Query SQL - singola tabella

Informatica@SEFA 2018/2019 - Lezione 22

Massimo Lauria <massimo.lauria@uniroma1.it>
http://massimolauria.net/courses/infosefa2018/

Venerdì, 07 Dicembre 2018

Tabelle in SQL

Tabelle nel database relazionale

Un DB relazionale è costituito da una serie di tabelle.

- Ogni colonna ha un nome e un tipo di dato
- Ogni riga contiene un valore per ogni colonna, del tipo di dato indicato.
- Il comando `select` permette vederne il contenuto

```
select * from Combustibili;                                    1
```

Cod_Combustibile	Descrizione_Combustibile
1	Benzina
2	Gasolio
3	GPL
4	Metano

Terminologia: database VS algebra

Algebra	Database
Relazione	Tabella
Attributo	Colonna
Tupla	Riga

Tipi di dato delle colonne

- Testi (non unicode): `char`, `varchar`
- Testi (unicode): `nchar`, `nvarchar`
- Numeri interi: `integer`
- Numeri decimali esatti: `decimal`, `numeric`
- Numeri reali approssimati: `real`, `float`
- Date/Orari: `date`, `time`

Alcuni dei tipi richiedono dei parametri

E.g. `nchar(20)` un testo di 20 caratteri.

E.g. `nvarchar(100)` un testo di al più 100 caratteri.

SQLite e tipi di dati

In SQL ogni colonna/attributo ha un tipo di dato, ma in SQLite ogni cella ha il suo tipo e SQLite si occupa delle convesioni.

```
drop table if exists A;
create table if not exists A (
     Numeri integer,
     Caratteri nvarchar(4) );

insert into A values ('01','01'),('BBB','BBB'),
                     (1,1),       (1.33,1.33);

select * from A;
```

Numeri	Caratteri
1	01
BBB	BBB
1	1
1.33	1.33

Lo schema di una tabella

Lo schema viene descritto da SQLite come la sequenza di istruzioni SQL usate per generare le tabelle.

```
.schema Veicoli
```

```
CREATE TABLE Veicoli (
       Targa             nvarchar(10) primary key,
       Cilindrata        integer,
       Cavalli_Fiscali   integer,
       Velocità          integer,
       Posti             integer,
       Immatricolazione  date,

       [...omissis...]
);
```

Creazione di una tabella

```
create table if not exists NomeTabella (
   descrizione_colonna1,
   descrizione_colonna2,
   descrizione_colonna3,
   ...

   vincolo_di_integrità1,
   vincolo_di_integrità2,
   ...
);
```

Questo comando crea la tabella `NomeTabella` se questa non esiste. La prima riga sarebbe anche potuta essere

```
create table NomeTabella (
```

ma in questo caso se la tabella fosse esistita si sarebbe verificato un errore.

Descrizione di una colonna

La descrizione più semplice di una colonna è

```
<NoneColonna> <TipoColonna>
```

ad esempio le due colonne

```
Nome nvarchar(20),
Cognome nvarchar(30)
```

esistono molte altre cose che possono essere specificate per ogni colonna (e.g. se la colonna accetta valori NULL, se è parte di una chiave, ecc…).

Colonne a valori non nulli

Le celle di una tabella possono avere un valore nullo. Il significato di un valore nullo non è specificato.

- dato non valido
- dato assente
- dato ignoto

Delle colonne possono avere il vincolo di non nullità

```
create table DatiAnagrafici (                                    1
       CodFiscale char(15) not null,                             2
       Nome nvarchar(30) not null,                               3
       Cognome nvarchar(30) not null,                            4
       Indirizzo nvarchar(100)  -- può essere nullo              5
);                                                               6
```

Valori unici

É possibile definire il vincolo che il valore contenuto in una o più colonne appaia al massimo su un unica riga.

```
create table DatiAnagrafici (
       CodFiscale char(15) not null unique,
       Nome nvarchar(30) not null,
       Cognome nvarchar(30) not null,
       Indirizzo nvarchar(100),

       unique(Nome,Cognome) -- non associato ad un attributo
     );
```

- La coppia `Nome,Cognome` non può essere ripetuta.
- Il codice fiscale non può essere ripetuto.

Osservazione

Notate la differenza tra i due comandi SQL

```
create table A (                                    1
       Nome nvarchar(30) not null,                  2
       Cognome nvarchar(30) not null,               3
       unique(Nome,Cognome)                         4
     );                                             5
```

```
create table B (                                    1
       Nome nvarchar(30) not null unique,           2
       Cognome nvarchar(30) not null unique         3
     );                                             4
```

Nella tabella B non sono permesse neppure persone con lo stesso nome (E.g. Marco Bianchi e Marco Rossi).

Valori unici e NULL

Se si usa `unique` senza `not null` si permettono comunque più righe con elementi nulli. Due valori nulli sono considerati differenti.

```
create table T ( campo1 integer,                                    1
                 campo2 integer,                                    2
                 unique(campo1,campo2) );                           3
insert into T values (1,2),(1,NULL),(NULL,NULL),(NULL,NULL);        4
select * from T;                                                    5
```

campo1	campo2
1	2
1	NULL
NULL	NULL
NULL	NULL

Chiavi della tabella

Attributi che siano

- simultaneamente `not null`
- collettivamente `unique`

formano una superchiave della tabella. Ovvero ogni riga della tabella è identificata univocamente da questi valori.

Chiave: una superchiave minimale.

(nella lezione avevo sbagliato terminologia, scusate)

Chiave primaria

Una tabella può avere una chiave primaria, ovvero un chiave principale per la tabella.

Molteplici chiavi con `unique` e `not null`, ma solo una chiave primaria.

```
create table Studente (
  Matricola integer primary key,
  Nome    nvarchar(60),
  Cognome nvarchar(60)
);
```

Il vincolo `primary key` implica `not null` e `unique`.

Chiave primaria (II)

La chiave primaria può essere costituita da più attributi.

```
create table Studente (                                    1
  Matricola       integer,                                 2
  CodUniversità   nchar(5)                                 3
  Nome     nvarchar(60),                                   4
  Cognome  nvarchar(60),                                   5
                                                           6
  primary key(Matricola,CodUniversità)                     7
);                                                         8
```

Con i seguenti vincoli implicati.

```
  Matricola       integer not null,                        1
  CodUniversità   nchar(5) not null,                       2
  unique (Matricola,CodUniversità)                         3
```

Vincoli intrarelazionali

I vincoli di

- chiave primaria
- unicità
- non nullità

riguardano una singola tabella (i.e. intrarelazionali).

Quando vedremo query SQL che coinvolgono più tabelle allora discuteremo di vincoli interrelazionali.

Eliminazione di una tabella

```
drop table NomeTabella;
```

oppure

```
drop table if exists NomeTabella;
```

La prima variante solleva un errore se la tabella non esiste.

Comando `select`

Le tre parti principali

Il comando `select` è lo strumento principale per eseguire query (i.e. interrogazioni) in un database.

```
select <ListaColonne>    -- clausola select                    1
from <Tabella>           -- clausola from                      2
where <Condizione>       -- clausola where (opzionale)         3
```

La clausola `select` seleziona una lista di colonne `<ListaColonne>` dalla tabella `<Tabella>`. La clausola

```
where <Condizione>
```

filtra le righe della tabella prima di selezionare colonne.

Stampa di una tabella

Se l'argomento della clausola `select` è `*` allora stampa tutte le colonne.

```
select * from Modelli;                                    1
```

Cod_Modello	Nome_Modello	Cod_Fabbrica	Numero_Versioni
1	Panda	1	3
2	Vespa	4	4
3	Brava	1	2
4	Mondeo	3	3
5	V-10	5	2
6	Ducato	1	5
7	Clio	6	5
8	Corolla	7	4
9	Coupè	1	1
10	Golf	8	4
11	Megane	6	2
12	Seicento	1	2
13	Laguna	6	2
14	Civic	9	3

Proiezione (selezione colonne)

```
select Cod_Modello,Nome_Modello from Modelli;                    1
```

Cod_Modello	Nome_Modello
1	Panda
2	Vespa
3	Brava
4	Mondeo
5	V-10
6	Ducato
7	Clio
8	Corolla
9	Coupè
10	Golf
11	Megane
12	Seicento
13	Laguna
14	Civic

Scelta dell'ordine delle colonne

```
select Nome_Modello,Cod_Modello from Modelli;                    1
```

Ridenominare e/o ripetere le colonne

```
select Nome_Modello as 'Nome del Modello',                    1
       Cod_Modello as [Codice del Modello],                   2
       Nome_Modello, Nome_Modello                             3
       from Modelli;                                          4
```

Nome del Modello	Codice del Modello	Nome_Modello	Nome_Modello
Panda	1	Panda	Panda
Vespa	2	Vespa	Vespa
Brava	3	Brava	Brava
Mondeo	4	Mondeo	Mondeo
V-10	5	V-10	V-10
Ducato	6	Ducato	Ducato
Clio	7	Clio	Clio
Corolla	8	Corolla	Corolla
Coupè	9	Coupè	Coupè
Golf	10	Golf	Golf
Megane	11	Megane	Megane
Seicento	12	Seicento	Seicento
Laguna	13	Laguna	Laguna
Civic	14	Civic	Civic

Parentesi quadre o apici

Servono per i nomi che contengono spazi o parole speciali. Possono essere omesse altrimenti.

```
select Nome_Modello as 'select',                    1
       Cod_Modello as Codice from Modelli;          2
```

select	Codice
Panda	1
Vespa	2
Brava	3
Mondeo	4
V-10	5
Ducato	6
Clio	7
Corolla	8
Coupè	9
Golf	10
Megane	11
Seicento	12
Laguna	13
Civic	14

Espressioni

La clausola select accetta cose anche più articolate del nome di una colonna. Anche espressioni e costanti.

```
select Targa,Velocità as 'Vel. km/h',                              1
       Velocità*1000/3600 as 'Vel. m/s', 'noia' as 'Commento'      2
    from Veicoli;
```

Espressioni (II)

L'uso principale della `select` è di selezionare colonne, tuttavia potere vedere la cosa più in generale: ogni riga è in effetti la sequenza del valore di funzioni $f_1, f_2, f_3, \ldots$ calcolate sulle righe della tabella.

```
drop table if exists Demo;                                          1
create table Demo (                                                 2
     x integer,                                                     3
     y integer );                                                   4
insert into Demo values (1,1),(2,-3),(7,1);                         5
select *,x,*,y,x+y,x-y,x*y,21-2 from Demo;                          6
```

x	y	x	y	x	x+y	x-y	x*y	0
1	1	1	1	1	2	0	1	0
2	-3	2	-3	2	-1	5	-6	0
7	1	7	1	7	8	6	7	0

Senza clausola `from`

Se l'espressione non contiene riferimenti alla tabella possiamo anche omettere la clausola `from`.

```
select 10 as 'Prima colonna',                                    1
       20 as 'Seconda colonna';                                  2
```

Prima colonna	Seconda colonna
10	20

Modificatore `distinct`

L'opzione `distinct` elimina le righe doppie dall'output, che di default sono mantenute. La tabella `Veicoli` ha 15 righe, ma la query successiva ne produce 14.

```
select distinct Cod_Modello as Modelli from Veicoli;          1
```

Cod_Modello
4
1
9
6
7
2
3
14
11
13
10
12
5
8

Clausola `where`

Selezione di righe

Si possono filtrare le righe in base ad una condizione sui valori delle colonne.

```
select Targa,Immatricolazione,Velocità as 'km/h' from Veicoli 1
    where Cilindrata > 1000;
```

Targa	Immatricolazione	km/h
A123456X	1998-12-30	195
C76589AG	1998-08-13	130
C78905GT	1994-11-06	212
CFT340VB	1995-01-12	170
DD4567XX	1997-06-05	NULL
DGH789JC	1995-10-05	170
DH79567H	NULL	170
ERG567NM	1997-12-18	175
F96154NH	1992-03-08	185
XCH56GJK	1998-09-04	210
XF5789CY	1996-05-05	175

Condizione della clausola `where`

La clausola `where` vuole una espressione a valori booleani.

1. Operatori di confronto tra costanti, attributi, espressioni. `=`, `<>`, `>`, `<`, `<=`, `>=`
2. Connettori logici `AND`, `OR`, `NOT`
3. Operatore `BETWEEN`
4. Operatore `IN`
5. Operatore `LIKE` (pattern matching)
6. Test di valore nullo `IS NULL`, `IS NOT NULL`

Risultato vuoto

Una selezione può produrre una tabella senza righe.

```
select Targa,Immatricolazione,Velocità as 'km/h' from Veicoli 1
    where Posti = 6;
```

Targa	Immatricolazione	km/h

Selezione di una riga attraverso una chiave

Qui la clausola `where` chiede che gli attributi chiave siano uguale ad uno specifico valore.

```
select * from Veicoli where Targa = 'A123456X';
```

Targa	Cilindrata	Cavalli_Fiscali	Velocità	Posti	Immatricolazione	Cod_C
A123456X	1796	85	195	5	1998-12-30	

Confronto tra due colonne della tabella

```
select Targa,Velocità,Posti,                                    1
       Cavalli_Fiscali as CF,Cilindrata from Veicoli            2
where  Cilindrata>CF*20;                                        3
-- Cilindrata > Cavalli_Fiscali*20 sarebbe stato lo stesso      4
```

Targa	Velocità	Posti	CF	Cilindrata
A123456X	195	5	85	1796
B256787Y	120	5	10	708
C76589AG	130	5	54	1106
DD4567XX	NULL	5	17	1581
FGH673FV	140	5	39	899

Uso di operatori logici

```
 select Targa, Velocità, Cod_Categoria, Cod_Modello,      1
Cilindrata from Veicoli
   where not Cod_Categoria='01';                          2
```

Targa	Velocità	Cod_Categoria	Cod_Modello	Cilindrata
C845905Z	NULL	4	6	NULL
D239765W	NULL	3	2	NULL

NOT precede AND che precede OR

```
select Targa, Velocità, Cod_Categoria, Cilindrata from      1
 Veicoli
where not (Cod_Categoria='01' and Cilindrata > 1500);       2
```

Targa	Velocità	Cod_Categoria	Cilindrata
B256787Y	120	1	708
C76589AG	130	1	1106
C845905Z	NULL	4	NULL
CFT340VB	170	1	1390
D239765W	NULL	3	NULL
FGH673FV	140	1	899

```
select Targa, Velocità, Cod_Categoria, Cilindrata from      1
 Veicoli
where not Cod_Categoria='01' and Cilindrata > 1500;         2
```

Targa	Velocità	Cod_Categoria	Cilindrata

BETWEEN <min> AND <max>

Un'espressione compresa in un intervallo.

```
select Targa,Velocità,Cod_Combustibile, Cilindrata from      1
    Veicoli
where Cilindrata between 1500 and 1800;                      2
```

Targa	Velocità	Cod_Combustibile	Cilindrata
A123456X	195	1	1796
DD4567XX	NULL	1	1581
DGH789JC	170	2	1590
DH79567H	170	4	1589
ERG567NM	175	4	1598
F96154NH	185	3	1781
XF5789CY	175	1	1587

Realizzazione alternativa di BETWEEN

L'operatore BETWEEN può essere realizzato con confronti e operatori booleani, ma è meno leggibile.

```
select Targa,Velocità,Cod_Combustibile, Cilindrata from      1
    Veicoli
where Cilindrata >= 1500 and Cilindrata <=1800;              2
```

Targa	Velocità	Cod_Combustibile	Cilindrata
A123456X	195	1	1796
DD4567XX	NULL	1	1581
DGH789JC	170	2	1590
DH79567H	170	4	1589
ERG567NM	175	4	1598
F96154NH	185	3	1781
XF5789CY	175	1	1587

Operatore IN

Un’espressione inclusa in una lista di valori.

```
select Targa,Velocità,Cod_Combustibile, Cilindrata from      1
    Veicoli
where Targa in ('A123456X','XCH56GJK','D239765W');           2
```

Targa	Velocità	Cod_Combustibile	Cilindrata
A123456X	195	1	1796
D239765W	NULL	1	NULL
XCH56GJK	210	1	1918

Realizzazione alternativa di `IN`

```
select Targa,Velocità,Cod_Combustibile, Cilindrata from      1
    Veicoli
where Targa = 'A123456X' or                                  2
      Targa = 'XCH56GJK' or                                  3
      Targa = 'D239765W';                                    4
```

Targa	Velocità	Cod_Combustibile	Cilindrata
A123456X	195	1	1796
D239765W	NULL	1	NULL
XCH56GJK	210	1	1918

Algebra dei valori NULL

`Velocità < 180 or Velocità >= 180` **sembrerebbe essere sempre vera, è invece restituisce 12 delle 15 righe.**

```
select Targa, Velocità, Cilindrata from Veicoli
  where Velocità < 180 or Velocità >= 180;
```

Targa	Velocità	Cilindrata
A123456X	195	1796
B256787Y	120	708
C76589AG	130	1106
C78905GT	212	1998
CFT340VB	170	1390
DGH789JC	170	1590
DH79567H	170	1589
ERG567NM	175	1598
F96154NH	185	1781
FGH673FV	140	899
XCH56GJK	210	1918
XF5789CY	175	1587

Valori NULL e logica a tre valori

- Espressioni che contengo attributi `NULL` sono *unknown*.
- Gli operatori logici seguono una logica a 3 valori.

X	Y	not X	X and Y	X or Y
V	V	F	V	V
V	U	F	U	V
V	F	F	F	V
U	V	U	U	V
U	U	U	U	U
U	F	U	F	U
F	V	V	F	V
F	U	V	F	U
F	F	V	F	F

- La riga viene inclusa solo se la condizione è vera.

IS NULL e IS NOT NULL

Testano se un attributo ha valore NULL o meno. La query restituisce tutte le 15 righe.

```
select Targa, Velocità, Cilindrata from Veicoli
  where Velocità < 180 or Velocità >= 180 or Velocità is
null;
```

Targa	Velocità	Cilindrata
A123456X	195	1796
B256787Y	120	708
C76589AG	130	1106
C78905GT	212	1998
C845905Z	NULL	NULL
CFT340VB	170	1390
D239765W	NULL	NULL
DD4567XX	NULL	1581
DGH789JC	170	1590
DH79567H	170	1589
ERG567NM	175	1598
F96154NH	185	1781
FGH673FV	140	899
XCH56GJK	210	1918
XF5789CY	175	1587

Ordinamento dell'output

Non esiste un ordine predefinito per l'output delle righe.

Possiamo ordinare rispetto ad una colonna

- ascendente (dal più piccolo al più grande)
- discendente (dal più grande al più piccolo)

utilizzando la clausola `ORDER BY`

Esempio: ordinamento ascendente

```
select Targa,Cilindrata,Velocità from Veicoli                    1
where Velocità is not null                                       2
order by Velocità;                                               3
```

Targa	Cilindrata	Velocità
B256787Y	708	120
C76589AG	1106	130
FGH673FV	899	140
CFT340VB	1390	170
DGH789JC	1590	170
DH79567H	1589	170
ERG567NM	1598	175
XF5789CY	1587	175
F96154NH	1781	185
A123456X	1796	195
XCH56GJK	1918	210
C78905GT	1998	212

Parola chiave `ASC`

L'ordinamento di default è ascendente, ma può essere specificato dalla parola chiave `ASC`

```
select Targa,Cilindrata,Velocità from Veicoli
where Velocità is not null
order by Velocità asc;
```

Targa	Cilindrata	Velocità
B256787Y	708	120
C76589AG	1106	130
FGH673FV	899	140
CFT340VB	1390	170
DGH789JC	1590	170
DH79567H	1589	170
ERG567NM	1598	175
XF5789CY	1587	175
F96154NH	1781	185
A123456X	1796	195
XCH56GJK	1918	210
C78905GT	1998	212

Esempio: ordinamento discendente

```
select Targa,Cilindrata,Velocità from Veicoli                    1
where Velocità is not null                                       2
order by Velocità desc;                                          3
```

Targa	Cilindrata	Velocità
C78905GT	1998	212
XCH56GJK	1918	210
A123456X	1796	195
F96154NH	1781	185
ERG567NM	1598	175
XF5789CY	1587	175
CFT340VB	1390	170
DGH789JC	1590	170
DH79567H	1589	170
FGH673FV	899	140
C76589AG	1106	130
B256787Y	708	120

Ordinamenti a catena

```
select Targa,Cilindrata,Velocità from Veicoli                    1
order by Velocità desc, Targa asc;                               2
```

Targa	Cilindrata	Velocità
C78905GT	1998	212
XCH56GJK	1918	210
A123456X	1796	195
F96154NH	1781	185
ERG567NM	1598	175
XF5789CY	1587	175
CFT340VB	1390	170
DGH789JC	1590	170
DH79567H	1589	170
FGH673FV	899	140
C76589AG	1106	130
B256787Y	708	120
C845905Z	NULL	NULL
D239765W	NULL	NULL
DD4567XX	1581	NULL

Letture

Sessioni 4.1, 4.2, 4.3, 4.8 del manuale SQL.