	*ACorso*	*Codice*	*Cf*
M3	Maria Mei	BO	1	D2	Cf2
M4	Ugo Rossi	MO	2	D1	Cf1
M4	Ugo Rossi	MO	2	D3	Cf3
M5	Sara Neri	MO	2	D1	Cf1
M5	Sara Neri	MO	2	D3	Cf3

Semi-Join: è un prodotto cartesiano nel quale compaiono solo gli attributi dell'entità dalla parte del l
date 2 relazioni, si definisce --> $r \times s = \pi_X(r \times s)$
ES: Studenti della stessa città dei Docenti --> Studenti x Docenti

Matr	***Nome***	***Città***	***ACorso***
M3	Maria Mei	BO	1
M4	Ugo Rossi	MO	2
M5	Sara Neri	MO	2

OuterJoin: sono le tuple che partecipano al Join, più le tuple che non partecipano al Join, nelle quali metto null i valori mancanti, ossia comprende le tuple dangling.
Il risultato di un Join r x s comprende solo tuple t_R di r (t_S di s) che possono essere messe in corrispondenza, in base al predicato di Join, con una tupla t_S di s (t_R di r).
Una tupla si t_R di r (t_S di s) che non partecipa a tale corrispondenza, e che quindi non contribuisce al Join, è detta dangling.

- Left-OuterJoin: r = x s comprende le tuple dangling t_R di r
- Right-OuterJoin: r x = s comprende le tuple dangling t_S di s
- Full-OuterJoin: r = x = s comprende sia le tuple dangling t_R di r che le tuple dangling t_S di s

Divisione: è un operatore derivato e serve per individuare le tuple del dividendo associate a tutte le tuple del divisore.
Date 2 relazioni r e s, con schemi R(X) e S(Y) tali che $Y \subset X$ (Y contenuto strettamente in X), l'operazione di divisione tra r e s, $r \div s$, ha come risultato una relazione che ha schema $(X - Y)$ ed è definita da:

$r \div s = \{t_D \mathbin{/} \forall t_S \in S, t_D t_S \in r\}$

NB: l'operatore come detto precedentemente può essere derivato dagli operatori di base

$r \div s = \pi_{X-Y}(r) - \pi_{X-Y}((\pi_{X-Y}(r) \times s) - r)$

ES: Studenti che frequentano tutti i corsi di D1

Matr
M1