

La strutturazione dei database (1/3): La chiave primaria



Iniziamo oggi una serie di articoli sulla **strutturazione dei database**. In anni di militanza nei vari forum di supporto, mi sono reso conto che ci sono molti utenti perfettamente in grado di manipolare i database, con query anche molto complesse, ma quando chiedono aiuto per qualche ragione e si va a vedere la struttura delle tabelle del database, capita di trovarsi di fronte a delle realizzazioni confuse e che non tengono conto di alcune semplici e basilari norme.

Questo articolo è dunque destinato a chi si è addentrato da poco nel mondo dei database e vuole partire con il piede giusto, come pure a chi ha già una certa esperienza e desidera migliorare la qualità dei propri prodotti.

I principi che vedremo sono fondamentali per realizzare dei database efficienti, mantenibili ed aderenti agli standard della programmazione. Gli argomenti che tratteremo sono i seguenti:

- La chiave primaria
- La prima, la seconda e la terza forma normale
- Il concetto di relazionalità e l'applicazione della normalizzazione

Assimilati questi concetti, ti risulterà più facile creare delle strutture di database in modo corretto. Iniziamo dunque con il primo argomento di fondamentale importanza in quanto ruota quasi tutto intorno a quello.

Cos'è la chiave primaria?

In una tabella la chiave primaria è quel dato che **ci permette di identificare univocamente una riga** (quindi un insieme di dati) e per convenzione prende solitamente il nome *id*.

Una tabella contenente ad esempio degli indirizzi, può avere più righe nelle quali il nome è "Maurizio" ed il cognome è "Tarchini". Infatti possono esserci diverse persone che si chiamano "Maurizio Tarchini". La chiave primaria ci permette di distinguere ogni singolo record indipendentemente dalle ambiguità che possono facilmente crearsi (come persone con lo stesso nome e lo stesso cognome).

id	Nome	Cognome	Città
34	Maurizio	Tarchini	Balerna
57	Nando	Pappalardo	Catania
144	Maurizio	Rossi	Catania

Come vedi in questa tabella i campi *nome*, *cognome* e *città* non sono univoci. Il nome Maurizio compare due volte come pure la città Catania. Dobbiamo quindi creare un dato che sia certamente unico, quindi una chiave primaria.

La presenza in una tabella di una chiave primaria correttamente impostata è una precondizione indispensabile all'applicazione della relazionalità (concetto che approfondiremo in un altro articolo).

Se tutto questo viene a mancare diventa alto il rischio che si produca il fenomeno dell'*inconsistenza*, che in poche parole significa la non affidabilità dei dati (vedremo perchè e in che modo).

Come impostare la chiave primaria?

Normalmente la chiave primaria è impostata in fase di creazione della tabella.

Il caso più classico è impostare il campo come numero intero, annunciarlo come chiave primaria ed attribuirgli la proprietà `auto_increment`.

In questo modo lasciamo al database il compito di gestire le chiavi primarie. Questa è una modalità molto sicura.

Il database attribuirà automaticamente un numero univoco ad ogni record. Anche qualora cancellassimo una riga, la chiave primaria di questa riga non sarà più utilizzata successivamente.

Ma questa strada, a volte, può rivelarsi un errore.

Serve un solo dato univoco

Capita di essere tanto abituati ad impostare una chiave primaria da non rendersi conto che **a volte**

esiste già senza doverla creare. Quello che segue è un errore molto comune.

Id	Codice assicurato	Nome	Cognome
1	45T7PL3	Maurizio	Tarchini
2	EF6U098	Marco	Rossi

Una regola fondamentale dei database (verrà fuori altre volte) è che **non dobbiamo avere dati ridondanti**. In questo caso abbiamo però un dato ridondante.

Infatti non ha senso creare il campo *id* come dato univoco. Il codice assicurato è già, per sua stessa natura, un dato univoco.

Possiamo quindi utilizzare questo dato come chiave primaria. In questo caso non dovremo però ovviamente definire la proprietà *auto_increment*; il dato sarà inserito così com'è, a noi basta la certezza che sia un dato univoco. Possiamo dunque fare a meno della colonna *id*.

Similmente potremo utilizzare il numero di matricola come chiave primaria nell'elenco degli studenti di un'università oppure il numero di partita IVA per un elenco di aziende.

Il dato univoco può essere una combinazione di dati?

Se vogliamo proprio dirla tutta, non necessariamente la chiave primaria dovrà essere un unico dato. Qualora la combinazione di due o più dati possono identificare univocamente una riga, possiamo utilizzare quella che viene definita una **chiave primaria composta**.

Prendiamo ad esempio una tabella che mostri la provincia, il numero di targa ed il proprietario dell'automobile.

provincia	numero	proprietario
MI	457465RT6	Rossi
CO	4211ZR4D6	Verdi
RO	L06453	Bianchi

La combinazione tra *provincia* e *numero* da luogo ad un dato univoco e quindi alla possibile chiave primaria per questa tabella.

Conclusione

In questo articolo abbiamo iniziato ad esplorare le tecniche grazie alle quali è possibile realizzare dei database efficienti e mantenibili. Abbiamo dunque analizzato brevemente cosa sia una chiave primaria. Quanto questo dato sia fondamentale lo vedremo nei prossimi articoli. La chiave primaria

è infatti la base per realizzare la normalizzazione (che vedremo nel prossimo articolo) e la normalizzazione è la base per implementare una robusta relazionalità tra le tabelle (argomento dell'ultimo articolo di questa serie).

Ringrazio capobecchino per aver segnalato un errore importante in questo articolo ed avermi dato modo così di correggerlo