



Il Linguaggio SQL-DML SQL nei linguaggi di programmazione

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

Ultime note sulla "select"



- Operatori sugli insiemi
 - **UNION** (unione)
 - **INTERSECT** (intersezione)
 - **EXCEPT** (differenza)
 - Sintassi
- ```
SelectExpr UNION [ALL | DISTINCT] SelectExpr;
SelectExpr INTERSECT SelectExpr;
SelectExpr EXCEPT SelectExpr;
```
- MySQL supporta solo "UNION"

Corrado Santoro, Basi di Dati (A-L), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DML)

2

---

---

---

---

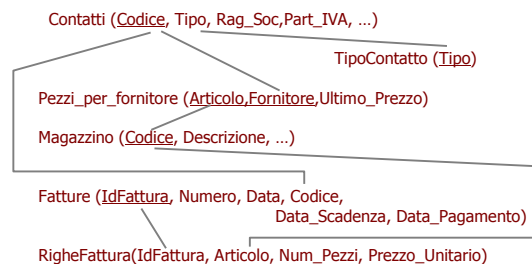
---

---

---

---

## Esempio di "UNION"



Tutti i codici degli articoli venduti ad un certo cliente, più gli articoli forniti da un certo fornitore (ma non necessariamente acquistati da questo)

Corrado Santoro, Basi di Dati (A-L), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DML)

3

---

---

---

---

---

---

---

---

### Esempio di "UNION"

Pezzi\_per\_fornitore (Articolo,Fornitore,Ultimo\_Prezzo)

Fatture (IdFattura, Numero, Data, Codice,  
Data\_Scadenza, Data\_Pagamento)

RigheFattura(IdFattura, Articolo, Num\_Pezzi, Prezzo\_Unitario)

Tutti i codici degli articoli venduti ad un certo cliente, più gli articoli forniti da un certo fornitore (ma non necessariamente acquistati da questo)

```
select articolo from pezzi_per_fornitore
 where fornitore = 2
union
select articolo from righe_fattura natural join fatture
 where codice = 1;
```

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DML) 4

---

---

---

---

---

---

---

---

---

---

---

---

### Viste (Views)

- Spesso ci si trova ad usare una stessa espressione di "select" in diverse parti di un programma
- In altri casi, può rendersi necessario che utenti diversi possano avere accesso (in visualizzazione e/o modifica) ad uno schema relazionale "ristretto" o "modificato" rispetto a quello originale
- Questo accade quando la modalità con cui dobbiamo trattare i dati (**vedere i dati**) è diversa da quella rappresentata dallo schema relazionale usato
- In questi casi è possibile usare le **viste (views)** le quali permettono di creare delle tabelle "fittizie" derivate da una manipolazione dello schema relazionale originale

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DML) 5

---

---

---

---

---

---

---

---

---

---

---

---

### Esempio di viste

Contatti (Codice, Tipo, Rag\_Soc,Part\_IVA, ...)

TipoContatto (Tipo)

Pezzi\_per\_fornitore (Articolo,Fornitore,Ultimo\_Prezzo)

Magazzino (Codice, Descrizione, ...)

Fatture (IdFattura, Numero, Data, Codice,  
Data\_Scadenza, Data\_Pagamento)

1. Tutte le fatture emesse (Numero, Data e cliente associato) scadute e non pagate
2. Tutte le fatture emesse (Numero, Data e cliente associato) pagate secondo la scadenza

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DML) 6

---

---

---

---

---

---

---

---

---

---

---

---

### Esempio di viste

```

1. select * from Fatture natural join Contatti where
Tipo='Cliente' and Data_Scadenza < now() and
Data_Pagamento is null;

2. select * from Fatture natural join Contatti where
Tipo='Cliente' and Data_Pagamento is not null and
Data_Pagamento <= Data_Scadenza;

```

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DML) 7

---

---

---

---

---

---

---

---

---

---

---

---

### Esempio di viste

```

1. select * from Fatture natural join Contatti where
Tipo='Cliente' and Data_Scadenza > now() and
Data_Pagamento is null;

2. select * from Fatture natural join Contatti where
Tipo='Cliente' and Data_Pagamento is not null and
Data_Pagamento <= Data_Scadenza;

```

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DML) 8

---

---

---

---

---

---

---

---

---

---

---

---

### Esempio di viste

```

create view Fatture_Clienti as
select * from Fatture natural join Contatti
where Tipo='Cliente';

• select * from Fatture_Clienti where Data_Scadenza >
now() and Data_Pagamento is null;
• select * from Fatture_Clienti where Data_Pagamento is
not null and Data_Pagamento <= Data_Scadenza;

```

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DML) 9

---

---

---

---

---

---

---

---

---

---

---

---

## Transazioni

- A volte una singola query non è sufficiente per effettuare una determinata operazione
- Cioè, per alcune operazioni è necessario eseguire diversi comandi SQL in sequenza
- Tali comandi sono però legati fra loro in modo "stretto"
- Esempio: Trasferimento di una somma di denaro da un conto corrente all'altro
  - `UPDATE conti_correnti SET totale = totale + 2000 WHERE NumConto = 321245;`
  - `UPDATE conti_correnti SET totale = totale - 2000 WHERE NumConto = 1098;`
- **Cosa accade se tra la prima e la seconda operazione avviene un errore o se la seconda operazione fallisce?**

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DML) 10

---

---

---

---

---

---

---

---

---

---

## Transazioni

- Esempio: Trasferimento di una somma di denaro da un conto corrente all'altro
  - `UPDATE conti_correnti SET totale = totale + 2000 WHERE NumConto = 321245;`
  - `UPDATE conti_correnti SET totale = totale - 2000 WHERE NumConto = 1098;`
- **Cosa accade se tra la prima e la seconda operazione avviene un errore o se la seconda operazione fallisce?**
- Ci saranno 2000 € che si saranno materializzati dal nulla, dato che non saranno tolti dal conto corrente del debitore
- In altre parole, il sistema si troverà in uno stato **inconsistente**
- Occorre incapsulare le istruzioni SQL "legate fra loro" in una **transazione**

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DML) 11

---

---

---

---

---

---

---

---

---

---

## Transazioni... a piccoli passi

- Esempio: Trasferimento di una somma di denaro da un conto corrente all'altro
  - `UPDATE conti_correnti SET totale = totale + 2000 WHERE NumConto = 321245;`
  - `UPDATE conti_correnti SET totale = totale - 2000 WHERE NumConto = 1098;`
- **Prima proprietà:**
  - Occorre garantire che le operazioni siano fatte **tutte o nessuna**
  - L'insieme delle operazioni deve essere dunque **Atomico** (cioè o tutto o niente)

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DML) 12

---

---

---

---

---

---

---

---

---

---

## Transazioni... a piccoli passi



- Esempio: Trasferimento di una somma di denaro da un conto corrente all'altro
  - `UPDATE conti_correnti SET totale = totale + 2000 WHERE NumConto = 321245;`
  - `UPDATE conti_correnti SET totale = totale - 2000 WHERE NumConto = 1098;`
- **Seconda proprietà:**
  - Occorre garantire che il sistema venga lasciato sempre in uno stato **Consistente**
  - Se dunque un'istruzione SQL fallisce, tutte le altre devono essere "annullate" (disfatte)
  - In base all'esempio, se il secondo aggiornamento fallisce, anche il primo aggiornamento deve essere annullato

---

---

---

---

---

---

---

---

---

---

## Transazioni... a piccoli passi



- Esempio: Trasferimento di una somma di denaro da un conto corrente all'altro
  - `UPDATE conti_correnti SET totale = totale + 2000 WHERE NumConto = 321245;`
  - `UPDATE conti_correnti SET totale = totale - 2000 WHERE NumConto = 1098;`
- **Terza proprietà:**
  - Occorre garantire che l'esecuzione delle operazioni di una transazione non influenzino altre eventuali transazioni concorrenti fintanto che la prima non abbia concluso la sua esecuzione
  - Ogni transazione deve essere **Isolata**
  - In base all'esempio, il primo aggiornamento non dovrà essere visibile fino a che anche il secondo aggiornamento non sarà completato.
  - Oppure... eventuale altre transazioni/comandi concorrenti dovranno attendere che la transazione in corso sia completata

---

---

---

---

---

---

---

---

---

---

## Transazioni... a piccoli passi



- Esempio: Trasferimento di una somma di denaro da un conto corrente all'altro
  - `UPDATE conti_correnti SET totale = totale + 2000 WHERE NumConto = 321245;`
  - `UPDATE conti_correnti SET totale = totale - 2000 WHERE NumConto = 1098;`
- **Quarta proprietà:**
  - Occorre garantire che il risultato di una transazione duri nel tempo, cioè sia **Persistente (Durable)**

---

---

---

---

---

---

---

---

---

---

## Transazioni

- Quattro Proprietà
  - Atomicity
  - Consistency
  - Isolation
  - Durability
- ACID properties
- Una transazione è un insieme di operazioni che soddisfa le ACID properties

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DML) 16

---

---

---

---

---

---

---

---

---

---

---

---

## Transazioni in SQL

- Le transazioni in SQL sono evidenziate dal fatto che i comandi sono racchiusi tra
  - **BEGIN;**
  - **COMMIT | ROLLBACK;**

```

BEGIN;
UPDATE conti_correnti SET
 totale = totale + 2000
WHERE NumConto = 321245;
UPDATE conti_correnti SET
 totale = totale - 2000
WHERE NumConto = 1098;
COMMIT;

```

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DML) 17

---

---

---

---

---

---

---

---

---

---

---

---

## DBMS, Utenti ed SQL

- L'accesso al DBMS deve avvenire in modo "controllato"
- Chi accede al DBMS deve essere conosciuto
- Deve avere i privilegi per fare le operazioni richieste

```
mysql> select * from clienti;
```

Software Applicativo  
(scritto in Java, C, C++)  
Libreria di accesso al DB  
**DBMS Client**

Comandi in SQL  
digitati  
direttamente  
dall'utente

Comandi in SQL

DBMS (Server)

↔

Database

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DML) 18

---

---

---

---

---

---

---

---

---

---

---

---

## DBMS, Utenti ed SQL

- Prima di inviare comandi SQL occorre fare una **connessione** al DBMS, specificando il proprio **nome utente** e la propria **password**
- Il DBMS verificherà se l'utente è presente
- Per ogni comando inviato, il DBMS verificherà se l'utente ha i privilegi per eseguirlo
- Al termine dell'uso del DBMS, è necessario fare una **disconnessione**

```
mysql> select * from clienti;
```

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DBMS) 19

---

---

---

---

---

---

---

---

---

---

## DBMS, Utenti ed SQL

- Queste operazioni sono effettuate in modo automatico dai software applicativi
- L'uso di un command-line client prevede invece che l'utente venga specificato in modo esplicito

```
C:\>mysql -u corrado
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 5 to server version: 4.0.21-debug

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select * from clienti;
```

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DBMS) 20

---

---

---

---

---

---

---

---

---

---

## DBMS, Utenti ed SQL

- Esiste un utente denominato **"Database Administrator (SYSDBA)"** che possiede tutti i permessi, compreso la possibilità di creare altri utenti
- In MySQL questo utente si chiama **"root"**

```
C:\>mysql -u root
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 5 to server version: 4.0.21-debug

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select * from clienti;
```

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DBMS) 21

---

---

---

---

---

---

---

---

---

---

## Gestione Utenti in SQL

- Creazione utenti
  - CREATE USER username;
  - Oppure tramite un'utility a riga di comando
- Cancellazione utenti
  - DROP USER username;
  - Oppure tramite un'utility a riga di comando

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DBMS) 22

---

---

---

---

---

---

---

---

---

---

---

---

## Gestione Utenti in SQL

- Impostazione privilegi
  - GRANT privilegio ON risorsa TO utente;
  - GRANT ALL ON risorsa TO utente;
  
  - GRANT SELECT ON ARTICOLI TO MAGAZZINIERE;
  - GRANT ALL ON ARTICOLI TO CONTABILE;
- Revoca privilegi
  - REVOKE privilegio ON risorsa FROM utente;
  - REVOKE ALL ON risorsa FROM utente;
  
  - REVOKE SELECT ON ARTICOLI FROM MAGAZZINIERE;
  - REVOKE ALL ON ARTICOLI FROM CONTABILE;

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DBMS) 23

---

---

---

---

---

---

---

---

---

---

---

---

## SQL nei linguaggi di programmazione

- L'uso di SQL nei comuni linguaggi di programmazione (C, Java, Visual Basic, etc.) avviene attraverso l'uso di opportune librerie che permettono l'accesso al DBMS
- Le librerie sono dipendenti del tipo di DBMS che si sta usando
- Le funzioni fornite permettono la connessione al DBMS, l'invio di comandi SQL e la traduzione dei tipi SQL in tipi nativi del linguaggio utilizzato

```

 graph TD
 subgraph Client_Stack [DBMS Client]
 direction TB
 A[Software Applicativo
(scritto in Java, C, C++)]
 B[Libreria di accesso al DB]
 end
 Client_Stack --> C[DBMS Server]
 C <--> D[(Database)]

```

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DBMS) 24

---

---

---

---

---

---

---

---

---

---

---

---



## Esempio: Programma C che usa un database Postgres

```
#include <libpq-fe.h>

int main (int argc, char * argv[])
{
 PGconn *conn;
 PGresult *result;
 int num, i;
 conn = PQconnectdb ("user=root dbname=azienda");
 result = PQexec(conn,
 "select codice, descrizione from magazzino");
 num = PQntuples(result);
 printf ("Number of rows: %d\n", num);
 for (i = 0; i < num; i++) {
 printf ("Codice %s, Descrizione %s\n",
 PQgetvalue (result, i, 0),
 PQgetvalue (result, i, 1));
 }
 PQfinish (conn);
}
```

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DMS)

25

---

---

---

---

---

---

---

---

## Accesso Indipendente dal DB

- L'esempio precedente mostra l'uso di un DBMS Postgres con linguaggio C
- Se però vogliamo usare lo stesso programma C per accedere ad un DB con la stessa struttura, ma gestito da un altro DBMS, non possiamo farlo
- Questo perché stiamo usando le librerie specifiche di un DBMS
- Tuttavia sarebbe opportuno introdurre un **livello di astrazione** che consenta di realizzare l'accesso ad un DB in modo **indipendente** dal DBMS sottostante

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DMS)

26

---

---

---

---

---

---

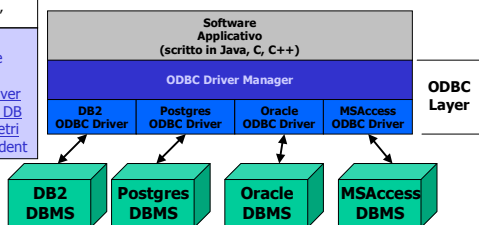
---

---

## Open Data Base Connectivity (ODBC)

- Lo standard ODBC introduce questo livello di astrazione consentendo lo sviluppo di applicazioni in modo indipendente dal DBMS

Tabella dei "DataSource"  
Per ogni DataSource è specificato  
• il tipo di driver  
• il nome del DB  
• altri parametri driver-dependent



Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DMS)

27

---

---

---

---

---

---

---

---

## Open Data Base Connectivity (ODBC)

- L'applicazione specifica la connessione con un Data Source Name (DSN)
- Il Driver Manager cerca, nella tabella dei DataSource, il DataSource con il nome specificato dall'applicazione

Tabella dei "DataSource"

Per ogni DataSource è specificato:

- il tipo di driver
- il nome del DB
- altri parametri driver-dependent

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DBMS) 28

---

---

---

---

---

---

---

---

---

---

---

---

## Open Data Base Connectivity (ODBC)

- Se il DSN esiste, vengono determinati il **driver** ed il **database** da usare
- Il driver viene selezionato in modo che quest'ultimo possa effettuare la connessione reale con il DBMS

Tabella dei "DataSource"

Per ogni DataSource è specificato:

- il tipo di driver
- il nome del DB
- altri parametri driver-dependent

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DBMS) 29

---

---

---

---

---

---

---

---

---

---

---

---

## Open Data Base Connectivity (ODBC)

- Una volta stabilita la connessione, l'applicazione dialoga direttamente con il DBMS attraverso il driver manager ed il driver specifico

Tabella dei "DataSource"

Per ogni DataSource è specificato:

- il tipo di driver
- il nome del DB
- altri parametri driver-dependent

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 12 - Il Linguaggio SQL (DBMS) 30

---

---

---

---

---

---

---

---

---

---

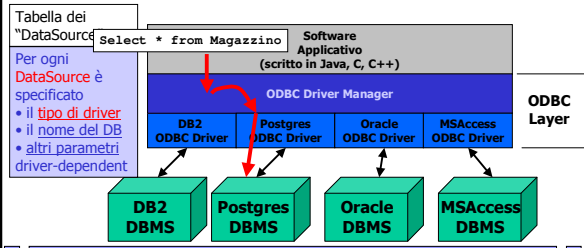
---

---

# Open Data Base Connectivity (ODBC)



- Se poi vogliamo usare un altro DBMS, sarà sufficiente cambiare il driver nella tabella dei DataSource, in corrispondenza con il nostro DSN



---

---

---

---

---

---

---

---