



Il Linguaggio SQL (DDL)

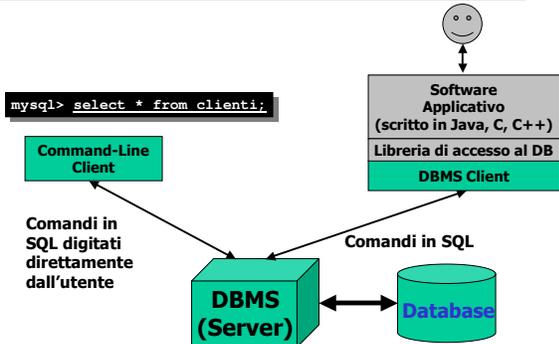
Basi di Dati (Corso A-L)
Ingegneria Informatica
Ing. Corrado Santoro

Introduzione a SQL



- SQL = Structured Query Language
 - E' il linguaggio di riferimento per i DBMS
- E' un linguaggio **dichiarativo**
 - Specifica l'obiettivo da raggiungere, non indica in che modo raggiungerlo
- Si suddivide in
 - DDL: Data Definition Language
 - DML: Data Manipulation Language
- Riferimenti: Libro di testo e manuale di MySQL
www.mysql.org
- Comandi **case-insensitive** e **sempre** terminati dal carattere ";"

Dov'è SQL?



SQL Data Types: Interi

- **Intero** `INTEGER [(N)] [UNSIGNED]`
 - Range $[-2^{31}, +2^{31}-1]$ (signed)
 - Range $[0, 2^{32}-1]$ (unsigned)
 - "(N)" può specificare opzionalmente il numero massimo di cifre

- **Esempi**
 - Un intero con segno `INTEGER`
 - Un intero senza segno `INTEGER UNSIGNED`
 - Un intero con segno a 8 cifre `INTEGER (8)`
 - Un intero senza segno a 5 cifre `INTEGER (5) UNSIGNED`

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 8 - Il Linguaggio SQL 4

SQL Data Types: Interi

- **Intero corto** `SMALLINT [(N)] [UNSIGNED]`
 - Range $[-2^{15}, +2^{15}-1]$ (signed)
 - Range $[0, 2^{16}-1]$ (unsigned)

- **Intero lungo** `BIGINT [(N)] [UNSIGNED]`
 - Range $[-2^{63}, +2^{63}-1]$ (signed)
 - Range $[0, 2^{64}-1]$ (unsigned)

- **Booleano** `BOOL`
`BOOLEAN`

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 8 - Il Linguaggio SQL 5

SQL Data Types: Reali

- **Reale** `DECIMAL [(N[,D])] [UNSIGNED]`
 - Range dipendente dalla specifica
 - Viene memorizzato in forma "unpacked"
 - Es: un numero reale a 10 cifre con 4 decimali
 - `DECIMAL (10,4)`

- **Packed Real** `FLOAT [(N,D)] [UNSIGNED]`
`DOUBLE [(N,D)] [UNSIGNED]`
 - Range dipendente dalla specifica
 - Viene memorizzato in forma "packed"
 - Float: Singola precisione
 - Double: Doppia precisione

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 8 - Il Linguaggio SQL 6

SQL Data Types: Caratteri & Stringhe

- **Carattere** CHAR
 - Memorizza un singolo carattere

- **Strighe a lunghezza fissa** CHAR (N)
 - "N" = Massimo numero di caratteri della stringa
 - Usa sempre "N" caratteri nella memorizzazione del campo

- **Strighe a lunghezza variabile** VARCHAR (N)
 - "N" = Massimo numero di caratteri della stringa
 - Usa sempre i caratteri effettivamente presenti nella memorizzazione del campo

- **Le stringhe sono sempre indicate tra apici singoli.**
 - Es: 'Una Stringa.....'

Corrado Santoro, Basi di Dati (A-1), Ing. Informatica, Lezione 8 - Il Linguaggio SQL
7

SQL Data Types: Data & Ora

- **Data** DATE
 - Formato 'YYYY-MM-DD'
 - Range da '1000-01-01' a '9999-12-31'

- **Ora** TIME
 - Formato 'HH:MM:SS'

- **Data + Ora** DATETIME
 - Formato 'YYYY-MM-DD HH:MM:SS'
 - Range da '1000-01-01 00:00:00' a '9999-12-31 23:59:59'

- **Orario** TIMESTAMP
 - Formato 'YYYY-MM-DD HH:MM:SS'
 - Range da '1970-01-01 00:00:00' a '2037-12-31 23:59:59'

- **Intervallo temporare** INTERVAL
 - Non supportato da MySQL

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

SQL DDL: Creazione & uso di un DB

- **In SQL Standard**
 - CREATE SCHEMA *nome-schema* ;
- **In MySQL**
 - CREATE DATABASE [IF NOT EXISTS] *db_name* [*spec* [, *spec* ...] ...] ;
 - *nome-schema*, *db_name* è il nome del database;
 - *spec* sono delle opzioni aggiuntive (in genere poco usate)

- **Uso di un DB (in MySQL):**
 - USE *db_name* ;

- **Cancellazione di un DB (in MySQL):**
 - DROP DATABASE [IF EXISTS] *db_name* ;

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

SQL DDL: Creazione di un tabella



Versione "base" del comando di creazione di una tabella

```
CREATE TABLE table_name (  
    NomeAttr1 tipoAttr1,  
    NomeAttr2 tipoAtt2,  
    ... );
```

Comando di cancellazione di una tabella

```
DROP TABLE table_name [IF EXISTS];
```

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

10

SQL DDL: Esempio di creazione di un tabella



```
Amici (Nome, Cognome, Soprannome,  
       NumTelefono, Email)
```

```
create table Amici (  
    nome          varchar(30),  
    cognome       varchar(30),  
    soprannome    varchar(20),  
    numtelefono   varchar(20),  
    email         varchar(30));
```

Ma non abbiamo specificato né i vincoli (es. valori null) né le chiavi primarie

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

11

SQL DDL: Esempio di creazione di un tabella



```
Amici (Nome, Cognome, Soprannome,  
       NumTelefono, Email)
```

```
create table Amici (  
    nome          varchar(30) not null,  
    cognome       varchar(30) not null,  
    soprannome    varchar(20)  
                not null primary key,  
    numtelefono   varchar(20),  
    email         varchar(30));
```

"not null" specifica che il campo non può assumere valori null
"primary key" indica che il campo è una chiave primaria

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

12

SQL DDL: Superchiavi

Se la superchiave è formata da più di un campo, l'opzione "unique" viene posta alla fine della specifica dei campi

```
create table Amici (  
  nome          varchar(30) not null,  
  cognome       varchar(30) not null,  
  soprannome    varchar(20) not null primary key,  
  numtelefono   varchar(20) not null unique,  
  email         varchar(30),  
  unique (nome, cognome);
```

"nome" + "cognome" è univoco

SQL DDL: Vincoli interrelazionali

I legami tra tabelle vengono espressi tramite l'opzione "references" nel campo che costituisce il legame



```
create table Amici (  
  nome          varchar(30) not null,  
  cognome       varchar(30) not null,  
  soprannome    varchar(20) not null primary key,  
  numtelefono   varchar(30) not null unique,  
  email         varchar(30));  
  
create table Prestiti (  
  soprannome    varchar(20) references Amici(soprannome),  
  titololibro   varchar(30) not null,  
  DataRest      date not null,  
  primary key (soprannome, titololibro));
```

SQL DDL: Politiche sui vincoli interrelazionali

Cosa accade alla tabella "Prestito" se cancello o modifico un record dalla tabella "Amici"?



- Politica 1: Non consentire l'operazione (**no action**)
- Politica 2: Propagare l'azione a cascata (**cascade**)
 - Cancellazione/modifica delle tuple correlate
- Politica 3: Introduzione di valori NULL (**set null**)
 - Serve quando devo comunque conservare l'informazione anche se non più correlata con altre informazioni
- Politica 4: Uso del valore di default (**set default**)
 - Serve quando devo comunque conservare l'informazione anche se non più correlata con altre informazioni

SQL DDL: Campi Sequenza



- Lo standard SQL prevede il tipo "sequence"
- MySQL non lo supporta ma usa la keyword "auto_increment"
- In ogni caso, un campo sequenza deve essere definito come chiave

```
create table Amici (  
  id_amico      integer auto_increment unique,  
  nome          varchar(30) not null,  
  cognome       varchar(30) not null,  
  numtelefono  varchar(30) not null unique,  
  email         varchar(30),  
  primary key (id_amico));
```
