Il Linguaggio SQL (DML)

Basi di Dati (Corso A-L)

Ingegneria Informatica

Ing. Corrado Santoro

Ripartiamo dallo schema relazionale ...

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

Marche (Nome, Estera)

Motorizzazioni (Tipo)

Stati (NomeStato)

CambioStati (IdCambioStato, Matricola, NuovoStato, Data_Ora)

FoglioDiRiparazione (IdCambioStato, Intervento, Pezzo, Data, Meccanico, IdRiparazione)

Meccanici (Matricola, Cognome, Nome, CodiceFiscale)

Pezzi(Nome)

Realizziamo la query...

- "elenco dei pezzi meccanici sostituiti negli automezzi FIAT"
- Serve il Join di 3 tabelle

$\pi_{Pezzo}\sigma_{Marca='FIAT' \wedge \neg\ Pezzo\ =\ null}$ (Automezzi $\triangleright\triangleleft_{automezzi.matricola=cambiostati.matricola}$ CambioStati $\triangleright\triangleleft_{cambiostati.idcambiostato=foglipdiriparazione.idcambiostato}$ FoglioDiRiparazione)

```
select distinct pezzo from
      automezzi join cambiostati
      on cambiostati.matricola = automezzi.matricola
      join fogliodiriparazione
      on cambiostati.idcambiostato =
            fogliodiriparazione.idcambiostato
      where marca='FIAT' and pezzo is not null;
```


Il Join multiplo

```
SELECT NomeAttr1 [AS Alias1], NomeAttr2 [AS Alias2], …
      FROM
      Tabella1 [JOIN Tabella2 ON Condizione1
      [JOIN Tabella3 ON Condizione2]]
      [ WHERE Condizione3 ] ;
```

- Se la query è però troppo lunga, possiamo usare la "ridenominazione" delle tabelle

```
select distinct pezzo from
      automezzi as a join cambiostati as c
      on c.matricola = a.matricola
      join fogliodiriparazione as f
      on c.idcambiostato = f.idcambiostato
      where marca='FIAT' and pezzo is not null;
```

Realizziamo la query...

- "l'elenco dei pezzi meccanici sostituiti nel 2003 con l'indicazione degli automezzi associati e del meccanico che ha fatto la sostituzione"

$\pi_{\text{Pezzo,automezzi.matricola,cognome,nome}}\sigma_{\text{year(Data)=2003} \wedge \neg \text{ Pezzo = null}}$
(Automezzi $\triangleright\triangleleft_{\text{automezzi.matricola=cambiostati.matricola}}$ CambioStati
$\triangleright\triangleleft_{\text{cambiostati.idcambiostato=foglipdiriparazione.idcambiostato}}$ FoglioDiRiparazione
$\triangleright\triangleleft_{\text{foglipdiriparazione.meccanico=meccanici.matricola}}$ Meccanici)

```
select distinct pezzo,a.matricola,cognome,nome from
      automezzi as a join cambiostati as c
      on a.matricola = c.matricola
      join fogliodiriparazione as f
      on c.idcambiostato = f.idcambiostato
      join meccanici as m
      on f.meccanico = m.matricola
      where year(data)=2003 and pezzo is not null;
```


Semplifichiamo la query...

- "elenco dei pezzi meccanici sostituiti negli automezzi FIAT"
- Se i campi su cui fare il join hanno lo stesso nome è possibile usare "`using (campo1, campo2, …)`"

$\pi_{\text{Pezzo}}\sigma_{\text{Marca='FIAT'} \wedge \neg \text{ Pezzo = null}}$
(Automezzi $\triangleright\triangleleft_{\text{automezzi.matricola=cambiostati.matricola}}$ CambioStati
$\triangleright\triangleleft_{\text{cambiostati.idcambiostato=foglipdiriparazione.idcambiostato}}$ FoglioDiRiparazione)

```
select distinct pezzo from
      automezzi join cambiostati using (matricola)
      join fogliodiriparazione
      on cambiostati.idcambiostato =
            fogliodiriparazione.idcambiostato
      where marca='FIAT' and pezzo is not null;
```

Semplifichiamo la query…

- "l'elenco dei pezzi meccanici sostituiti nel 2003 con l'indicazione degli automezzi associati e del meccanico che ha fatto la sostituzione"
- Se il join è naturale è possibile usare l'operatore "**natural**"

$$\pi_{\text{Pezzo,automezzi.matricola,cognome,nome}}\sigma_{\text{year(Data)=2003} \wedge \neg \text{ Pezzo = null}} (\text{Automezzi} \rhd\lhd \text{CambioStati} \rhd\lhd \text{FoglioDiRiparazione} \rhd\lhd \text{Meccanici})$$

```
select distinct pezzo,a.matricola,cognome,nome from
      automezzi as a natural join cambiostati as c
      natural join fogliodiriparazione as f
      join meccanici as m
      on f.meccanico = m.matricola
      where year(data)=2003 and pezzo is not null;
```


Differenza tra "using" e "natural"

- ***t1* NATURAL JOIN *t2***
  - Esegue il join usando tutti gli attributi con lo stesso nome
- ***t1* JOIN *t2* USING (*a1*, *a2*, …)**
  - Esegue il join usando solo gli attributi specificati (posto che essi abbiano lo stesso nome in *t1* e *t2*)

Ordinamenti

- E' possibile specificare l'ordine in cui i record devono apparire nel risultato della query
- L'ordinamento è in base ad uno o più attributi
- L'ordinamento è in base al **tipo** degli attributi specificati
- E' possibile indicare se l'ordinamento è crescente o descrescente

```
SELECT NomeAttr1 [AS Alias1], NomeAttr2 [AS Alias2], …
       FROM Tabella1 [JOIN Tabella2 ON Condizione1
       [JOIN Tabella3 ON Condizione2]]
       [ WHERE Condizione3 ]
       [ ORDER BY NomeAttr1 [DESC], NomeAttr2 [DESC], …];
```


Ordinamenti: esempio

```
mysql> select * from marche;
+---------+--------+
| nome    | estera |
+---------+--------+
| FIAT    |      0 |
| Opel    |      1 |
| Iveco   |      0 |
| Citroen |      1 |
+---------+--------+
```

```
mysql> select * from marche
    -> order by nome;
+---------+--------+
| nome    | estera |
+---------+--------+
| Citroen |      1 |
| FIAT    |      0 |
| Iveco   |      0 |
| Opel    |      1 |
+---------+--------+
```

```
mysql> select * from marche order by nome desc;
+---------+--------+
| nome    | estera |
+---------+--------+
| Opel    |      1 |
| Iveco   |      0 |
| FIAT    |      0 |
| Citroen |      1 |
+---------+--------+
```

Selezione, Join e Ordinamenti

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

CambioStati (IdCambioStato, Matricola, NuovoStato, Data_Ora)

FoglioDiRiparazione (IdCambioStato, Intervento, Pezzo, Data, Meccanico, IdRiparazione)

- L'elenco degli interventi dell'automezzo con matricola "1", ordinato per data di intervento

```
select intervento, pezzo, data from
  automezzi natural join cambiostati
  natural join fogliodiriparazione
  where automezzi.matricola = 1
  order by data;
```


Selezione, Join e Ordinamenti

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

CambioStati (IdCambioStato, Matricola, NuovoStato, Data_Ora)

FoglioDiRiparazione (IdCambioStato, Intervento, Pezzo, Data, Meccanico, IdRiparazione)

- L'elenco degli interventi di tutti gli automezzi ordinati per matricola e data di intervento

```
select automezzi.matricola, intervento, pezzo, data from
  automezzi natural join cambiostati
  natural join fogliodiriparazione
  order by automezzi.matricola, data;
```

Limitazione dell'output

- E' possibile selezionare i record di una query in base alla loro posizione (num. di riga)

```
SELECT NomeAttr1 [AS Alias1], NomeAttr2 [AS Alias2], …
      FROM Tabella1 [JOIN Tabella2 ON Condizione1
      [JOIN Tabella3 ON Condizione2]]
      [ WHERE Condizione3 ]
      [ ORDER BY NomeAttr1 [DESC], NomeAttr2 [DESC], …]
      [ LIMIT [start,]count ];
```

- **LIMIT *Count***
  - Seleziona dall'output le prime "count" righe
- **LIMIT *Start, Count***
  - Seleziona dall'output "count" righe a partire dalla riga "start" (le righe sono numerate a partire da 0)

Esempi di uso di "LIMIT"

```
mysql> select * from cambiostati;
+-----------------+-----------+-----------------+----------------+
| Id_Cambio_Stato | matricola | nuovo_stato     | data_ora       |
+-----------------+-----------+-----------------+----------------+
|               1 |         1 | In Manutenzione | 20031010120000 |
|               2 |         1 | In Rimessa      | 20031011120000 |
|               3 |         1 | In Servizio     | 20031011131500 |
|               4 |         2 | In Manutenzione | 20041010120000 |
|               5 |         2 | In Rimessa      | 20041011120000 |
+-----------------+-----------+-----------------+----------------+
mysql> select * from cambiostati limit 2;
+-----------------+-----------+-----------------+----------------+
| Id_Cambio_Stato | matricola | nuovo_stato     | data_ora       |
+-----------------+-----------+-----------------+----------------+
|               1 |         1 | In Manutenzione | 20031010120000 |
|               2 |         1 | In Rimessa      | 20031011120000 |
+-----------------+-----------+-----------------+----------------+
mysql> select * from cambiostati limit 3,1;
+-----------------+-----------+-----------------+----------------+
| Id_Cambio_Stato | matricola | nuovo_stato     | data_ora       |
+-----------------+-----------+-----------------+----------------+
|               4 |         2 | In Manutenzione | 20041010120000 |
+-----------------+-----------+-----------------+----------------+
```

Altro esempio di uso di "LIMIT"

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

CambioStati (IdCambioStato, Matricola, NuovoStato, Data_Ora)

FoglioDiRiparazione (IdCambioStato, Intervento, Pezzo, Data, Meccanico, IdRiparazione)

- L'ultimo intervento (in ordine di data) effettuato nell'automezzo con matricola "2"

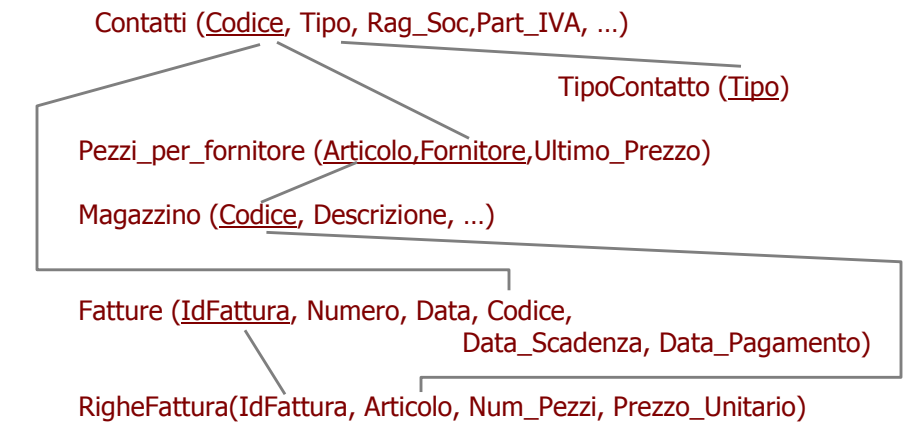
```
select intervento, pezzo, data from
  automezzi natural join cambiostati
  natural join fogliodiriparazione
  where automezzi.matricola = 2
  order by data desc limit 1;
```


Ritorniamo alla lezione 7…

- **Si desidera informatizzare un'azienda commerciale. Occorre gestire l'archivio degli clienti, dei fornitori, il magazzino e la fatturazione in base alle seguenti specifiche:**

1. **per ogni cliente e fornitore occorre registrare il codice, la ragione sociale, la partita iva/codice fiscale, l'indirizzo, la città, etc.;**
2. **per ogni articolo di magazzino occorre indicare il codice, la descrizione, l'ultimo prezzo di acquisto, il prezzo di vendita, la disponibilità corrente, la scorta minima;**
3. **ogni articolo di magazzino può essere fornito da più di un fornitore, ognuno dei quali lo vende ad un prezzo diverso;**
4. **per ogni fornitore occorre mantenere un archivio di fatture ricevute con numero fattura, data fattura, importo totale, scadenza pagamento, data effettuato pagamento, più tutte le varie voci presenti nella fattura;**
5. **ogni voce della fattura fornitori riporta il codice articolo, la quantità acquistata e il prezzo di acquisto;**
6. **per ogni cliente occorre mantenere un archivio di fatture emesse con numero fattura, data fattura, importo totale, scadenza pagamento, data ricevuto pagamento, più tutte le varie voci presenti in fattura;**
7. **ogni voce della fattura clienti riporta il codice articolo, la quantità venduta e il prezzo di vendita.**

Un possibile schema



Operazioni sui campi

- Elenco delle righe della fattura con ID = 1 indicando *articolo, numero pezzi, prezzo unitario* e *prezzo totale per riga (num*prez_un)*

```
mysql> select
    -> articolo,num_pezzi,prezzo_unitario,
    -> num_pezzi * prezzo_unitario as prezzo_totale
    -> from righe_fattura where id_fattura = 1;
+----------+-----------+-----------------+---------------+
| articolo | num_pezzi | prezzo_unitario | prezzo_totale |
+----------+-----------+-----------------+---------------+
|        1 |        15 |          950.00 |      14250.00 |
|        2 |        15 |          140.00 |       2100.00 |
+----------+-----------+-----------------+---------------+
```

Funzioni di Aggregazione

- SQL mette a disposizione le seguenti funzioni di aggregazione
- **`Count(*)`**
  - Conta le righe di una query
- **`Sum(campo)`**
  - Somma i valori di un campo in una query
- **`Avg(campo)`**
  - Determina la media dei valori di un campo in una query
- **`Max(campo)`**
  - Determina il massimo dei valori di un campo in una query
- **`Min(campo)`**
  - Determina il minimo dei valori di un campo in una query

Esempi di funzioni di aggregazione

- Qual è il numero di articoli del mio magazzino?

```
mysql> select count(*) as num_articoli from magazzino;
+---------------+
| num_articoli |
+---------------+
|            2 |
+---------------+
```

- Qual è il prezzo massimo di vendita tra tutti gli articoli?

```
mysql> select max(prezzo_vendita) as prezzo_max from magazzino;
+------------+
| prezzo_max |
+------------+
|    1400.00 |
+------------+
```

Esempi di funzioni di aggregazione

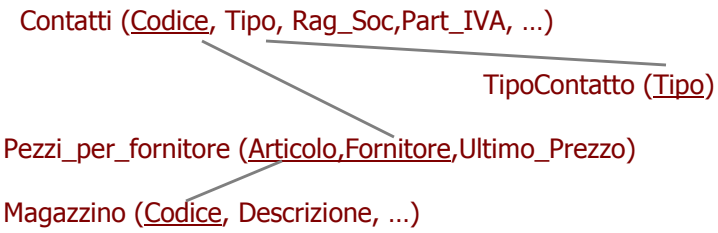
- Qual è il prezzo medio a cui acquisto un certo articolo?

```
mysql> select avg(prezzo) from pezzi_per_fornitore
    -> where articolo = 1;
+-------------+
| avg(prezzo) |
+-------------+
|  900.000000 |
+-------------+
```

- Qual è il prezzo minimo a cui ho venduto un certo articolo?

```
mysql> select min(prezzo_unitario) from righe_fattura
    -> where articolo = 2;
+----------------------+
| min(prezzo_unitario) |
+----------------------+
|               140.00 |
+----------------------+
```


Operazioni di raggruppamento



- Il prezzo di acquisto minimo di ogni articolo
  - **Raggruppare tutti i fornitori di un certo articolo**
  - Determinare il prezzo più basso
  - Restirure le informazioni determinate

Operazioni di raggruppamento

Contatti (Codice, Tipo, Rag_Soc,Part_IVA, …)

TipoContatto (Tipo)

Pezzi_per_fornitore (Articolo,Fornitore,Ultimo_Prezzo)

Magazzino (Codice, Descrizione, …)

```
select magazzino.*, min(prezzo) from
      magazzino join pezzi_per_fornitore
      on magazzino.codice = pezzi_per_fornitore.articolo
      group by articolo;
```


Operazioni di raggruppamento (step 1)

Contatti (Codice, Tipo, Rag_Soc,Part_IVA, …)

Pezzi_per_fornitore (Articolo,Fornitore,Ultimo_Prezzo)

Magazzino (Codice, Descrizione, …)

```
mysql> select codice,descrizione,fornitore,prezzo from magazzino as m
    -> join pezzi_per_fornitore as p on m.codice = p.articolo;
+--------+-------------------+-----------+--------+
| codice | descrizione       | fornitore | prezzo |
+--------+-------------------+-----------+--------+
|      1 | Personal Computer |         1 | 900.00 |
|      1 | Personal Computer |         2 | 850.00 |
|      1 | Personal Computer |         3 | 750.00 |
|      2 | Monitor LCD       |         1 | 144.30 |
|      2 | Monitor LCD       |         2 | 145.50 |
|      2 | Monitor LCD       |         3 | 180.30 |
+--------+-------------------+-----------+--------+
```

Operazioni di raggruppamento (step 2)

Contatti (Codice, Tipo, Rag_Soc,Part_IVA, …)

Pezzi_per_fornitore (Articolo,Fornitore,Ultimo_Prezzo)

Magazzino (Codice, Descrizione, …)

```
mysql> select codice,descrizione,min(prezzo) from
    -> magazzino as m join pezzi_per_fornitore as p
    -> on m.codice = p.articolo
    -> group by codice;
+--------+-------------------+-------------+
| codice | descrizione       | min(prezzo) |
+--------+-------------------+-------------+
|      1 | Personal Computer |      750.00 |
|      2 | Monitor LCD       |      144.30 |
+--------+-------------------+-------------+
```


Selezione sul raggruppamento

- Non è possibile usare **WHERE** come comando per la selezione su un raggruppamento o funzione aggregata
- Si usa invece la clausola **HAVING**

```
mysql> select codice,descrizione,min(prezzo) as min_prezzo from
    -> magazzino as m join pezzi_per_fornitore as p
    -> on m.codice = p.articolo
    -> group by codice having min_prezzo < 700;
+--------+-------------------+------------+
| codice | descrizione       | min_prezzo |
+--------+-------------------+------------+
|      2 | Monitor LCD       |     144.30 |
+--------+-------------------+------------+
```

Subquery

- In SQL, in taluni casi, è possibile specificare una ulteriore query al posto di un parametro
- Esempio:

```
SELECT * FROM MAGAZZINO WHERE CODICE IN
  (SELECT ARTICOLO FROM RIGHE_FATTURA)
```

- Questa query restituisce gli articoli che compaiono in un fattura
- MySQL supporta le subquery a partire dalla versione 4.1
- Però tutti gli altri DBMS le supportano
- In genere è possibile usare dei JOIN, ma sono più costosi
- … ritorneremo sull'argomento

Cancellazione record

- La cancellazione di un o più record si effettua con il comando SQL "DELETE"

```
DELETE FROM tbl_name
       [WHERE where_definition]
       [ORDER BY ...]
       [LIMIT row_count];
```

- Esempio: cancellare tutte le fatture pagate

```
DELETE FROM Fatture
       WHERE Data_pagamento is not null;
```

- Esempio: cancellare tutti gli articoli

```
DELETE FROM Magazzino;
```

Modifica record

- La modifica di uno o più record si effettua con il comando SQL "UPDATE"

```
UPDATE tbl_name
       SET col_name1=expr1 [, col_name2=expr2 ...]
       [WHERE where_definition]
       [ORDER BY ...]
       [LIMIT row_count];
```

- Esempio: modificare la ragione sociale del cliente 3

```
UPDATE CONTATTI SET RAG_SOC='PROVINCIA CT'
       WHERE CODICE=3 AND TIPO='CLIENTE';
```

- Esempio: incrementare del 10% tutti i prezzi di vendita

```
UPDATE MAGAZZINO
       SET PREZZO_VENDITA = PREZZO_VENDITA * 1.1;
```