



PEGASO
Università Telematica



Indice

1. INTRODUZIONE.....	3
2. SQL: STORIA.....	4
3. DEFINIZIONE DEI DATI.....	6
4. COMMENTI.....	8
5. DOMINI	9
5.1 CARATTERI	9
5.2 TIPI NUMERICI ESATTI	9
6. VINCOLI INTRARELAZIONALI	11
6.1 VINCOLO NOT NULL	11
6.2 VINCOLO UNIQUE	11
6.3 VINCOLO PRIMARY KEY	11
BIBLIOGRAFIA	12
SITOGRAFIA	13

1. Introduzione

SQL è un linguaggio specifico utilizzato nella programmazione e progettato per la gestione dei dati contenuti in un sistema di gestione di database relazionali (RDBMS) o per l'elaborazione del flusso in un sistema di gestione del flusso di dati relazionale (RDSMS).

Originariamente basato sull'algebra relazionale e sul calcolo relazionale sulle tuple, SQL consiste di molti tipi di istruzioni, che possono essere classificate informalmente come sotto linguaggi:

- Un linguaggio di query dei dati (DQL);
- Un linguaggio di definizione dei dati (DDL);
- Una lingua di controllo dei dati (DCL);
- Un linguaggio di manipolazione dei dati (DML).

L'ambito di SQL include query sui dati, manipolazione dei dati (inserimento, aggiornamento ed eliminazione), definizione dei dati (creazione dello schema e modifica) e controllo dell'accesso ai dati.

Sebbene SQL sia spesso descritto come, e in larga misura, un linguaggio dichiarativo (4GL), esso include anche elementi procedurali.

SQL è stato uno dei primi linguaggi commerciali per il modello relazionale di Edgar F. Codd. Il modello è stato descritto nel suo autorevole articolo del 1970: "Un modello relazionale di dati per grandi banche dati condivise". Nonostante non aderisca completamente al modello relazionale descritto da Codd, è diventato il linguaggio di database più utilizzato¹.

SQL è diventato uno standard dell'American National Standards Institute (ANSI) nel 1986 e dell'Organizzazione internazionale per la standardizzazione (ISO) nel 1987. Da allora, lo standard è stato rivisto per includere un set più ampio di funzionalità. Nonostante l'esistenza di tali standard, la maggior parte del codice SQL non è completamente portabile tra diversi sistemi di database senza modifiche.

¹ <https://en.wikipedia.org/wiki/SQL>

2. SQL: Storia

SQL è stato inizialmente sviluppato in IBM da Donald D. Chamberlin e Raymond F. Boyce dopo aver appreso del modello relazionale di Ted Codd nei primi anni '70. Questa versione, inizialmente denominata SEQUEL (Structured English Query Language), fu progettata per manipolare e recuperare i dati memorizzati nel sistema di gestione del database quasi-relazionale originale IBM, *System R*, che un gruppo presso l'IBM San Jose Research Laboratory aveva sviluppato durante gli anni '70.

Il primo tentativo da parte di Chamberlin e Boyce di un linguaggio di database relazionale era *Square*, ma era difficile da usare a causa della notazione degli iscritti. Dopo essersi trasferiti nel laboratorio di ricerca di San Jose nel 1973, iniziarono a lavorare su SEQUEL. L'acronimo SEQUEL è stato in seguito modificato in SQL perché "SEQUEL" era un marchio registrato della società Hawker Siddeley Dynamics Engineering Limited con sede nel Regno Unito.

Dopo aver testato SQL presso i siti di test dei clienti per determinare l'utilità e la praticità del sistema, IBM iniziò a sviluppare prodotti commerciali basati sul prototipo System R inclusi System /38, SQL/DS e DB2, commercialmente disponibili nel 1979, 1981 e 1983, rispettivamente.

Alla fine degli anni '70, Relational Software, Inc. (ora Oracle Corporation) vide il potenziale dei concetti descritti da Codd, Chamberlin e Boyce e sviluppò il proprio RDBMS basato su SQL con l'intento di venderlo alla US Navy, Central Intelligence Agency e altre agenzie governative degli Stati Uniti. Nel Giugno del 1979, Relational Software, Inc. introdusse la prima implementazione disponibile in commercio di SQL, Oracle V2 (Versione 2) per computer VAX della Digital.

Nel 1986, i gruppi di standard ANSI e ISO adottarono ufficialmente la definizione di linguaggio standard *SQL Database Language*. Nuove versioni dello standard sono state pubblicate nel 1989, 1992, 1996, 1999, 2003, 2006, 2008, 2011 e più recentemente, nel 2016 (si suggerisce di consultare il sito: <http://troels.arvin.dk/db/rdbms/> per un'analisi accurata ed approfondita delle varie versioni. Per un quadro riassuntivo si veda la Tabella 1.

Tabella 1: le tappe fondamentali del linguaggio SQL:

Nome informale	Nome ufficiale	Caratteristiche
SQL base	SQL-86	Costrutti base
	SQL-89	Integrità referenziale
SQL-2	SQL-92	Modello relazionale

		Vari costruttori nuovi 3 livelli entry: entry, intermediate, full
SQL-3	SQL:1999	Modello relazionale a oggetti Organizzato in diverse parti Trigger, funzioni esterne...
	SQL:2003	Estensioni del modello a oggetti Eliminazione di costrutti non usati Nuove parti: SQL/JRT, SQL/XML...
	SQL:2006	Estensione della parte XML
	SQL:2008	Lievi aggiunte (trigger, ...)
	SQL:2011	Ricco supporto per dati temporali
	SQL:2016	Gestione del formato Json

3. Definizione dei dati

Iniziamo lo studio del linguaggio SQL con l'istruzione che crea una tabella ovvero una rappresentazione di una relazione:

- Istruzione **CREATE TABLE**:
 - definisce uno schema di relazione e ne crea un'istanza vuota;
 - specifica attributi, domini e vincoli.

Esempio 1

Si supponga di scrivere l'istruzione:

```
CREATE TABLE Impiegato(
    Matricola CHAR(6) PRIMARY KEY,
    Nome CHAR(20) NOT NULL,
    Cognome CHAR(20) NOT NULL,
    Dipart CHAR(15),
    Stipendio NUMERIC(9) DEFAULT 0,
    FOREIGN KEY(Dipart) REFERENCES
    Dipartimento(NomeDip),
    UNIQUE (Cognome,Nome)
)
```

Tale istruzione crea lo schema di relazione Impiegato(matricola, nome, cognome, **dipartimento**, stipendio), con vincolo di foreign key con la relazione Dipartimento(**NomeDip**, responsabile), come illustrato in Figura 1.

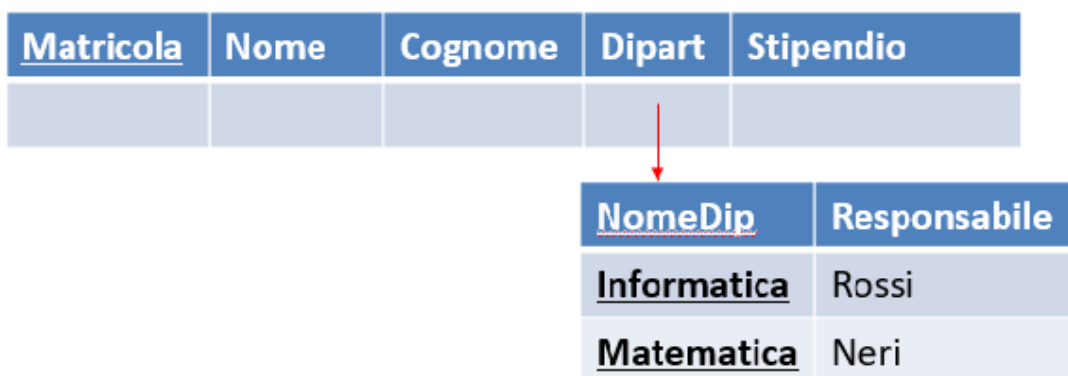


Figura 1: lo schema risultato dell'istruzione di esempio ed il suo legame con la relazione Dipartimento.

Esempio 2

Si supponga di voler creare lo schema della relazione Libro(Num_inv, Titolo, Anno_Pubbl, Prezzo, Ubicazione). Utilizzando l'istruzione CREATE TABLE si ha:

```
CREATE TABLE libro(  
  Num_Inv CHAR(10) PRIMARY KEY,  
  Titolo CHAR(30),  
  Autore CHAR(20),  
  Anno_Pubbl NUMERIC(4),  
  Prezzo NUMERIC(4,2),  
  Ubicazione CHAR(6)  
  }
```

Tabella 2: la tabella creata con un record inserito.

Num_Inv	Titolo	Autore	Anno_Pubbl	Prezzo	Ubicazione
AB-123	Reti di Calcolatori	A.S. Tanenbaun	2018	39,00	12-A

4. Commenti

In questo breve paragrafo riassumiamo alcuni concetti fondamentali sull'istruzione CREATE TABLE (Atzeni et al., 2018 pag. 98):

1. Una tabella SQL è costituita da una collezione ordinata di attributi e da un insieme (eventualmente vuoto) di vincoli.
2. Si adotta la convenzione che si userà il termine *tabella* al posto del termine *relazione* ed al posto di *tupla* il termine *riga*.
3. Ogni tabella viene quindi definita associandole un nome ed elencando gli attributi che ne compongono lo schema.
4. Per ogni attributo si definiscono:
 - a. Nome.
 - b. Dominio.
 - c. Un insieme di vincoli (eventualmente vuoto), che deve essere rispettato dai valori dell'attributo.
5. Dopo aver definito gli attributi, si possono definire i vincoli che coinvolgono più attributi della tabella.
6. Una tabella è inizialmente vuota e il creatore possiede tutti i privilegi sulla tabella, cioè i diritti di accedere ai suoi contenuti e di modificarli.

5. Domini

Il linguaggio SQL mette a disposizione alcune famiglie di domini elementari attraverso i quali definire i domini degli attributi dello schema.

5.1 Caratteri

Il dominio elementare *character* permette di rappresentare singoli caratteri o stringhe. La lunghezza delle stringhe di caratteri, può essere fissa o variabile; per le stringhe di lunghezza variabile si indica la lunghezza massima. In Tabella 3 i tipi di dati più importanti costruiti attraverso i caratteri.

Data type	Description
CHAR(size)	Holds a fixed length string (can contain letters, numbers, and special characters). The fixed size is specified in parenthesis. Can store up to 255 characters
VARCHAR(size)	Holds a variable length string (can contain letters, numbers, and special characters). The maximum size is specified in parenthesis. Can store up to 255 characters. Note: If you put a greater value than 255 it will be converted to a TEXT type
BLOB	For BLOBs (Binary Large Objects). Holds up to 65,535 bytes of data

Tabella 3: Il dominio elementare carattere.

5.2 Tipi numerici esatti

Questa famiglia contiene i domini che permettono di rappresentare valori esatti, interi o con una parte decimale di lunghezza prefissata. SQL mette a disposizione diversi tipi numerici esatti², due dei quali riportati in Tabella 4.

² https://www.w3schools.com/sql/sql_datatypes.asp

Tabella 4: tipi numerici esatti.

Data type	Description
INT(size)	-2147483648 to 2147483647 normal. 0 to 4294967295 UNSIGNED*. The maximum number of digits may be specified in parenthesis
DECIMAL(size,d)	A DOUBLE stored as a string, allowing for a fixed decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter

6. Vincoli intrarelazionali

I vincoli intrarelazionali sono quei vincoli sugli attributi che coinvolgono una sola relazione. I più semplici vincoli intrarelazionali sono: NOT NULL, UNIQUE e PRIMARY KEY.

6.1 Vincolo NOT NULL

Questo vincolo indica semplicemente che il valore NULL per l'attributo non è consentito. In tal caso il valore dell'attributo deve essere sempre specificato, tipicamente in fase di inserimento.

6.2 Vincolo UNIQUE

Questo vincolo si applica ad un attributo oppure ad un insieme di attributi di una tabella ed impone che i valori dell'attributo (o le ennuple di valori sull'insieme di attributi) siano una (super) chiave, cioè righe differenti della tabella non possono avere gli stessi valori. Il valore NULL può invece apparire su diverse righe senza violare il vincolo in quanto si assume che i valori nulli siano tutti diversi tra loro.

Il caso di attributo unico prevede la seguente sintassi:

Matricola CHAR(6) UNIQUE

Mentre nel caso di attributo multiplo:

Nome CHAR(20) NOT NULL,
Cognome CHAR(20) NOT NULL,
UNIQUE (Cognome, Nome)

6.3 Vincolo PRIMARY KEY

Il linguaggio permette di specificare, una sola volta, per ogni tabella, uno o più attributi come chiave primaria attraverso il vincolo PRIMARY KEY:

Nome CHAR(20) NOT NULL,
Cognome CHAR(20) NOT NULL,
PRIMARY KEY (Cognome, Nome)

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