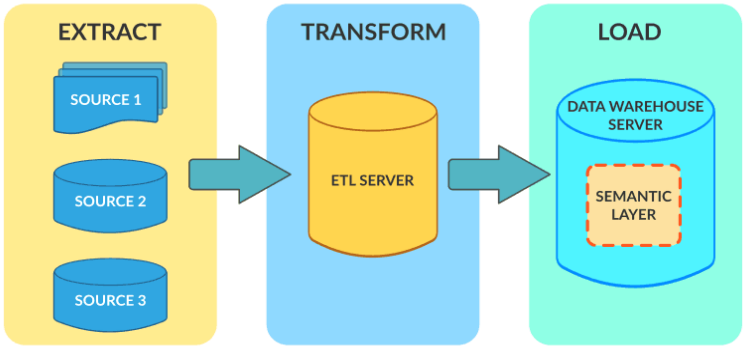
Data Science

Francesco Pugliese, PhD

neural1977@gmail.com

ETL – Extract, Transform and Load

- ✓ **ETL**, è un processo di Data Integration (Integrazione Dati) che combina i dati provenienti da diverse sorgenti di dati all'interno di una singola data store consistente che è in genere caricato in un data warehouse o un sistema Target.
- ✓ Man mano che i database sono cresciuti in popolarità intorno al 1970, **l'ETL** fu introdotto come processo di integrazione e caricamento dati per elaborazione ed analisi, e alla fine è divenuto il metodo primario per processare dati per i progetti di data **warehousing**.
- ✓ Un Enterprise Data Warehouse (EDW) è un sistema che aggrega dati provenienti da differenti sorgenti in un singolo data store che supporti processi come: data analysis, data mining or Artificial Intelligence (AI, ML)



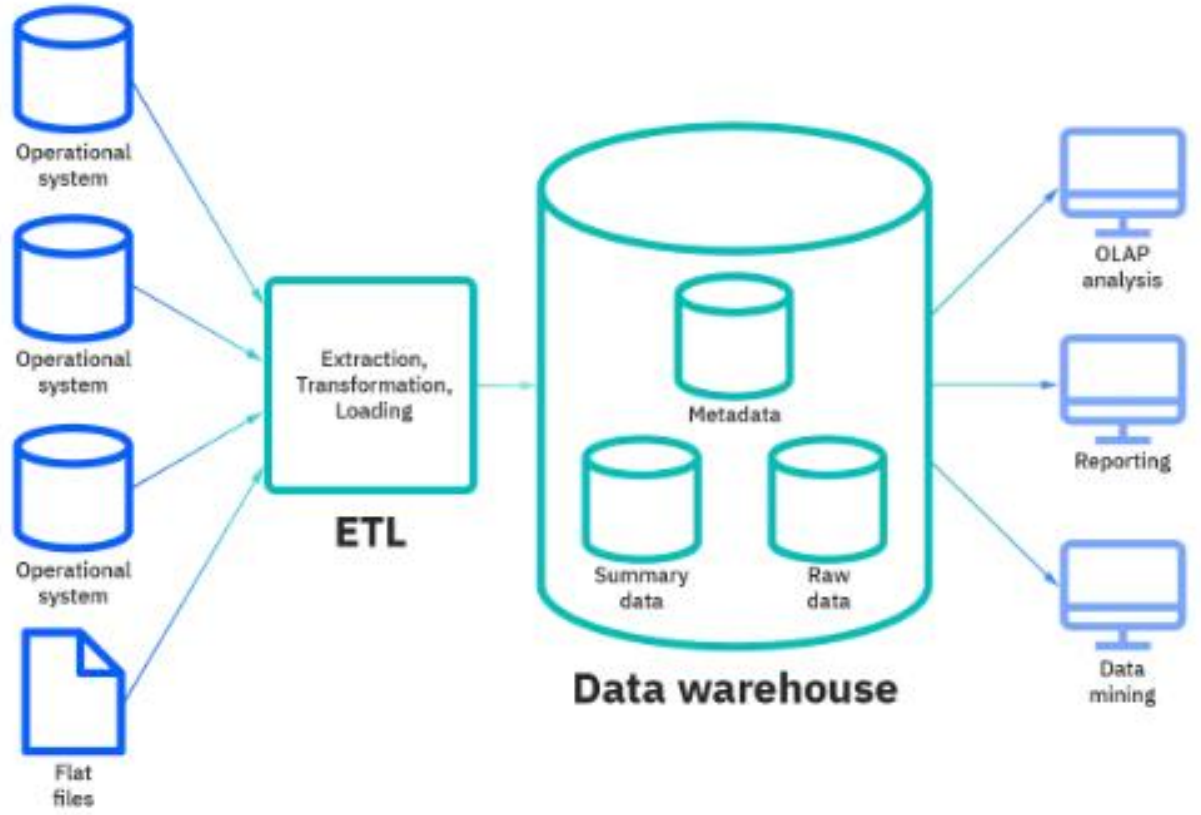
ETL – Data Warehouse

- Un **Data Warehouse** è un sistema che permette ad un'organizzazione di eseguire potenti analisi su elevati volumi (petabytes e petabytes) di dati in modalità che un database standard non è in grado di eseguire.
- I sistemi di **Data Warehouse** sono stati una parte dei sistemi di Business Intelligence per oltre 3 decenni, ma si sono evoluti solo di recente mediante nuovi titpi di dati e metodi di hosting.
- Originariamente un Data Warehouse veniva ospitato on-premises su un computer mainframe, e le sue funzionalità si focalizzavano sull'estrazione dei dati da varie sorgenti, pulizia e preparazione dei dati, caricamento e immagazzinamento dei dati all'interno di un database relazionale.
- Più recentemente, un Data Warehouse può essere ospitato su un dispositivo dedicato o su un cloud, e alla maggior parte dei sistemi di Data Warehouse sono state aggiunte capacità analitiche, di visualizzazione dati e tool di presentazione.

ETL – Architettura di un Data Warehouse

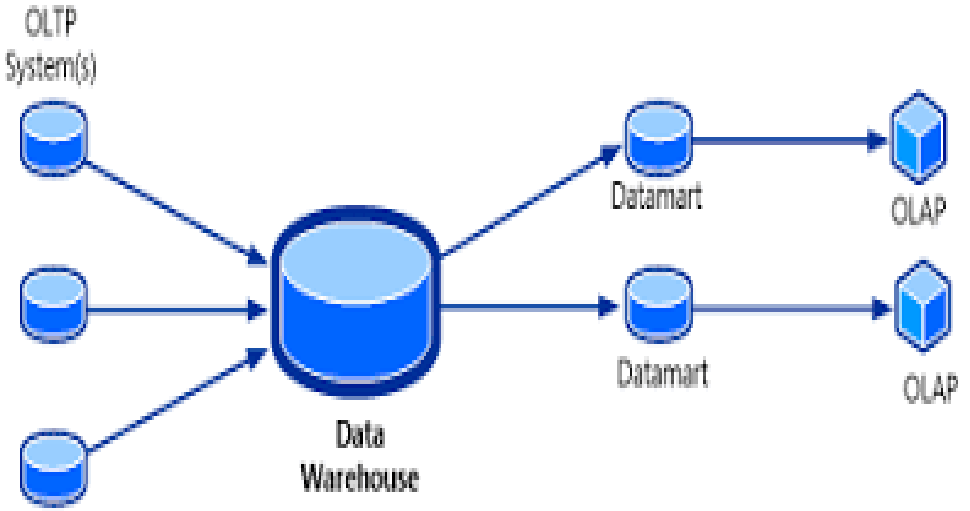
- ✓ Generalmente parlando ha una architettura three-tier (3 livelli):

Bottom tier: E' costituito da una data warehouse server, di solito si tratta di un sistema database relazionale, il quale colleziona, ripulisce e trasforma i dati provenienti da sorgenti di dati multiple attraverso un processo conosciuto come ETL (Extract, Transform and Load) o un processo conosciuto come Extract, Load and Transform (ELT).



ETL – Architettura di un Data Warehouse

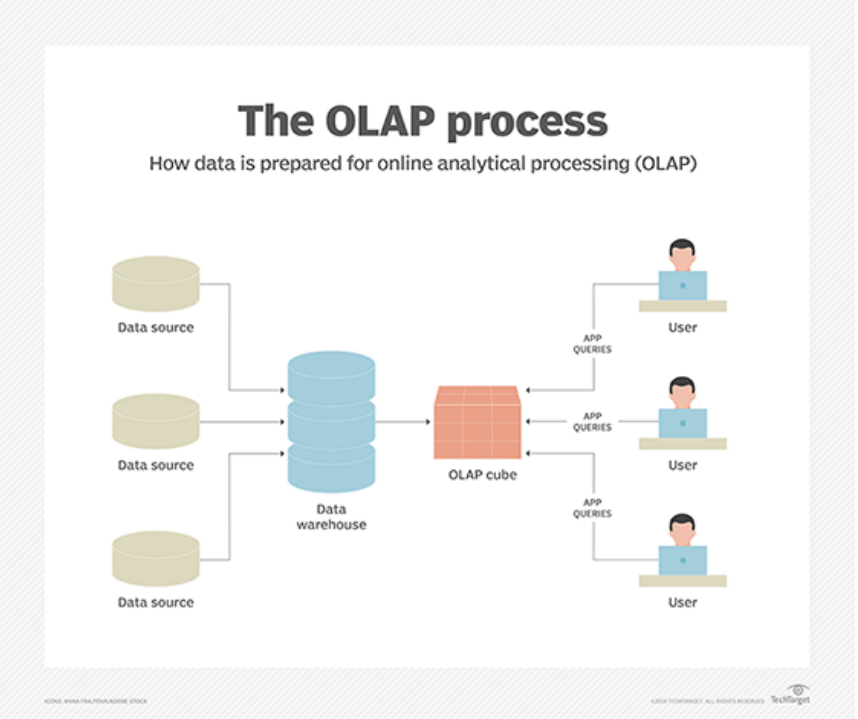
Middle tier: Questo è costituito da un **OLAP** (ossia un **OnLine Analytical Processing**) Server che abilita l'utente ad avere delle velocità di query elevate. Esistono 3 tipi di modelli OLAP che possono essere usati in questo tier conosciuti come: **ROLAP**, **MOLAP** e **HOLAP**. Il tipo di modello OLAP usato è dipendente dal tipo di sistema database che esiste.



Top tier: Questo livello è rappresentato da qualcosa del tipo interfaccia **front-end user** o vari tool di reportistica, che abilita l'utente finale a condurre analisi ad-hoc sui loro dati di business.

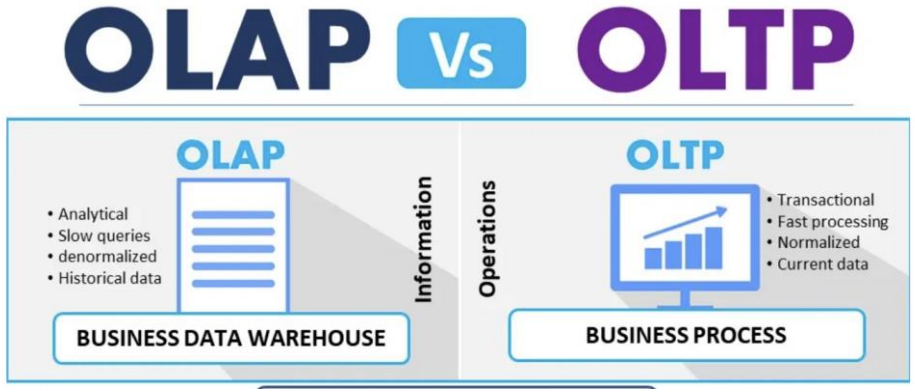
Data Warehouse – Comprendere OLAP e OLTP

- **OLAP (Online Analytical Processing)** è un software per eseguire analisi multidimensionali ad alta velocità su grandi volumi di dati provenienti da data store unificati e centralizzati come i **Data Warehouse**. **OLTP (Online Transactional Processing)** abilita l'utente ad avere un'esecuzione in tempo reale su grandi numeri di transazioni su database effettuate da un gran numero di persone, tipicamente su Internet.
- La principale differenza tra **OLAP** e **OLTP** è il nome: **OLAP** è analitico per natura mentre **OLTP** è transazionale.



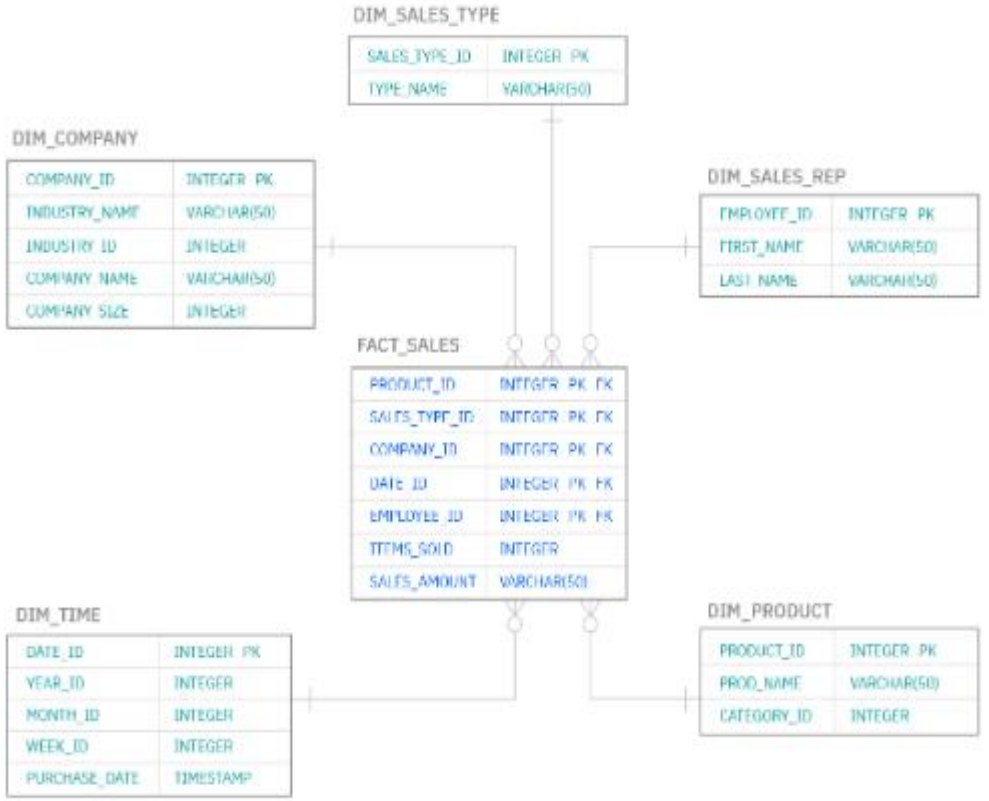
Data Warehouse – Comprendere OLAP e OLTP

- ✓ I tool **OLAP** sono progettati per l'analisi multidimensionale dei dati all'interno di un Data Warehouse, il quale contiene sia dati storici che transazionali. I comuni usi di OLAP sono il **Data Mining** ed altre applicazioni di Business Intelligence, calcoli analitici complessi, scenari predittivi, come anche funzioni di reportistica di business come analisi finanziaria, budgeting e forecast plannig.
- ✓ **OLTP** è progettato per supportare applicaziomni orientate alle transazioni elaborando transazioni recenti il più rapidamente e accurato possibile. Comuni usi di OLTP includono ATMs, software e-commerce, elaboraziojni di pagamenti di carte di credito, prenotazioni online, sistemi di prenotazioni, strumenti di record-keeping, ecc.



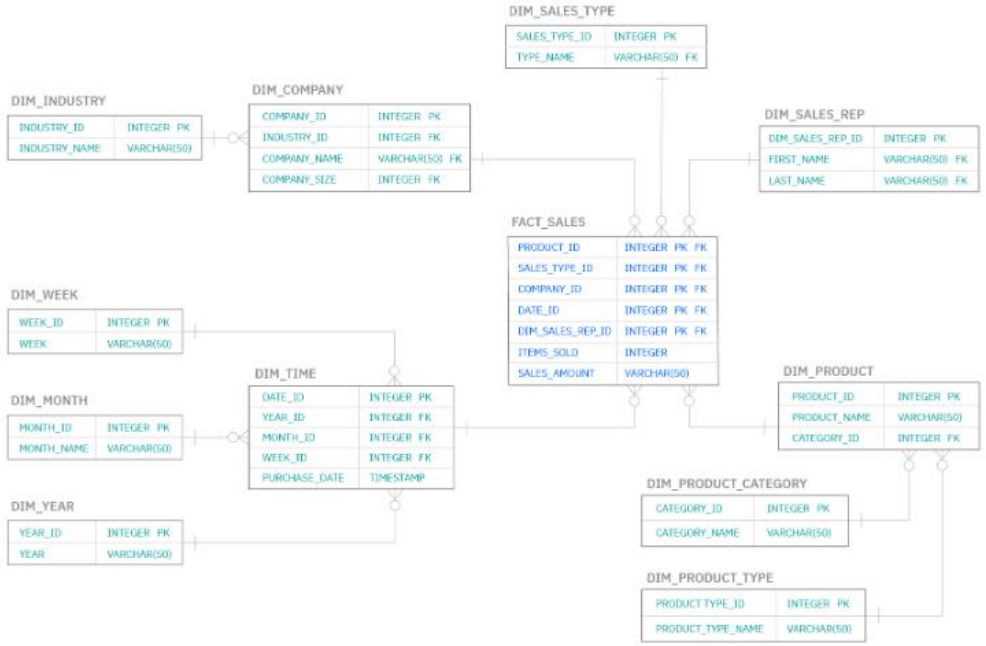
Data Warehouse –Schema

- ✓ Gli schemi sono i modi in cui i dati sono organizzati all'interno dei database o data ware house. Ci sono due principali strutture degli schemi: lo **star schema** e lo **snowflake**, che influenza il progetto del modello dei dati.
- ✓ **Star Schema:** Questo schema è costituito da una «fact table» che può essere unita ad un certo numero di «dimension table» denormalizzate. Esso è considerato il più comune tipo di schema, e i suoi utenti beneficiano delle sue rapide velocità durante il processo di query.



Data Warehouse –Schema

- **Snowflake Schema:** Anche se non è ancora ampiamente usato, lo snowflake schema è un'altra organizzazione della struttura in un data warehouse. In questo caso la fact table è connessa a un numero normalizzato di dimension table. Questo schema è costituito da una «fact table» che può essere unita ad un certo numero di «dimension table» denormalizzate. Esso è considerato il più comune tipo di schema, e i suoi utenti beneficiano delle sue rapide velocità durante il processo di query.

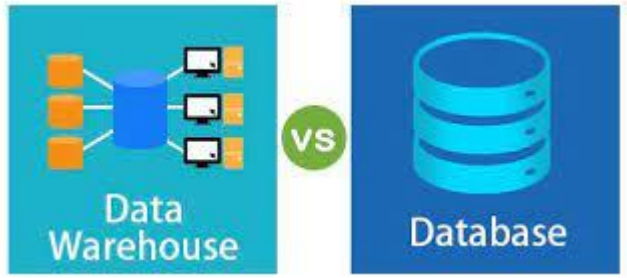


Data Warehouse vs Database, Data Lake, Data Mart

- **Data Warehouse vs Data Lake:** Un **Data Warehouse** colleziona e riunisce dati grezzi da sorgenti multiple di dati verso un repository centrale, strutturato usando uno schema predefinito di dati espressamente progettato per l'analisi dei dati. Mentre un **Data Lake** è un data warehouse senza uno schema predefinito. Come risultato, il data lake permette più tipi di analisi che un data warehouse semplice. I data lake sono comunemente costruiti su piattaforme di Big Data come Apache Hadoop.
- **Data Warehouse vs Data Mart:** Un **Data Mart** è un sottoinsieme di un data warehouse che contiene dati specifici per una particolare linea di business o dipartimento. Dal momento che il data mart contiene un più piccolo sottoinsieme di dati, i data mart permettono ai dipartimenti o linee di business di scoprire informazioni di valore più focalizzati e più rapidamente di quanto sia possibile lavorando con un dataset più ampio presente in un data warehouse.

Data Warehouse vs Database, Data Lake, Data Mart

- ✓ **Data Warehouse vs Database:** Un database è costruito principalmente per soddisfare rapide query ed elaborazioni di transazioni, non per fare analytics.
- ✓ Un database tipicamente viene utilizzato come **data store** focalizzato ad una specifica applicazione, mentre un data warehouse immagazzina dati provenienti da qualsiasi applicazione o persino tutte quelle appartenenti in una organizzazione.
- ✓ Un database si focalizza su aggiornamento dati in tempo reale mentre un data warehouse ha un obiettivo più ampio, catturando serie storiche di dati o correnti per effettuare analisi predittiva, machine learning e altri tipi di analisi avanzate.



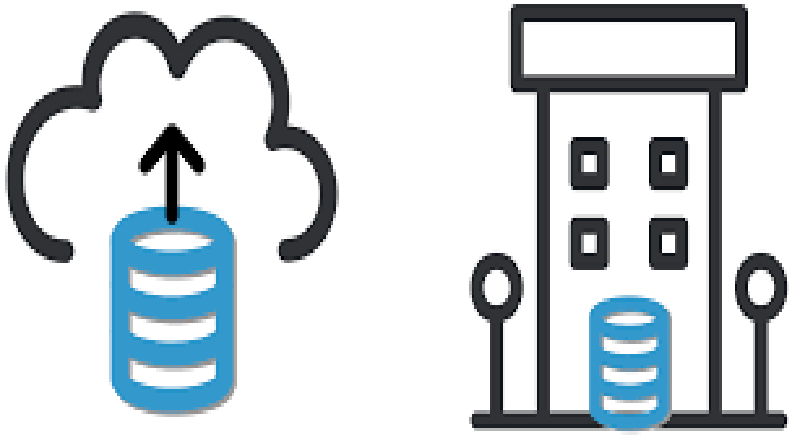
Tipi di Data Warehouse

- **Data Warehouse su Cloud:** Un Cloud Data Watehouse è un data warehouse specificatamente costruito per essere eseguito su cloud, e viene offerto ai clienti come un servizio gestito dal cloud. I data warehouse basati su cloud sono divenuti sempre più popolare negli ultimi 5 anni dal momento che molte compagnie usano servizi cloud per cercare di ridurre sempre più l'impatto dei loro **data center** on-premises.

- Con il termine **software on premise** (od on premises, come sarebbe più corretto) si fa riferimento alla fornitura di programmi informatici installati e gestiti attraverso computer locali. Deriva dall'inglese “on the premises”: nelle sedi, nei locali (del titolare della licenza).

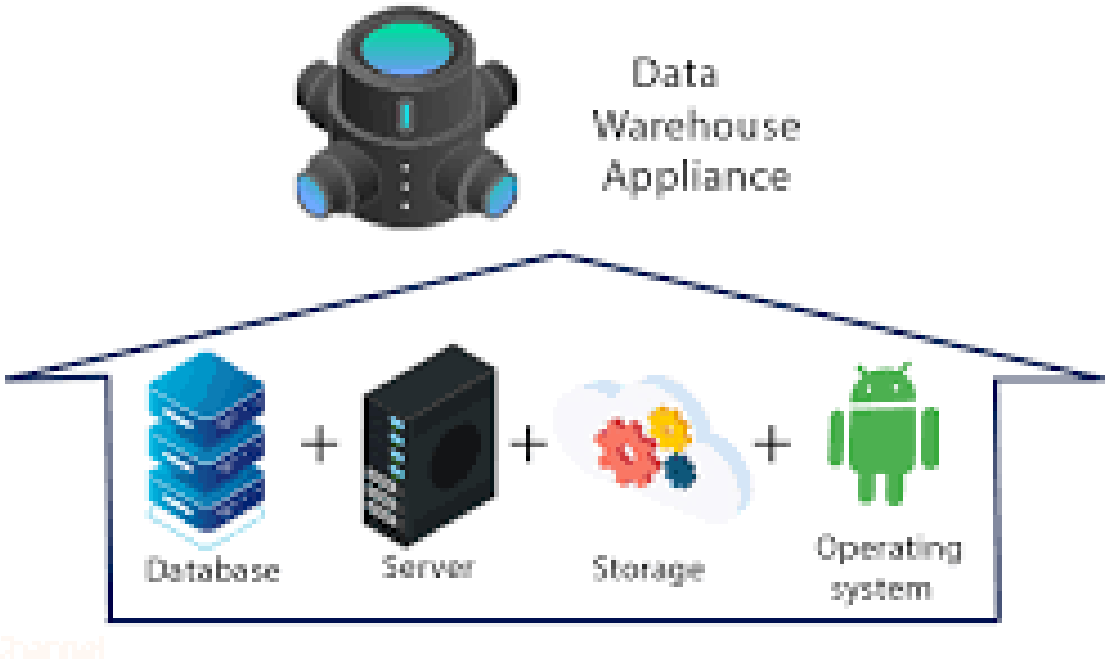
Tipi di Data Warehouse

- **Data Warehouse Software (on-premises / su licenza):** Una organizzazione può acquistare un data warehouse sotto licenza e poi deployare il data warehouse sulla propria infrastruttura on-premises. Sebbene questo sia tipicamente più costoso di un servizio di data warehouse su cloud, può essere una scelta migliore per entità governative (come **l'ISTAT**), istituzione finanziarie o altre organizzazioni che vogliono avere più controllo sui loro dati o anno la necessità di soddisfare rigide norme di sicurezza o standard di privacy dei dati o regolamentazioni varie.



Tipi di Data Warehouse

- **Apparati di Data Warehouse (Data Warehouse Appliance):** Un'apparato di Data Warehouse è un insieme di sistemi hardware e software come CPU storage, sistema operativo e data warehouse software che un'organizzaione può connettere alla su rete e usarla come parte di essa. Un data warehouse appliance si colloca tra cloud e le implementazioni on-premises in termini di costi, velocità di deployment, scalabilità, e controllo di gestione.



Benefici di un Data WarehouseTipi di Data Warehouse

1. **Migliore qualità dei dati:** Un data warehouse centralizza i dati da una varietà di sorgenti di dati, come sistemi transazionali, database operazionali, e file piatti. Dunque, ripulisce i dati, elimina i duplicati e li standardizzato per creare un unica sorgente di dati.
2. **Più veloce e informazioni di business:** I dati provenienti da disparate sorgenti limitano il potere decisionale dei decision makers per avviare strategie di business con una certa affidabilità. I data warehouse permettono la **Data Integration** (Integrazione Dati), permettendo agli utenti del business di estrarre tutte le informazione necessarie dai dati della compagnia durante ciascuna decisione di business.

Benefici di un Data WarehouseTipi di Data Warehouse

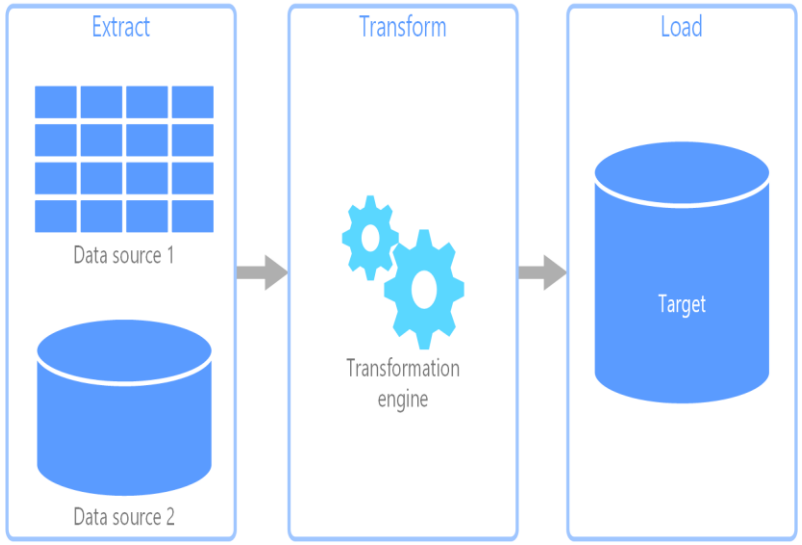
3. **Decision-making più intelligente:** Un data warehouse supporta funzioni di Business Intelligence ad ampia scala come il **data mining** (che cerca pattern e relazioni nei dati), intelligenza artificiale e machine learning. I professionisti e i leader di business possono usare i dati per prendere decisioni smart in virtualmente ogni area dell'organizzazione, dai processi di business al management finanziario all'inventory management.

4. **Guadagnare e far crescere un vantaggio competitivo:** Tutti i benefici visti si combinano per aiutare un organizzazione a trovare più opportunità nei dati, più rapidamente di quanto sia possibile con data store displocati in luoghi diversi e disparati.

ETL – Extract, Transform and Load

- ✓ **ETL** fornisce le fondamenta per la data analytics e i workstream di machine learning. Attraverso una serie di regole, l'ETL purifica e organizza i dati in un modo che incontra specifici bisogni di business intelligence, come report mensili ma può anche migliorare i processi di back-end o l'esperienza dell'utente finale.
- ✓ In genere **l'ETL** è utilizzato dalle organizzazioni per:

1) Estrarre dati da sistemi legacy
2) Ripulire i dati per migliorarne la qualità e renderli consistenti
3) Caricare i dati all'interno di un database target



ETL – Sistemi Legacy

- Un sistema **legacy**, in informatica, è un sistema informatico, un'applicazione o un componente obsoleto, che continua ad essere usato poiché l'utente (di solito un'organizzazione) non intende o non può rimpiazzarlo. Legacy equivale a versione "retrodatata" (rispetto ai sistemi/tecnologie correnti). Un esempio sono il **Cobol** o i **Mainframes** dei sistemi bancari.

ETL versus ELT

- ✓ La più semplice differenza tra **ETL** e **ELT** è in termini di operazioni. **ELT** copia ed esporta i dati dalle sorgenti, ma invece di caricarli in su un'area per la trasformazione successiva, **l'ELT** carica i dati grezzi direttamente sullo store di target dei dati per poter essere trasformati alla bisogna.
- ✓ Mentre entrambi **ETL** e **ELT** fanno leva su una varietà di repository di dati, quali database, data warehouse e data lake, ciascuno dei due processi possiede i suoi vantaggi e svantaggi.

1. **ETL** è particolarmente utile per dataset ad alto volume non strutturati dal momento che il caricamento può avvenire direttamente dalla sorgente. Questo processo richiede più definizione all'inizio, le regole di business per la data transformation hanno bisogno di essere costruite.
2. **ELT** è più ideale per nel mondo dei Big Data dal momento che non richiede una progettazione anticipata per la data extraction e lo storage dei dati. **ELT** è divenuto più popolare con l'adozione dei database su cloud, anche se non ci sono ancora molte best practices su **ELT**.

Trasformazione dei Dati (Data Transformation)

- ✓ L'Analisi dell'informazione richiede di solito dati accessibili e ben strutturati per ottenere i migliori risultati possibili. La Data Transformation rende alle organizziazioni possibile l'alterazione della struttura e del formato dei dati grezzi secondo le necessità. La Data Analytics più efficiente deriva anche dal modo in cui l'impresa trasforma i suoi dati.
- ✓ La **Data Transformation** è il processo di cambiamento del formato, struttura, e valori dei dati. Per i progetti di data analytics, i dati possono essere trasformati a due livelli della pipeline dei dati. Le organizzazioni che usano i **data warehouse** on-premises generalmente usano un processo ETL, in cui la trasformazione dei dati è il processo intermedio. Oggigiorno, la maggior parte delle organizzazioni usano i data warehouse basati su cloud, i quali possono scalare le risorse di computazione e di storage con una latenza misurata in secondi o minuti.

Trasformazione dei Dati (Data Transformation)

- ✓ La scalabilità delle piattaforme di cloud permette alle organizzazioni di evitare delle trasformazioni precaricate e di caricare i dati grezzi all'interno del data warehouse, per poi trasformarli in una query successiva, si tratta del modello chiamato **ELT**, dove in pratica la trasformazione viene postposta rispetto all' **ETL**.
- ✓ Tutti i seguenti processi implicano la **Data Transformation**:

1. **Data Integration**
2. **Data Migration**
3. **Data Warehousing**
4. **Data Wrangling**

Processi collegati alla Data Transformation: Data Integration

- ✓ La **Data Integration** è la combinazione di processi tecnici ed economici usati per combinare dati da diverse sorgenti verso una informazione significativa e preziosa.
- ✓ Una soluzione di **Data Integration** completa fornisce dati affidabili provenienti da varie sorgenti.
- ✓ La **Data Integration** combina dati da multipli sistemi distinti in una vista unificata. Questa vista unificata è tipicamente immagazzinata in un repository centrale di dati conosciuto come **data warehouse**.
- ✓ La **Data Integration** è spesso un prerequisito per altri processi che includono, analisi, reportistica e predizione.

Processi collegati alla Data Transformation: Data Integration

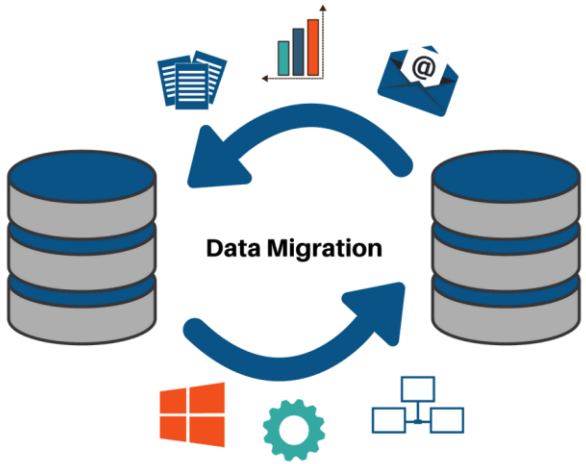
- La **Data Integration** supporta l'elaborazione analitica di grandi dataset mediante allineamento, combinazione e presentazione dei data set provenienti da dipartimenti organizzativi e sorgenti esterne remote al fine di soddisfare gli obiettivi dell'integrazione.
- Per esempio, un data set completo appartenente ad un utente potrebbe includere dati estratti e combinati dai reparti di marketing, vendite e operazioni in generale. Questo dataset può essere combinato in un modo tale che esso può essere reso consistente, completo, aggiornato e con informazioni corrette per la reportistica di business e analisi.
- I sistemi sorgente possono essere vari tipi dispositivi
- I dati sorgente possono essere di vari formati differenti.

Processi collegati alla Data Transformation: Data Migration

- La **Data Migration** è il processo di selezione, preparazione, estrazione, trasformazione e trasferimento permanente dei dati da uno storage di un computer ad un altro.
- Mentre la **Data Integration** implica la collezione dei dati da sorgenti differenti fuori da un organizzazione di analisi, la migrazione si riferisce allo spostamento di dati già immagazzinati all'interno di diversi sistemi.
- Le società tipicamente migrano i dati quando devono implementare un nuovo sistema o devono fondere il vecchio con un nuovo ambiente. Le tecniche di migraizone sono spesso eseguite da un insieme di programmi o script automatizzati che automaticamente trasferiscono i dati.
- Addizionalmente, la validazione dei dati migrati

Processi collegati alla Data Transformation: Data Migration

- ✓ Addizionalmente, la validazione dei dati migrati per completezza e il decomissionamento dei data storage di tipo legacy (obsoleti) sono considerati fasi integranti del processo di migrazione dei dati.
- ✓ Tuttavia, "**trasferire**" non è il solo aspetto della metodologia di migrazione dei dati. Se il dato è vario e diversificato, il processo di migrazione include anche operazioni di mapping e trasformazione tra sorgente dei dati e destinazione.
- ✓ Il tasso di successo di qualsiasi progetto di migrazione dati è direttamente dipendente dalla diversità, volume e qualità dei dati che devono essere trasferiti.



Processi collegati alla Data Transformation: Data Migration

✓ La maggior parte dei processi di migrazione dati avvengono attraverso 5 fasi:

1. **Estrazione**: si rimuovono i dati la sistema corrente per iniziare a lavorare su di essi
2. **Trasformazione**: convertire i dati nelle sue nuove forme assicurandosi che i metadata riflettano i dati in ciascun campo.
3. **Pulizia**: rimozione duplicati, avvio di test, e gestione di qualsiasi tipi di dato corrotto.
4. **Validazione**: testa e ritesta che spostando i dati alla locazione di target fornisca la risposta attesa.
5. **Caricamento**: trasferimento dei dati dentro un nuovo sistema e revisione degli errori nuovamente.

Processi collegati alla Data Transformation: Data Wrangling

- ✓ Il **Data Wrangling** è il processio di collezione, selezione e trasformazione dei dti che risponde ad una domanda analitica. Anche conosciuto come processo di **Data Cleaning** o **«data munging»** (cambio formato dei dati).
- ✓ Secondo una ricerca di **Elder Research** sembra che il **Data Wrangling** richieda costi analitici pari al 80% del tempo, lasciando solo il 20% **all'esplorazione** e alla **modellazione**.
- ✓ Se si vuole creare una pipeline **ETL** efficiente o creare una bella Data Visualization, è necssare fare molto data wrangling.
- ✓ Il **Data Wrangling** in sostanza è il processo di trasformazione dei dati in un formato che lo renda più facile lavorarci. Questo potrebbe significare trasformare tutti i valori di una data colonna in un certo modo o fondendo colonne multiple insieme.

Processi collegati alla Data Transformation: Data Wrangling

- ✓ La necessità del **Data Wrangling** è spesso dipendente dal dominio o dal prodotto, di dati collezionati o presentati.
- ✓ I dati vengono manualmente introdotti da umani e sono spesso caricati con errori.
- ✓ I dati collezionati dai siti web sono spesso ottimizzati per essere visualizzati sui siti web, non per essere ordinati o aggregati.
- ✓ Se lavoriamo con **SQL** regolarmente, abbiamo la necessità di diventare forti con queste capacità di data wrangling.

Trasformazione dei Dati (Data Transformation)

- ✓ La **Data Transformation** può essere:
  1. **costruttiva:** aggiungere, copiare e replicare i dati.
  2. **distruttiva:** cancellare campi e record**.**
  3. **estetica:** standardizzare i nomi delle strade
  4. **strutturale:** rinominare, spostare e combinare più colonne in un DB

- ✓ Un'organizzazione può scegliere tra una varietà elevata di **tool per ETL** che automatizzino il processo della trasformazione dei dati. Gli analisti dei dati, gli ingegneri dei dati, e i data scientist trasformano i dati anche usando **script in Python** e **linguaggi specifici del dominio come SQL**.

Benefici e Sfide della Trasformazione dei Dati (Data Transformation)

- La **Data Transformation** produce i seguenti benefici:

1. Il dato è trasformato per essere meglio organizzato. I Dati Trasformati possono essere più facili da interpretare sia per umani che per i computer.
2. Appropriatamente formattati e validati i dati migliorano la qualità dei dati e proteggono le applicazioni da problemi potenziali come: valori nulli, duplicati inattesi, indicizzazione errata e formati incompatibili.
3. La Trasformazione dei Dati facilità la compatibilità tra applicazioni, sistemi e tipi di dati. I dati usati per scopi multipli potrebbero avere bisogno di essere trasformati in differenti modi.

Benefici e Sfide della Trasformazione dei Dati (Data Transformation)

- ✓ La **Data Transformation** invece deve affrontare le seguenti sfide:
- ✓ La **Data Transformation** può essere molto costosa. Il costo è dipendente dalla infrastruttura specifica, dal software e dagli strumenti per elaborare i dati.
- ✓ La **Data Transformation** può richiedere un numero elevato di risorse. Le trasformazioni che vengono eseguite nei data warehouse on-premises dopo il caricamento dei dati e prima di immetterli i alcune applicazioni, possono creare un carico computazionale che rallenta le altre operazioni. Se si usa un data warehouse basato su cloud, è possibile effettuare le trasformazioni solo dopo aver fatto il caricamento dei dati perchè la piattaforma di cloud può scalare espandendosi per incontrare le esigenze della domanda di risorse.

Benefici e Sfide della Trasformazione dei Dati (Data Transformation)

- ✓ Mancanza di **expertise** e non curanza possono introdurre problemi durante la data transformation. Gli analisti dei dati senza appropriata conoscenza della materia in oggetto meno probabilmente noterano errori di digitazione o dati incorretti perchè non sono esperti di dominio e quindi hanno meno familiarità con gli intervalli dei valori che sono considerati accurati e permessi. Per esempio, qualcuno che lavora su dati medici che non è abituato a termini potrebbe sbagliare nell'etichettare nomi di malattie che potrebbero essere mappate su un singolo valore.
- ✓ Le imprese possono eseguire trasformazioni che non si adattano ai loro bisogni. Un'attività potrebbe cambiare informazione a uno specifico formato per un'applicazione solo per reinvertire poi l'informazione a formato orginale per una differente applicazione.

Come trasformare i dati con la Data Transformation

- ✓ La **Data Transformation** può incrementare l'efficienza dei processi di business e analitici ed rendere capaci di un migliore decision-making orientato ai dati.
- ✓ La **prima fase** della **Data Tranformation** dovrebbe includere cose come la **conversione di tipi di dati** e l'appiattimento (flattening) dei dati gerarchici. Queste operazioni modellano i dati in modo da incrementarne la compatibilità con i sistemi di analisi.
- ✓ Gli analisti dei dati e i **Data Scientist** possono implementare ulteriori trasformazioni aggiungendo alla bisogna un **processo a strati individuali** come avviene per il software **Talend**.
- ✓ Ciascuno strato di elaborazione dovrebbe essere progettato per eseguire uno specifico set di task che va incontro ad un'attività conosciuta o una richiesta tecnica.

Talend

- ✓ **Talend** è un software che fornisce una piattaforma unificata di **Data Integration, Data Integrity e Data Governance.** Inoltre **Talend** offre **data delivery** in tempo reale.
- ✓ **Talend** è inoltre una compagnia privata Data Driven che fornisce soluzioni di **Data Integration** per ottenere valore istantaneo dai dati e per distribuirli in tempo e renderli di facile accesso a tutti.
- ✓ **Talend** viene eseguito nativamente su **Hadoop** usando le ultime innovazioni dell'ecosistema Apache. **Talend** combina componenti di di big data per **Hadoop MapReduce 2.0 (YARN), Hadoop, HBase, HCatalog, Sqoop, Hive, Oozie e Pig** verso ambiente **Open Source** unificato, per elaborare enormi quantità di dati rapidamente.
- ✓ I prodotti di **Data Integration di Talend** sono disponibili sulla base di sottoscrizioni con free trial. La compagnia offre anche consulenza e supporto.

Funzioni della Data Transformation

- **Estrazione e Parsing (Extraction and Parsing):** nei processi moderni di **ETL,** la **Data Ingestion**, inizia con l'estrazione di informazione da una sorgente dati, seguita da una copia dei dati nella sua destinazione. Le trasformazioni iniziali sono focalizzate sulla **modellazione del formato e della struttura dei dati** per assicurare la sua compatibilità sia con i sistemi di destinazione ma anche con i dati giù presenti li. Per esempio, parsare i campi all'interno di **dati di log** separati da virgola (CSV) per trasferirli su un database relazionale è un chiaro caso di **Data Transformation** appartenente a questa fase di estrazione e parsing.
- **Traduzione e Mapping (Translation and Mapping):** Alcune delle trasformazioni dei dati di base riguardano il mapping e translation dei dati. Per esempio una colonna contentente interi che rappresentano codici di errore può essere mappata nelle relative descrizioni di errori rilevanti, facendo si che la colonna più facile da comprendere è anche la più utile da visualizzazione.

Funzioni della Data Transformation

- ✓ **Filtraggio, aggregazione e sintesi (Filtering, aggregation e summarization):** spesso la **Data Transformation** ha a che fare con sminuzzamento dei dati in modo da renderli più maneggevoli in un secondo momento. I dati possono essere consolidati filtrandoli dai campi, colonne e record non necessari. I dati omessi potrebbero includere indici numerici nei dati intesi per i grafi e le dashboard o i record provenienti da regioni di attività che non sono di interesse per un determinato studio. I dati potrebbero anche essere aggregati e sintetizzati attraverso per esempio la trasformazione di una serie storica di clienti in conteggi giornalieri o a ore.

Funzioni della Data Transformation

- ✓ **Arricchimento e Imputazione (Enrichment e Imputatiom):** I dati provenienti da diverse sorgenti possono essere mergiati per creare un informazione arricchita e denormalizzata. Le transazioni dei clienti possono essere arrotolate in una globale e fornite all'utente sottoforma di tabella informativa. Campi lunghi o a forma libera possono essere divisi in colonne multiple e i valori mancanti possono essere imputati e i dati corrotti possono essere rimpiazzati come risultato di queste trasformazioni.
- ✓ **Indicizzazione e ordinamento (Indexing e Ordering):** I dati possono essere trasformati in modo tale che possono essere ordinati logicamente oppure possono adattarsi allo schema di immagazzinamento dati. Nei Sistemi di Gestione di Database Relazionali (RDBMSs) per esempio, la creazione degli indici può migliorare la performance o migliorarre la gestione delle relazioni tra differenti tabelle.

Funzioni della Data Transformation

- ✓ **Anonimizzazione e Crittografia (Anonymization e encryption):** I dati contenenti un'informazione personalmente identificabile, o altre informazioni che potrebbero compromettere la privacy o la security, dovrebbero essere anonimizzati prima della diffusione. La crittografia di dati privati è un requisito in molte industrie, e i sistemi possono eseguire una cifratura a livelli multipli, da celle di database individuali a interi record o campi.
- ✓ **Modellazione, conversione di tipi, formattazione e rinomina (Modeling, typecasting, formatting e renaming):** Infine, un intero set di trasformazioni può rimodellare i dati senza cambiare il contenuto. Questo include la conversione dei tipi di dati per garantire la compatibilità, aggiustando le date e orari con offset e formati di localizzazione, e schemi di rinomina, e colonne per chiarezza.

RDF - Resource Description Framework

- ✓ Il **Resource Description Framework (RDF)** è lo strumento di base proposto da W3C per la codifica, lo scambio e il riutilizzo di metadati strutturati e consente l'interoperabilità semantica tra **applicazioni** che condividono le informazioni sul **Web**.
- ✓ **Interoperabilità**: E' la capacità di due o più sistemi, reti, mezzi, applicazioni o componenti, di scambiarsi informazioni e di essere poi in grado di utilizzarle. In ambito informatico è la capacità di un sistema o di un prodotto informatico di cooperare e di scambiare informazioni o servizi con altri sistemi o prodotti in maniera più o meno completa e priva di errori, con affidabilità e con ottimizzazione delle risorse. Obiettivo dell'interoperabilità è dunque quello di facilitare l'interazione tra sistemi differenti, nonchè lo scambio e il riutilizzo delle informazioni anche fra sistemi informativi non omogenei (sia per software che per hardware).

RDF - Resource Description Framework

- ✓ Il **Resource Description Framework (RDF)** è lo strumento di base proposto da **W3C** per la codifica, lo scambio e il riutilizzo di metadati strutturati e consente l'interoperabilità semantica tra **applicazioni** che condividono le informazioni sul **Web**.
- ✓ **RDF** è costituito da due componenti:

1. **RDF Model and Syntax:** che espone la struttura del modello RDF e descrive una possibile sintassi.
2. **RDF Schema**: espone la sintassi per definire schemi e vocabolari per i metadati.

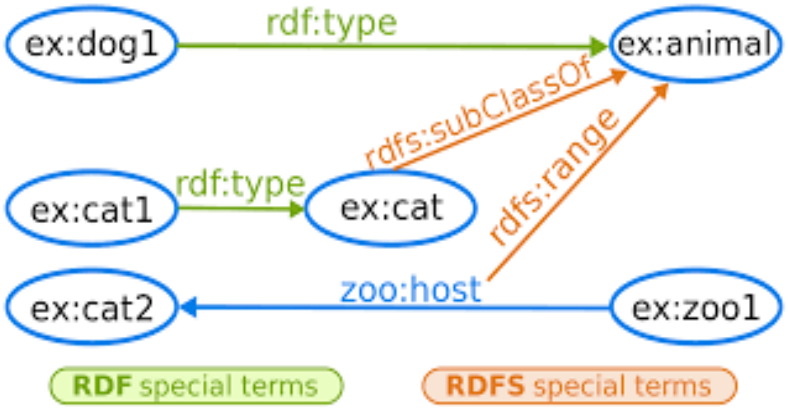
- ✓ L'**RDF Data Model** si basa su 3 principi chiave: 1) Qualunque cosa può essere identificata da un URI (Uniform Resource Identifier); 2) The least power, ovvero utilizzare il linguaggio meno espressivo possibile per definire qualunque cosa; 3) Qualunque cosa può dire qualunque cosa su qualunque cosa.

RDF - Principi e modello dei dati

- ✓ Un **URI** può essere classificato come qualcosa che definisce posizioni (**URL**) o nomi (URN) o entrambi. Un **URL** (Uniform Resource Locator) è un **URI** che identifica una risorsa tramite la sua "collocazione" ("location") in un grafo. Di fatto, non identifica la risorsa per nome, ma con il modo con cui la si può reperire.
- ✓ Qualunque cosa descritta da **RDF** è detta **risorsa**. Una risorsa è sostanzialmente reperibile su **web,** ma RDF può descrivere anche risorse che non si trovano direttamente sul **web.** Ogni risorsa è identificata da un **URI**.
- ✓ Il modello **RDF** è formato da risorse, proprietà e valori:
- ✓ **Le proprietà:** sono delle relazioni che legano tra loro risorse e valori e sono anche esse identificate da un **URI**.
- ✓ **Valore**: è un tipo di dato primitivo, che può essere una stringa contenente **l'URI** di una risorsa.

RDF - Principi e modello dei dati

- L'unità di base per rappresentare un'informazione in **RDF** è lo statement. Uno statement è una tripla del tipo **Soggetto-Predicato-Oggetto,** dove il soggetto è una risorsa, il predicato è una proprietà e l'oggetto è un valore (e quindi anche un **URI** che punta ad un'altra risorsa).
- Il data model **RDF** permette di definire un modello semplice per descrivere le relazioni tra le risorse, in termini di proprietà identificate da un nome e relativi valori.
- Tuttavia **RDF** non fornisce nessun meccanismo per dichiarare queste proprietà, nè per definire le relazioni tra queste proprietà ed altre risorse. Tale compito è definito da **RDF Schema.**



RDF - Container

- ✓ **RDF** quando deve far riferimento a più di una risorsa, per esempio per descrivere il fatto che la risorsa è associata a più proprietà, definisce dei contenitori (container), ossia liste di risorse.
- ✓ Tre sono i tipi di contenitori:
- ✓ **Bag:** è una lista non ordinata di risorse o costanti. Viene utilizzato per dichiarare che una proprietà ha valori multipli. Per esempio i componenti di un convegno.
- ✓ **Sequence:** differisce da Bag per il fatto che l'ordine delle risorse è significativo. Per esempio si vuole mantenere l'ordine alfabetico di un insieme di nomi, gli autori di un sito.
- ✓ **Alternative:** è una lista di risorse che definiscono un'alternativa per il valore singolo di una proprietà. Per esempio per fornire titoli alternativi in varie lingue.

RDF - Rappresentazione fisica del modello

- ✓ Un modello **RDF** è rappresentato da un **grafo** orientato sui cui nodi ci sono risorse o tipi primitivi e i cui archi rappresentano le proprietà.
- ✓ Un grafo **RDF** è rappresentato fisicamente mediante una serializzazione.
- ✓ In Informatica, una **serializzazione** è un processo per salvare un oggetto su un supporto di memorizzazione lineare (ad esempio, un file o un'area di memoria) o per trasmetterlo attraverso una connessione di rete. La serializzazione può essere in forma **binaria** o può utilizzare codifiche testuali (ad esempio il formato **XML**) direttamente leggibili dagli esseri umani. Lo scopo della serializzazione è quello di trasmettere l'intero stato dell'oggetto in modo che esso possa essere successivamente ricreato nello stesso identico stato dal processo inverso, chiamato **deserializzazione.** I vantaggi della **serializzazione** sono poter usare gli oggetti persistenti, fare chiamate di procedura remota (rpc) o distribuire oggetti con software come **CORBA (Common Object Request Broker Architecture).**

RDF - Esempio

- ✓ Si supponga di voler serializzare la frase "Mario_Rossi" "è_autore_di" "Rosso_di_sera_bel_tempo_si_spera": il risultato in **RDF/XML** sarà:

```
 <rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#"
   xmlns:au="http://description.org/schema/">
   <rdf:Description
about="http://www.book.it/Rosso_di_sera_bel_tempo_si
_spera/">
     <au:author>Mario_Rossi</au:author>
   </rdf:Description>
 </rdf:RDF>
```

RDF Schema

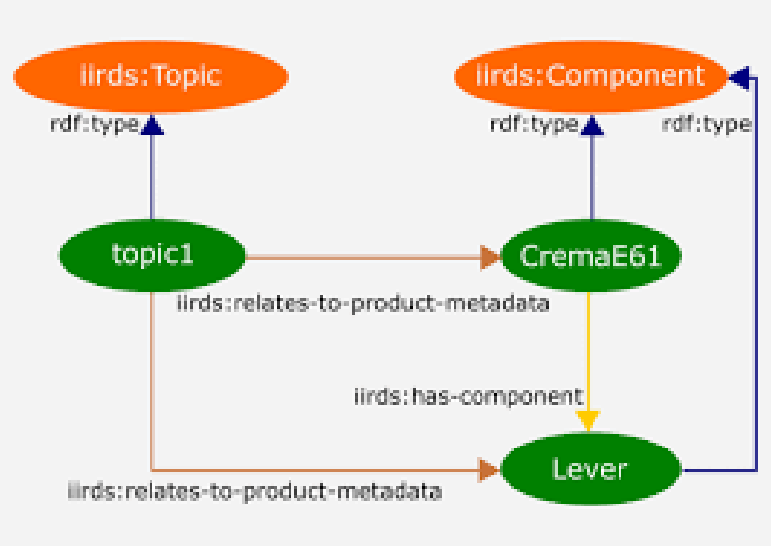
- In **RDF Schema (RDFS)** ogni predicato è in relazione con altri predicati e permette di dichiarare l'esistenza di proprietà di un concetto, che permettano di esprimere con metodo sistematico affermazioni simili su risorse simili. RDF Schema permette di definire nuovi tipi di classe, inoltre specificando il concetto di classe e sottoclasse, consente di definire gerarchiedi classi. In **RDF** si possono rappresentare le risorse come istanze di classi e definire sottoclassi e tipi.
- **Classi RDF:** Ogni risorsa descritta in **RDF** è istanza della classe *rdfs:Resource*.
- Le sottoclassi di *rdfs:Resource* sono:
- ***rdfs:Literal***: rappresenta un letterale, una stringa di testo
- ***rdfs:Property:*** rappresenta le proprietà
- ***rdf:Class:*** rappresenta una classe dei linguaggi orientati agli oggetti

RDF Schema

- **Proprietà RDF:**
- ***rdf:type***: indica che una risorsa è del tipo della classe che viene specificata.
- ***rdfs:subPropertyOf***: indica che una proprietà è una specializzazione di un'altra.
- ***rdfs:seeAlso:*** Specifica che la risorsa è danche descritta in altre parti.

RDF - Rappresentazione fisica del modello

- ✓ Le principali serializzazioni adottabili con un **grafo RDF** sono:
- ✓ **RDF/XML:** documento **RDF** è serializzato in un file **XML**
- ✓ **N-Triples:** serializzazione del grafo come un insieme di triple **soggetto - predicato - oggetto.**
- ✓ **Notation3:** serializzazione del grafo descrivendo, una per volta, una risorsa e tutte le sue proprietà.
- ✓ In particolare la serializzazione in **XML** può avvenire secondo due metodi, quello classico e quello abbreviato, più leggibile per l'uomo.



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.