

Workshop 9 ottobre

Nota bene:

*Nel lavoro reale molto spesso **non possiamo permetterci di modificare la struttura di un Database quando esso è già in produzione**. Infatti esisteranno tantissime procedure interne ed esterne al Database che sono state costruite utilizzando la sua struttura, anche se non efficiente. Andando a cambiare lo schema delle tabelle, le nostre modifiche provocherebbero dunque una serie lunghissima di malfunzionamenti. Questo ci fa capire quanto **importante e delicata** sia la parte iniziale di progettazione! In questo Workshop proveremo comunque, a scopo didattico, a risolvere il problema a monte, normalizzando il Database in modo da rendere la sua struttura più efficiente.*

Creare sul proprio schema (studente1, studente2, eccetera) una versione normalizzata delle tabelle *normalizzazione.Prestiti*, *normalizzazione.Clienti* e *normalizzazione.ElencoRate*

Traccia della soluzione

- 1) La relazione tra *prestiti* e *clienti* è di tipo **molti a molti**. Iniziamo con creare tre nuove tabelle:
 - Prestiti
 - Clienti
 - PrestitiClienti

Usiamo l'istruzione CREATE, facendo attenzione a chiavi primarie e chiavi esterne.

Come saranno distribuite le colonne tra le tabelle?

La colonna *Residenza* rispetta la prima forma normale?

- 2) Popoliamo le tre tabelle con le tre relative istruzioni INSERT. Qualche suggerimento:
 - Dovremmo probabilmente ricorrere a una DISTINCT nella SELECT relativa alla tabella prestiti.
 - Per la tabella Clienti servirà una JOIN, ma anche qui, facciamo attenzione alle duplicazioni!
 - Sempre per la tabella dei Clienti, devo trovare un modo per dividere il contenuto delle due colonne.

- 3) Creiamo una quarta tabella **TipologiaPrestiti** con due colonne:

- Una IdTipologia con la proprietà Identity(1,1)
 - Una Tipologia di tipo Varchar
- 4) Popoliamo la tabella con l'elenco dei valori distinti presenti nella colonna Tipologia della tabella Prestiti
- 5) Aggiungiamo alla tabella Prestiti la colonna IdTipologia con un ALTER TABLE
- 6) Eseguiamo un "Update con join" per popolare la colonna IdTipologia della tabella prestiti con il valore della **TipologiaPrestiti**
- 7) Cancelliamo la colonna Tipologia
- 8) Consideriamo la tabella *normalizzazione.ElencoRate* contenente
- IdPrestito INT
 - Rate JSON

L'obiettivo finale è creare una nuova tabella con le colonne

- IdPrestito INT
- IdRata INT
- Importo DECIMAL(18,4)

Traccia della soluzione del punto 8

- a) Partiamo eseguendo una outer apply tra
- *normalizzazione.ElencoRate* as r
 - OpenJson (r.rata)

Cosa dobbiamo specificare all'interno del WITH? (Osserva che la colonna Rate è un array di JSON)

- b) Eseguiamo una seconda outer apply con l'OpenJson della colonna calcolata nel punto a.

Cosa inseriamo questa volta nel WITH?

- c) Hai notato che la chiave importo è scritta a volte con la i minuscola e a volta con la i maiuscola? Come possiamo gestire questo caso