Business Intelligence

Alberto Filosa

30/9/2020

Indice

1 Sistemi Informativi

> I Sistemi Informativi di oggi rendono certamente possibili i Big Data, che in realtà si iscrivono nel costante tentativo dell'uomo di misurare,comprendere e analizzare il mondo. La rivoluzione dell'IT è evidente in molti aspetti della nostra vita, ma l'enfasi è stata posta quasi sempre sulla T di Tecnologia. Adesso è ora di rifocalizzarci sulla *I* di Informazioni.

Per *Sistema Informativo* si intende l'insieme delle informazioni utilizzate, memorizzate ed elaborate in una organizzazione per perseguire i propri fini. Per *Sistema Organizzativo*, invece, si intende l'insieme delle risorse e regole per l'utilizzo coordinato di queste risorse. Per *Sistema Informatico* si intende quella parte del sistema informativo le cui informazioni sono raccolte, elaborate e scambiate mediante l'uso delle tecnologie della informazione e della comunicazione.

Gli elementi primari di una organizzazione aziendale sono:

- *Risorsa Organizzativa*, tutto ciò con cui l'organizzazione opera per perseguire i propri obiettivi (Es: prodotti, servizi, materiali);
- *Processo*, insieme delle attività che l'organizzazione svolge per gestire il ciclo di vita delle risorse.

La ***B***usiness ***I***ntelligence (**BI**) è l'insieme dei metodi e tecniche basate su tecnologie di elaborazione di informazione per analisi di dati business, effettuati su fatti caratteristici (Es: prestazione, vendite, costi) osservati sotto diverse dimensioni. Le *Business Analytics* sono tecnologie basate su competenze e tecnologie applicative per una continua esplorazione e studio delle performance del business passato per conoscere e pianificare il business del futuro.

Le soluzioni principali della BI sono:

- Reporting, in modo da accedere in tempi veloci ai dati nei Data Warehouse per rispondere a domande specifiche;

- Cubi Multidimensionali e Analisi OLAP, ai fini di una navigazione dei dati secondo logiche dinamiche e gerarchiche;
- Dashboard, fornendo informazioni grafiche e sintetiche allo studio in questione;
- Alerting, per segnalare allarmi azionati da regole determinate ed avvisare il superamento di alcuni valori di prestazione.

Le soluzioni principali, invece, per la Business Analytics sono:

- Forecast, studio di serie storiche per individuare la tendenza e stagionalità di valori;
- Prediction, metodi di Data Mining con l'obiettivo di identificare le relazioni tra le variabili esplicative e la variabile target (Es: classificazione, regressione, clustering);
- Optimization, che permettono di identificare la decisione ottimale da effettuare in un'ampia scelta di azioni alternative (Es: massimizzazione ricavi e minimizzazione costi).

L'organizzazione deve avere delle tecniche che permettono di ottenere risposte rapide alle domande ottenendo informazioni immediatamente ed analizzare in termini efficaci le informazioni dei diversi dipartimenti.

Il valore della BI di una azienda dipende principalmente da 3 fattori:

- Livello di Disponibilità delle soluzioni all'interno dell'organizzazione;
- Livello di Responsabilizzazione, numero di utenti autorizzati ad effettuare richieste specifiche;
- Propensione Culturale a superare i compartimenti stagni della organizzazione.

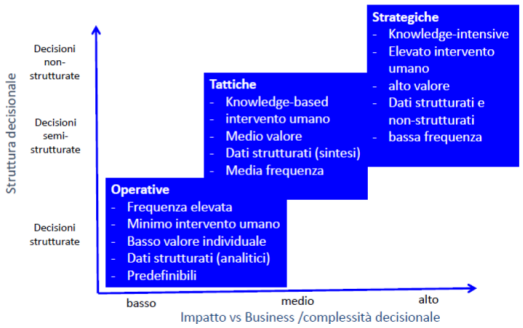
2 Decision Making

L'azienda è vista come un sistema decisionale articolato su:

1. Processi fisici, produttivi e distributivi;
2. Decisioni operative che controllano le operazioni ordinarie (processi fisici);
3. Decisioni che valutano il risultato delle decisioni operative e che ne variano le regole.

La Griglia di Scott Morton prende spunto dalla suddivisione delle tipologie decisionali di della piramide di Simon e la semplifica in tre classi:

- Decisioni Strutturate, le principali azioni operative predefinite che hanno un basso intervento umano;
- Decisione semi-Strutturate, le principali azioni tattiche che richiedono l'intervento umano su dati strutturati;
- Decisioni non Strutturate, le principali azioni strategiche che richiedono un elevato intervento umano che richiedono una alta complessità decisionale.



3 Big Data

Per Big Data si intende il processo di elaborazione veloce di grandi moli di dati di diverse tipologie e provenienti da diverse sorgenti. Esistono diversi servizi tecnologici e cloud all'interno di una azienda:

- *PaaS*, che fornisce ed opera su alcuni software tipicamente organizzate, gestite e mantenute su cloud;
- *IaaS*, che possiede ed opera su applicazioni che risiedono su server remoti e gestite in cloud;
- *SaaS*, che possiede ed opera su software che risiedono tramite *pay per use* gestite in cloud.

Prima dei Big Data gran parte delle fonti da cui si attingono i dati (prevalentemente strutturati) erano interne all'azienda, si applicano delle applicazioni di ***E***xtraction ***T***ransformation ***L***oading (**ETL**) e si immagazzinano in un Data Warehouse, solitamente con uno schema a stella. Sostanzialmente è un approccio Top-Down, che genera la possibilità di identificare in termini deduttivi le modalità con cui si interviene sui dati.

Prima dei Big Data le analisi si limitavano a testare un numero limitato di ipotesi predefinite prima della raccolta dati. Adesso è possibile far emergere collegamenti inaspettati tra le variabili. Ad esempio, alcuni fondi speculativi consultano Twitter per prevedere le performance del mercato azionario.

Dopo i Big Data è possibile intraprendere la strada precedentemente analizzata, oppure estrarre da diversi fonti di dati, principalmente sul Web e spesso non strutturati, e creare un Data Lake che immagazzina i dati nel loro formato naturale. Successivamente verranno utilizzate tecniche di manipolazione di dati per lo storage in un Data Warehouse e per elaborali. Questo è un approccio Bottom-Up, che genera la possibilità di identificare in termini induttivi le modalità di intervento dei dati (si parte dal dato per formulare domande che il dato genera).

I task principali dei Big Data sono:

- *Data Availability*, il livello di disponibilità dei dati e se sono disponibili;
- *Data Quality*, la rilevanza e consistenza dei dati, il livello di copertura e quanto sono aggiornati;
- *Data Discovery*, individuare una qualità di dati alta da una vasta collezione di essi;
- *Completezza dei Dati*, aree con poche informazioni;
- *Privacy*, estrazione di informazioni personali sufficienti ad effettuare analisi a supporto dei clienti senza compromettere la privacy.

3.1 Data Process

La Data Ingestion è il processo di estrazione ed importazione di dati dal Web in un database. In base all'importazione dei dati esistono diversi tipi di estrazioni dati:

- *Scraping*, estrazioni di dati strutturati di una parte della pagina Web. I problemi principali di questo processo sono la bassa scalabilità e il continuo aggiustamento dei dati;
- *Crawling*, estrazione della completa pagina Web (in pratica è il download in formato HTML). I problemi sono la presenza di dati non strutturati e di conseguenza il rumore per l'estrazione dei dati di interesse;
- *API*, estrazione di dati direttamente collegato al database sorgente. Il vantaggio principale dell'utilizzo delle API rispetto allo scraping è la possibilità di scaricare enormi quantità di dati in poco tempo, in più i dati sono scalabili, in quanto strutturati e con una alta qualità. I problemi riguardano gli accordi tra le parti, utente che richiede l'utilizzo ed azienda, e le differenti strutture dei dati e la conseguente sistemazione dei dati.

3.2 Dark Side

Sono stati accumulati anni di esperienza nello studio del comportamento umano, oggi una delle domande principali riguardo i Big Data è come è possibile regolamentare un algoritmo. Dalla nascita della informatica sono state regolamentate azioni per la tutela della privacy. Con i Big Data gli utenti condividono volentieri le informazioni on-line. Un altro problema principale è passare dalla privacy alla probabilità, ad esempio la probabilità di commettere un crimine ed arrestare preventivamente il soggetto. Inoltre, bisogna domandarsi quale sarà il ruolo della libertà e volontà rispetto alla dittatura dei dati e quale sarà il ruolo dell'intuito e della incertezza in contraddizione al dato empirico.

4 Database

Un modello di dati è un insieme di concetti utilizzati per organizzare i dati di interesse e descriverne la struttura in modo da risultare comprensibile a tutti. Il modello relazionale dei dati, attualmente il più diffuso, permette di definire tipi per mezzo della relazione, che consente di organizzare i dati in insiemi di record e struttura fissa.

Le fasi della progettazione di un Database sono divise in:

- *Progettazione Concettuale*, nella quale si rappresentano i dati nella realtà di interesse ad alto livello indipendentemente dal DBMS costruendo uno schema concettuale (Es: Modello Entità-Relazioni);
- *Progettazione Logica*, nella quale si rappresentano i dati in termini dei costrutti logici di una classe di DBMS. Lo schema del Database viene interpretato da una macchina, il DBMS per l'appunto. Si costruisce uno schema logico rappresentando i dati in un modello logico (Es: Modello Relazionale);
- *Progettazione Fisica*, rappresentando i dati attraverso strutture dati di uno specifico DBMS. Si costruisce uno schema fisico che consiste in uno schema logico con alcune scelte di ottimizzazione per implementare il DBMS (Es: Modello Relazionale con strutture fisiche).

Una delle peculiarità dei modelli concettuali è l'astrazione, la capacità di individuare caratteristiche comuni in

un insieme di oggetti. Le principali astrazioni di base per la rappresentazione della conoscenza sono:

- Classificazione, la capacità di definire classi di oggetti, caratterizzati da una proprietà comune, o fatti del mondo reale;
- Aggregazione, classi di oggetti incluse in altre classi;
- Generalizzazione, costruzione di gerarchie tra elementi di due o più classi.

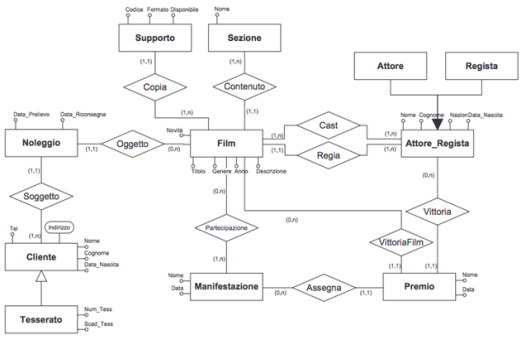
4.1 Modello Entità Relazioni

Il *Modello Entità Relazioni* è un modello concettuale di dati che fornisce una serie di costrutti che descrivono la realtà di interesse in maniera comprensibile astraendo i concetti di organizzazione dei dati nei calcolatori. Per *Entità* si intende una classe di oggetti (o di fatti) che hanno proprietà comuni rilevanti per l'applicazione. Ognuna di essa può essere caratterizzata da proprietà, chiamati *Attributi*. Per distinguere le istanze bisogna definire un insieme di attributi univoci per ogni istanza, chiamate chiavi. Essi possono essere:

- Scalari, associato un solo valore;
- Multipli, associati n valori.

Alle volte è utile decomporre concettualmente una entità in gerarchia di entità con diverso livello di dettaglio. Si tratta di gerarchia di *Generalizzazione* tra le entità, che può essere parziale o totale. In questo modo il diagramma è più comprensibile e si osserva il legame tra la classe padre e figlia.

Una *Relazione* rappresenta un legame logico tra due o più entità di interesse. Ognuna di essa presenta una cardinalità, ovvero un vincolo sul numero di istanze con cui un'entità partecipa ad una relazione.



4.2 Structured Query Language

Lo ***S***tructured ***Q***uery ***L***anguage (**SQL**) è un linguaggio di interrogazione dati dei database relazionali. I dati sono organizzati in tabelle, ognuna composta da attributi, nome delle colonne, e da tuple, le osservazioni. Il software che gestisce i dati strutturati è chiamato DBMS, che può gestire più database contemporaneamente.

Una *chiave primaria* nel modello relazionale delle basi di dati è un insieme di attributi che permette di individuare univocamente un record in una tabella. Una *chiave esterna*, invece, è un insieme di attributi che fa riferimento a una chiave di un'altra tabella per unire tramite una join due tabelle diverse. Una tabella deve obbligatoriamente possedere una e una sola chiave primaria, e nessun record nella tabella può lo stesso valore di un altro record: questo vincolo è chiamato vincolo di unicità.

Gli operatori principali di una interrogazione SQL sono:

- Proiezione (π), selezione di alcuni attributi di una tabella (*Select*);
- Selezione (σ), selezione di alcune tuple di una tabella (*Where*);
- Prodotto Cartesiano (x), l'insieme delle di tutte le possibili coppie ordinate di elementi;
- Join ( ), combinazione delle tuple di due o più relazioni di un database (Operatore composta da Proiezione e Selezione).

Una interrogazione di una tabella SQL avviene nel seguente modo:

```
SELECT <attributo/i>
FROM <tabella/e>
WHERE <condizione/i>
GROUP BY <attributo/i>
ORDER BY <attributo/i> ASC/DESC
HAVING <condizione/i>;
```

Il comando *Select* ha il compito di selezionare le colonne di una o più colonne all'interno di un DataBase. Se sono presenti due colonne da tabelle diverse, è necessario specificare da quale tabella è stata presa. Il comando *Where* specifica una condizione di ricerca per filtrare le righe selezionate nella *Select*: tramite operatori logici è possibile utilizzare più condizioni. Il comando *Group by* è utilizzato assieme alla select per raggruppare le righe di una tabella. Se si vogliono fare delle condizioni sul raggruppamento, si utilizza il comando *Having*.

5 Data Warehouse

La disponibilità di enormi quantità di dati rende molto complicato estrarre informazioni importanti. Il fenomeno del Data Warehousing nasce dall'enorme accumulo

di dati ed utilizzarli per scopi che superino quelli legati all'elaborazione giornaliera. L'obiettivo fondamentale è supportare le decisioni da prendere estraendo le informazioni da un insieme di dati del passato.

> Il Data Warehouse è una collezione di dati di supporto al processo decisionale orientata ai soggetti di interesse, integrata e consistente, rappresentativa dell'evoluzione temporale e non volatile (accessibile in sola lettura).

La costruzione di un sistema di Data Warehouse non comporta l'inserimento di nuove informazioni, bensì la riorganizzazione di quelli esistenti.

Le principali differenze tra un Database ed un Data Warehouse è che il primo è una collezione di dati che rappresenta alcuni elementi, mentre il secondo permette l'immagazzinamento storico dei dati da diverse sorgenti. Inoltre, nei Database le unioni di tabelle sono complicate in quanto normalizzate, mentre nei Data Warehouse no.

5.1 Architettura

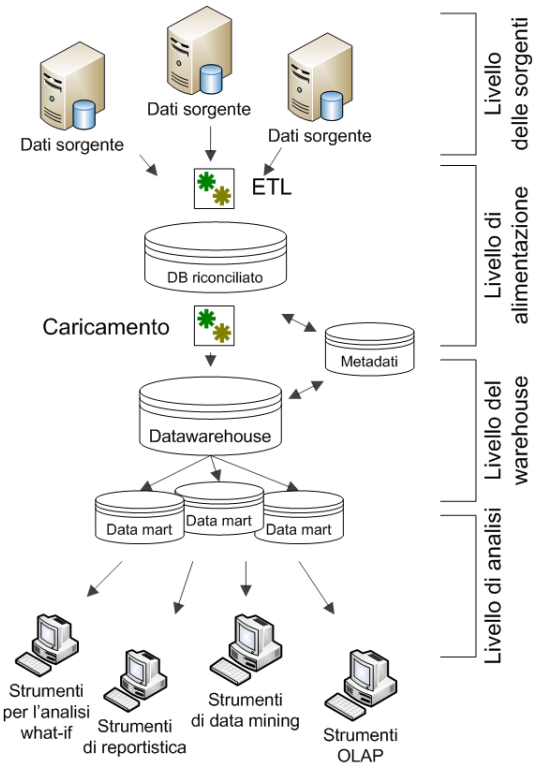
Le principali caratteristiche di un Data Warehouse sono:

- Separazione, l'elaborazione analitica e transazionale devono essere mantenute il più possibile separate;
- Scalabilità, le componenti hardware devono essere ridimensionati in base alla variazione dei dati;
- Estendibilità, integrabilità di nuove applicazioni senza riprogettare il sistema;
- Sicurezza, gli accessi devono essere controllati;
- Amministrabilità, la complessità dell'amministrazione del sistema non deve essere troppo elevata.

Generalmente un Data Warehouse è composto da 3 livelli:

1. Livello delle Sorgenti, i database delle transazione (una sequenza di operazioni che produce una variazione nello stato di un Database) dei processi operativi per estrarre i dati e caricarli nel Data Warehouse;
2. Livello del Warehouse, il livello di costruzione della base di dati che è capace di integrare i dati da diverse sorgenti;
3. Livello di analisi, gli strumenti di reportistica ed OLAP multidimensionale.

In particolare, è possibile introdurre nel Data Warehouse uno (o più) *Data Mart*, ovvero un sottoinsieme di dati contenenti informazioni rilevanti per una particolare area, ed i *Metadati*, dei dati che spiegano il dato considerato (in un DW indicano sorgenti, valori ed uso dei dati memorizzati). Essi descrivono come i dati sono alterati e trasformati durante i passaggi attraverso i livelli dell'architettura.



Il Data Warehouse si basa sul concetto di *multidimensionalità* dei dati: una volta che i dati sono stati ripuliti, integrati e trasformati bisogna capire come trarne il massimo vantaggio. Gli oggetti che influenzano le decisione sono **Fatti** del mondo di interesse, le occorrenze di esse corrispondono ad un **Evento** e per ciascun fatto interessano i valori di un insieme di **Misure** che descrivono quantitativamente gli eventi in diverse **Dimensioni**.

Esistono due tipologie di modelli per il DW:

- Modello Logico, utilizzando un modello a Stella, ovvero costruito da una tabella centrale di Fatti che referenzia su diverse tabelle di dimensioni, oppure un modello a Fiocco di Neve, che possiede un livello di normalizzazione delle tabelle maggiore ed una tabella delle dimensione può essere ulteriormente scomposta in un'altra tabella;
- Modello Operazionale, utilizzando un Data Cube che descrive tutte le possibili aggregazioni effettuabili in base al tipo di dimensioni selezionate.

I fatti di interesse sono rappresentati in *cubi* in cui ogni cella contiene misure che quantificano il fatto da diversi punti di vista, ogni asse una dimensione di interesse e ogni dimensione può essere la radice di una gerarchia di attributi per integrare i dati memorizzati in cubi base. Le informazioni sono categorizzate e le categorie seguono delle gerarchie in base alle proprie esigenze. Inoltre, si possono compiere operazioni di aggregazione dei dati per successive analisi. Le funzioni base di uno strumento OLAP sono diverse:

- *Pivoting*: operazione di rotazione delle dimensioni di analisi per analizzare i dati in dimensioni diverse;
- *Slicing*: riduzione della dimensionalità del cubo fissando un valore per ogni dimensione;
- *Dicing*: riduzione della dimensionalità del cubo attraverso un criterio di selezione;
- *Roll-up*: aumenta l'aggregazione dei dati eliminando un livello di dettaglio da una gerarchia;
- *Drill-accross*: collegamento tra 2 o più cubi per comparare i dati;
- *Drill-down*: diminuzione di aggregazione dei dati aumentando il livello di dettaglio di una gerarchia.

6 Data Lake

Un *Data Lake* è un repository utilizzato per analizzare grandi quantità di dati in diversi formati nel loro formato nativo; la sua architettura è basata per contenere tutti i tipi di dati, come i dati prodotti dalle macchine, dalle persone e da quelli tradizionali. Infatti, lo schema ed i requisiti dei dati non sono definiti in partenza, ma vengono delineati al momento dell'interrogazione, processo chiamato *schema on read*. In questo modo si riduce la quantità di dati inseriti nel Data Warehouse. Il Data Lake può essere utilizzato come un'area di atterraggio per la fase di store dei dati nel DWS. Per implementare un data lake si possono utilizzare diverse tecnologie, come HDFS che è la più utilizzata, NoSQL, ecc. Lo schema di un Data Lake è il seguente:

- I dati sono presi ed immagazzinati nel loro formato nativo (*schema on read*);
- I dati vengono analizzati per determinare il proprio valore in base al tipo di analisi che si vuole fare con processi scalabili paralleli;
- Si catturano i dati originali e li si memorizzano sotto forma di analisi usata precedentemente (*schema on write*).

7 Data Quality

Per *Dato* si intende una rappresentazione elettronica della informazione, immagazzinato in diverse modalità:

- Strutturato, memorizzati in formato tabellare in un database;
- Semi-Strutturato, memorizzato in formato non tabellare ma associato un tag (XML o HTML);
- Non Strutturato, memorizzato in formato non tabellare e senza tag (Word).

7.1 Applicazioni

Si applicano diverse procedure di *Qualità dei Dati* in quanto per analizzare un fenomeno specifico è necessario che siano completi (no dati mancanti), non anomali (no errori o anomalie tipo valori multipli o refusi) e consistenti (no incongruenze tra diversi database). Una dimensione della qualità del dato cattura e descrive un aspetto particolare dello stesso. Esse sono una metrica qualitativa che descrivono una proprietà di interesse. Inoltre sono dipendenti tra di loro (correlazioni positive e negative). I dati possono essere analizzati secondo diversi livelli:

7.1.1 Istanza

Il livello di Istanza identifica le righe di un database. I problemi possono avvenire su singoli record (Es: valori nulli, anomali e ambigui) oppure su record multipli (valori duplicati, contraddittori e non strutturati).

Per **Accuracy** si intende la vicinanza tra la rappresentazione v e quella corretta del fenomeno v' che si intende rappresentare. Essa può essere *sintattica*, ovvero la vicinanza tra due rappresentazioni contenuti in un dominio finito, oppure *semantica*, confrontando la rappresentazione con la controparte reale. Essa fornisce una valutazione dicotoma. La accuratezza può essere espressa a diversi livelli di granularità: ad un singolo valore di un attributo, ad una colonna, ad una relazione od all'intero Database:

Per **Completeness** si intende la completezza di informazioni di un dato. La completezza è caratterizzata secondo la presenza di valori nulli, ovvero valori non disponibili, non osservati o persi. Ne esistono di 3 tipologie:

- Relativo allo Schema, quanto i concetti sono modellati nello schema;
- Relativo alla Colonna, valutando i valori mancanti per una proprietà all'interno di una tabella;

- Relativa alla popolazione di riferimento.

Per **Currency**, **Timeless** e **Volatility** si intendono le dimensioni correlate con il tempo. In particolare:

- Currency descrive la velocità di aggiornamento di un dato;
- Timeless descrive l'appropriatezza dei dati per i propri scopi;
- Volatility descrive la frequenza di aggiornamento di un dato.

Per **Consistenza** si intende l'insieme delle regole semantiche definite all'interno di un Database relazionale, ad esempio i vincoli, le dipendenze funzioni e le chiavi primarie-esterne.

7.1.2 Schema

Il livello di Schema identifica la progettazione logica del database, alla struttura che conterrà i dati. I RDBMS definiscono una struttura che permette di evitare i problemi di data quality (Es: valori nulli, multipli e categorizzazione dati). Problemi di questo tipo portano anche a problemi di qualità nelle istanze; di conseguenza è necessario costruire un database duraturo e consistente per ottenere un miglioramento della qualità.

Il **Contenuto** è definito secondo i seguenti termini:

- Rilevanza, per i contesto di utilizzo;
- Ottenibilità, quanto i dati sono disponibili;
- Chiarezza, ad esempio la adozione di regole per eliminare ambiguità all'interno di un Database.

La **Copertura** deve essere:

- Comprensibile, lo schema deve soddisfare i requisiti dei potenziali utenti;
- Essenziale, lo schema deve soddisfare i soli requisiti dei potenziali utenti.

La **Composizione** di uno schema deve essere:

- Naturale, ovvero fatti correlati tra loro;
- Identificabile, necessita di una chiave primaria;
- Omogeneo, gli attributi devono essere applicati a tutte le entità di uno stesso tipo;
- Ridondanza Minima, i termini devono essere ripetuti solo necessariamente.

La **Consistenza** è definita in termini semantici e strutturali.

Il **Livello di Dettaglio** è definito in termini di granularità dell'attributo, in base alla disponibilità delle informazioni, ed alla precisione dei domini.

7.1.3 Formato

Il Formato comprende 7 differenti dimensioni:

- Appropriatezza, rispetto alle esigenze dell'utente;
- Interpretabilità, quanto è comprensibile un dato;
- Portabilità, la semiotica del dato;
- Precisione;
- Flessibilità rispetto ai requisiti dell'utente;
- Rappresentazione dei Valori Nulli;
- Uso efficiente Memoria.

7.2 Metodi di Miglioramento

I metodi per migliorare la qualità di un Database possono essere applicati sui dati oppure sui processi.

7.2.1 Dati

Per quanto riguarda i dati, i metodi si basano su:

- *Confronto con Controparte Reale*, il metodo più efficacie, ma anche quello più costoso in quanto richiede un controllo periodico;
- *Database Bashing*, che prevede il confronto di record in due o più Database interni con la tecnica di *Record Matching*, identificando se rappresentano la stessa entità;
- *Business Rule*, che prevede la verifica che i record rispettino determinati vincoli di business o semantici. Esso è il più conveniente in quanto più efficacie del Database Bashing e meno costoso del confronto con la controparte reale;
- Utilizzo dei *Metadati*, per memorizzare i valore delle dimensioni di qualità calcolati utilizzando approcci precedenti.

7.2.2 Processi

Il principale svantaggio dei metodi per il miglioramento della qualità basati sui dati è il non correggere le cause degli errori, perciò di non prevenire da errori futuri. I *Metodi basati sui Processi* prevedono invece un'analisi dei processi per individuare e correggere le cause di errore.

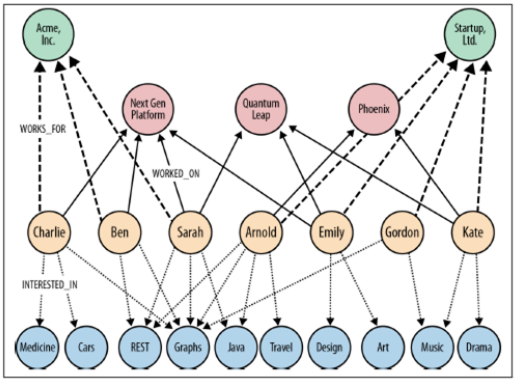
- *Controllo e Miglioramento*, controllo dei dati sui processi ed migliorare quelle parti del Database dove avvengono delle transazioni (metodo conservativo);
- *Reingegnerizzazione*, che prevede la ricostruzione dei processi delle transazioni dei dati (metodo radicale).

Per migliorare i processi è necessario individuare chi è il *Process Owner*, responsabile dei dati manipolati dal processo (Es: ministero delle finanza per accertamento fiscale), e chi è il *Data Steward*, responsabile dei processi di data governance. In questo modo è possibile individuare chi è responsabile nei processi.

8 Graph DB

Un *Graph Database* è un database utilizzato per lo storage efficiente di dati semi-strutturati. Permette di modellare le relazioni sociali tramite un modello a grafo.

Un nodo, rappresentato da una osservazione, può contenere diverse proprietà che specificano delle entità. Le relazioni, invece, rappresentano la connessione di entità. Devono obbligatoriamente avere un nome ed una direzione e devono avere un inizio ed una fine. Inoltre, anch'esse possono contenere delle proprietà, in modo tale da comprendere meglio la relazione.



Il linguaggio utilizzato in Neo4j è chiamato ***C***ypher ***Q***uery ***L***anguage (**CQL**), di tipo dichiarativo, perciò si descrive quello che si vuole, non come. Le principali funzioni sono tipo aggregativo e di ordinamento; inoltre, è possibile creare, aggiornare o eliminare elementi al grafo.

- Struttura query:

```
MATCH pattern_grafo
WHERE condizione/i
RETURN risultato
```

- Creazione nodo:

```
CREATE (:Person {name: "Charlie"})
```

- Creazione relazione:

```
MATCH (p:person), (s:skill)
WHERE p.name = 'charlie' AND s.name = 'medicine'
CREATE (p)-[r:INTERESTED_IN]->(s)
```

- Eliminare relazione e nodi:

```
match ()-[r:friends]-() delete r

match (n:person) delete n
```

- Caricare dati da CSV:

```
LOAD CSV WITH HEADERS FROM
"file:/skill.csv" AS row FIELDTERMINATOR ';'

CREATE (:skill
        {name: row.name});
```

- Eliminare indice dal grafo:

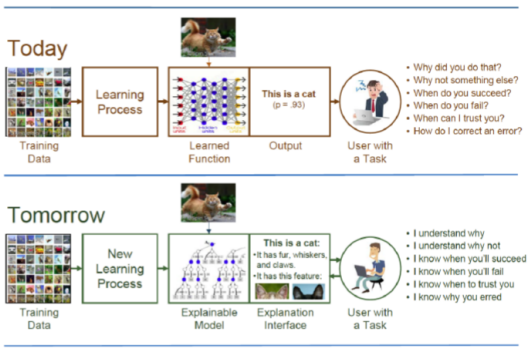
```
drop index [nome_indice]
```

- Schema del grafo:

```
:schema
CALL db.schema.visualization
```

9 Explainable AI

L'Intelligenza Artificiale è sempre più utilizzata in tutti gli ambiti lavorativi. La *Explainable AI* è un metodo efficace per far comprendere agli esperti di domini le decisioni prese da un algoritmo di Machine Learning. La necessità di utilizzo nella pratica risiede nella fiducia della decisione da prendere, nella interazione tra uomo e macchina per comprendere la decisione scelta e nella trasparenza delle decisioni.



Il problema della spiegazione dei modelli è correlata con la previsione della Accuracy. In particolare, i modelli di Neural Network hanno una Accuracy molto più elevata rispetto a semplici modelli, a discapito di una difficile interpretazione in base alla quantità di layer nascosti all'interno di essa.

La interpretabilità può essere determinata in termini:

- *Globale*, sulla rilevanza delle feature del modello considerato l'intero dataset;
- *Locale*, spiegano la decisione effettuata dall'algoritmo sulla singola istanza.

9.1 LIME

Il *LIME* è una tecnica di Explainable AI applicata a livello locale. Essa considera una singola osservazione e tutte le istanze in un intorno dello spazio delle feature (di conseguenza con caratteristiche simili). Il modello prevede lo spazio non lineare, campiona i punti vicini ed identifica un modello lineare che può essere fedele localmente. Successivamente massimizza la *local fidelity* che approssima le decisione del modello più complesso in quell'intorno.

9.2 SHAP

Lo ***SH***apley ***A***dditive ex***P***lanations è una tecnica di Explainable AI applicata a livello globale che determina quanto la previsione è determinata dalle feature. Per determinare l'importanza di una singola feature, è necessario considerare ogni possibile combinazione di essa per far apprendere un modello per volta. Lo *SHAP value* è calcolato come la somma pesata del contributo ad ogni livello della feature.