

13.8 Embedded SQL

I programmi C possono interagire con i sistemi di gestione di basi di dati attraverso il linguaggio SQL (*Structured Query Language*), in modalità *embedded* (letteralmente: “incastrato”), cioè accogliendo al loro interno pezzi di codice SQL, passando dei parametri e ricevendo dei risultati.

SQL è il linguaggio standard, utilizzato dai DBMS relazionali (RDBMS), che consente l'esecuzione di tutte le operazioni necessarie nella gestione e nell'utilizzo di una base di dati; permette infatti, oltre alla consultazione del database, anche la definizione e gestione dello schema, la manipolazione dei dati e l'esecuzione di operazioni amministrative.

✓ NOTA

Con il termine database (base di dati) si indica un insieme di dati rivolti alla rappresentazione di uno specifico sistema informativo di tipo aziendale, scientifico, amministrativo o altro, mentre con sistema di gestione di base di dati o Data Base Management System (DBMS) ci si riferisce a un componente di software di base che consente la gestione del database. I DBMS hanno permesso di superare l'approccio tradizionale al problema dell'archiviazione: l'utilizzo diretto dei file sulle strutture del file system, in cui le applicazioni accedono direttamente agli archivi dei dati – come nei programmi visti fino a qui in questo capitolo –, in cui ogni applicazione deve conoscere la struttura interna degli archivi e le relazioni tra i dati, deve preoccuparsi che siano rispettati i requisiti di ridondanza minima. Nell'approccio tradizionale rimangono poi da soddisfare i requisiti di utilizzo contemporaneo degli archivi da parte di più applicazioni e di permanenza dei dati, che spesso vengono in parte delegati a strati sottostanti di software non specializzato quali il sistema operativo.

Rispetto alle soluzioni tradizionali, l'utilizzo di un DBMS comporta una serie di vantaggi che si traducono in una gestione dei dati più affidabile e coerente. In particolare ne derivano: indipendenza dei dati dall'applicazione, riservatezza nell'accesso ai dati, gestione dell'integrità fisica dei dati, gestione dell'integrità logica dei dati, sicurezza e ottimizzazione nell'uso dei dati (si veda la Figura 13.1).

Perché l'interazione tra C e sistema di gestione di una base di dati sia possibile deve essere disponibile un apposito precompilatore – generalmente messo a disposizione dalla casa madre del database in oggetto – che preprocessa il programma applicativo e produce un programma C che può essere poi normalmente compilato (Figura 13.2). Per esempio, il precompilatore della Oracle è conosciuto come Pro-C mentre quello della Sybase come Embedded SQL/C. In qualsiasi parte del programma possono essere inseriti comandi SQL immersi all'interno del comune codice C; nel caso di Sybase i comandi devono essere preceduti dalle parole chiave `exec sql` e terminati da un punto e virgola (Listato 13.10).

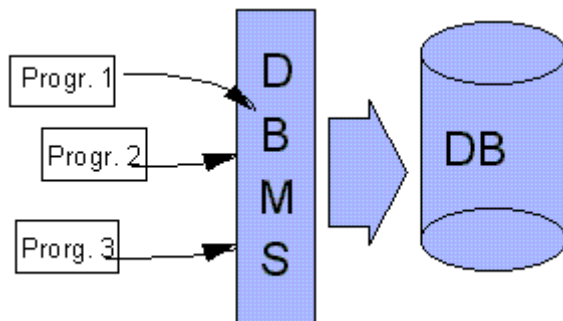


Figura 13.1 I programmi accedono ai dati attraverso il gestore di basi di dati

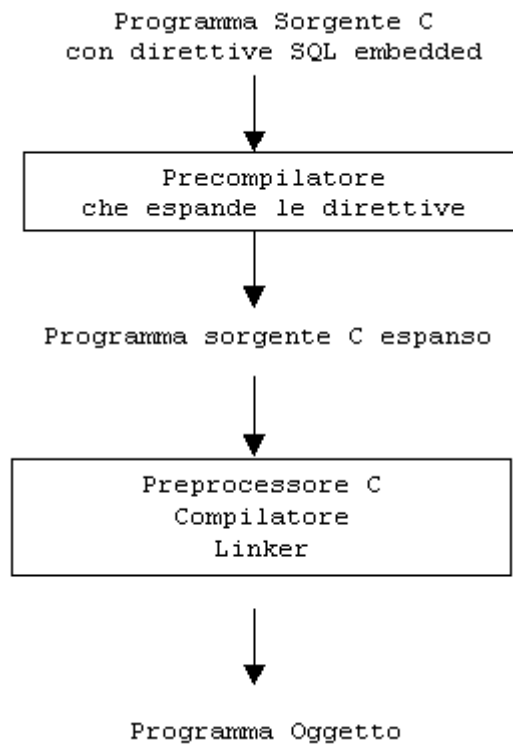


Figura 13.2 La prima fase dell'interazione fra C e DBMS è l'espansione delle direttive Embedded SQL

```
/* Esempio di Embedded SQL */

exec sql include sqlca;

main()
{
/* Dichiarazione delle variabili */
exec sql begin declare section;
CS_CHAR utente[31]; password[31];
exec sql end declare section;

/* Inizializzazione del sottoprogramma che
   gestisce gli errori SQL */
exec sql whenever sqlerror perform err_p();

/* Avvia la connessione con il server SQL */
printf("\nInserisca l'identificativo utente ");
gets(utente);
printf("\npassword ");
gets(password);
exec sql connect :user identified by :password;

/* Esempio di esecuzione di un comando SQL */
exec sql update auto set prezzo = prezzo *1.10;

/* Chiude la connessione con il server SQL */
exec sql disconnect;
}

/* Sottoprogramma che gestisce gli errori SQL */
```

```
err_p()
{
/* Stampa il codice di errore, un messaggio e il
   numero di linea dove si è prodotto l'errore */
printf("\nerrore nella linea %d, codice errore %d.\ns"
       sqlca.sqlcode, sqlca.sqlerrm, sqlerrmc);
}
```

Listato 13.10 Memorizzazione all'interno di un file di informazioni su un gruppo di alunni, inserite da tastiera

Nel caso di Sybase – per gli altri database la sintassi si discosta un po' ma i concetti rimangono validi – ogni direttiva inserita nel programma C ha la forma

```
exec sql direttivaSQL
```

Nel programma del Listato 13.10 inizialmente viene incluso il file `sqlca` con la direttiva

```
exec sql include sqlca;
```

in modo analogo a come facciamo con la direttiva `#include` del preprocessore C. Il file `sqlca` contiene strutture dati necessarie al sistema per eseguire i comandi SQL. Successivamente vengono dichiarate le variabili `utente` e `password` di tipo array di `CS_CHAR`, che è equivalente in C a un array di `char`, dunque a una stringa.

```
exec sql begin declare section;
CS_CHAR utente[31]; password[31];
exec sql end declare section;
```

Il programma richiede l'identificativo utente e la password e la comunica al database con altra direttiva:

```
exec sql connect :user identified by :password;
```

Per ragioni di sicurezza, infatti, si inizia normalmente una sessione di lavoro facendosi riconoscere dal gestore di basi di dati. Ed ecco che finalmente eseguiamo un'operazione direttamente sulla base di dati con un comando SQL:

```
exec sql update auto set prezzo = prezzo *1.10;
```

Il comando `update` (aggiorna) modifica la tabella `auto` aumentando il valore della colonna `prezzo` del 10%. L'ultima direttiva chiude la connessione con il database:

```
exec sql disconnect;
```

Naturalmente tra l'apertura e la chiusura della sessione di lavoro avremmo potuto inserire a piacere codice C e altre direttive Embedded SQL ■.