Databases
Sql and NoSql

Francesco Pugliese, PhD

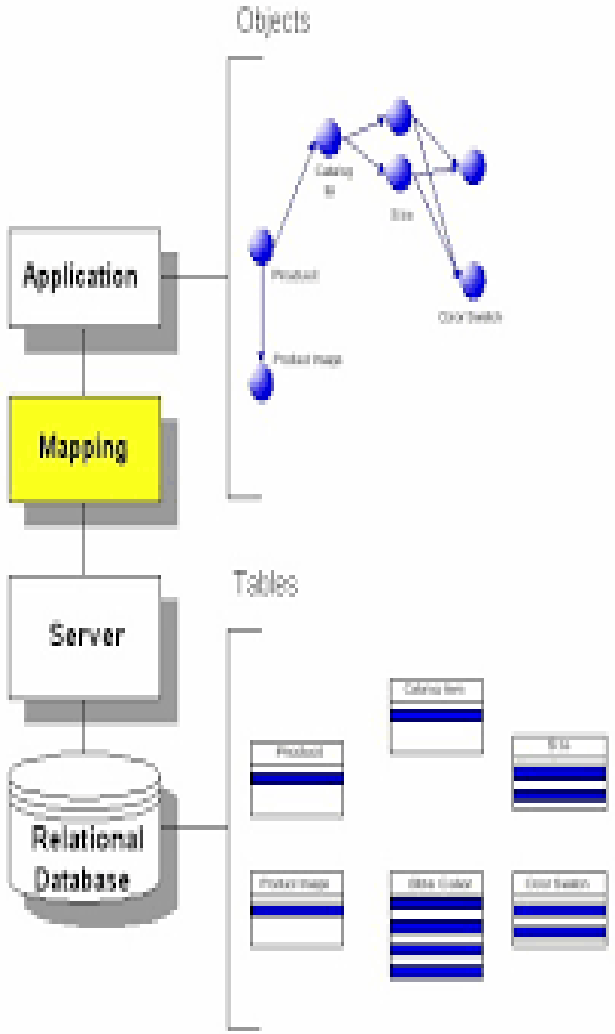
neural1977@gmail.com

Database Transazionali

- I **Database Transazionali** sono ottimizzati per l'esecuzione di sistemi di produzione, dai siti web alle banche fino ai negozi al dettaglio. Questi database si distinguono per la rapidità di lettura e scrittura di singole righe di dati senza alterarne l'integrità. Questi database transazionali sono archivi di righe, dunque i dati sono archiviati su disco come righe anzichè colonne. Gli archivi di righe sono molto utili quando è necessario sapere tutto di un cliente nella tabella utente, per esempio. Tuttavia non sono ottimali quando per esempio si cerca di conteggiare i clienti di un determinato codice postale, in quanto è necessario caricare in memoria nonsolo la colonna codice postale ma anche le colonne nome, indirizzo, ecc. Dunque i DB transazionali non sono creati per l'analisi.

Impedence Mismatch

- **L'Impedence Mismatch** è il termine che si usa quando ci si riferisce ai problemi che si verificano a causa delle differenze tra il modello di database e il modello del linguaggio di programmazione.
- A causa dell'Impedance Mismatch possono verificarsi i seguenti problemi:
- Per esempio il tipo dei dati degli attributi nel linguaggio di programmazione potrebbe differire da quello nel modello dei dati, quindi è necessario avere un collegamento per ciascun linguaggio di programmazione host che specifichi per ciascun attributo il tipo compatibile nel linguaggio di programmazione. Per esempio i tipi in C sono differenti da Java ed entrambi differiscono dai tipi SQL.



Impedence Mismatch

- Il secondo problema che potrebbe verificarsi è dovuto al fatto che i risultati della maggior parte delle query sono insiemi o multi-insiemi di tuple e ciascuna tupla è formata da una sequenza di valori di attrubuti. Nel programma è necessario accedere ai singoli valori dei dati all'interno di tuple per stampare o elaborare. Quindi c'è la neccessita di un collegamento per mappare le strutture dati risultanti da una query (che è una tabella in pratica) ad una appropriata struttura dati nel linguaggio di programmazione. E' necessario un meccanismo per ciclare su tutte le tuple in un risultato di una query al fine di accedere ad ogni singola tupla alla volta e estrarre i singoli valori dalla tupla. I valori estratti sono tipicamente copiati in appropriate variabili per ulteriori elaborazioni del programma. Un cursore o un iteratore è una variabile usata per ciclare sulle tuple di un risultato di una query, i valori singoli all'interno di ciascuna tupla vengono estratti per essere inseriti all'interno di differenti o uniche variabili di programma che devono avere un appropriato datatype. Questo problema può essere mitigato quando il linguaggio di programmazione del db usa lo stesso data model e lo stesso data type del modello di database come per Oracle PL/SQL.

Proprietà ACID dei DB - Atomicity, Consistency, Isolation e Durability

- ✓ Nell'ambito dei **Database, ACID** deriva dall'acronimo inglese **Atomicity, Consistency, Isolation e Durability** che significano **Atomocità, Coerenza, Isolamento e Durabilità** ed indicano le proprietà logiche che devono avere le transazioni.
- ✓ **Atomica:** il processo deve essere suddivisibile in un numero finito di unità indivisibili, chiamate **transazioni**. L'esecuzione di una transazione perciò deve essere per definizione o totale o nulla, e non sono ammesse esecuzioni parziali, un processo, anche parziale, invece in quanto insieme di transazioni può non essere elementare.
- ✓ **Coerente:** il database rispetta i **vincoli di integrià**, sia a inizio che a fine transazione. Non devono verificarsi contraddizioni (incoerenza dei dati) tra i dati archiviati nel DB.

Proprietà ACID dei DB - Atomicity, Consistency, Isolation e Durability

- ✓ **Isolata:** ogni transazione deve essere eseguita in modo isolato e indipendente da altra transazioni, l'eventuale fallimento di una transazione non deve interferire con le altre transazioni in esecuzione.
- ✓ **Durevole:** detta anche **persistenza,** si riferisce al fatto che una volta che una transazione abbia richiesto un **commit work,** i cambiamenti apportati non dovranno essere più persi. Per evitare che nel lasso di tempo tra il momento in cui la base di dati si impegna a scrivere le modifiche e quello in cui li scrive effettivamente si verifichino perdite di dati dovuti a malfunzionamenti, vengono tenuti dei registri di log dove sono annotate tutte le operazioni sul DB.

Database a Grafo

- ✓ Un **grafo** è una rappresentazione visuale di un insieme di oggetti dove alcune coppie di oggetti sono connesse tra loro attraverso dei collegamenti.
- ✓ Un **grafo** è composto da due elementi: i **nodi** anche detti **vertici** e le **relazioni** o **edge (frecce).**
- ✓ Un **Database a Grafo** è un database usato per modellare i dati nella forma di un **grafo**. In questo tipo di database, i nodi del grafo rappresentano le entità mentre le relazioni rapprosentano le associazioni tra i nodi.
- ✓ I più popolare **Database a Grafo** è **Neo4j**. Altri Database a Grafo sono Database Orale NoSQL, OrientDB, HypherGraphDB, GraphBase, InfiniteGraph e AllegroGraph.

Perchè un Database a Grafo?

- ✓ Oggigiorno, la maggior parte dei dati esiste nella forma di relazioni tra differenti oggetti e più spesso, la relazione tra i dati è più importante che i dati stessi.
- ✓ I **database relazionali** immagazzinano dati altamente strutturati che hanno parecchi record che immagazzinano lo stesso tipo di dati così che essi possano essere usati per memorizzare dati strutturati, tuttavia essi non storano le relazioni tra i dati.
- ✓ A differenza degli altri DB, i Graph DB immagazzinano le relazioni e le connessioni come entità di prima classe.
- ✓ Il modello dei dati di un

TurtleDB e Triplestore

turtleDB is a framework for developers to build offline-first, collaborative web apps. It provides a user-friendly API for developers, empowering them with the ability to create apps with in-browser storage, effective server synchronization, document versioning, and flexible conflict resolution for any document data.

Web applications will work seamlessly online or offline, and developers can leave the backend to turtleDB - it will handle all data synchronization and conflict resolution between users. Works with MongoDB out of the box!

Bibliografia

https://www.stitchdata.com/resources/data-transformation

https://www.ibm.com/cloud/learn/data-warehouse#:~:text=A%20data%20warehouse%2C%20or%20enterprise,AI)%2C%20and%20machine%20learning.