

Caratteristiche generali del linguaggio SQL

Il linguaggio SQL è il linguaggio usato per la gestione dei database relazionali, cioè dei database creati con un DBMS di tipo relazionale. Esso nacque nella seconda metà degli anni '70 ad opera di IBM e successivamente nacquero altre versioni dell'SQL ad opera di altre aziende e tutte diverse le une dalle altre. Successivamente nel 1986 l'ANSI (American National Standard Institute) definì un linguaggio standard¹ che prese il nome di SQL/86 e successivamente gli standard SQL/89 e SQL/92. Tale linguaggio standard ha come obiettivo quello di fornire alle diverse aziende un linguaggio comune da usare per creare database compatibili con i diversi DBMS in circolazione come: Access SQL, MySQL, SQL Informix, DBII SQL, Postgre SQL, ecc.

²A che serve il linguaggio SQL?

Esso serve a definire la struttura delle tabelle di un database, sfruttando i comandi del linguaggio DDL (Data Definition Language); a manipolare i dati contenuti nel database, sfruttando i comandi del DML (Data Manipulation Language); a porre interrogazioni al database sfruttando i comandi del QL (Query Language) e, infine, a gestire gli accessi e i permessi per gli utenti, sfruttando i comandi del DCL (Data Control Language). Esso, quindi, sembrerebbe che serve a fare le stesse cose che ad es. il DBMS Access ci permette di fare. In realtà non è così.

Cosa offre in più il linguaggio SQL rispetto ad Access?

Il linguaggio Sql offre la possibilità di creare interrogazioni che con Access non si riescono a creare, offre la possibilità di modificare una query creata precedentemente in modo facile e rapido direttamente modificando il codice SQL nella finestra dell'Editor, ecc.

Identificatori e tipi di campi

Gli identificatori sono i nomi delle tabelle e i nomi dei campi delle tabelle del database. Essi sono costituiti da una sequenza di caratteri di lunghezza massima 18, devono iniziare con una lettera e possono anche contenere il carattere _ ("trattino basso"), come ad es. : Nome_Cognome, ecc.

Il tipo di un campo è l'insieme dei possibili valori che un campo di una tabella può assumere.

I tipi standard del linguaggio SQL e che noi useremo nelle esercitazioni sono:

- CHAR (N) per i campi di tipo testo al più di dimensione n = 255 caratteri
- MEMO per i campi di tipo testo al più di dimensione n = 65536 caratteri
- DATE per le date nella forma MM/GG/AAAA. (Ad es. per le date 12/05/06)
- TIME per le ore nella forma HH:MM. (Ad es. per le ore 16:45)
- BYTE per numeri interi di al più da p = 3 cifre (da 0 a 255)
- SMALLINT per i numeri interi costituiti al più da p = 5 cifre (da -32768 → 32767)
- INTEGER per i numeri interi costituiti al più da p = 10 cifre (da -2.147.483.648 → 2.147.483.647)
- REAL per i numeri con la virgola
- BIT per i valori di tipo si/no o vero/falso
- MONEY per i valori di tipo valuta di Access
- IMAGE per i valori che in Access sono di tipo Oggetto Ole e cioè immagini, suoni e video.

I comandi del linguaggio DDL (Data Definition Language): CREATE E ALTER

Il comando CREATE

In SQL le tabelle sono definite con il comando CREATE TABLE seguito dal nome della tabella e dall'elenco dei campi. Per ogni campo occorre specificare il nome e il suo tipo. La sua sintassi³ è la seguente:

¹ Un es. di linguaggio standard è l'italiano, mentre es. delle versioni non standard sono i dialetti regionali. I linguaggio standard, in tal caso, serve a far comunicare tra loro gli abitanti delle diverse regioni .

² Nota al Prof.: ricordare agli allievi che DDL, DML, QL e DCL sono i linguaggi di comandi usati in ogni DBMS

```
CREATE TABLE nome_tabella
( nome_campo1 tipo_campo1 clausola,
.....
nome_campon tipo_campon clausola
);
```

Qui vicino all'ultimo campo non ci vuole la virgola

dove nome-campo1, tipo-campo1,, nome-campon, tipo-campon sono i nomi dei campi e dei loro tipi e pertanto sono obbligatori, mentre la scritta clausola sottintende un valore facoltativo scelto tra i seguenti: **NULL, NOT NULL, DEFAULT, PRIMARY KEY, FOREIGN KEY, UNIQUE O REFERENCE.**

Le clausole in dettaglio

- la clausola **NULL** è da attribuire ad un campo che ha un valore non disponibile o non definito
- la clausola **NOT NULL** è da attribuire ad un campo che obbligatoriamente deve essere riempito
- la clausola **DEFAULT** definisce il valore da attribuire al campo al momento della creazione.
- la clausola **PRIMARY KEY** definisce un campo chiave primaria
- la clausola **UNIQUE** vieta la presenza in un campo di valori duplicati⁴
- la clausola **REFERENCE** definisce un vincolo di integrità referenziale.

Esempio

Per creare con SQL la tabella Dipendenti avente la seguente struttura:

ID_Dip	Cognome	Nome	Nascita	Residenza	CAP	Telefono	Retribuzione
--------	---------	------	---------	-----------	-----	----------	--------------

Creato con Access un database **azienda.mdb**, selezionato l'oggetto query e poi selezionata la voce crea query in visualizzazione struttura, chiusa la finestra mostra tabella, in alto a sinistra cliccare sul pulsante **SQL** per aprire la finestra editor in cui è possibile scrivere i comandi del linguaggio SQL.

In tale finestra scrivere il codice SQL⁵ seguente:

```
CREATE TABLE Dipendenti
( ID_Dip SMALLINT PRIMARY KEY,
  UNIQUE (ID_Dip),
  Cognome CHAR (30) NOT NULL,
  Nome CHAR (20) NOT NULL,
  Nascita DATE,
  Residenza CHAR (30),
  CAP CHAR (15),
  Telefono CHAR(10),
  Retribuzione MONEY
);
```

La clausola **UNIQUE** vieta la presenza di valori duplicati nel campo chiave primaria **ID_DIP**. Quando per un campo, come per il campo **ID_DIP** si vuole usare una clausola aggiuntiva, oltre a quella usata precedentemente, **PRIMARY KEY**, bisogna usare la seguente sintassi:

Qui vicino all'ultimo campo non ci vuole la virgola

Terminato di scrivere il codice SQL cliccare su salva e nominare la query così: "creazione tabella Dipendenti". Cliccato due volte la query creata, rispondere si alla domanda: "eseguire questa query SQL?" Infine andare nell'oggetto tabella per verificare che effettivamente la tabella richiesta è stata creata il cui nome è quello specificato nel comando **CREATE TABLE**.

³ Nota al Prof.: la frase: "la sua sintassi è" significa la regola con cui tale comando si usa nel linguaggio Sql.

⁴ Nota al Prof.: scegliendo tale clausola per un campo in fase di riempimento è vietato l'inserimento per lo stesso campo di due valori uguali. Se si provano a inserire lo stesso due valori uguali Access ci avvisa con un messaggio di errore.

⁵ Nota al Prof.: conviene far scrivere agli allievi le parole chiavi del linguaggio in maiuscolo per metterle in risalto. Esse possono essere scritte anche in minuscolo o in maiuscolo e minuscolo poiché l'SQL di Access non è case sensitivi.

Il comando ALTER

La struttura di una tabella può essere modificata in un secondo tempo con il comando ALTER TABLE seguito da una clausola che sottintende un valore facoltativo scelto tra ADD, per aggiungere un nuovo campo alla tabella, e tra DROP, per eliminare un campo dalla tabella.

Esempio1: Aggiunta del campo e-mail alla tabella Dipendenti

Nella finestra dell’editor scrivere il codice SQL seguente:

```
ALTER TABLE Dipendenti  
ADD e_mail CHAR (30);
```

salvare la query come: ‘aggiunta campo e-mail’ e eseguire la query. Fatto ciò andare a controllare che la tabella Dipendenti precedentemente creata presenta un nuovo campo, il campo e-mail richiesto.

Esempio2: Eliminazione del campo Retribuzione dalla tabella Dipendenti

Nella finestra dell’editor scrivere il codice SQL seguente:

```
ALTER TABLE Dipendenti  
DROP CAP;
```

salvare la query come: ‘eliminazione campo CAP’ e eseguire la query. Fatto ciò andare a controllare che la tabella Dipendenti precedentemente creata non presenta più il campo CAP come richiesto.

Esempio3: Eliminazione dell’intera tabella Dipendenti

Creata una copia della tabella Dipendenti (per evitare di perderla), nella finestra dell’editor scrivere il codice SQL seguente:

```
DROP TABLE Dipendenti;
```

Salvare la query come: ‘eliminazione tabella Dipendenti’ e eseguire la query. Fatto ciò andare a controllare che la tabella Dipendenti precedentemente creata non esista più.

I comandi del linguaggio DML: INSERT, UPDATE E DELETE

I valori dei campi nei record della tabella possono essere inseriti, aggiornati o cancellati rispettivamente con i comandi INSERT, UPDATE E DELETE.

Il comando INSERT

La sintassi è:

1° modo:
INSERT INTO nome_tabella
(nome_campo1,, nome_campon)
VALUES
(valore_campo1, ..., valore_campon)

2° modo:
INSERT INTO nome_tabella
VALUES
(valore_campo1, ..., valore_campon)

6° Esempio 1° modo

Nella finestra dell’editor scrivere il codice SQL seguente:

```
INSERT INTO Dipendenti  
(ID_Dip, Nome, Cognome, Nascita, Residenza, Telefono, Retribuzione)  
VALUES  
(1, 'Rossi', 'Marco', '12/12/1977', 'Saronno', '021234567', 2500);
```

Bisogna mettere tra virgolette tutti i valori i cui campi sono di tipo char, memo e date e senza virgolette gli altri.

Salvare la query come: ‘insert 1°record’ e eseguire la query, che Access chiama di accomodamento. Fatto ciò andare a controllare che la tabella Dipendenti precedentemente creata presenta un record popolato.

Esempio 2 modo

Nella finestra dell’editor scrivere il codice SQL seguente:

```
INSERT INTO Dipendenti  
VALUES (2, 'Bianchi', 'Alessandro', '12/01/1971', 'Napoli', '08121234567', 1500);
```

Salvare la query come: ‘insert 2° record’ e eseguire la query. ⁷

Esercizio

Aggiungere alla tabella Dipendenti, usando il comando INSERT INTO (1° modo o 2° modo) i seguenti record:

3	Tannini	Marco	01/01/68	Milano	021234567	1300
4	Bassi	Jessica	02/02/69	Roma	061234567	1350
5	De Vincenti	Rosa	03/03/72	Varese	0332123456	1400
6	De Bernardi	Luca	04/04/73	Varese	0332321234	1500

chiamando le query rispettivamente: insert 3° record, insert 4° record, ... , insert 6° record.

Osservazione

Poiché per inserire n nuovi record nella tabella Dipendenti creata sono necessarie n query di accomodamento tutte con la stessa sintassi ma con valori diversi dei campi, i DBMS moderni, come Access, possiedono un’interfaccia utente visuale conviene, in termini di tempo, effettuare l’inserimento di nuovi record manualmente e cioè senza l’utilizzo del comando Insert di SQL

Il comando UPDATE

La sintassi è:

```
UPDATE nome_tabella SET  
nome_campo = valore  
WHERE nome_campo = valore
```

⁶ Nota al Prof.: La tabella Dipendenti su creata, poiché abbiamo cancellato il campo CAP, è qui priva del campo CAP

⁷ Se ad es. volessimo inserire un record con id_dip = 1 già presente nella tabella Dipendenti Access si rifiuta di accodare il record richiesto. D’altra parte, nella creazione della tabella abbiamo esplicitamente richiesto che il campo ID_Dip contenga valori unici come l’uso della clausola UNIQUE

Esempio1

```
UPDATE Dipendenti SET
```

```
Retribuzione = 2000
```

```
WHERE ID_DIP = 1; | ← Stabilisce una condizione ovvero quale record bisogna aggiornare
```

Tale query si legge così: “ aggiorna al valore 2000 il campo Retribuzione del record individuato dal valore 1 della dalla chiave primaria ID_DIP (della tabella Dipendenti)

E’ possibile aggiornare anche più campi insieme con un solo comando UPDATE

Esempio2

```
UPDATE Dipendenti SET
```

```
Residenza = 'Busto Arsizio', Telefono = '0331123456'
```

```
WHERE ID_DIP=2;
```

Non vuole la virgola. Va tra virgolette perché il campo Telefono è di tipo char(10). Analogamente anche il campo Residenza è di tipo char(30)

Tale query si legge così: “ aggiorna al valore ‘Busto Arsizio’ il campo Residenza e al valore ‘0331123456’ il campo Telefono del record individuato dal valore 2 della dalla chiave primaria ID_DIP (della tabella Dipendenti)

Il comando DELETE

⁸La sintassi è:

```
DELETE *
```

```
FROM nome_tabella
```

```
WHERE nome_campo = valore
```

Esempio1

```
DELETE *
```

```
FROM Dipendenti
```

```
WHERE ID_Dip = 1;
```

Tale query si legge così: “ elimina il record individuato dal valore 1 della chiave primaria ID_DIP (della tabella Dipendenti)

Esempio2

```
DELETE *
```

```
FROM Dipendenti
```

```
WHERE Residenza='Città';
```

Tale query si legge così: “ elimina il record individuato dal valore specificato nel campo Residenza (della tabella Dipendenti) all’atto dell’esecuzione della query. E’ possibile immettere una qualsiasi città tra quelle presenti nella tabella Dipendenti. Se i record con la città richiesta ce ne sono più di uno tale query permette di cancellarli tutti in una sola volta.

I comandi del linguaggio QL

Essi sono i comandi usati per porre delle interrogazioni ad un database dopo che è stato definito e popolato.

⁸ Per selezionare tutti i campi del record possiamo o elencarli o sottintenderli con l’asterisco

Il comando SELECT⁹

Esso permette di selezionare i campi di una tabella e/o i record di una tabella che soddisfano ad un ben determinata condizione. La sua sintassi, in generale è:

```
SELECT nome_campo1, ..., nome_campon ← I campi possono essere, in base alla query da realizzare, uno, due o n  
FROM nome_tabella                       Anche la condizione, in base alla query da realizzare, potrebbe esserci o non  
WHERE condizione
```

Esempio1: selezione di un campo senza condizioni

```
SELECT Cognome  
FROM Dipendenti;
```

Esempio2: selezione di tutti i campi senza condizioni

```
SELECT *  
FROM Dipendenti;
```

Esempio3: selezione di tre 4 campi dei record che soddisfano ad una condizione semplice

```
SELECT Cognome, Nome, Residenza, Retribuzione  
FROM Dipendenti  
10WHERE Residenza = 'Varese';
```

Esempio4: selezione di tutti i campi dei record che soddisfano ad una condizione semplice

```
SELECT *  
FROM Dipendenti  
WHERE Residenza = 'Varese';
```

Inoltre nel comando Select è possibile specificare non solo una condizione semplice ma anche una condizione composta, cioè costituita da due o più condizioni semplici unite tra loro attraverso gli operatori logici: AND e OR.

L' operatore AND congiunge due o più condizioni e ritorna tutti e solo i record che soddisfano a tutte le condizioni. L' operatore OR collega anch'esso due o più condizioni, ma ritorna tutti i record che soddisfano ad almeno una delle condizioni semplici inserite.

Esempio5: selezione di tutti i campi dei record che soddisfano ad una condizione composta

```
SELECT *  
FROM Dipendenti  
WHERE Residenza = 'Varese' AND Retribuzione>1000;
```

Esempio6: selezione di tutti i campi dei record che soddisfano ad una condizione composta

```
SELECT *  
FROM Dipendenti  
WHERE Residenza = 'Varese' OR Residenza='Milano';
```

⁹ Esso corrisponde all'operazione di selezione studiato nei database dopo i concetti fondamentali del mod. relazionale.

¹⁰ Tale es. usa una condizione semplice. Tali condizioni sono realizzate con gli operatori relazionali: <, <=, >, >=, = e ≠.