A drawing of a face

Description automatically generated

**BACKEND SYSTEM INTEGRATOR**

**BASI DI DATI - SQL**

**Docente: Alberto Venditti**

**Titolo argomento: Normalizzazione nei database relazionali**

**Cenni sulla Normalizzazione**

**nei Database Relazionali**

*Alberto Venditti*

La normalizzazione di un database relazionale è un'operazione che consiste nell'applicare una serie di trasformazioni strutturali alla base dati, secondo alcune regole formali mirate a prevenire anomalie ed inconsistenze dei dati. Quando la struttura del database soddisfa a ciascuna di queste regole formali, si dice che esso assume la corrispondente "forma normale".

Il processo di normalizzazione verrà qui presentato in modo intuitivo, senza fare riferimento alla teoria relazionale ed omettendo le definizioni e le dimostrazioni che ne giustificano le regole.

# La prima forma normale

La prima forma normale si può esprimere dicendo che:

***Tutti i record di una tabella devono contenere lo stesso numero di campi.***

La prima forma normale esclude dunque la presenza di campi opzionali (che possono cioè contenere il valore nullo) e di campi multipli. L'esempio seguente mostra una tabella che non rispetta la prima forma normale: essa contiene campi vuoti e non è utilizzabile per memorizzare i dati un padre con più di tre figli. A destra viene riportata la corrispondente tabella corretta, che non presenta questi inconvenienti.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Padre** | **Figlio 1** | **Figlio 2** | **Figlio 3** |  | **Padre** | **Figlio** |
| Paolo Rossi | Alberto | Lucia | Rinaldo |  | Paolo Rossi | Alberto |
| Michele Bianchi | Luca | - | - |  | Paolo Rossi | Lucia |
|  |  |  |  |  | Paolo Rossi | Rinaldo |
|  |  |  |  |  | Michele Bianchi | Luca |

# La seconda forma normale

Data una tabella e la sua chiave primaria (che, come tale, ne individua univocamente i record), la seconda forma normale afferma che:

***Tutti i campi non chiave devono contenere informazioni attinenti all'intera chiave primaria.***

La seconda forma normale richiede inoltre che sia rispettata la prima forma normale.

Nella tabella seguente, la chiave primaria è costituita dai campi Padre e Figlio; il campo Indirizzo è però riferito al solo padre e quindi la seconda forma normale risulta violata.

|  |  |  |
| --- | --- | --- |
| **Padre** | **Figlio** | **Indirizzo** |
| Paolo Rossi | Alberto | Via degli Orti 4 |
| Paolo Rossi | Lucia | Via degli Orti 4 |
| Paolo Rossi | Rinaldo | Via degli Orti 4 |
| Michele Bianchi | Luca | Corso Traiano 5 |

Gli inconvenienti di una simile struttura sono molteplici: l'indirizzo è ripetuto in ogni record, il che porta facilmente ad inconsistenze ed incoerenze; non si può memorizzare l'indirizzo di un "padre" che non ha figli (lasciare vuoto il campo Figlio significherebbe violare la prima forma normale).

La soluzione sta nello spezzare la tabella in due sottotabelle, come illustrato di seguito.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Padre** | **Figlio** |  | **Padre** | **Indirizzo** |
| Paolo Rossi | Alberto |  | Paolo Rossi | Via degli Orti 4 |
| Paolo Rossi | Lucia |  | Michele Bianchi | Corso Traiano 5 |
| Paolo Rossi | Rinaldo |  |  |  |
| Michele Bianchi | Luca |  |  |  |

# La terza forma normale

La terza forma normale (molto vicina alla seconda forma normale) richiede che siano rispettate le precedenti regole e afferma che:

***Ogni campo non chiave deve contenere un'informazione riguardante la chiave primaria.***

Nella tabella seguente, la chiave primaria è NomeImpiegato; il campo CittàSocietà riguarda più propriamente la società (dunque il campo NomeSocietà) che l'impiegato. Gli inconvenienti di questa struttura sono simili a quelli evidenziati nel caso in cui sia violata la seconda forma normale.

Anche qui, la soluzione sta nello spezzare la tabella in due sottotabelle, aventi come colonne rispettivamente: NomeImpiegato e NomeSocietà, NomeSocietà e CittàSocietà.

|  |  |  |
| --- | --- | --- |
| **NomeImpiegato** | **NomeSocietà** | **CittàSocietà** |
| Paolo Rossi | FIAT | Torino |
| Mario Verdi | Fininvest | Milano |
| Michele Bianchi | FIAT | Torino |

# La quarta forma normale

Secondo la quarta forma normale:

***Una tabella non può contenere due o più fatti multivalore indipendenti.***

Un fatto multivalore è un campo che può assumere più di un valore in corrispondenza dello stesso valore di un altro campo. Per esempio, nella seguente tabella si sono elencati i Figli e le Lingue conosciute dal sig. Rossi e dal sig. Verdi.

|  |  |  |
| --- | --- | --- |
| **Nome** | **Figlio** | **Lingua** |
| Paolo Rossi | Alberto | - |
| Paolo Rossi | Gloria | - |
| Paolo Rossi | - | Inglese |
| Paolo Rossi | - | Tedesco |
| Gianni Verdi | Beatrice | - |
| Gianni Verdi | - | Francese |

Tale rappresentazione è inaccettabile poiché viola la prima forma normale (vista la presenza di campi vuoti). Per ovviare ai valori nulli, si pensi ad una rappresentazione come la seguente:

|  |  |  |
| --- | --- | --- |
| **Nome** | **Figlio** | **Lingua** |
| Paolo Rossi | Alberto | Inglese |
| Paolo Rossi | Gloria | Tedesco |
| Gianni Verdi | Beatrice | Francese |

Anch'essa risulta inadatta poiché non risolve il problema nel caso in cui il numero di figli e il numero di lingue conosciute non coincidano. Inoltre, i campi Figlio e Lingua sono arbitrariamente accoppiati: tra i due campi vi è una connessione indiretta (perché sono riferiti alla stessa persona), ma il loro accoppiamento non porta alcuna informazione supplementare. La quarta forma normale tende ad evitare accoppiamenti non significativi di dati.

La struttura della tabella va modificata creando due sottotabelle, aventi come colonne rispettivamente: Nome e Figlio, Nome e Lingua.

# La quinta forma normale

Si dice che ***una tabella è in quinta forma normale quando l'informazione che essa contiene non può essere ricostruita da un numero maggiore di tabelle più semplici*** (cioè costituite da un minor numero di campi).

Supponendo di avere una tabella composta dai campi Venditore, Prodotto e Ditta, i suoi record indicheranno quali prodotti vende un venditore e di quali ditte egli è rappresentante.

|  |  |  |
| --- | --- | --- |
| **Venditore** | **Prodotto** | **Ditta** |
| Paolo Rossi | Auto | FIAT |
| Paolo Rossi | Auto | Renault |
| Paolo Rossi | Camion | FIAT |
| Paolo Rossi | Camion | Renault |
| Michele Bianchi | Auto | FIAT |
| Michele Bianchi | Auto | Renault |

Nell'ipotesi che valga la regola che, se un venditore rappresenta una ditta e vende un certo prodotto, e la ditta produce quel prodotto, allora il venditore vende quel prodotto di quella ditta, la tabella finale può essere derivata da tre tabelle più semplici. La prima indicherà quale ditta rappresenta un venditore; la seconda quali prodotti vende un venditore; la terza quali prodotti fabbrica una ditta.

|  |  |
| --- | --- |
| **Venditore** | **Prodotto** |
| Paolo Rossi | Auto |
| Paolo Rossi | Camion |
| Michele Bianchi | Auto |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Venditore** | **Ditta** |  | **Prodotto** | **Ditta** |
| Paolo Rossi | FIAT |  | Camion | FIAT |
| Paolo Rossi | Renault |  | Camion | Renault |
| Michele Bianchi | FIAT |  | Auto | FIAT |
| Michele Bianchi | Renault |  | Auto | Renault |

La quinta forma normale permette dunque di eliminare molte ridondanze.

# Conclusioni

Anche se può essere non evidente dagli esempi forniti, un database normalizzato (pur necessitando di più tabelle dell'equivalente non normalizzato) richiede meno spazio fisico per la memorizzazione dei dati poiché è composto da un numero minore di record. Tale vantaggio si realizza quando le dimensioni della base dati aumentano, poiché il database normalizzato cresce in modo additivo, mentre l'equivalente non normalizzato cresce in modo moltiplicativo.

Se la normalizzazione da un lato evita la duplicazione di informazioni riducendo il rischio di anomalie ed inconsistenze nella base dati, dall'altro essa porta in generale ad un peggioramento delle prestazioni. In un database relazionale normalizzato, infatti, le interrogazioni necessiteranno di un maggior numero di *join* tra tabelle differenti, e il *join* è tra le operazioni più critiche in un database relazionale. È questo il motivo per cui talvolta si ricorre volutamente ad una parziale denormalizzazione del database, per ottimizzarne le prestazioni evitando alcuni *join* critici.