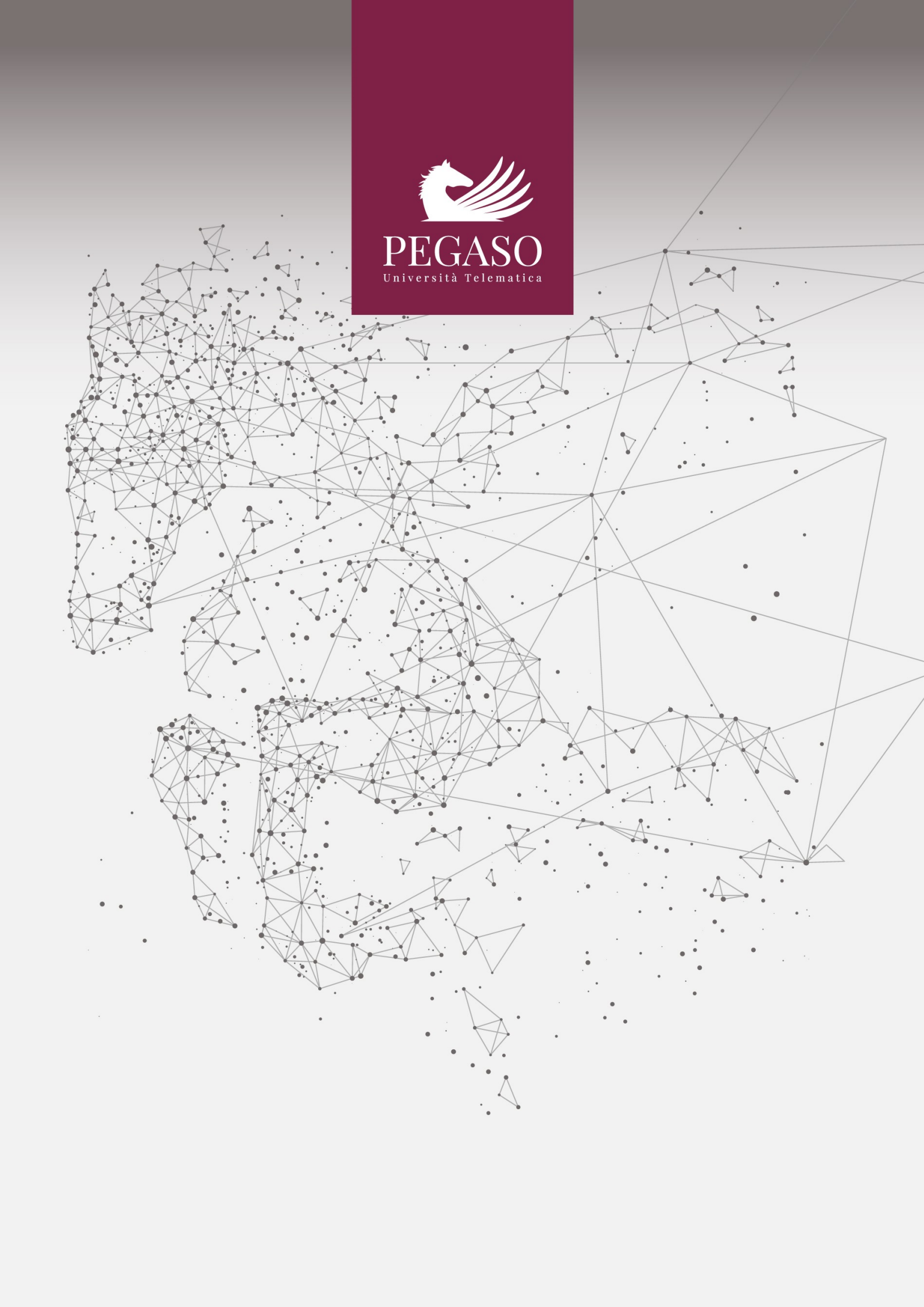
PEGASO
Università Telematica

Indice

1. Introduzione

Le query sono una delle cose che rendono i database così potenti. Una "query" si riferisce all'azione di recupero dei dati da un database. Una query è una domanda, spesso espressa in modo formale. Una query di database può essere una query di selezione o una query di azione. Una query di selezione è una query di recupero dati, mentre una query di azione richiede ulteriori operazioni sui dati, come l'inserimento, l'aggiornamento o la cancellazione. Una query è una richiesta di dati o informazioni da una tabella di database o una combinazione di tabelle. Questi dati possono essere generati come risultati restituiti da Structured Query Language (SQL) o come illustrazioni, grafici o risultati complessi, ad es. Analisi delle tendenze da strumenti di data mining. È possibile utilizzare uno dei numerosi linguaggi di query per eseguire una serie di query di database semplici o complesse. SQL, il linguaggio di query più noto e ampiamente utilizzato, è familiare alla maggior parte degli amministratori di database (DBA). In questa unità didattica ne studiamo le principali caratteristiche.

2. Dichiaratività del linguaggio SQL

Abbiamo coperto molte informazioni sulla tecnologia dei database, i termini e le idee fino ad ora.

Presto ci immergeremo nell'uso del linguaggio di manipolazione dei dati per inserire, aggiornare ed eliminare i dati in vari database che creeremo. Prima di arrivare a quel punto, dobbiamo dare un'occhiata all'aspetto chiave del Structured Query Language in generale. Questo sarebbe il fatto che SQL è una lingua dichiarativa, che è molto diversa da un linguaggio procedurale. È più probabile che tu abbia maggiore familiarità con gli approcci procedurali nell'informatica. Questo è il motivo per cui esamineremo più da vicino cosa significhi essere una lingua dichiarativa e come avremo bisogno di adattare il nostro pensiero per utilizzarlo al meglio. Il linguaggio SQL esprime le interrogazioni in modo *dichiarativo*, ovvero in una istruzione di questo linguaggio:

- Si specifica l'obiettivo dell'interrogazione e non il modo in cui ottenerlo;
- Si seguono i principi del calcolo relazionale.

Tale linguaggio, si contrappone ai linguaggi di interrogazione *procedurali* come l'algebra relazionale.

Vale la pena approfondire tale importante concetto attraverso la lettura di quanto espresso a tal proposito da una fonte come *Wikipedia*[1]: "Programmazione dichiarativa significa un programma che descrive a *cosa* una certa entità assomiglia, piuttosto che prescrivere *come* un'entità può essere creata. Ad esempio le pagine web HTML sono dichiarative, perché descrivono *cosa* la pagina dovrebbe contenere — titolo, testo, immagini — ma non *come* si deve fare per visualizzare la pagina sullo schermo del computer.

Altri linguaggi, come il Fortran, il C e Java si basano su un diverso approccio, richiedendo al programmatore di implementare specifici algoritmi esecutivi. In sintesi, i programmi imperativi definiscono in modo esplicito un algoritmo per conseguire uno scopo, mentre i programmi dichiarativi definiscono in modo esplicito soltanto lo scopo da raggiungere, lasciando che l'implementazione dell'algoritmo sia realizzata dal software di supporto (per esempio, un'istruzione **SELECT di SQL** specifica le proprietà dei dati che devono essere estratti da un database, ma NON i dettagli del processo di estrazione vero e proprio)."

Infatti, l'interrogazione SQL per essere eseguita viene passata all'ottimizzatore di interrogazioni (*Query Optimizer*) e il modo con cui lavora è nascosto all'utente. Per tale motivo, chi scrive interrogazioni in SQL può trascurare gli aspetti di traduzione e ottimizzazione, poiché la maggior parte dei DBMS relazionali ha comunque un buon motore di ottimizzazione.

[1] https://it.wikipedia.org/wiki/Programmazione_dichiarativa

Inoltre, esistono molti modi diversi per esprimere la stessa interrogazione in SQL ed il programmatore deve effettuare la scelta basandosi sulla leggibilità e la modificabilità dell'interrogazione, senza occuparsi di altri fattori legati all'ambiente tecnologico sottostante.

In un linguaggio dichiarativo, un vantaggio è quindi che è il database stesso a capire la procedura.

Tutto quello che si deve fare è dire al database quello che si vuole, sarà il suo motore interno a lavorare. È l'Optimizer che determina quale algoritmo otterrà i dati più richiesti. Questo è un approccio molto diverso da quello a cui si potrebbe essere abituati utilizzando il classico approccio procedurale alla programmazione. Bisogna pensare a ciò che si vuole e non a come si vuole.

3. Interrogazioni semplici

Nella Unità Didattica precedente abbiamo iniziato a studiare l'istruzione SELECT la quale specifica l'operazione di interrogazione, così come illustrato in Figura 1.

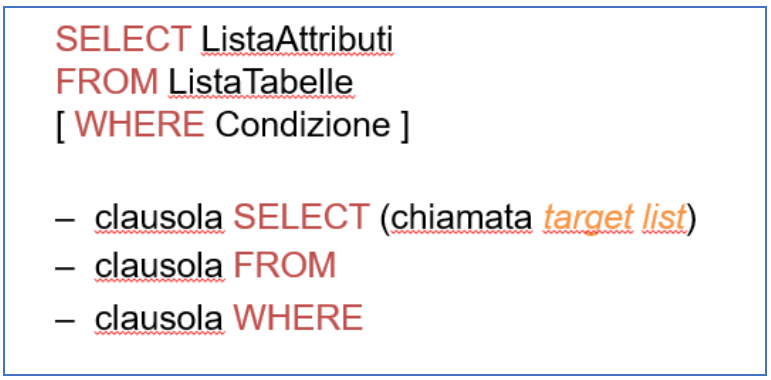

Figura 1: istruzione SELECT.

3.1 Uso della SELECT

L'interrogazione SQL:

- Seleziona, tra le righe che appartengono al prodotto cartesiano delle tabelle elencate nella clausola FROM, quelle che soddisfano le condizioni espresse nell'argomento della clausola WHERE.

Mentre il risultato di una SELECT, chiamato resource-set è rappresentato da una tabella con una riga per ogni riga prodotta dalla clausola FROM e filtrata dalla clausola WHERE.

Esempi:

Date le due tabelle: Impiegato (Nome,Cognome,Dipart,Ufficio,Stipendio,Città), raffigurata in Figura 2 e Dipartimento(Nome,Indirizzo,Città), raffigurata in Figura 3. Utilizziamo queste tabelle per effettuare alcune interrogazioni.

Interrogazione 1: estrarre lo stipendio degli impiegati di cognome «Rossi».

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giovanni	Verde	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Gialli	Direzione	7	73	Genova
Paola	Rosati	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Figura 2: tabella Impiegato

Nome	Indirizzo	Città
Amministrazione	Via Livio 32	Milano
Produzione	Via Lavier 4	Torino
Distribuzione	Via Segre 9	Roma
Direzione	Via Livio 32	Milano
Ricerca	Via Venosa 6	Milano

Figura 3: tabella Dipartimento.

La corrispondente istruzione SQL è la seguente:

- **SELECT Stipendio alias Salario**
- **FROM Impiegato**
- **WHERE Cognome='Rossi'**

Ed il risultato è quello illustrato in Figura 4. Se non ci sono impiegati con Cognome='Rossi', l'interrogazione tornerà un insieme vuoto altrimenti un insieme con tante righe quanti sono tali impiegati.

Alias Salario cambia il nome dell'attributo nel resource–set. Questa possibilità può far comodo in alcuni casi che analizzeremo in seguito.

Salario
45
80

Figura 4: il resource-set risultato della query dell'interrogazione 1.

Come argomento della clausola SELECT può apparire anche il carattere speciale ***** (asterisco).

Rappresenta la selezione di tutti gli attributi delle tabelle elencate nella clausola FROM. Per comprendere l'utilizzo di questo carattere speciale, formuliamo un'altra interrogazione.

Interrogazione 2*:* estrarre tutte le informazioni degli impiegati di cognome *Rossi.*

- **SELECT ***
- **FROM Impiegato**
- **WHERE Cognome='Rossi'**

Il risultato della interrogazione 2 è riportato in Figura 5. Come si può notare, nel resource-set sono presenti tutti gli attributi.

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Rossi	Direzione	14	80	Milano

Figura 5: resource-set risultato della interrogazione 2.

Un altro caso è quello in cui nella clausola SELECT possono comparire generiche espressioni sul valore degli attributi di ciascuna riga selezionata. Vediamo un altro esempio.

Interrogazione 3: estrarre lo stipendio mensile dell'impiegato che ha cognome 'Bianchi'. L'istruzione SQL che realizza tale query è la seguente:

- **SELECT Stipendio/12 as StipendioMensile**
- **FROM Impiegato**
- **WHERE Cognome='Bianchi'**

In questo caso, il resource–set è espresso nella:

StipendioMensile
3,00

Figura 6: resource-set interrogazione 3.

3.2 Clausola From

La clausola FROM può essere utilizzata in una interrogazione che coinvolge righe appartenenti a più tabelle: si pone come argomento della clausola FROM l'insieme di tabelle alle quali si vuole accedere e sul prodotto cartesiano delle tabelle elencate vengono applicate le condizioni contenute nella clausola WHERE.

Vediamo adesso un esempio di quanto detto, costruendo una interrogazione con il JOIN, dove le due tabelle sono collegate dai due nomi dei dipartimenti uguali.

Interrogazione 4: estrarre i nomi degli impiegati e le città in cui lavorano.

Istruzione SQL:

- SELECT Impiegato.Nome, Impiegato.Cognome, Dipartimento.Città
- FROM Impiegato.Dipartimento
- WHERE Impiegato.Dipart=Dipartimento.Nome

Il risultato è quello illustrato in Figura 7. Come si può notare, la sintassi della clausola SELECT aiuta a distinguere le varie tabelle che possono comparire.

Impiegato.Nome	Impiegato.Cognome	Dipartimento.Città
Mario	Rossi	Milano
Carlo	Bianchi	Torino
Giovanni	Verde	Milano
Franco	Neri	Roma
Carlo	Rossi	Milano
Lorenzo	Gialli	Milano
Paola	Rosati	Milano
Marco	Franco	Torino

Figura 7: resource-set risultato della interrogazione 4.

Vediamo adesso un esempio più complesso. Come database prendiamo lo schema illustrato in Figura 8.

Interrogazione 5: per ogni infrazione vogliamo sapere i dati del vigile che l'ha rilevata.

L'istruzione SQL è la seguente:

- **SELECT Infrazioni.Codice, Infrazioni. Vigile, Vigili.Cognome**
- **FROM Infrazioni, Vigili**
- **WHERE Infrazioni.Vigile=Vigili.matricola**

Il risultato è quello illustrato nel resource-set di Tabella 1.

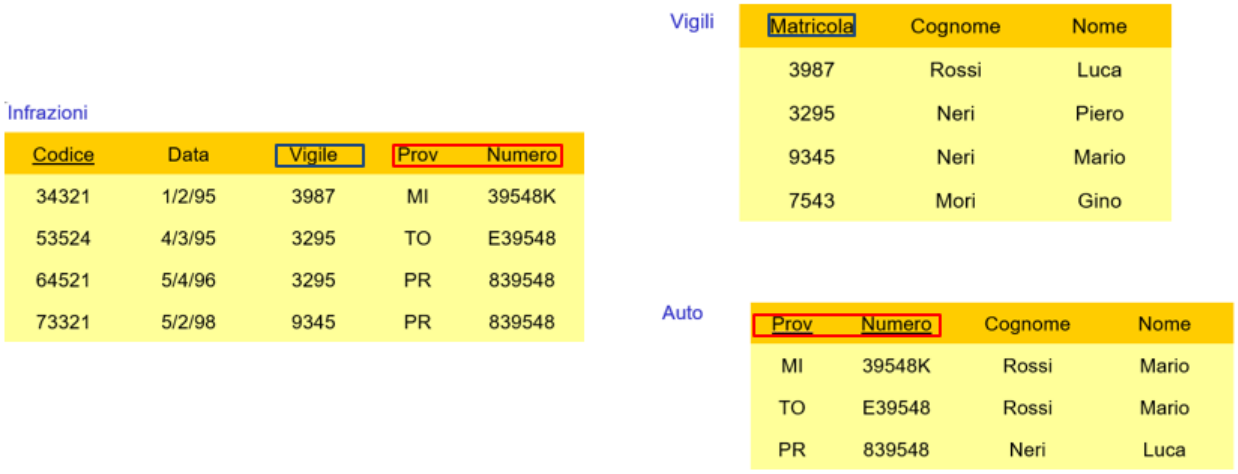

Figura 8: database di esempio.

Tabella 1: resource-set interrogazione 5.

Infrazioni.Codice	Infrazioni.Vigili	Vigili.Cognome
32321	3987	Rossi
53524	3295	Neri
64521	3295	Neri
73321	9345	Neri

Bibliografia

- Atzeni P., Ceri S., Fraternali P., Paraboschi S., Torlone R. (2018). Basi di Dati. McGraw-Hill Education.
- Batini C., Lenzerini M. (1988). Basi di Dati. In Cioffi G. and Falzone V. (Eds). Calderini. Seconda Edizione.

Sitografia

- https://www.w3schools.com/sql/sql_datatypes.asp
- https://forum.html.it/forum/showthread/t-896814.html
- https://www.youtube.com/watch?v=rtmeNwn4mEg