



Il Linguaggio SQL (DML)

Basi di Dati (Corso A-L)
Ingegneria Informatica
Ing. Corrado Santoro

Un passo indietro...



- Si desidera informatizzare l'autoparco di un'azienda di trasporti urbani. Occorre gestire l'archivio degli automezzi ed i movimenti di ciascun mezzo in base alle seguenti specifiche:
 1. per ogni automezzo occorre memorizzare, marca (indicare anche se la marca è estera), modello, motorizzazione (benzina, diesel, elettrica), cilindrata, matricola, targa (la matricola è univoca)
 2. ogni automezzo può trovarsi in uno dei seguenti stati:
 1. in servizio
 2. in autorimessa
 3. in manutenzione
 3. gestire i cambi di stato di ogni automezzo, memorizzando, in un apposito archivio storico, la data e l'ora del cambio di stato, ed il nuovo stato raggiunto
 4. ogni volta che un automezzo entra "in manutenzione", gestire un "foglio di riparazione" in cui sono riportati tutti gli interventi eseguiti e gli eventuali pezzi sostituiti, insieme alla data in cui l'intervento/sostituzione è stata effettuata ed al meccanico che ha effettuato l'operazione

Lo schema relazionale...



Un pezzo dello script SQL

```
create database autoparco; use autoparco;
create table marche (nome varchar (20) primary key,
                    estera bool not null default 0);
create table motorizzazioni (tipo varchar (20) primary key);
create table stati (nomestato varchar (20) primary key);
create table automezzi (
    matricola integer primary key,
    marca varchar (20)
        references marche(nome)
        on delete cascade on update cascade,
    modello varchar (30),
    motorizzazione varchar (20)
        references motorizzazioni(tipo)
        on delete cascade on update cascade,
    cilindrata integer,
    targa varchar (20) not null unique,
    stato_corrente varchar (20)
        references stati(nomestato)
        on delete cascade on update cascade);
```

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 9 - Il Linguaggio SQL (DML)

4

Inseriamo i primi dati

- L'inserimento di record su tabelle avviene, in SQL, tramite il comando

```
INSERT INTO
    Nome_Tabella ( NomeAttr1, NomeAttr2, ..., NomeAttrN )
VALUES ( Valore1, Valore2, ..., ValoreN );
```

- Esempio: inserimento valori nella tabella "Marche"

```
create table marche (nome varchar (20) primary key,
                    estera bool not null default 0);
```

```
insert into marche(nome,estera) values ('FIAT',0);
insert into marche(nome,estera) values ('Opel',1);
insert into marche(nome) values ('Iveco');
insert into marche(nome,estera) values ('Citroen',1);
```

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 9 - Il Linguaggio SQL (DML)

5

Interrogiamo le tabelle

- L'interrogazione avviene tramite il comando "select"
- E' un comando molto potente in quanto consente di applicare tutti gli operatori principali dell'algebra relazionale (ridenominazione, proiezione, selezione, join e varianti)
- Ha quindi una sintassi complessa
- La forma "base":

```
SELECT NomeAttr1, NomeAttr2, ..., NomeAttrN
FROM Nome_Tabella1, ..., Nome_TabellaN
[ WHERE Condizione ] ;
```

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 9 - Il Linguaggio SQL (DML)

6

Alcuni Esempi

- Tutti i record della tabella **marche**

```
mysql> select nome, estera from marche;
```

nome	estera
FIAT	0
Opel	1
Iveco	0
Citroen	1

```
mysql> select * from marche;
```

Forma equivalente
 * = Tutti gli attributi

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 9 - Il Linguaggio SQL (DML) 7

Alcuni Esempi

- Tutti i record della tabella **marche** che si riferiscono a marche italiane

```
mysql> select * from marche where estera=0;
```

nome	estera
FIAT	0
Iveco	0

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 9 - Il Linguaggio SQL (DML) 8

Alcuni Esempi

- tutti gli automezzi (matricola, marca, modello, targa) "in servizio"

```
create table automezzi (
  matricola integer primary key,
  marca varchar (20) references ...,
  modello varchar (30),
  motorizzazione varchar (20) references ...,
  cilindrata integer,
  targa varchar (20) not null unique,
  stato_corrente varchar (20) references ...);
```

```
mysql> select matricola, marca, modello, targa from automezzi
-> where stato_corrente = 'In Servizio';
```

matricola	marca	modello	targa
1	FIAT	AAAA	KM123JJ
3	Iveco	ZZZZ	KM125JJ

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 9 - Il Linguaggio SQL (DML) 9

Select e Proiezione

- Però la proiezione ...

```
mysql> select estera from marche;
```

estera
0
1
0
1

~~←→~~ π_{estera} Marche

```
mysql> select distinct estera from marche;
```

estera
0
1

←→ π_{estera} Marche

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 9 - Il Linguaggio SQL (DML) 13

Select e Ridenominazione

- La select supporta anche l'operatore di ridenominazione p attraverso la seguente sintassi

```
SELECT NomeAttr1 AS Alias1, NomeAttr2 AS Alias2, ...
FROM Nome_Tabella1, ..., Nome_TabellaN
[ WHERE Condizione ] ;
```

- Esempio: selezionare solo le marche estere, usando però "marca_estera" come nome dell'attributo risultante

$\rho_{Marca_estera} \leftarrow \text{Nome } \pi_{\text{Nome}} \sigma_{\text{Estera}=1} \text{ Marche}$

```
mysql> select nome as marca_estera from marche where estera=1;
```

marca_estera
Opel
Citroen

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 9 - Il Linguaggio SQL (DML) 14

Arricchiamo lo schema relazionale...

Automezzi (Matricola, Marca, Modello, Motorizzazione, Cilindrata, Targa, StatoCorrente)

Marche (Nome, Estera)

Motorizzazioni (Tipo)

Stati (NomeStato)

CambioStati (IdCambioStato, Matricola, NuovoStato, Data, Ora)

```
create table CambioStati (
  IdCambioStato integer auto_increment primary key,
  matricola integer
  references automezzi(matricola)
  on delete cascade on update cascade,
  nuovo_stato varchar (20) not null
  references stati(nomestato)
  on delete cascade on update cascade,
  data_ora timestamp not null default 'now()');
```

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 9 - Il Linguaggio SQL (DML) 15

I Join Esterni



```
SELECT NomeAttr1 AS Alias1, NomeAttr2 AS Alias2, ...  
FROM Tabella1  
LEFT [OUTER] JOIN Tabella2 ON Condizione  
[ WHERE Condizione ] ;
```

```
SELECT NomeAttr1 AS Alias1, NomeAttr2 AS Alias2, ...  
FROM Tabella1  
RIGHT [OUTER] JOIN Tabella2 ON Condizione  
[ WHERE Condizione ] ;
```

```
SELECT NomeAttr1 AS Alias1, NomeAttr2 AS Alias2, ...  
FROM Tabella1  
FULL [OUTER] JOIN Tabella2 ON Condizione  
[ WHERE Condizione ] ;
```
