

Interrogazioni: operazioni su tabelle

- Le risposte alle domande sono sempre tabelle ottenute combinando altre tabelle
- L'algebra relazionale definisce una serie di operazioni su tabelle che danno come risultato altre tabelle
- Algebra relazionale
 - uso degli schemi e delle chiavi (primarie ed esterne) per recuperare i dati

Operazioni possibili

- Ritagliare "sottotabelle"
- Combinare e unire tabelle
- Specificare condizioni
- Modificare dati
- Inserire dati
- Eliminare dati

Ritagliare sotto-tabelle

- Proiezione: si selezionano solo alcuni attributi
 - “taglio verticale”, selezione “per colonne”
- Restrizione: si selezionano solo alcune ennuple, che soddisfano una certa condizione
 - “taglio orizzontale”, selezione “per righe”

Ritagliare sotto-tabelle: proiezione

Id_provincia	nome_provincia	id_ambito	cod_ambito	nome_ambito	id_settore	fk_ambito
BS	Brescia	1	TE	Temù	1	1
BS	Brescia	1	TE	Temù	2	1
BS	Brescia	1	TE	Temù	3	1
BS	Brescia	1	TE	Temù	4	1
BS	Brescia	1	TE	Temù	5	1
BS	Brescia	1	TE	Temù	6	1
BS	Brescia	1	TF	Temù	7	1
BZ	Bolzano	2	GC	Gomagoi	8	2
BZ	Bolzano	2	GC	Gomagoi	9	2
BZ	Bolzano	2	GC	Gomagoi	10	2
BZ	Bolzano	2	GC	Gomagoi	11	2
BZ	Bolzano	2	GC	Gomagoi	12	2

$\Pi_{id_ambito, nome_ambito, id_settore, fk_ambito}$ (Ambito)

id_ambito	nome_ambito	id_settore	fk_ambito
1	Temù	1	1
1	Temù	2	1
1	Temù	3	1
1	Temù	4	1
1	Temù	5	1
1	Temù	6	1
1	Temù	7	1
2	Gomagoi	8	2
2	Gomagoi	9	2
2	Gomagoi	10	2
2	Gomagoi	11	2
2	Gomagoi	12	2

Ritagliare sotto-tabelle: restrizione (selezione)

Id_provincia	nome_provincia	id_ambito	cod_ambito	nome_ambito	id_settore	fk_ambito
BS	Brescia	1	TE	Temù	1	1
BS	Drescia	1	TE	Temù	2	1
BS	Brescia	1	TE	Temù	3	1
BS	Brescia	1	TE	Temù	4	1
BS	Brescia	1	TE	Temù	5	1
BS	Brescia	1	TE	Temù	6	1
BS	Brescia	1	TF	Temù	7	1
BZ	Bolzano	2	GC	Gomagoi	8	2
BZ	Bolzano	2	GC	Gomagoi	9	2
BZ	Dolzano	2	GC	Gomagoi	10	2
BZ	Bolzano	2	GC	Gomagoi	11	2
BZ	Bolzano	2	GC	Gomagoi	12	2

σ nome_ambito='Sondalo' (Ambito)

id_ambito	nome_ambito	id_settore	fk_ambito	cod_settore	Nome_settore
7	Sondalo	35	7	01	Sondalo 1
7	Sondalo	36	7	02	Sondalo 2
7	Sondalo	37	7	03	Sondalo 3
7	Sondalo	38	7	04	Sondalo 4
7	Sondalo	39	7	05	Sondalo 5
7	Sondalo	40	7	06	Sondalo 6
7	Sondalo	41	7	02	Val di Rezzalo
7	Sondalo	42	7	01	Serottini-Sasso Maurone
7	Sondalo	43	7	01	Cepina-Mondadizza

Ritagliare sotto-tabelle: restrizione

- Condizioni booleane elementari
 - $A_i = B_j$, $A_i > B_j$, $A_i = \text{un valore}$, $A_i > \text{un valore}$...
- Condizioni booleane composte
 - AND
 - (Sesso = 'M' **AND** Età <> 'J')
 - OR
 - (Età = 'A' **OR** Età = 'S')
 - NOT
 - **NOT** (Età = 'J')

Combinare tabelle: prodotto

id_specie	nome_ita	genere	specie
00070	Tuffetto	Tachybaptus	ruficollis
00090	Svasso maggiore	Podiceps	cristatus
02070	Poiana	Buteo	buteo
00980	Larabusino	Ixobrychus	minutus
01040	Nitticora	Nycticorax	nycticorax
01190	Garzella	Egretta	garzella
01220	Airone cinereo	Ardea	cinerea
01240	Airone rosso	Ardea	purpurea
01520	Cigno reale	Cygnus	olor
01840	Alzavola	Aras	crecca
01860	Germano reale	Aras	platyrhynchos
01910	Marzaiola	Aras	querquedula
01980	Morigiane	Aythya	farina
02020	Moretta tabaccata	Aythya	nyroca
02310	Falco pecchiaiolo	Pernis	apivorus
02380	Nibbia bruna	Mivus	migrans
02560	Biancone	Circus	gallus
02600	Falco di peluce	Circus	aeruginosus

id_quadrato	sezione	codice	primario
1 A3c3	D5	T	
2 A3c3	E5	F	
3 A3c3	F5	T	
4 A3c3	G5	F	
5 A3c4	C2	F	
5 A3c4	D3	F	
7 A3c4	D1	T	
8 A3c4	U2		
9 A3c4	U1		
10 A3c4	U4	I	
11 A3c4	U1		
12 A3c4	L1	I	
13 A3c4	L2		
14 A3c4	L1	I	

Specie | x | Quadrato

id_specie	nome_ita	genere	specie	id_quadrato	sezione	codice	primario
00070	Tuffetto	Tachybaptus	ruficollis	1 A3c3	D5	T	
00070	Tuffetto	Tachybaptus	ruficollis	2 A3c3	E5	F	
00070	Tuffetto	Tachybaptus	ruficollis	3 A3c3	F5	T	
00070	Tuffetto	Tachybaptus	ruficollis	4 A3c3	G5	F	
00070	Tuffetto	Tachybaptus	ruficollis	5 A3c4	C2	F	

Combinare tabelle: giunzione

Non tutte le combinazioni possibili (prodotto cartesiano) sono realistiche...

- Giunzione (join):
 - prodotto cartesiano tra chiave esterna e chiave primaria

Specie |X| Quadrato =

$$\sigma_{\text{Specie.id_specie} = \text{Censimento.id_specie}} (\text{Specie |X| Censimento})$$

Combinando le operazioni è possibile formulare interrogazioni articolate e complesse

SQL: Structured Query Language

- 1974: IBM San Jose Research Lab (System R)
- 1981: IBM SQL/DS (IBM DB2)
- 1983: standard *de facto*
- 1986, 1989, 1992: standardizzazione ANSI
- Linguaggio non procedurale (*cosa* anziché *come*)
- DML e anche DDL
- Numerosissimi "dialetti"
 - standard ISO-ANSI SQL 1999 (SQL 3)
 - uso di *cheat sheet*

SQL: funzionalità

- DDL
- DML interattivo
 - SELECT
 - INSERT, DELETE, UPDATE
- Viste
- Autorizzazioni
- Integrità
- Controllo transazione
- Strutturato come “linguaggio naturale”
- Rappresentazione di operazioni di algebra relazionale
- Clausole standard
- Possibilità di *embedding*
- Utilizzato dalla maggior parte dei DBMS

SELECT

- Clausola SELECT: effettua la maggior parte delle operazioni dell'algebra relazionale (e qualcuna in più)
- Consente interrogazioni sofisticate
- Consente di elaborare, anche in senso statistico i dati

Tipicamente:

```
SELECT <attributi>
```

```
FROM <tabelle>
```

```
[WHERE <condizioni>]
```

```
;
```

Il risultato è sempre una relazione (tabella)

SELECT: dettagli

- SELECT corrisponde all'operatore di proiezione
SELECT genere,specie FROM tassonomia;
 $\Pi_{\text{genere,specie}}(\text{tassonomia})$
- Caratteri "jolly"
 - * tutti gli attributi
- Modificatori
 - ALL tutti i *record*, nessuno escluso
 - DISTINCT elimina i record duplicati
- È possibile effettuare calcoli
SELECT peso * 0.1 AS pct10peso FROM biometrie;

WHERE

- WHERE consente di specificare condizioni (filtri)
- Condizioni booleane
- Selezione su intervalli ('BETWEEN')
- Caratteri jolly: '_' e '%'

```
SELECT id_quadrato, n_individui FROM rilievi  
WHERE (specie = 'Emberiza cia');
```

```
select AttrExpr [ [as] Alias ] { , AttrExpr [ [as] Alias ] }  
from Tabella [ [as] Alias ] { , Tabella [ [as] Alias ] }  
[ where Condizione ]
```

```
select *  
from Impiegato  
where Cognome like '_o%i'
```

Casi particolari

SELECT [DISTINCT] a_1, a_2, \dots, a_k FROM R
equivale a $\Pi_{a_1, a_2, \dots, a_k} R$

SELECT * FROM R WHERE c
equivale a $\sigma_c R$

SELECT * FROM R, S
equivale a $R \times S$

SELECT * FROM R, S WHERE c
equivale a $\sigma_c (R \times S)$

CREATE

- **CREATE DATABASE**
 - Crea una nuova base di dati (collezione di tabelle)
 - Opposto: DROP DATABASE
- **CREATE TABLE**
 - Creare una nuova tabella, definendone la struttura

```
CREATE TABLE Radiotracking (  
  id TEXT(20) NOT NULL UNIQUE,  
  data DATE NOT NULL,  
  ora TIME NOT NULL,  
  x_utm INTEGER NOT NULL,  
  y_utm INTEGER NOT NULL);
```

- Opposto: DROP TABLE

Definizione di chiavi e di vincoli di integrità

```
CREATE TABLE Radiotracking (  
  id TEXT(20) NOT NULL UNIQUE,  
  fk_rilevatore INTEGER NOT NULL,  
  data DATE NOT NULL,  
  ora TIME NOT NULL,  
  x_utm INTEGER NOT NULL,  
  y_utm INTEGER NOT NULL,  
  PRIMARY KEY id,  
  FOREIGN KEY rilevatore(id_rilevatore)  
    REFERENCES fk_rilevatore  
    ON DELETE RESTRICT);
```

Modifiche

- INSERT INTO <tabella> <campi> VALUES (<valori>);
 - INSERT INTO corsi VALUES ('F75001');
- DELETE FROM studenti WHERE nome='Rossi';
- UPDATE tassonomia
SET specie='macrobullaris'
WHERE genere='Plecotus' AND specie='alpinus';
- ALTER TABLE tassonomia ADD statusIUCN TEXT;

Rinominare i campi

<nome vecchio> AS <nome nuovo>

SELECT

ageclass AS eta,

sex AS sesso,

tarsoavg AS (dxtarso + sxtarso)/2

FROM

biometrie

WHERE

(eta = 'S') OR (eta = 'A');

Interrogazione su più tabelle

```
SELECT DISTINCT
  nome_specie, nome_genero
FROM specie AS S, genere AS G
WHERE S.id_genero = G.id_genero;
```

Ordinamento

```
SELECT DISTINCT
  nome_specie, nome_generere
FROM specie AS S, genere AS G
WHERE S.id_generere = G.id_generere
ORDER BY G.nome_generere, S.nome_specie ASCENDING;
```

- AVG
- MIN
- MAX
- SUM
- COUNT

```
SELECT AVG(dx_Vdito) AS VFINGER  
FROM biometrie  
WHERE  
    id_specie = '110.0.121.0.001' AND sesso = 'M';
```

```
SELECT COUNT(*) FROM rilievi;
```

Raggruppamento

Utile insieme alle funzioni di aggregazione

```
SELECT AVG(dx_Vdito) AS VFINGER
FROM biometrie, specie
WHERE
  ((eta = 'A') AND (sesso = 'M')) AND
  (biometrie.fk_specie = specie.id_specie)
GROUP BY specie;
```

```
SELECT COUNT(*) FROM rilievi GROUP BY sito HAVING
  anno > 1983;
```

Firebird SQL

