




Consorzio per la formazione e la ricerca in Ingegneria dell'Informazione

Structured Query Language *parte 3*


Non solo interrogazioni...

Docenti: Gennaro Pepe & Alberto Castelli
CEFRIEL
Information & Communication TEchnology for Medicine Area

Basi di Dati



SQL: non solo *SELECT*



Comandi SQL (Structured Query Language)		
DDL	DML	DCL
CREATE	INSERT	GRANT
	UPDATE	REVOKE
	DELETE	
	SELECT	

- DDL: Data Definition Language definizione schema DB
- DML: Data Management Language modifica istanze
- DCL: Data Control Language controllo accessi

Structured Query Language - parte 3

- 2 -

Basi di Dati



MySQL by examples - Console⁽¹⁾



Come lanciare la console di MySQL

- Nello Start Menu -> Programs cercare *MySQL*
- La voce *MySQL* contiene 3 comandi:

MySQL Client

Start MySQL Server

Stop MySQL Server

- Per prima cosa lanciare *Start MySQL Server*:
si aprirà una finestra DOS (la "console")

NON CHIUDERLA FINO ALLA FINE



MySQL by examples - Console⁽²⁾



Come lanciare la console di MySQL (continua)...

- Sempre da *Start->Programs->MySQL*
lanciare *MySQL Client*:
si apre una sessione col server (una nuova
finestra DOS) **NON CHIUDERLA MAI**
- Nella finestra si vede il prompt di *MySQL*:
mysql>

POSSIAMO INIZIARE...





MySQL: Creazione DB



NB: tutte le istruzioni devono terminare con **;** o **\G**

- Al momento dell'installazione, MySQL contiene il DB "MySQL" e il DB "Test", vuoto
`mysql> SHOW DATABASES;`
- Per prima cosa andiamo a creare il DB su cui lavoreremo:
`mysql> CREATE DATABASE cognome;`
`mysql> SHOW DATABASES;`
- Per lavorare con il DB appena creato dobbiamo selezionarlo:
`mysql> USE cognome;`



SQL come DDL



- SQL presenta tutti i costrutti necessari per la definizione e le modifica degli schemi di un DB
 - ▶ Normalmente si affianca alle interfacce proprietarie fornite dai produttori di DBMS relazionali
- Il comando **CREATE TABLE** permette di creare tabelle specificando i nomi ed i tipi delle colonne
- Il comando **ALTER TABLE** consente di aggiungere (attraverso l'ADD) e, talvolta, cancellare (attraverso il DROP) colonne di una tabella
- Il comando **DROP TABLE** consente di cancellare tabelle



MySQL by examples - Creare DB



Tabelle:

Esistono diversi tipi di tabelle con caratteristiche diverse, soprattutto riguardo ai tipi di dati ammessi ed alla gestione delle transazioni. I tipi più comuni sono:

MyIsam (Default) – Heap – Merge	non transazionali
InnoDB – Gemini – Berkley	transazionali



Creazione di tabelle



CREATE TABLE *NomeTabella*

```
(
  NomeAttributo Dominio [ValoreDiDefault] [Vincoli],
  NomeAttributo Dominio [ValoreDiDefault] [Vincoli],
  AltriVincoli
)
```

- Ogni tabella viene definita con un nome e un elenco di attributi che ne definiscono lo schema
- Ogni attributo è caratterizzato da un nome, un dominio e dei vincoli



MySQL by examples - Tabelle (1)



Creare una tabella:

Si usano le istruzioni:

```
mysql> CREATE TABLE nome_tabella (  
mysql> nome_campo TIPO,  
mysql> nome_campo TIPO,  
mysql> nome_chiave TIPO NOT NULL,  
mysql> PRIMARY KEY (nome_chiave));
```

Se l'operazione va a buon fine appare una scritta del tipo:

```
mysql> Query OK, 0 rows affected (0.10 sec)
```



Domini: Column Types



- MySQL supporta diversi tipi di domini, che possono essere raggruppati in tre categorie:
 - ▶ numeric types,
 - ▶ date and time types,
 - ▶ string (character) types



Domini: Numeric Types



- **INT[(M)] [UNSIGNED] [ZEROFILL]**
 - ▶ Intero da -2.147.483.648 a 2.147.483.647
 - ▶ Se UNSIGNED da 0 a $2^{32}-1$
 - ▶ es: CAP INT(5) UNSIGNED ZEROFILL
- **FLOAT(precision) [ZEROFILL]**
- **FLOAT[(M,D)] [ZEROFILL]**
 - ▶ Floating-point
- **NUMERIC(M,D) [ZEROFILL]**
 - ▶ Floating-point "unpacked": salvato come stringa di caratteri
 - ▶ es: prezzo NUMERIC(5,2) numeri tra -999,99 e 999,99



Domini: Date & Time Types



- **DATE**
 - ▶ Data: ammessi valori da '1000-01-01' a '9999-12-31'
 - ▶ es: data_di_nascita DATE
- **TIME**
 - ▶ Viene visualizzato nel formato 'HH:MM:SS'
- **DATETIME**
 - ▶ Visualizzato nel formato 'YYYY-MM-DD HH:MM:SS'
- **TIMESTAMP[(M)]**
 - ▶ Visualizzato nel formato YYYYMMDDHHMMSS
 - ▶ Comodo per memorizzare il momento dell'inserimento o aggiornamento di record, in quanto può essere assegnato automaticamente



Domini: String Types



- CHAR(N) [BINARY]
 - ▶ Permette di rappresentare singoli caratteri o stringhe di lunghezza fissata (N caratteri < 256)
 - ▶ Se non è specificato BINARY maiuscole e minuscole sono indifferenti (case-insensitive)
 - ▶ es: cognome CHAR(20)
- VARCHAR(N) [BINARY]
 - ▶ Stringhe di N < 256 caratteri
 - ▶ Se vengono inserite stringhe più corte di N viene allocata solo la memoria necessaria
 - ▶ es: note VARCHAR(200) (MySQL le preferisce!!)
- TEXT
 - ▶ Stringhe fino a $2^{16}-1$ caratteri



Domini: String Types



- ENUM('value1','value2',...)
 - ▶ Stringa che può avere un solo valore, scelto tra quelli proposti, oppure NULL o ""
 - ▶ es: auto ENUM('opel', 'fiat', 'renault')
- SET('value1','value2',...)
 - ▶ Può assumere zero o più valori tra quelli proposti



Specifica di valori di default



- Il valore di default consente di specificare per ogni attributo il valore che assume automaticamente all'inserimento di una nuova riga nella tabella
 - DEFAULT (*GenericoValore* | *user* | *null*)**
 - ▶ *GenericoValore* → valore del dominio
 - ▶ *user* → identificativo dell'utente
 - ▶ *null* → è il valore inserito se non si indica nulla
- Esempio:
 - ▶ **NumeroFigli INTEGER DEFAULT 0**



Vincoli intrarelazionali (1)



- NOT NULL
 - ▶ Indica che non è ammesso un valore nullo per un certo attributo
 - **Cognome CHAR(20) NOT NULL**
- UNIQUE
 - ▶ Si specifica per un attributo o un insieme di attributi e impone che i valori siano una superchiave
 - Esempio 1
 - Nome CHAR(20) NOT NULL UNIQUE,
 - Cognome CHAR (20) NOT NULL UNIQUE,
 - Esempio 2
 - Nome CHAR(20) NOT NULL,
 - Cognome CHAR(20) NOT NULL,
 - UNIQUE(Nome, Cognome)



Vincoli intrarelazionali (2)



- Chiave Primaria
 - ▶ E' possibile impostare il vincolo di chiave primaria una sola volta per tabella
 - E' invece possibile usare più volte UNIQUE e NOT NULL
 - ▶ Il vincolo può essere definito su un attributo o su un insieme di attributi
- Vincoli di integrità generici
 - ▶ SQL-2 ha introdotto una clausola per esprimere ulteriori vincoli: **CHECK(condizione)**
- Esempio
 - Nome CHAR(20),
 - Cognome CHAR(20),
 - Stipendio INTEGER(9) CHECK(Stipendio>0),
 - PRIMARY KEY (Cognome, Nome)



Vincoli interrelazionali (1)



- E' possibile definire *vincoli di integrità referenziale*
 - ▶ Crea un legame fra i valori di un attributo della tabella corrente (interna) e i valori di un attributo di un'altra tabella (esterna)
 - ▶ Impone che per ogni riga della tabella interna il valore, se diverso da nullo, sia presente nei valori dell'attributo della tabella esterna
 - ▶ L'attributo della tabella esterna deve essere **UNIQUE** anche se tipicamente è chiave primaria
 - ▶ Il tutto può essere imposto anche su un insieme di attributi
- Si usano i costrutti **REFERENCES** e **FOREIGN KEY**



Vincoli interrelazionali (2)



```
CREATE TABLE Impiegato
(
  Matricola char(6) primary
    key,
  Nome char(20) not null,
  Cognome char(20) not null,
  Dipart char(15) references
    Dipartimento(NomeDip),
  Unique (Nome,Cognome)
);
```

```
CREATE TABLE Impiegato
(
  Matricola char(6) primary key,
  Nome char(20) not null,
  Cognome char(20) not null,
  Dipart char(15) references
    Dipartimento(NomeDip),
  Unique (Nome,Cognome)
  Foreign key
    (Nome, Cognome)
  references
    Anagrafica(Nome, Cognome)
);
```



Vincoli interrelazionali (3)



- Per i vincoli intrarelazionali quando c'è una violazione il sistema segnala l'errore
- Per i vincoli interrelazionali è possibile indicare al sistema quale reazione adottare
 - ▶ Le reazioni sono indicate per rispondere a modifiche sulla tabella esterna (master) e si applicano sulla tabella interna (slave)
 - ▶ Alle modifiche o cancellazioni è possibile reagire in diversi modi
 - CASCADE
 - SET NULL
 - SET DEFAULT
 - NO ACTION



Vincoli interrelazionali (4)



```
CREATE TABLE Impiegato
(
  Matricola CHAR(6) PRIMARY KEY,
  Nome CHAR(20) NOT NULL,
  Cognome CHAR(20) NOT NULL,
  Dipart CHAR(15) REFERENCES Dipartimento(NomeDip),
  UNIQUE (Nome,Cognome)
  FOREIGN KEY
  (Nome, Cognome) REFERENCES Anagrafica(Nome,
  Cognome)
  ON DELETE SET NULL
  ON UPDATE CASCADE,
);
```



MySQL by examples - Tabelle (2)



Esempio:

```
mysql> CREATE TABLE ristorante (
mysql> NomeRist VARCHAR(15) NOT NULL,
mysql> Localita VARCHAR(20) NOT NULL,
mysql> Indirizzo VARCHAR(20),
mysql> Telefono VARCHAR(15),
mysql> #Posti int,
mysql> PRIMARY KEY (NomeRist, Localita));
```



MySQL by examples - Tabelle (4)



Una volta create le tabelle necessarie per la nostra base di dati, è possibile visualizzare l'elenco delle tabelle del database corrente col comando:

```
mysql> SHOW TABLES;
```

mentre è possibile visualizzare l'elenco dei campi costituenti una tabella col comando:


```
mysql> SHOW COLUMNS FROM nome_tabella;
```




Inserimento e modifica di dati



- SQL permette anche di modificare il contenuto di una base di dati (modificare l'istanza)
- **INSERT** permette di inserire in una tabella un'intera nuova riga, oppure una parte di una nuova riga, oppure i valori ricavati da una query
- **UPDATE** permette di modificare i valori di una singola riga oppure di un insieme di righe
- **DELETE** permette di cancellare alcune o tutte le righe di una tabella



Inserimento e modifica di dati - es. (1)



persone

Cod_fisc	Nome	Cognome
cod1	Marco	Pace
cod3	Sara	Gelo

clienti

Cod_fisc	Nome	Cognome	Prodotto
cod8	Luigi	Strada	re456
cod7	Lara	Santi	f54t

insert into persone
(Cod_fisc, Nome)
values (cod9, Lucia);

insert into persone
select Cod_fisc, Nome,
Cognome
from clienti;

persone

Cod_fisc	Nome	Cognome
cod1	Marco	Pace
cod3	Sara	Gelo
cod9	Lucia	


clienti

Cod_fisc	Nome	Cognome
cod1	Marco	Pace
cod3	Sara	Gelo
cod8	Luigi	Strada
cod7	Lara	Santi


Structured Query Language - parte 3

- 25 -

Basi di Dati



Inserimento e modifica di dati - es. (2)



persone

Cod_fisc	Nome	Cognome	Compenso_mensile
cod1	Marco	Pace	85
cod4	Luigi	Gioia	105
cod3	Sara	Gelo	102

update persone
set Cod_fisc=cod11
where (Nome=Marco) AND
(Cognome=Pace);

delete from persone
where
Compenso_mensile<100;

persone

Cod_fisc	Nome	Cognome	Compenso_mensile
cod11	Marco	Pace	85
cod4	Luigi	Gioia	105
cod3	Sara	Gelo	102

persone

Cod_fisc	Nome	Cognome	Compenso_mensile
cod4	Luigi	Gioia	105
cod3	Sara	Gelo	102

Structured Query Language - parte 3

- 26 -

Basi di Dati



MySQL by examples - Inserire dati



(1)

Popolare le tabelle:

le tabelle vengono popolate inserendo, uno alla volta, i record (tuple) che le compongono con le seguenti istruzioni

```
mysql> INSERT INTO nome_tabella VALUES
mysql> ('contenuto_1°_campo', 'contenuto_2°_campo',...);
```

i valori di tipo stringa vanno inseriti tra apici, i valori numerici non vanno tra apici



MySQL by examples - Inserire dati



(2)

Esempio:

la tabella inserita era:

```
RISTORANTE(NomeRist, Località, Indirizzo,
Telefono,#Posti)
```

un record si inserisce (considerando l'ordine dei campi) con:

```
mysql> INSERT INTO ristorante VALUES
mysql> ('Da Peppino', 'Milano', 'via Toselli 10',
mysql> '0234765545', 134);
```

se l'operazione va a buon fine risulta:

```
Query OK, 1 row affected (0.02 sec.)
```



MySQL by examples - Query (1)



Esempi di query MySQL:

- estrazione di un record (tupla) dalla tabella

```
mysql> SELECT * from ristorante
mysql> WHERE (NomeRist = "Da Peppino");
```

il risultato avrà la forma:

NomeRist	Localita	Indirizzo	Telefono	#Posti
Da peppino	Milano	Via Toselli 10	0234765545	134



MySQL by examples - Query (2)



Esempi di query MySQL:

- eliminazione di un record da una tabella

```
mysql> DELETE FROM ristorante
mysql> WHERE (Localita = "Milano");
```

tutti i record che contengono il valore Milano nel campo Località vengono eliminati dalla tabella.

Il risultato è una tabella come quella mostrata nell'esempio precedente dove non compaiono i record eliminati.



MySQL by examples - Query (3)



Esempi di query MySQL:

- modifica del valore di un campo di un record già presente nella tabella

```
mysql> UPDATE ristorante SET #Posti = 150
mysql> WHERE #Posti = 134;
```

modifica tutti i record che avevano #Posti pari a 134 modificando il #Posti a 150.

Il risultato è ancora la nuova tabella visualizzata come mostrato nel primo esempio.



MySQL by examples - Query (4)



Esempi di query MySQL:

- ricerca del valore di un campo di un record identificato da una chiave (o altro campo)

```
mysql> SELECT #Posti FROM ristorante
mysql> WHERE (NomeRist = "Da Peppino");
```

il risultato sarà:

#Posti
150

Il valore era stato modificato precedentemente!



Altre operazioni



- **Modifica:**

Cod_fisc	Nome	Compenso_mensile
----------	------	------------------


```
alter table persone
add Mesi_di_lavoro INTEGER(2);
```

Cod_fisc	Nome	Compenso_mensile	Mesi_di_lavoro
----------	------	------------------	----------------


- **Cancellazione:**

```
drop table persone;
```

Structured Query Language - parte 3
- 33 -
Basi di Dati



MySQL by examples - Modificare DB



- Rinominare una tabella


```
mysql> ALTER TABLE ristorante RENAME mio_ristorante;
```

 ora la tabella si chiama mio_ristorante.
- Aggiungere un nuovo campo ad una tabella


```
mysql> ALTER TABLE mio_ristorante ADD Cuoco varchar(20);
```

 aggiunge il campo Cuoco di tipo VARCHAR(20) alla tabella
- Eliminare un campo


```
mysql> ALTER TABLE mio_ristorante DROP Cuoco;
```

Structured Query Language - parte 3
- 34 -
Basi di Dati



Controllo dell'accesso (1)




- Un DBMS consente di proteggere tutte le risorse di una base di dati (tabelle, viste,...)
- Il controllo dell'accesso di basa sul concetto di **privilegio** che viene specificato da
 - ▶ **Risorsa**
 - ▶ **Utente che concede il privilegio**
 - ▶ **Utente che riceve il privilegio**
 - ▶ **Azione che viene consentita**
 - ▶ **Se può propagare il privilegio ad altri**
- Il creatore di una risorsa ha tutti i privilegi
- I possibili privilegi comprendono: insert, update, delete, select, references, usage
- I privilegi vengono concessi con **GRANT** e **REVOKE**




Controllo dell'accesso (2)



- **Esempi**
 - ▶ **GRANT SELECT ON dipartimento TO Stefano**
 - ▶ **GRANT ALL PRIVILEGES ON Impiegato TO Paolo, Riccardo**
 - ▶ **GRANT DELETE ON Persone TO Cesare WITH GRANT OPTION**
 - ▶ **REVOKE DELETE ON Persone TO Cesare**
 - Il comando **REVOKE** può essere attuato solo da chi ha concesso il privilegio

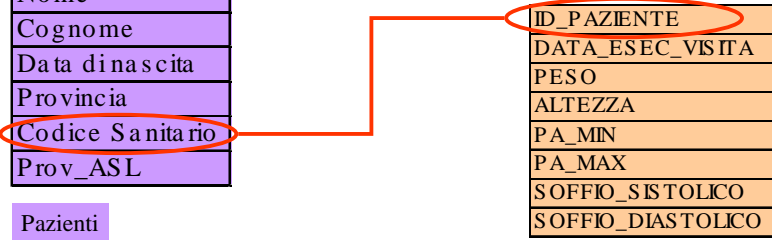


Esercizio₍₁₎



Crea il DB mio_cognome

Nome
Cognome
Data di nascita
Provincia
Codice Sanitario
Prov_ASL




ID_PAZIENTE
DATA_ESEC_VISITA
PESO
ALTEZZA
PA_MIN
PA_MAX
SOFFIO_SISTOLICO
SOFFIO_DIASTOLICO

Pazienti


Visite

Crea queste tabelle...

Structured Query Language - parte 3
- 37 -
Basi di Dati



Esercizio₍₂₎



Inserisci dei dati di esempio...

Nome	Cognome	Data di nascita	Provincia	Codice Sanitario	Provincia ASL
BRUNO	ACARNICI	04/04/32	VR	000SB198	MI
FRANCESCO	APICELLA	18/07/40	SA	001KC198	MI
ALDO	RAGAZZI	12/09/56	PN	001SC111	BG
MARIA	GIORGINI	28/02/25	BG	001TJ212	BG
ROBERTO	AGAZZI	14/09/46	BG	001TT862	BG
GIOVANNA	RINALDI	22/12/11	MI	002WL651	MI
ANGELA	GALBANI	31/07/39	BG	003CW863	BG

Pazienti

ID_PAZIENTE	DATA_ESEC_VISITA	PESO	ALTEZZA	PA_MIN	PA_MAX	SOFFIO_SISTOLICO	SOFFIO_DIASTOLICO
000SB198	27/03/01	80	180	80	120	N	S
001KC198	22/09/00	74	160	65	140	N	N
001KC198	15/12/00	74	160	60	150	N	S
001SC111	23/03/01	80	155	75	140	S	N
001TJ212	15/03/01	65	165	80	120	S	N
001TT862	29/03/01	74	162	85	140	N	N
002WL651	22/02/01	90	180	80	140	N	N
003CW863	20/11/00	90	177	80	140	N	N

Visite

Structured Query Language - parte 3
- 38 -
Basi di Dati



Esercizio₍₃₎



Esegui le query...

- Visualizza i nomi e cognomi delle persone cui è stato riscontrato un soffio sistolico
- Visualizza il numero di visite realizzate nel 2001
- Visualizza il valore medio della pressione max (PA_MAX) e della pressione minima (PA_MIN) delle persone con più di 60 anni

- Modifica la tabella Pazienti inserendo il campo Cod_ASL. Inserisci in tale colonna i valori 001 se la PROV_ASL è Bergamo e 008 se è Milano




MySQL by examples - Exit



Per concludere: come scollegarsi da MySQL

- nella finestra di client digitare

mysql> EXIT

- andare in Start->Programs->MySQL e scegliere l'opzione "Stop MySQL Server"
- appena compare la scritta - Shutdown Complete - chiudere la finestra dal pulsante  in alto a destra.



Bibliografia



- Paolo Atzeni, Stefano Ceri,
Stefano Paraboschi, Riccardo Torlone
Basi di Dati - Seconda edizione
settembre 1999 - McGraw Hill
 - ▶ 4.1, 4.3, 4.4, 4.5
- Teach Yourself SQL in 10 minutes
second edition
Ben Forta
2001, SAMS

