**Master Thesis**

# Studio di un framework di Data Management che governa processi ETL in ambito bancario

**Autore: Marco Ronco**

**Relatori**
Relatore Didattico: Tania Cerquitelli
Relatore Aziendale: Stefano Magni

Laurea magistrale in Ingegneria Gestionale


Dipartimento di Ingegneria Gestionale e della Produzione
Politecnico di Torino
Italia, Torino
Marzo 2020

# Prefazione

Con la seguente tesi si andrà a riassumere ed analizzare le attività compiute durante il periodo di tirocinio svolto, focalizzandosi sugli aspetti fondamentali del progetto in cui sono stato inserito; l'elaborato avrà quindi un carattere narrativo differente da tesi sperimentali in quanto si descriveranno attività operative svolte concretamente sul campo.

Il periodo di tirocinio in esame è iniziato nel mese di settembre 2019 e si è protratto fino ad ora ed è stato svolto presso Deloitte Srl. Trattandosi di una società di consulenza, le quali lavorano tipicamente a progetti in capo ad un cliente, la mia figura è stata inserita all'interno di un team presente su una realtà del settore bancario con l'obiettivo di ampliare il framework esistente per la modellazione e l'archiviazione delle informazioni provenienti dalle banche acquisite dalla capogruppo verso il sistema centrale. In concomitanza alle attività compiute su tale fronte è stata implementata una dashboard per dare agli utenti la possibilità di accedere alla reportistica annessa ai dati immessi nel modello.

L'elaborato darà una panoramica generale su tutto ciò che riguarda la gestione e la fruizione di mole di dati aziendali a fini manageriali, esplicitando come le pratiche fino ad ora sviluppate stiano plasmando il modo di prendere le decisioni imprenditoriali, focalizzandosi sui pilastri di questa scienza, le architetture alla base ed i fondamenti necessaria per una sua corretta applicazione. Dopodiché si andrà a riportare come questi aspetti siano stati implementati nel corso del progetto, esaminando gli strumenti che rendono possibile la concretizzazione del risultato atteso e come essi si relazionino tra loro.

L'intero lavoro è stato supervisionato dal manager di riferimento Stefano Magni, il quale, oltre ad offrire un supporto per lo studio del modello e la stesura del documento finale, si è occupato di garantire la tutela dei dati relativi al cliente in questione, assicurandosi che l'elaborato non violasse le normative sulla privacy e non

andasse in conflitto con i vincoli imposti in fase contrattuale.

# Indice

# Capitolo 1

# Alla scoperta degli Analytics

L'evoluzione dello scenario in cui si muove la società a noi circostante ha reso necessario lo sviluppo di approcci innovativi in ambito scientifico volti a governare l'ingente aumento del volume di informazioni prodotte. Il progresso tecnologico che ha portato alla costante presenza di dispositivi elettronici in ogni aspetto delle vite che conduciamo ha conseguentemente creato ingenti volumi di dati preziosi, tanto preziosi da essere considerati, in ambito aziendale, la nuova vera ricchezza.

Davanti a questo fenomeno in piena crescita sorge spontaneo chiedersi come i grandi players del mondo imprenditoriale, cioè esattamente colore che generano l'informazione attraverso il normale utilizzo dei propri prodotti e servizi immessi sul mercato, possano sfruttare questi patrimoni informativi di valore cambiando drasticamente il modo di fare business, indipendentemente dal settore di appartenenza.

Data la consistenza delle metriche che delineano le dimensioni di questo fenomeno prende piede la necessita di fondare metodologie con lo scopo di estrarre quanto più contenuto informativo partendo da un set di dati provenienti come fattore di input. A colmare questa esigenza ecco che entrano in gioco gli **Analytics**, termine questo che va ad indicare la scienza che applica modelli matematici ai dati per rispondere a quesiti di natura imprenditoriale, scoprendo relazioni e prevedendo risultati sconosciuti al fine di automatizzare le decisioni facendole prendere dalla macchina. Questo diverso campo dell'informatica viene utilizzato per trovare modelli significativi nei dati e scoprire nuove conoscenze basate su matematica applicata, statistica, modellazione predittiva e tecniche di apprendimento automatico.

La differenza sostanziale dalla comune *Analysis* deriva dal fatto che essa è foca-

lizzata sulla comprensione del passato, mentre gli *Analytics* attingono alcune informazioni dalla precedente per comprendere le motivazioni di ciò che è accaduto al fine di proiettarle su un orizzonte futuro.

L'enorme versatilità di cui gode questa materia è indubbiamente legata al fatto che la sua applicazione non prescinde dalla natura dell'informazione trattata, quindi in base al settore in cui andremo ad implementare le sue metodologie e i suoi strumenti potremmo avere soluzioni a problemi profondamente differenti: applicandoli ad esempio in uno scenario finanziario potremmo determinare il rischio legato al credito, implementandolo nel settore farmaceutico potremmo fabbricare nuovi medicinali rilasciandoli in minor tempo, in ambito automobilistico potremmo impartire l'apprendimento alla guida di un automobile affinché essa guidi autonomamente, nel ramo del marketing potremmo attuare campagne di profilazione al fine di fidelizzare il cliente: insomma, le possibilità risultano infinite.

In passato svariati limiti tecnici imposti dalle specifiche dei dispositivi hardware non concedevano la possibilità di applicare modelli per l'analisi avanzata; velocità di archiviazione e tempestività nell'elaborazione dell'epoca non avrebbero potuto reggere i volumi di dati e calcoli desiderati. Oggi queste restrizioni sono state superate grazie ad un'evoluzione esponenziale delle caratteristiche tecniche delle macchine, che le ha rese estremamente performanti, affidabili ed in grado di sostenere carichi computazionali richiesti dagli algoritmi più complessi per la gestione di grandi quantità di dati in più passaggi.

Il mercato creatosi attorno ad ogni fattore componente il mondo degli Analytics è in costante crescita da anni con ampi margini di miglioramento. Sempre più organizzazioni consolidate iniziano ad internalizzare le competenze necessarie avviando interi reparti dedicati allo studio del dato, mentre un numero crescente di imprese esordienti si affida al supporto di competenze esterne. Anche il mercato del lavoro alla base è in un vortice di profondo cambiamento, con un aumento dei requisiti minimi all'ingresso come solide preparazioni basate su una didattica aggiornata e capacità trasversali in grado di coniugare più elementi assieme; recenti studi condotti dall'Osservatorio Big Data Analytics & Business Intelligence della School Management del Politecnico di Milano hanno dimostrato come le aziende stiano investendo in figure professionali aventi tali caratteristiche, con particolare riguardo alle realtà

che non possiedono risorse legate agli analytics che dichiarano di voler assumere un Data Scientist per una percentuale del 30% delle società intervistate, un Data Analyst per il 31%, un Data Engineer per l'17% e l'11% un Data Visualization Expert. Tali statistiche portano ad intuire il fatto che il mondo industriale abbia compreso quanto indispensabile sia un'incessante presenza su tale fronte per non rischiare di perdere il valore che il dato possiede e di come vedano gli analytics come una risorsa strategica al pari degli asset standard in ambito industriale, ritenendola fondamentale per molti ruoli e competenze funzionali.

Se considerassimo gli Analytics come guscio perimetrale dell'insieme di discipline riferite all'Information Management, al suo interno potremmo trovare:

- **Business Intelligence**, cioè un agglomerato di processi aziendali focalizzati ad ottenere informazioni utili riguardo il miglioramento e l'efficienza delle decisioni strategiche imprenditoriali.

- **Enterprise Data Management**, pilastro fondamentale del sistema gestionale di contenuti aziendali con un'impronta distintiva sul dato. Essa rappresenta un sistema di software, infrastrutture, logiche e politiche destinate alla rimozione delle problematiche organizzative provenienti da una cattiva gestione dei dati al fine di creare un canale di consegna strutturato dalla sorgente all'utente finale.

- **Performance Management**, l'insieme di modalità con le quali vengono monitorati i processi svolti da un'organizzazione verificando che i risultati che essi permettono di ottenere siano allineati con i valori target desiderati. Le principali mansioni all'interno di quest'area sono rappresentate da analisi quantitative e qualitative basate su attività interne.

Nei capitoli seguenti andremo a dettagliare quelle che sono state aree d'interesse all'interno del progetto svolto.

## 1.1 La Business Intelligence

In un panorama industriale in cui le figure presenti sul mercato assumono i connotati di Data-Driven Company prendenti decisioni basate su elementi oggettivi come i dati e non intuiti soggettivi, diventa primario definire processi tecnologici utili ad agire partendo dall'analisi del dato stesso. Per racchiudere la totalità delle metodologie usate a tale scopo viene coniato il termine **Business Intelligence**; essa permette di attuare un ciclo completo di gestione dei dati e comprendere lo status quo della situazione attuale, consentendo alle aziende che la applicano al proprio interno efficacia dal punto di vista operativo. Un'attività imprenditoriale di successo deve possedere capacità di Business Intelligence, supportando la Business Analytics e le sue analisi avanzata che forniscono maggiori vantaggi competitivi attraverso modelli di studio delle opportunità o delle tendenze proiettate nel futuro.

Ognuna delle attività di Business Intelligence ha inizio con l'estrazione del fattore di input del problema; il dato.

### 1.1.1 Il fenomeno dei Big Data

Il dato rappresenta l'unità fondamentale trattata all'interno di questo settore. Il raggruppamento di essi, inserito nel panorama degli Analytics, assume caratteristiche tali per cui prenda il nome di **Big Data**, cioè un'intera nuova classe di dati guidata da un'esplosione del volume creato da social media, automazione di macchinari, dispositivi intelligenti e dall'incessante digitalizzazione e interconnessione degli oggetti presenti attorno a noi che compongono le nostre routine quotidiane chiamata *"Internet of Things"*.

Identificando le quattro aree di maggior rilevanza andanti a formare l'universo dei big data troviamo i dati generati da social network, che comprendono tutti le estrazioni da piattaforme social, blogging e feedback dei clienti, inclusi testi, immagini, audio e video, i dati transazionali quali denunce in ambito assicurativo, chiamate registrate nel campo delle telecomunicazioni, fatturazioni, transazioni finanziarie e trading on-line, i dati individuali provenienti da call center, e-mail, dispositivi medici elettronici e altri device che trasmettono informazioni relative ai singoli individui

ed infine dati prodotti da sensori generati da macchine e rilevatori come i weblogs, i log di attrezzature e i sensori di macchinari industriali.

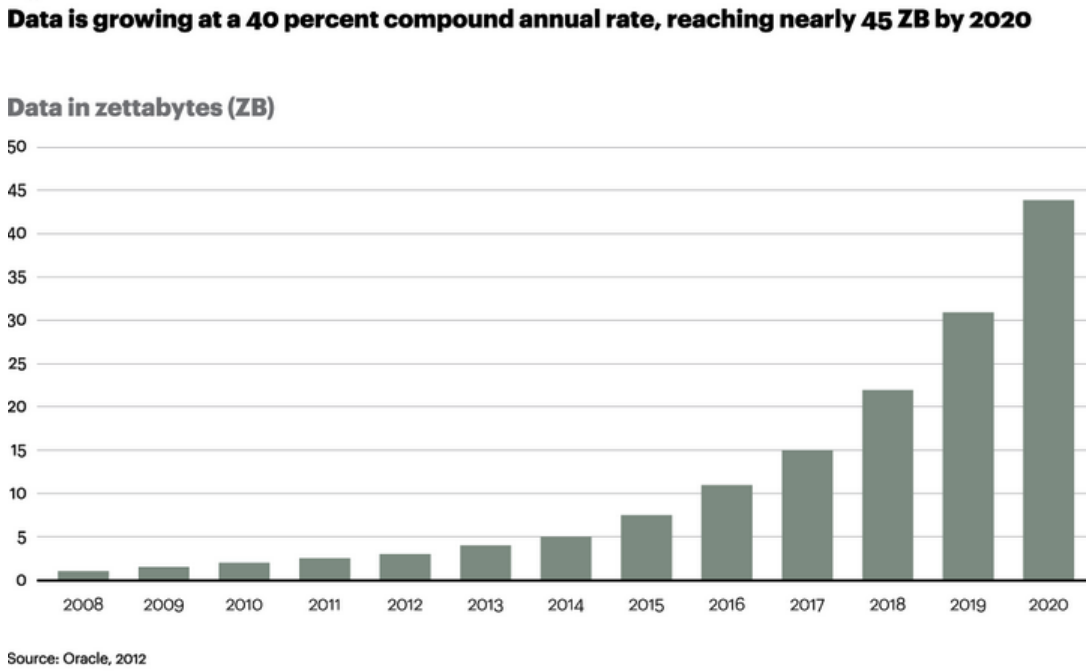



Figura 1.1: Crescita annuale del volume dati prodotto a livello globale.

I tratti distintivi che caratterizzano i big data vengono sintetizzati dalle *"3V"*, acronimo di *Velocity*, *Volume* and *Variety*. Con la velocità si identifica la frequenza di generazione dei dati da parte delle sorgenti, la quale raggiunge valori troppo elevati per essere gestita tramite metodi tradizionali basati su database relazionali; per avere una percezione numerica della rapidità con cui vengano creati i dati al giorno d'oggi basti pensare alla ricerca svolta dalla pagina *GO-Globe.com* dimostrante che in capo al famoso motore di ricerca Google pendono oltre 2 milioni di interrogazioni a livello mondiale ogni minuto oppure, sempre ogni 60 secondi, tramite i nostri smartphone inviamo più di 44 milioni di messaggi. Le tecnologie adattate per elaborazioni aventi queste tempestività hanno cambiato il paradigma di lavoro, passando da elaborazioni basate su lotti aventi in carico i dati raccolti su intervalli regolari e predefiniti, ossia modalità Batch, ad analisi Real Time, in cui il sistema è alimentato con informazioni raccolte in tempo reale e la loro elaborazione avviene istantaneamente.

La crescita esponenziale dei dati mondiali ha fatto sì che il volume rientrasse tra le dimensioni caratterizzanti questo fenomeno; le cifre valorizzanti tale cubatura hanno reso necessario l'uso di unità di misura non ancora mai utilizzate come ZettaBytes

(ZB = $10^{21}$ byte), portando in poco più di 10 anni un aumento del valore dei dati prodotti a livello mondiale da 1 ZB (nell'anno 2008) a 45 ZB (nell'anno 2020), con un crescita del 4400% circa.

Per originare grandezze di stazze tali è necessario che siano molte le fonti che producano le informazioni. Essendo il numero di fonti estremamente elevato è naturale che la loro forma risulti eterogenea; questa diversità porta al crearsi di differenti formati sotto cui viene rilasciato il dato, di cui alcuni di essi risultano avere una struttura ben definita ed organizzati secondo schemi e tabelle rigide, altri senza alcuna struttura o schema, come per esempio file contenenti elementi testuali o multimediali ed infine altri solo parzialmente, dove si incontrano alcune delle caratteristiche dei dati strutturati e alcune delle caratteristiche dei non strutturati, come per esempio nei file html.


Figura 1.2: Somma dei tratti distintivi dei Big Data che portano alla creazione del loro valore.

A volte, in aggiunta alle comuni *3V*, vengono inserite ulteriori *2V* aggiuntive per accrescere il valore che i big data generano denominando quelle che sono la *Veracity* per stabilire l'affidabilità del dato dal momento in cui le decisioni prese in base alla loro analisi potrebbe essere affetta da errori di incoerenza, ambiguità, latenza e approssimazione e la *Viability*, per comprendere quanto il dato risulti rilevante.

L'avvento dei Big Data pone sfide tecnologiche alle organizzazioni che richiedono di prendere in considerazione approcci alternativi per l'archiviazione, la gestione e l'elaborazione dei dati. Il cambiamento dei requisiti tecnici ha portato mutamenti significativi nella progettazione delle soluzioni, come ad esempio un maggior riguardo alla scalabilità e all'efficienza per mantenere un livello di prestazione adeguato anche al crescere del volume, l'indipendenza dei database da schemi fissi per accogliere dati con strutture flessibili nel tempo ed una maggiore tolleranza ai guasti garantita dal ripristino dei componenti hardware e software per mantenere alta la soglia di affidabilità. Inoltre un particolare riguardo spetta al costo di archiviazione ed elaborazione, il quale deve essere tenuto ad un livello economicamente vantaggioso ed

infine all'integrazione con set di sistemi interni ed esterni in maniera sicura.

Dal punto di vista dell'elaborazione, le soluzioni di Big Data possono essere classificate in base ai requisiti aziendali che vengono affrontati e al tipo di dati sottostanti; in contrapposizione ai sistemi di elaborazione tradizionali avremo l'elaborazione definita In-Memory, un approccio basato sulla permanenza dei dati nella memoria primaria, riducendo così i tempi di elaborazione eliminando i trasferimenti dalla memoria secondaria. Questo metodo è abilitato all'interno delle nuove architetture che hanno una memoria primaria indirizzabile più grande, di solito in TB. In alternativa vi sono le architetture di calcolo distribuito distinguibili tra le Massive Parallel Processing, ossia modelli seguenti un approccio di elaborazione parallela in cui un processo viene eseguito da più processori disposti in parallelo ognuno avente le proprie risorse allocate, e i Distributed Cluster, ovvero soluzioni costruite su Hadoop per sfruttare lo storage distribuito ed il suo framework di elaborazione parallela MapReduce.

L'implementazione delle soluzioni Big Data varia secondo il problema in base ai requisiti unici che vengono affrontati da ciascuna impresa. Tuttavia, esistono tre categorie principali di opzioni di distribuzione:

- **Software and Hardware**, cioè software distribuiti interamente su hardware nuovi od esistenti presenti sul mercato. Configurazioni di questo tipo permettono di avere un software predisposto a funzionare sulla maggior parte dell'hardware standard dopo aver effettuato una serie di test durante la distribuzione. Attraverso il loro utilizzo si acquisisce elevata scalabilità a costi ottimali.

- **Dispositive**, ovvero soluzioni hardware e software integrate, appositamente costruite, preconfigurate, testate e facilmente integrabili con le infrastrutture esistenti. Essendo un elemento rilasciato "chiavi in mano" non necessita di configurazione al momento della distribuzione.

- **Cloud**, ossia una combinazione di servizi Software (SaaS), Platform (PaaS) e Infrastruttura (IaaS) che consentono agli utenti di accedere ed utilizzare i componenti attraverso applicazioni in cloud basate su internet. Queste tecniche permettono di soddisfare la maggior parte degli standard aziendali in

materia di replicabilità, backup e sicurezza. In tale ottica scalabilità e provisioning avvengono in maniera rapida con un'amministrazione minima, anche se la personalizzazione dei servizi standard potrebbe risultare impegnativa.

Ovviamente il dato grezzo proveniente direttamente dalla sorgente non potrà essere utilizzato direttamente per le analisi desiderate in quanto potrebbe contenere errori o discrepanze rispetti al modello target; dovrà quindi essere processato attraverso tecniche di pulizia ed archiviazione che lo predispongano alle lavorazioni.

### 1.1.2 L'architettura del Data Warehouse

#### Il Data Warehouse

La comparsa dei fenomeni che hanno portato al formarsi del Big Data ha imposto un cambiamento rivoluzionario nei framework architetturali concepiti per gestire le informazioni. Le sfide tecnologiche che il progresso ha messo in atto hanno portato a concepire modelli innovativi con caratteristiche ben precise per far fronte alle peculiarità che questi dati possiedono. Analizzando i componenti strutturali ideati per costituire un sistema di gestione del dato avanzato come quello richiesto per l'implementazione delle metodologie proprie della business intelligence, il primo elemento che va citato è sicuramento il **Data Warehouse**. Con la parola Data Warehouse si va a racchiudere una raccolta di tecnologie costituenti il cuore dei meccanismi di definizione delle strategie aziendali. Esso infatti ha senso esclusivamente nel momento i cui i dati in esso immagazzinati vengono raffinati, organizzati e interpretati al fine di produrre un'analisi: per racchiudere il suo significato in una frase, si potrebbe identificare come "l'archivio informativo aziendale". Questa globalità è la sostanziale differenza che lo distingue da un RDBMS tradizionale, il quale viene installato con il semplice scopo di registrare istanze relative all'applicazione a cui è legato proprio per il suo orientamento ai programmi.

La costruzione di un sistema di data warehousing viene intrapresa orientando il focus finale verso l'utilizzatore ultimo del dato, figura spesso ricoperta da analisti non facenti parte del mondo digitale o informatico che non possiedono quindi competenze tali da interpretare linguaggi tecnici. Emerge quindi il bisogno di fornire una visione semplice e concisa su argomenti specifici, strutturando le modellazioni sul

concetto di trasformazioni subject-oriented (ossia orientate ai soggetti utilizzatori) ed escludendo così i dati che non risultano utili nel processo di analisi decisionale.

Restando sempre sulle differenze che può avere rispetto ad un database relazionale, un data warehouse contiene i dati ricavati da un insieme di sorgenti differenti sia per la tecnologia che le gestisce, sia per il modello attraverso il quale rappresentano la realtà aziendale. Le varie fonti operazionali possono quindi essere fortemente indipendenti tra loro, dunque è fondamentale che il progettista acquisisca una conoscenza quanto più profonda possibile delle sorgenti dati al fine di integrare informazioni sotto un unico nodo, compiendo così un corretto processo in riconciliazione del dato. Inoltre all'interno di un DWH noi andremo ad inserire le informazioni ad un elevato livello di profondità storica; essendo esso il fulcro centrale del patrimonio informatico utilizzato per prendere decisioni avrà una validità solamente nel momento in cui possiederà dati appartenenti ad uno specchio temporale di ampiezza tale da permettere di eseguire le analisi su cui basare previsioni future.

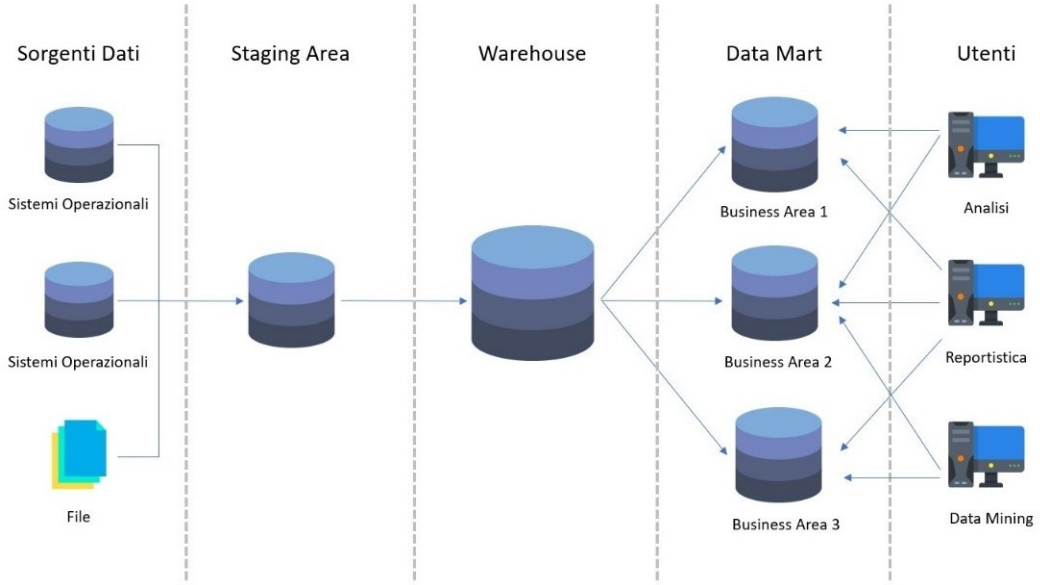



Figura 1.3: Schema concettuale di architettura Data Warehouse comprendente Staging Area e Data Mart.

Come già precedentemente citato, in un sistema di Data Warehousing è necessario pulire ed elaborare i dati operativi prima di immetterli all'interno delle tabelle contenute nel repository storicizzante i dati che spesso si presentano in maniera caotica, con una struttura eterogenea dipendente dai sistemi sorgente di provenienza. I processi aventi il compito di effettuare le operazioni di pulizia del dato vengono

generalmente collocati attorno ad una *Staging Area* (letteralmente "area di sosta"), ovvero aree di salvataggio preliminare del dato proveniente dalla fonte alimentante. Nelle staging area, in cui accedono solamente i professionisti SQL in quanto i contenuti non sono ancora semplificati per gli utenti business, i dati operazionali in arrivo vengono ribaltati e pre-elaborati prima di essere archiviati in una forma appropriata per il loro utilizzo, formando quello che è il livello di acquisizione ed integrazione del dato. Al momento del popolamento di questo primo step le tabelle presenti vengono tipicamente depurate del loro contenuto per far spazio ai nuovi flussi dati in modo che essi compongano solamente l'insieme più recente dei dati in arrivo.

Per la fase successiva, ovvero quella di archiviazione del dato storico, vengono utilizzate strutture memorizzanti i dati operativi chiamate *Operational Data Store* (ODS), soprattutto nelle casistiche richiedenti un aggiornamento della fotografia storica in tempo reale. La logica alla base di questa tecnologia si fonda sul concetto di Slow Changing Dimensione di tipo 2 (SCD2) dove i record presenti nella staging area vengono ribaltati all'interno dell' ODS in modalità differenziale rispetto a ciò che è già contenuto nelle tabelle di quest'ultima. La verifica della presenza di un record produrrà tre tipologie di risultato possibile:

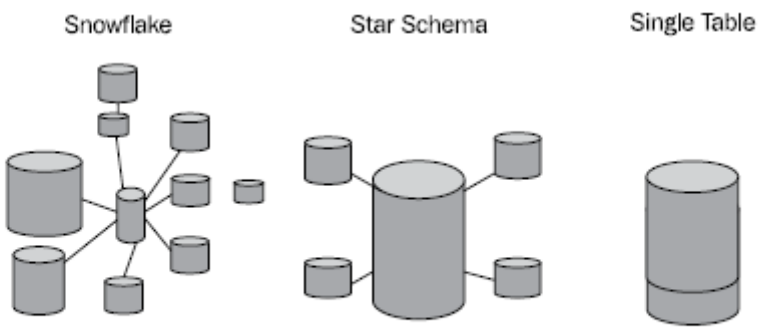
- Nella casistica in cui il record di staging sia già presente all'interno dell'ODS, esso verrà scartato per non produrre errori di chiave duplicata.

- Se invece il record risulta presente in base alle chiavi ma con valori differenti sui campi secondari verrà allora aggiornato il record presente in tabella valorizzando il suo timestamp di fine validità con il momento in cui viene eseguito il caricamento e successivamente inserito il record nuovo con una fine validità approssimabile all'arco temporale di vita del DW. Operando secondo tale schema il timestamp di fine validità del record verrà aggiunto al set componente le chiave primaria.

- Infine, se il record risulta non presente secondo la storia scritta su DB, il record verrà aggiunto con la fine validità posta alla fine del tempo di vita del DW.

Per ridurre il carico elaborativo posto in capo al livello appena descritto è spesso opportuno personalizzare l'architettura del warehouse per i diversi gruppi all'interno dell'azienda utilizzanti le sue informazioni, garantendo inoltre sicurezza del dato e fornendo direttamente l'informazione desiderata all'utente. Il raggiungimento dell'efficienza è possibile trovarlo inserendo un livello finale di presentazione del dato tramite l'uso di *Data Mart*. Spesso singole business area compiono le medesime operazioni nel tempo per portare a termine le proprie analisi, sfruttando costantemente la stessa porzione di database ogniqualvolta vi operino; sarebbe quindi inefficiente occupare l'intero data warehouse per estrapolare informazioni solamente da una minima parte di esso. È prassi comune impartire il modello secondo uno *"Star Schema"*, un tipo di diagramma entità-relazione dove le entità legate sono la tabella dei fatti contenti i risultati registrati su base continuativa e le tabelle delle dimensioni, contenti gli attributi utilizzabili per descrivere i dati presenti nella tabella dei fatti, vale a dire una raccolta di informazioni relative agli eventi memorizzati. Ogni schema a stella è costituito da una tabella dei fatti intorno alla quale sono raggruppate, a forma di stella, diverse tabelle delle dimensioni. Un modello di questo tipo consente di avere accessi efficienti settando i privilegi di logging a determinati gruppi di lavoro e di migliorare le prestazioni del data warehouse poiché si vanno a diminuire le interrogazioni fatte direttamente su quest'ultimo.

Un'estensione dello schema a stella è lo *"Snowflake Schema"*. Mentre le tabelle delle dimensioni di uno schema a stella hanno forma denormalizzata, in uno schema a fiocco di neve le informazioni sono presentate secondo la terza forma normale. Si dà luogo quindi a una classificazione e gerarchizzazione dei dati, in cui le informazioni ridondanti vengono trasferite in nuove tabelle più esterne rispetto al singolo fatto. Ne risulta che le informazioni ridondanti vengono conservate, classificate e gerarchizzate in tabelle dimensionali separate. Rispetto ai modelli a stella, i modelli a fiocco di neve sono caratterizzati da un minore consumo di spazio di archiviazione, conseguenza della memorizzazione normalizzata dei dati. La normalizzazione consiste nella rimozione dalle tabelle di colonne ridondanti, al fine di evitare la duplicazione delle voci. La riduzione delle ridondanze riduce anche lo sforzo necessario per la manutenzione dei dati: nella migliore delle ipotesi, ogni informazione è presente solo una volta e, se necessario, modificata solo una volta nell'intero sistema.




| | Snowflake | Star Schema | Single Table |
|---|---|---|---|
| Response Time | Excellent | Good | Excellent |
| RAM consumption | Good | Good | Bad |
| Script run time | Good | Excellent | Bad |
| Flexibility Model | Poor | Excellent | Excellent |
| Complexity Script | Poor | Excellent | Excellent |

Figura 1.4: Confronto tra le tre possibili predisposizioni architetturali secondo la reattività nella risposta, il consumo di RAM, tempo di elaborazione degli script, flessibilità del modello e complessità degli script.

Nelle strutture appena descritte l'origine del nome delle tabelle delle dimensioni deriva dal fatto che ogni tabella può essere rappresentata come una dimensione di un particolare figura analitica chiamata **cubo OLAP** multidimensionale. Questa figura permette agli analisti di confrontare i fatti memorizzati nel DWH in base a tutti i criteri di riferimento disponibili, al fine di analizzare le informazioni chiave dell'attività salvate da diversi punti di vista e con diversi gradi di dettaglio.

### Il Data Lake

Parlare di Big Data però non induce a parlare solamente di grandi volumi come già citato precedentemente, bensì di dati con differenti caratteristiche e formati; quest'ultimo lineamento potrebbe portare alcune problematiche nel caso in cui la nostra scelta progettuale ricadesse su di una architettura basata su Data Warehouse,

perché a tutti i vantaggi che offre questa tecnologia si contrappone lo svantaggio di richiedere, al momento del popolamento, dati che risultino necessariamente strutturati, cioè organizzati secondo schemi rigorosi, requisito tipico ed indicato per i modelli di gestione relazionale delle informazioni. In supporto al progettista viene però in soccorso un altro elemento architetturale chiamato **Data Lake**, ovvero un repository per lo storage di volumi ingenti di dati con la particolarità di non necessitare di una strutturazione ex ante del dato. Il concetto di fondo del "salvataggio del dato nel suo formato originale" è divenuto il punto di forza più grande di questo modello, concedendo ai processi di impianto del dato di accogliere dati strutturati, semi-strutturati e destrutturati. Il Data Lake archivia i dati nella loro forma grezza mantenendo così lo scheletro originale ed evitando di perdere informazioni che nel momento iniziale non si ritengono interessanti ma potrebbero rivelarsi preziose in futuro per rispondere ad esigenze non emerse agli albori della progettazione. Essendo progettato su piattaforme distribuite, come lo è la tecnologia Hadoop per esempio, possiede capacità di archiviazione potenzialmente illimitata dettate dal fatto che possono essere incrementata secondo i bisogni ad un costo relativamente basso. I dati archiviati nel Data Warehouse, al contrario, rispondono ad esigenze di business ben precise e devono sposare il modello scelto per rappresentare l'informazione, rendendolo inadatto all'archiviazione da fonti estremamente eterogenee come quelle formanti il mondo dell'IoT.

Un ulteriore differenza sostanziale risiede nel fatto che nel Data Warehouse viene definita a priori la struttura del database, i dati vengono scritti nella struttura predefinita e poi letti nel formato desiderato seguendo uno *Schema-on-Write*. Nel Data Lake invece questa barriera viene oltrepassata seguendo uno *Schema-on-Read* in cui si acquisisce l'informazione nel formato nativo assegnando ad ogni elemento un identificatore e un insieme di metadati a corredo, ma lo schema che il dato deve seguire viene definito solamente nel momento del suo utilizzo facendo così in modo di scavalcare pre-elaborazioni con conseguente guadagno sulle tempistiche.

Un Data Lake, inoltre, consente di configurare e riconfigurare facilmente modelli, query e app live e di procedere alla Data Analytics in modo più flessibile, operazione più complessa e dispendiosa in un repository altamente strutturato. L'adozione di una soluzione non esclude però l'altra; entrambe possono essere implementate ed

integrate su diversi livelli aziendali per rispondere ovviamente a diverse esigente. In entrambi i casi però, solo grazie ad una stretta adozione di politiche di governance adeguate alle architetture i modelli riescono a rispettare le assunzioni fatte inizialmente.

### 1.1.3 Estrazione, Trasformazione e Caricamento del dato

Dal momento in cui quello che conta in un data warehouse sono i dati, non si può non considerare l'importanza delle procedure volte a preparare e caricare questi dati nel sistema. Un processo **ETL** rappresenta proprio quel meccanismo che permette di estrapolare e raffinare i dati direttamente dalle sorgenti, tipicamente disomogenee, per popolare le architetture descritte in precedenza con dati puliti. Il nome costituisce l'acronimo per le tre fasi principali del processo quali "**Extract**", "**Transform**" and "**Load**", importanti a tal punto da influenzare il funzionamento complessivo del sistema. Analizziamo quindi i tre step costituenti il meccanismo:

1. L'acquisizione avvia la procedura andando ad attingere l'informazione dalla fonte, la quale può rilasciare dati sotto diversi formati quali possono spaziare dal JSON, CSV, XML, TXT oppure essere in formato tabellare e provenire da basi dati su cui appoggiano sistemi informativi o gestionali. Risulta particolarmente prioritario in questa fase strutturare un'estrazione con timing, volumi e metodi tali da non impattare sull'operatività del sistema sorgente. Le estrazioni dalle sorgenti verso la staging area possono avvenire in due modalità:

    - *Statica*, in cui l'intero pacchetto dati viene acquisito senza logica alcuna.
    - *Incrementale*, in cui viene estrapolata solamente l'area risultate avere differenze dal dato immagazzinato in precedenza, andando così a ragionare per delta.

    Cruciale risulta in questa fase studiare la modalità di **Data Ingestion** con cui vengono inviati i dati dalla sorgente, dato che la via di acquisizione sarà

influenzata in base a che essa sia *batch* in cui si raggruppa periodicamente il dato, *real-time* dove il dato viene trasmesso in tempo reale in forma continua oppure *streaming*, cioè andando a comporre raggruppamenti di dimensione ridotta in un lasso di tempo estremamente breve.

2. La trasformazione rappresenta la centralità del processo, in cui vengono manipolate le unità dati acquisite all'interno dell'architettura, precisamente tra la staging area ed il corpo storicizzante del framework. All'intero della modellazione avvengono due passi sequenziali quali la pulizia e l'elaborazione del dato; la prima mira ad eliminare le incongruenze presenti nei file di input come la rimozione di duplicati, inesattezze, outliers oppure record inaspettati o presentanti dati mancanti, il tutto avente come fine l'aumento della qualità dell'oggetto trattato. Successivamente alla pulizia, il dato attraversa i criteri di elaborazione per andare a comporre ciò che gli utenti di business si aspettano di ottenere dal processo di ETL; elaborazioni classiche possono contenete la creazione di nuovi campi calcolati, raggruppamenti o partizionamenti per diminuire la granularità dell'informazione, join su elementi provenienti da sorgenti differenti, conversioni di formato su alcuni attributi e transcodifiche di valori.

   Ovviamente le trasformazioni applicate possono impattare drasticamente sul file di input come non essere applicate affatto in quanto questa fase può essere solamente una congiunzione trasparente tra l'acquisizione ed il caricamento, riportando in scala 1:1 le tabelle presenti nei due step.

3. In fine il dato viene caricato all'interno del sistema target predisposto ad accogliere il dato processato. In base alla modalità di estrazione scelta gli oggetti plasmati dal modello andranno a rimpiazzare i dati esistenti nell'architettura finale per intero oppure andranno in "append", cioè aggiunti ai dati presenti in quanto presentanti delle differenze.

La costruzione del sistema di ETL è un'attività la cui visibilità spesso risulta secondaria agli occhi degli utilizzatori finali, rischiando di non assegnarci la giusta priorità

sottovalutandone l'importanza; solamente i pianificatori esperti assegnano le giuste risorse alle attività di sviluppo di tale processo conoscendo l'enorme beneficio che porta un suo corretto impianto.

### 1.1.4 Data Visualization

Tra le varie ramificazioni che si creano nel campo della Data Science generate dai differenti utilizzi dei dati all'interno del livello di consumo nelle architetture presentate precedentemente possiamo trovare diversi filoni principali distinti dalle esigenze richieste per le fasi di analisi svolte dagli utenti; identifichiamo tra essi:

- *Production reporting*: in genere comportano l'interrogazione di un archivio dati all'interno di un data warehouse per creare documenti o report. Richiede spesso una programmazione personalizzata poiché le informazioni necessarie e il layout del report raramente possono cambiare.

- *Query and reporting*: noti anche come report ad hoc, sfruttano un solido livello semantico per consentire agli utenti di porre domande sui dati e ottenere risposte rapidamente, senza fare affidamento sull'organizzazione IT. Questi report vengono spesso creati dagli utenti aziendali.

- *OLAP*: analisi interattive e multidimensionali con dimensioni e livelli di dettaglio diversi. Ciò fornisce buone prestazioni delle query, ma limita esse ai dati, alle dimensioni e alle statistiche di riepilogo semplici di uno o più cubi

- *Dashboard*: consente la visualizzazione intuitiva di informazioni provenienti da più fonti dati ed è altamente visiva con illuminazioni ed avvisi per evidenziare le eccezioni così da avere un certo grado di interattività.

- *Predictive Analytics*: utilizza un set di dati di controllo per creare un modello che analizzi i dati passati per prevedere il futuro. Include lo studio di dati testuali o "non strutturati" insieme ai dati "strutturati" presenti nei database o nei registri delle transazioni per aggiungere profondità alle informazioni acquisite.

- *Data Mining*: fruizione delle informazioni al fine di effettuare analisi avanzate in cui viene sfruttato il parallelismo delle risorse di calcolo a disposizione per

compiere analisi dal carico computazionale molto elevato come per esempio classificazioni dei dati, clusterizzazioni, associazioni, regressioni, serie storiche e scoperte di sequenze

- *Data Visualization*: Tecniche di visualizzazione avanzate riepiloganti i dati attraverso spazio, tempo, relazioni e temi. Riassume e presenta una grande quantità di dati in un formato visivo per fornire informazioni più approfondite e distinte.

Soffermandoci su quest'ultima branca della business intelligente riusciamo a carpire la sua importanza dovuta al fatto che essa permette di far recepire le informazioni all'uomo; il cervello umano carpisce il contenuto informativo presente negli oggetti grazie a stimoli visivi in quanto non è in grado di processare migliaia di elaborazioni in parallelo come un calcolatore. La Data Visualization riesce a schematizzare il risultato delle elaborazioni ed esporlo in maniera visuale dando una panoramica sintetica sul business in oggetto che, all'occorrenza, può essere approfondita.

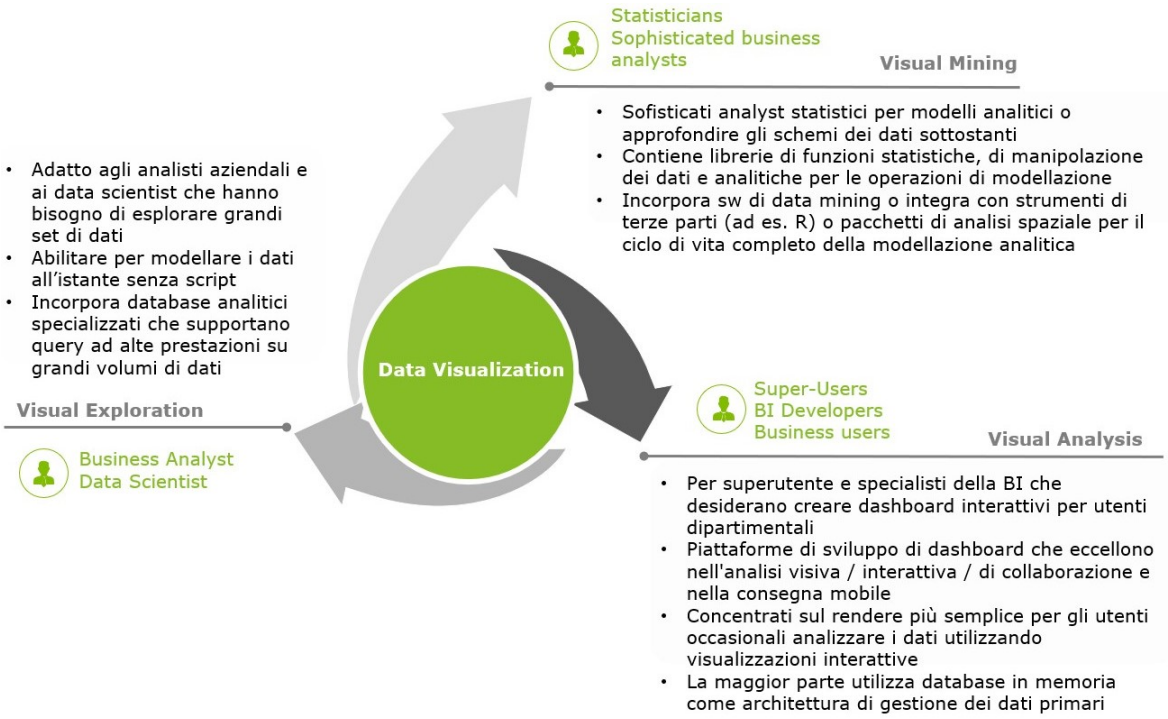



Figura 1.5: Descrizione dei tre filoni principali della Data Visualization: la Visual Exporation, la Visual Mining e la Visual Analysis.

La rappresentazione può avvenirre secondo una delle tipologie di esposizione dell'informazione che vengono utilizzate di norma per descrivere diverse famiglie di

dato; utilizzeremo una forma **relazionale** nei momento in cui cercheremo di far emergere connessioni e correlazioni tra più variabili, il **confronto** quando vorremo differenziare un insieme di variabili da un altro, l'**unione** tutte le volte che raggrupperemo diverse famiglie di informazioni al fine di esporre il valore in aggregato e la **distrubuzione** qualora dovessimo delineare le interazioni e le dipendenze tra più variabili.

Una cattiva decisione sulla rappresentazione rende estremamente difficile carpire il significato dei dati analizzati. Di conseguenza, oltre alla scelta sulla tipologia di esposizione sarà rilevante l'oggetto visuale che graficamente andrà a mostrare le analisi, che sià esso un grafico multidimensionale, una tabella o un cruscotto prestazionale.

## 1.2 Enterprise Data Management

Le organizzazioni hanno un crescente bisogno di eseguire i processi ed effettuare l'accesso ai dati aziendali in modalità real-time. Una maggiore collaborazione tra le business unit può portare l'azienda a migliorare le sue capacità di gestione dell'informazione rendendole vere e proprie estensioni delle iniziative strategiche aziendali. L'obiettivo primario dell'EDM è quello di garantire che gli utenti abbiano fiducia e confidenza con il dato ed inoltre che esso corrisponda effettivamente con quanto richiesto senza la necessità di effettuare operazioni manuali o trasformazioni multiple.

L'applicazione dei principi su cui si fonda il modello porta alla migrazione da un panorama aziendale in cui il dato non gode di alcuna affidabilità e non presiede sotto alcuno standard globale, in cui i processi di gestione dei dati sono limitati e privi di automazione verso uno scenario in cui le attività di business e i processi decisionali vengono migliorati per fornire dati precisi, affidabili, accessibili e coerenti, dove quindi viene stabilita una singola "versione della verità". Costruendo una base solida di dati sulla quale far lavorare al meglio le singole unità di business permette di raggiungere l'eccellenza aziendale, andando a ridurre quelli che sono i costi diretti relazionati alle attività di mantenimento dell'informazione e quelli indiretti imputati alle performance legate alla produttività.

Per raggiungere tali obiettivi vengono curati una serie di aspetti tematici aventi ognuna come fine ultimo il miglioramento di aree differenti della stessa materia, il dato. Tra le sezioni d'interesse curate dall'EDM troviamo:

- **Data Strategy and Architecture**: Identifica e definisce le componenti architetturali che forniscono un quadro per facilitare l'archiviazione, l'integrazione, l'utilizzo, l'accesso e la consegna dei dati in tutta l'azienda.
- **Data Conversion and Retention**: Gestisce la raccolta, la conservazione e la dismissione di dati aziendali per supportare migrazioni di applicazioni, la gestione dello storico del reporting direzionale e la conformità normativa.
- **Metadata Management**: Facilita la standardizzazione dei dati a livello aziendale per tutto il loro ciclo di vita.
- **Master Data Management**: Risolve l'armonizzazione e l'integrità dei dati aziendali, fattori essenziali per garantire una visione coerente e completa dei Master Data in tutta l'azienda.
- **Data Privacy and Security** Si concentra sulla protezione dei dati aziendali da qualsiasi violazione non autorizzata. Assicura che le politiche di sicurezza dei dati e di accesso siano adeguate e verificate con un monitoraggio continuo.
- **Data Governance**: Si concentra sulla creazione di policy, processi e tecnologie volte a garantire che i dati aziendali siano custoditi in modo accurato e coerente per soddisfare gli obiettivi di business.
- **Data Quality Management**: Stabilisce un quadro di processi e procedure di sostegno per diagnosticare in modo appropriato problemi di qualità del dato aziendali.

Queste ultime due tematiche rappresentano l'ombra sotto la quale il lavoro svolto nel progetto di tirocinio è risieduto: andiamole a scoprire.

### 1.2.1 L'importanza di un governo dati

Le architetture ed i concetti fino ad ora citati vanno a fondare le basi per un sistema di Data Management, concetto che racchiude al suo interno lo sviluppo, l'esecuzione

e la supervisione di progetti e politiche volte ad accrescere il valore delle risorse informative. Per una sua implementazione efficiente è doveroso impartire un modello di **Data Governance** ad alto livello che costituisca le fondamenta per le linee guida che dirigono le best practices svolte dagli operatori ruotanti attorno ai modelli dati. L'organo preposto a vigilare ed imporre le linee da seguire si occupa di gestire persone e creare metodologie al fine di realizzare un costante e corretto trattamento di tutti i dati che abbiano importanza per un'organizzazione.

Nella pratica la Data Governance punta a standardizzare la definizione dei dati a partire delle diverse funzioni aziendali, a stabilire regole di accesso e d'uso comuni e a identificare i soggetti coinvolti definendone le responsabilità. Per tanto rappresenta un insieme di regole da definire a monte dell'utilizzo dei dati, allo scopo appunto di esercitare un continuo controllo sui processi e sui metodi usati dagli amministratori per prevenirne gli errori e per suggerire gli interventi necessari a risolvere i problemi creati da dati di scarsa qualità.

Il suo primo vantaggio è sicuramente quello di permettere alle risorse (decisori o comunque dipendenti con mansioni operative che costituiscono la parte più ampia della forza-lavoro) di accedere e condividere informazioni fornite da applicazioni e database sulle quali ci si può contare perchè di qualità.

Ma un efficace governo dei dati ha anche notevoli effetti sulla sicurezza, con riduzione dei rischi derivanti dalle operazioni non permesse e dall'eventuale mancata osservanza di leggi e normative; tutto ciò ha effetto anche sul Business Process Management. Infatti, dovendosi realizzare processi di gestione dati (sui quali si basano tutti i processi di business) standardizzati e ripetibili, si stabiliscono linee guida e parametri atti a valutare sviluppo, gestione e prestazioni dei processi stessi.

Gli strumenti di Data Governance a supporto delle attività in azienda devono consentire di concentrare in un unico organo l'intero patrimonio informativo relativo ai metadati esistenti provenienti da applicativi differenti. Per convogliare la totalità delle conoscenze verso un unico nodo centrale il governo dati si avvale di un modello strumentale che permette il raggruppamento su appositi documenti di tutte le informazioni esistenti sui dati. Tra gli strumenti utilizzati sul campo possiamo individuare:

- **Naming Convention**: All'interno di quest'area si prevedono quali siano gli

standard da applicare nell'assegnazione dei nomi alle varie tipologie di oggetti.

- **Monitoring Scheduling**: È importante definire le logiche di gestione delle pianificazioni dei flussi di caricamento e della generazione dei metadati relativi proprio alle singole esecuzioni.

- **Modello Dati**: Al fine di avere un corretto controllo degli sviluppi è fondamentale stabilire le modalità e gli strumenti che dovranno essere adottati per la documentazione dei modelli dati. Possiamo dettagliare questi modelli in tre sottocategorie:

  - *Modello Concettuale*: questo modello descrive la realtà da modellare in termini di entità, attributi e relazioni tra entità, la cui rappresentazione concettuale consiste in un insieme di schemi di fatto. Gli elementi base modellati sono i *fatti*, che descrivono un'associazione molti a molti tra le dimensioni e soddisfano il requisito della dinamicità, le misure ovvero proprietà numeriche dei fatti che descrivono un aspetto quantitativo e le dimensioni ossia proprietà con dominio finito dei fatti che descrivono le coordinate di analisi

  - *Modello Logico*: questo modello trasforma le entità in tabelle, attributi in colonne e le relazioni in chiavi primarie e chiavi esterne. In altre parole le relazioni vengono de-normalizzate traducendo gli schemi concettuali, scegliendo quali viste verranno materializzate ed infine applicate forme di ottimizzazione.

  - *Modello Fisico*: questo modello eredita tabelle e colonne dal modello logico e permette di definirne l'implementazione fisica, sulla base del database scelto.

- **Modello Documentale**: I processi in ambito DWH/BI & Analytics prevedono una serie di documenti standard con formato e contenuti predefiniti quali:

- *Manuale Operativo*: Il manuale operativo è una guida a disposizione dell'utente dedicata agli strumenti di controllo di gestione e monitoraggio del software
- *User Acceptance Test*: UAT è una fase di test condotta per determinare se sono soddisfatti i requisiti specifici richiesti. Si compone di una serie di attività predefinite sviluppate per guidare l'esecuzione dei test per raggiungere gli obiettivi di prova, compresi corretta attuazione, identificazione degli errori e verifica della qualità.
- *Analisi Funzionale*: L'analisi Funzionale è la documentazione riguardante il processo di sviluppo del software, durante la quale vengono identificati e descritti i processi che compongono il sistema informativo.
- *Analisi Tecnica*: L'analisi tecnica corrisponde al progetto del software schematizzato. È redatta da un'analista informatico ed è una sorta di linea guida per il programmatore. L'analisi tecnica si focalizza su quattro elementi: Struttura dei dati, Architettura del software, Interfacce (grafiche o non) ed Algoritmi di dettaglio

- **Business Glossary**: L'area di Business Intelligence è il punto di incontro tra le esigenze di business e il mondo IT, da cui è opportuno dotarsi di un vocabolario comune, riconosciuto all'interno dell'azienda.

### 1.2.2 Data Quality per un'informazione affidabile

Quando trattiamo i dati, il rischio che si corre è quello di non avere una corrispondenza con la realtà d'azienda; questo ci dice che la qualità del dato non dipende solamente dalle caratteristiche del dato stesso ma anche dal contesto di business in cui viene utilizzato. Un incremento della fiducia degli utenti verso le informazioni esposte dai sistemi informativi è uno dei primi risultati che si ottengono attraverso un'estensione ed un miglioramento dei controlli di qualità apportati sul dato; stabilire il quadro di processi e procedure di sostegno per diagnosticare in modo appropriato problemi sul dato aziendale è compito della **Data Quality**, inteso come

insieme di metodi per certificare l'adeguatezza del dato al processo e non solo l'esattezza del dato in sé, ma quindi anche l'aggiornamento, l'adeguata rappresentazione e l'attenzione alla multidimensionalità.

Creare un sistema di monitoraggio e gestione della qualità richiede un approccio strutturato, progressivo e iterativo in grado di sanare quelle che sono le cause principali di una bassa qualità del dato quali cause intrinseche (molte tipologie di dati diventano obsolete), cause tecniche (vecchi sistemi legacy, vale a dire componenti obsoleti ma ancora in uso, dotati di controlli insufficienti sull'immissione dei dati, in unione con le fonti esterne, creano disallineamenti e incongruenze nei dati) e cause organizzative (inconsapevolezza sull'attuale qualità dei dati o mancanza di impegno per migliorarla) per riuscire a veicolare l'organizzazione verso il conseguimento di livelli crescenti di maturità in questo abito.

Operativamente il processo viene avviato dallo studio del perimetro dati da bonificare qualitativamente, definendo gli obiettivi da perseguire e le regole da applicare per raggiungerli. Superata questa fase iniziale si procede alle analisi del livello qualitativo del dato in esame, andando a scovare quelle che sono le fonti causanti li disallineamento rispetto gli obiettivi. In altri termini, l'accertamento della qualità dei dati viene predisposto applicando i metodi di misurazione definiti in fase funzionale e comparando il risultato alla soluzione target desiderata al fine di identificare le possibili problematiche ed una volta rilevate intraprendere le azioni richieste.

A valle di tutto ciò risulta indispensabile la stretta collaborazione tra le figure IT e quelle di business affinché venga svolta un efficiente fase preliminare di studio dello stato dell'arte e una successiva di definizione delle misure correttive più opportune da implementare.

Data l'importanza della materia per i motivi strategici già citati sarà essenziale una profonda conoscenza del contesto operativo per determinare le metriche di misura della bontà dei dati. A tal fine dovranno essere precisati degli appropriati **Key Quality Indicator** (KQI), cioè indicatori di sintesi della qualità del dato.

Un ulteriore complicazione consiste nel definire metriche universali. È comunque possibile tentare di definire delle metriche indipendenti dal contesto di utilizzo su cui basare i KQI sfruttando l'universalità di alcune dimensioni che risultano sempre necessarie sul dato quali:

| **Metriche** | **Descrizione** |
|---|---|
| **Accessibilità** | Facilità con cui un utente può ottenere l'informazione |
| **Comprensibilità** | Facilità di comprensione del dato |
| **Puntualità** | "Freschezza" del dato in termini di aggiornamento |
| **Oggettività** | Imparzialità del dato |
| **Accuratezza** | Differenza tra il valore reale del dato e la sua previsione |
| **Interpretabilità** | Presenza di documenti a supporto della comprensione sul dato |
| **Correttezza** | Esattezza del dato |
| **Utilità** | Beneficio portato dall'utilizzo |
| **Quantità** | Appropriatezza del volume in relazione alle necessità |
| **Manipolabilità** | Facilità con cui il valore dell'informazione può essere variati |
| **Integrità** | Garanzia di assenza di perdita di informazioni |
| **Consistenza** | Costanza della struttura tra i dati di uno stesso set |
| **Completezza** | Presenza della quantità adatta a spiegare la realtà rappresentata |
| **Conformità** | Rispetto degli standard formali appositamente definiti |
| **Coerenza** | Mutua non contradditorietà dei dati tra loro |
| **Unicità** | Non ridondanza dell'informazione |

Tabella 1.1: Elenco di caratteristiche essenziali per un dato di qualità.

Tutto questo insieme di caratteristiche richieste al dato rappresentano una condizione necessaria ma non sufficiente per consolidare il dato come qualitativamente accettabile. Da qui in poi spetterà ai singoli reparti definire le proprie metriche di

monitoraggio e controllo ad hoc che porteranno a classificare il dato come “buono” all’interno del contesto in cui verrà inserito.

La responsabilità di dirigere le fasi in questione è solitamente affidata al Data Quality Office predisposto dall’organizzazione.

# Capitolo 2

# Ambito progettuale

## 2.1 Panoramica del settore bancario

L'industria bancaria e dei servizi finanziari è un settore in cui il volume di dati generati non è paragonabile a nessun'alto. Ogni singola attività di questa sfera settoriale genera un'impronta digitale segnata dalla nascita di nuovi dati. Con l'aumentare del numero di record elettronici, i servizi finanziari utilizzano attivamente l'analisi dei dati per ricavare informazioni di business, archiviare dati e migliorare la scalabilità.

La tecnologia ha spinto le banche a compiere investimenti in maniera massiva per sfruttare i dati con cui prendere le migliori decisioni possibili. Ciò ha indotto svariate organizzazioni *BFSI* (Banking, Financial, Services and Insurance) a distruggere i loro precedenti metodi di analisi.

Dopo la grande recessione del 2008 che ha colpito drasticamente le banche globali, l'analisi dei big data ha goduto di una decennale popolarità nel settore finanziario. Quando le banche iniziarono a digitalizzare i propri processi operativi, dovettero assicurare diversi mezzi fattibili per analizzare tecnologie, come per esempio sistemi di gestione dati per i loro guadagni di business. Tali guadagni sono stati resi possibili grazie anche all'utilizzo di pratiche di analisi dati esistenti che hanno semplificato il monitoraggio e la valutazione della grande quantità di informazioni dei clienti. L'incremento della fiducia nella tecnologia per gestire i crescenti volumi relativi a clientela e transazioni ha portato ad uno stato migliorativo a livello globale dei servizi offerti dalle organizzazioni.

Operando attraverso il dato, le banche possono ora utilizzare le informazioni transazionali di un cliente per monitorare continuamente il proprio comportamento in tempo reale, fornendo il tipo esatto di risorsa necessaria in ogni momento. Que-

sta tempestività real time aumenta le prestazioni complessive e la redditività del mercato, spingendola a proseguire verso un ciclo di crescita incessante.

Il settore bancario è un'area che genera dati in ogni aspetto e, secondo quanto detto da Nazareno Lecis sulla rivista "Financecue", gli esperti del settore ritengono che la quantità di dati generati ogni secondo crescerà del 700% entro il 2021. I dati finanziari e bancari saranno uno dei capisaldi di questa alluvione dei Big Data e la possibilità di elaborarli permetterà di ottenere, per chi li analizzerà, un vantaggio competitivo sul resto delle istituzioni finanziarie.

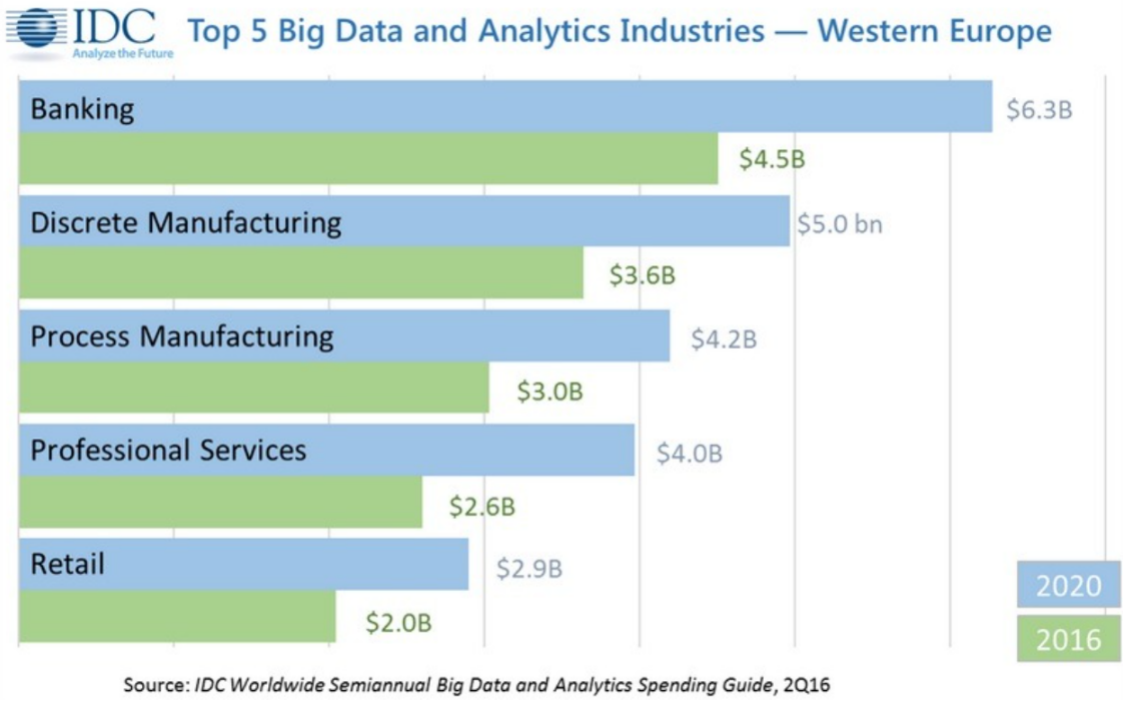



Figura 2.1: Panoramica dei settori in cui gli Analytics sono maggiormente presenti; in cima ad essi troviamo il settore bancario che, oltre ad essere il primo investitore riguardo lo studio dei Big Data, è colui che ha aumentanto maggiormente gli investimenti durante il corso degli anni.

I vantaggi che porta tale evoluzione nel settore sono molteplici, tra cui il miglioramento dell'esperienza del cliente attraverso la segmentazione dei loro profili, che consente di personalizzare le campagne marketing secondo i suoi tratti distintivi disponendo un servizio più curato e dedicato. Sempre attraverso i clienti e al loro feedback le banche migliorano l'erogazione dei propri servizi secondo quello che il cliente finale desidera maggiormente.

Studi alternativi basati sugli analytics interessano il mondo della finanza grazie alla costruzione di modelli di rischio ed investimento bancario o modelli di perfor-

mance e monitoraggio operativo, argomento primario tra quelli di ogni istituto di credito. Grazie alla potenza raggiunta dalla data science è possibile migliorare questi modelli e, di conseguenza, prendere decisioni oculate perché guidate dai dati, riducendo sempre più l'errore provocato dall'incertezza umana.

## 2.2 Stato dell'arte

Per sopperire la necessità della banca in esame di attingere alle informazioni da lei possedute sugli istituti acquisiti nel corso degli anni, ecco che in passato tali banche iniziarono ad inviare i primi flussi in modalità manuale o semi-automatica, allocando ai propri reparti tecnici dedicati al componimento di pacchetti informativi un ingente carico di lavoro, assoggettandosi inevitabilmente ad una serie di problematiche causate dagli errori umani. I flussi in questione venivano poi spediti direttamente agli applicativi, utilizzatori di tali informazioni, senza subire modellazione alcuna, scavalcando così, completamente o solamente in maniera parziale, ogni sorta di controllo che di norma viene eseguito prima di effettuare le esecuzioni. Tali flussi inviati dalle controllate direttamente alle filiere prendono il nome di **Flussi AS–IS** e costituiscono tutt'ora l'elemento somministrato da parte delle banche che non sono ancora dirette dagli standard imposti dal modello creato.

## 2.3 Finalità progettuale

Il progetto in questione oggetto della tesi nasce dall'esigenza sopra descritta di creare un network interconnesso con gli istituti inglobati dalla banca in esame nel corso della propria espansione a seguito di fusioni ed acquisizioni strategiche volte a rafforzare la propria posizione all'interno del mercato; grazie ad un canale diretto verso gli utenti finali passante dalla capogruppo, quest'ultima potrà avere il pieno possesso dei dati generati dalle acquisite in modo da poterli utilizzare per i propri fini strategici, commerciali e di controllo.

All'interno del programma di creazione dei canali verso la banca, il progetto nasce con l'obiettivo di realizzare un sistema di Data Governance per tutti gli istituti controllati che preveda l'identificazione e la creazione di un Data Office, un Data Owner, un Data Technology, una lista di Data User ed un Data Quality framework

per i seguenti scopi:

- Armonizzare la gestione informatica della Data Governance convergendo verso la soluzione di un modello unico di amministrazione dati per ogni gruppo bancario acquisito.
- Automatizzare tutti i flussi dati ad oggi inviati secondo modalità manuali o semi-automatiche.
- Ampliare e uniformare il patrimonio informativo tramite l'utilizzo di flussi informativi omogeneizzati e razionalizzati provenienti dalle sussidiarie.
- Creare un unico punto di accesso ai dati entranti nel modello.

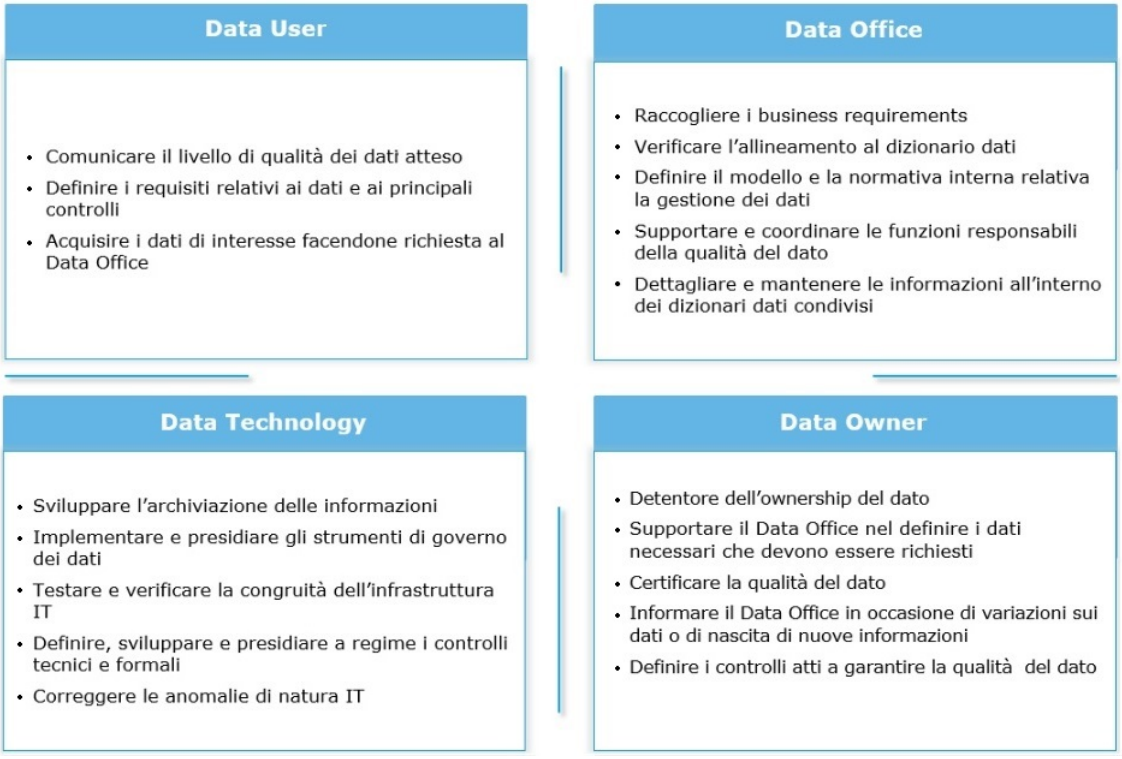



Figura 2.2: Figure chiave in materia di Data Governance con i relativi ruoli.

Il supporto offerto dal nostro team, basato su un approccio End-to-End (il quale punta a garantire una consulenza bilaterale sui fronti interessati quali lo sviluppo tecnologico con il sostegno costante all'interno del reparto tecnico e lato business user con supporto del reparto Data Office), si focalizza sull'obiettivo di attuare le attività di design, build e implementation delle nuove componenti integrandole

nell'attuale infrastruttura dopo aver eseguito una gap analysis tra l'attuale sistema di gestione e il sistema target definito. Un ulteriore finalità risiede nel supportare i business office che usufruirà dei dati elaborati nel mappare il perimetro e il data model dell'ambiente considerato ed assistere la banca nella definizione delle linee guida per la gestione della Data Governance.

Gli interventi atti al raggiungimento di tali obietti sono di due tipi:

1. Organizzativo, mirato cioè all'adozione formale delle linee guida in materia di Data Governance per l'implementazione del nuovo framework di Data Quality. Tali linee guida sono un documento redatto dall'ufficio Governo Dati che riassume tutti gli standard da adottare per lo sviluppo del modello.

2. Tecnologici: indirizzati alla gestione accentrata verso la capogruppo dei dati delle banche controllate al fine di favorire il pieno presidio delle informazioni e la loro diffusione verso tutti gli utenti e a garantire l'adozione di un approccio progressivo all'integrazione tecnologica, prevedendo cioè una fase tattica che predisponga l'integrazione dei flussi AS-IS delle sussidiarie all'interno dell'architettura creata e di una fase target che prevede un graduale passaggio delle alimentazioni dalle banche verso i nuovi flussi target generati dal modello, definiti Flussi TO-BE, che risulteranno razionalizzati e omogeneizzati.

## 2.4 Modello Archietturale

L'architettura a layer dati definita per creare il framework progettato inizialmente è composta da un *Lake*, uno *Strato Modellato* ed un *Modello Logico Applicativo*;

Il **Lake** rappresenta la staging area del framework in cui vengono inseriti i dati inviati dalle banche senza modellizzazione alcuna. Questi dati rappresentano i flussi informativi AS-IS che dovranno essere in futuro rimpiazzati dai flussi generati dal modello.

Il **Modello Logico Applicativo** è un layer dati in cui vengono immagazzinati i dati che successivamente verranno spediti alle filiere, che rappresentano gli applicativi che usufruiranno del risultato derivante dal processamento da parte del

modello: queste filiere possono essere immaginate come l'utente finale. Questa sezione è composta da viste che implementano la modellazione pro-filiera e tabelle che fisicizzano le viste dello strato modellato cosi da storicizzare le esecuzioni. Gli utenti accedono unicamente alle tabelle fisicizzate.

Lo **Strato Modellato** è costituito da viste che implementano la modellazione definita dall'utente. Nel progetto il layer modellato è creato in rapporto uno-a-uno con il lake. Questo è il layer tipicamente acceduto dall'utente di business per attuare le analisi finalizzate al miglioramento dei dati per predisporli alle linee guida. L'interazione avviene tramite un software integrato che genera, conseguentemente alla definizione dei requisiti voluti dall'utente per l'interrogazione, le query alle viste, permettendo cosi anche a figure non aventi conoscenze sul linguaggio SQL necessari per relazionarsi con databases di attingere informazioni in modalità più visiva e reportistica rispetto a modi più di basso livello.

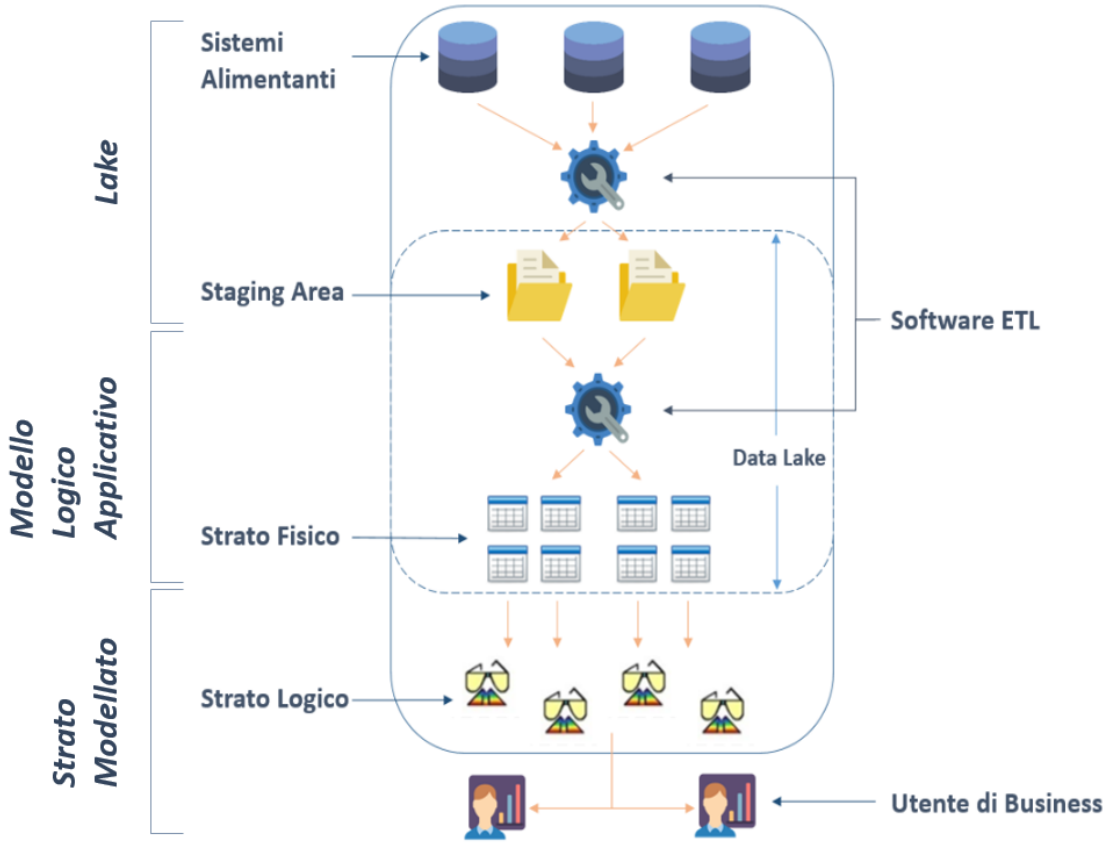



Figura 2.3: Grafico schematizzante il modello architetturale sviluppato.

Essendo un progetto avviato ormai da anni, durante l'arco temporale trascorso fino

ad ora parte del patrimonio informativo in perimetro è stato acquisito tramite il caricamento nell'architettura di flussi già in essere e inviati dalle banche alla capogruppo per l'alimentazione di applicativi. La parte rimanente è costituita da flussi creati ex-novo. I dati di tutte le aree tematiche sono stati storicizzati in un layer logico all'interno del data lake i cui requisiti sono stati definiti dall'ufficio di Data Office. I flussi AS-IS, non essendo pronti per rispettare le specifiche richieste dal governo dati hanno richiesto uno step di pre-processamento, per allinearne il formato alle linee guida architetturali. Inoltre, per tali flussi è stata predisposta una staging area persistente, con storicizzazione del singolo flusso, al fine di facilitare le trasformazioni verso tale layer e consentire agevolmente eventuali operazioni di recupero. D'altro canto, i flussi nuovi, essendo già in linea con il modello, non richiedono alcun tipo di elaborazione e transitano attraverso una staging area volatile. Successivamente, i flussi AS-IS sono stati progressivamente sostituiti da flussi creati ad-hoc, aderenti alle specifiche definite dal Data Office.

# Capitolo 3

# Applicativi utilizzati

In un panorama frastagliato e variegato come quello creatosi attorno al mercato dei prodotti e servizi informatici rivolti agli analytics risulta fondamentale analizzare la competitività ed i vantaggi che ognuno di essi può portare ad un'organizzazione.

La miglior via per scegliere quale tipo di strumenti necessita la nostra soluzione è chiedersi quali saranno le attività che essi dovranno andare a compiere e le principali criticità, dopodichè una volta ottenuta un'immagine verosimile della realtà operativa sarà possibile iniziare ad orientarsi nel panorama dei vendor presenti.

Andando con ordine il primo strumento da coinvolgere nella ricerca è una base dati solida che governa l'intero patrimonio informativo del sistema.

Prima di tutto una buona base dati utilizzata per rispondere ad esigenze di business deve essere soggetta ad una costante lavoro di ottimizzazione per rispondere positivamente a seri criteri di performance, soddisfando le interrogazioni degli utenti senza generare latenze nonstante il numero di dati che acquisisce si espanda senza preavviso. Deve poi disporre di misure di sicurezza facilmente implementabili ed attivabili, capaci di segnalare malfunzionamenti o potenziali rischi; non tenere conto di questo aspetto risulterebbe deleterio, sarebbe opportuno migrare la scelta verso prodotti che implementano soluzioni anche native.

Continuando con le caratteristiche imprescindibili che deve possedere un enterprise database troviamo la disponibilità costante di erogare il servizio che deve offrire. Le alte performance citate poco fa vanno mantenute nonostante ci si trovi in situazioni di crash del singolo DB server grazie a configurazioni a cluster offrenti più livelli di backup; operando in maniera distribuita si possono trasferire le elaborazioni su un nodo secondario se in quello di origine si genera un malfunzionamento,

garanendo in questo modo la business continuity.

I dati acquisiti devono essere sottoposti a processi di archiviazione che assicurino che nulla vada perso, che sia esso salvato in architetture relazionali, non relazionali o distribuite. Tale integrità del dato viene garantita dai cosiddetti sistemi **ACID** (Atomicity, Consistency, Isolation, Durability) predisponenti l'operatività delle transazioni in modo che seguano:

- **Atomicità**: la transazione è indivisibile nella sua esecuzione. Si avranno quindi solamente transazioni o totali o nulle, non è possibile avere esecuzioni parziali;
- **Consistenza**: quando inizia una transazione il database si trova in uno stato definito "coerente", esattamento come quando porta a termine tale operazione; è quindi possibile violare eventuali vincoli di integrità cosicchè non si creino incoerenze tra i dati archiviati nel database;
- **Isolamento**: ogni transazione deve essere eseguita in modo isolato e indipendente dalle altre. In caso di fallimento di una transazione essa non dovrà in alcun modo interferire con le altre;
- **Durabilità**: quando la transazione richiede una modifica al database, tale cambiamento non dovrà essere dimenticato o perso. Quindi per evitare perdite di dati i database moderni hanno a dispostone dei registi di log di tutte le operazioni.

Ultimo ma non per importanza, un database aziendale deve prestare particolare attenzione all'utilizzo delle risorse su cui viene installato. Per merito del deploy degli strumenti software su macchine virtuali cade la concezione per cui un database risulta performante solamente se allocato sui migliori hardware, ma tutto ciò comporta necessariamente la spremitura complessiva da parte della base dati delle risorse su cui viene impiantato per ogni istante in cui rimane in esecuzione.

Passando invece ad intraprende il confronto tra gli strumenti ETL presenti sul mercato, anche essi presentano una serie di caratteristiche da avere necessariamente per rispondere alle esigenze operative. Innanzitutto lo strumento dev'essere in grado di leggere e scrivere sull'intera gamma di sorgenti dato possedute in modo che le

attività eseguite possano svolgersi automaticamente, riuscendo ad operare in qualsiasi ambiente, da un'infrastruttura locale, in cloud oppure ibrida. Per una scelta ottimale esso deve presentare funzionalità di governo e qualità del dato integrate all'interno dando però la possibilità di inserirne di personalizzate dal cliente e deve consentire un agevole passaggio da un infrastruttura all'altra; nella fase iniziale di ideazione della soluzione potrebbe risultare conveniente, per esempio, un data lake fornito da Amazon Web Service ed il semestre successivo una configurazione differente erogata da Microsoft. È importante quindi disporre di uno strumento ETL in grado di passare da un provider di servizi ad un altro mediante il semplice scambio di alcuni parametri al suo interno, senza alterare la logica di trasformazione e del business.

Spesso la facilità di utilizzo ed il costo ridotto portano ad indirizzare la scelta su strumenti semplici ma carenti in termini di scalabilità; lavorare con programmi di questo tipo comporta restrizioni in quanto questi ultimi dipendono fortemente dalla macchina su cui vengono installati, compromettendo fortemente la portata dei processi analitici che non potranno espandersi a causa di limiti fisici il che, operando in un scenario variabile come quello dei Big Data, renderà l'ambiente sensibile ai frequenti mutamenti. Tra le caratteristiche essenziali da tenere in considerazione troviamo inoltre la portabilità spesso sottovalutata ma necessaria in un'ottica di migrazione del codice da una tecnologia deprecata ad una nuova.

Di seguito verranno elencati e descritti i software facenti parte del set di applicativi che permettono la copertura degli aspetti tecnici appena citati che hanno determinato gli sviluppi all'interno del progetto.

## 3.1 IBM DataStage

**Datastage** è un potente strumento ETL prodotto da IBM, spesso utilizzato nei progetti di Data Warehousing, che rappresenta per noi il cuore dell'intero processo attuato. Esso permette di creare soluzioni per l'integrazione dati tramite l'utilizzo di elementi grafici rappresentati da operatori a blocchi, connettibili fra loro per formare i *Job Parallel*, denominati semplicemente Job. Una moltitudine di job interconnessi a loro volta rappresenta l'unità fondamentali di una *Job Sequence*, che eseguono i job-parallel in sequenza.

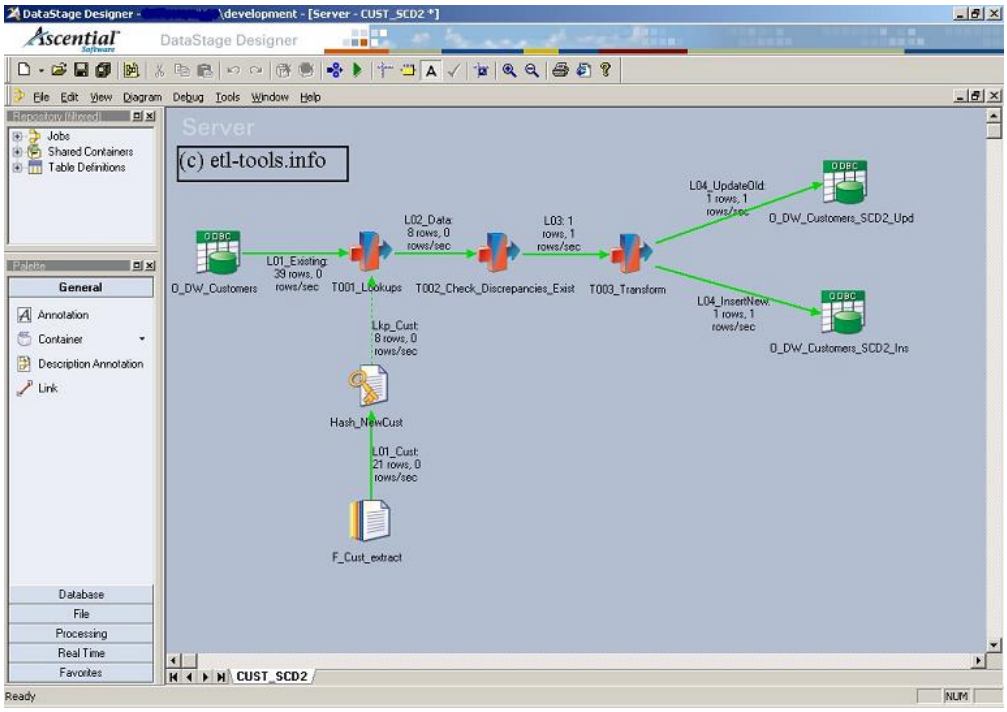



Figura 3.1: IBM Datastage: Versione Designer

Lo strumento viene distinto in due versioni, una versione *Designer* in cui è possibile comporre i job, impostare le logiche al loro interno e parametrizzare i dati di input secondo le necessità progettuali, ed una versione *Director*, in cui gli stessi job creati nel designer saranno di sola lettura e potranno essere monitorati nel corso della loro operatività, consultando i loro log nonché avviarli singolarmente in modalità manuale tramite l'inserimento dei parametri necessari.

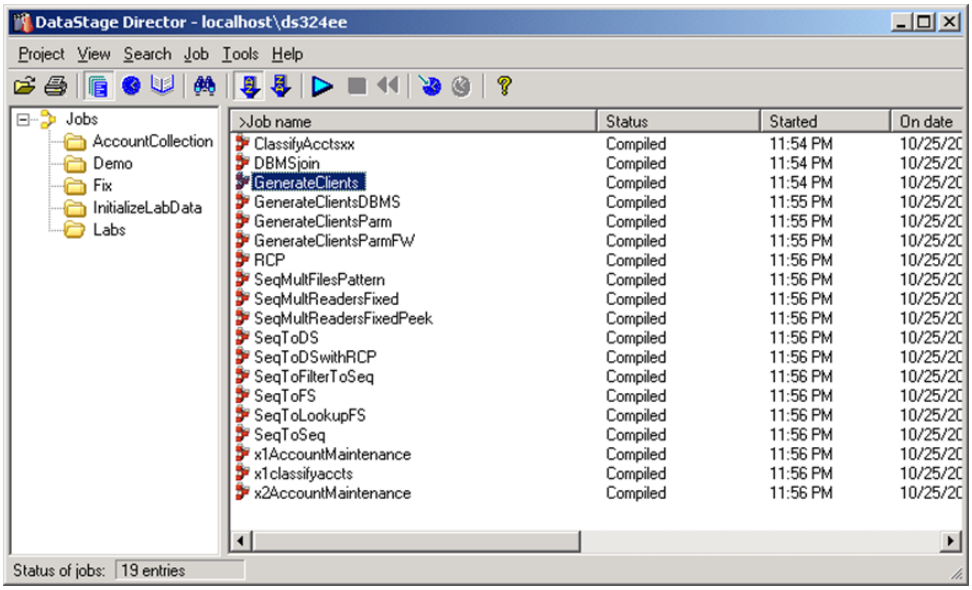



Figura 3.2: IBM Datastage: Versione Director

La piattaforma è basata su un'architettura client-server, lasciando la possibilità di impiantare il software su server Windows oppure Unix.

La popolarità di questo programma è dovuta alla sua elevata scalabilità e le sue molteplici possibili integrazioni con strumenti e basi dati di vario tipo.

## 3.2 Teradata

Per quanto riguarda lo storage dati si è scelto **Teradata**, una delle soluzioni preferite per le società operanti nella gestione dei Big Data tanto da ricevere il riconoscimento di miglior vendor per soddisfazione dei clienti e per la propria offerta tecnologica secondo l'annauale "Big data warehouse landscape report" stilato da The Information Difference. La scelta è ricaduta su tale DB data la sua perfetta integrazione in progetti di Data Analytics ed il suo potente motore alla base che permette di eseguire query molto complesse (possibilità di effettuare 256 join in una singola query).

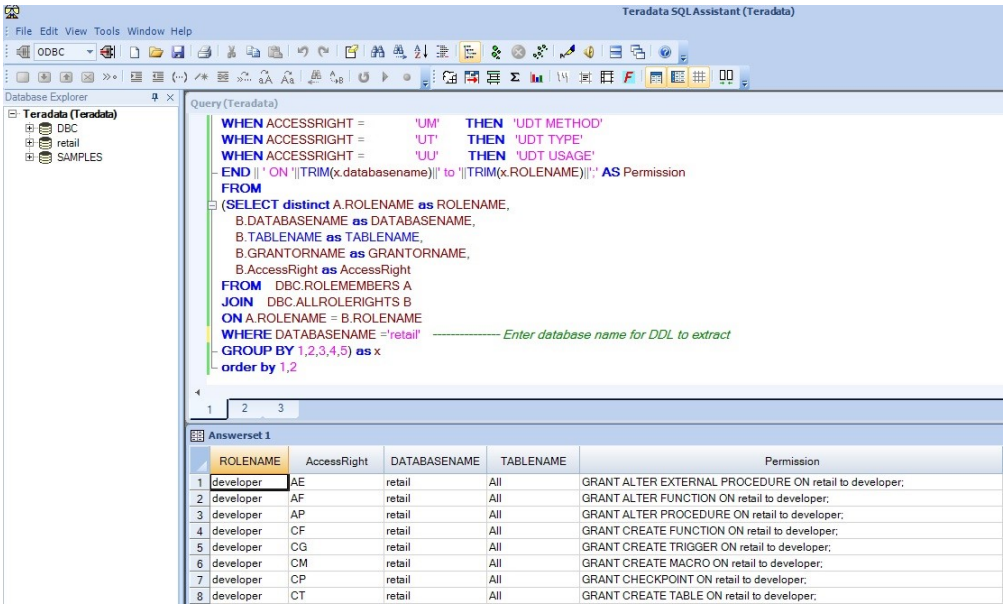



Figura 3.3: Interfaccia di assistenza per le interrogazioni al DB Teradata

Per interfacciarsi graficamente ai database allocati sui diversi server è disponibile *Teradata SQL Assistance*, che rappresenta il client SQL rilasciato insieme alla licenza.

# Capitolo 4

# Ciclo di vita del software

Per garantire un efficiente approccio all'implementazione di componenti software si è attuata una metodologia per la gestione dello sviluppo delle applicazioni basata sull'**Application Development Model**, finalizzato alla standardizzazione del ciclo di vita del software contribuendo all'integrazione dei servizi rilasciati con le continue evoluzioni richieste dal Business. Il processo si basa sulle seguenti fasi:


Figura 4.1: Processo seguito per l'implementare i componenti software

1) **Pianificazione**: La fase di pianificazione finalizza le attività necessarie alla consegna dei servizi sulla base dei vincoli di ambito, tempi e costi, con le risorse stabilite e secondo gli standard di qualità richiesti dall'organizzazione. Il team di sviluppo dovrà effettuare una valutazione del tipo di richiesta, determinare se il tipo di impatto risulta correttivo, evolutivo o conservativo e pianificare il rilascio; dovrà inoltre analizzare se i componenti da implementare avranno un impatto su terze parti; in tal caso dovrà pianificare un coinvolgimento con il fornitore e relativo piano di approvvigionamento.
   In questa fase dovrà essere effettuato un set-up del team di lavoro in base alla piattaforma di esecuzione, ai sistemi operativi, all'ambiente software, all'ambiente aziendale, ai database, e alle attività svolte.

2) **Raccolta e analisi dei requisiti**: In questa fase il gruppo di sviluppo procede alla raccolta dei requisiti funzionali e non funzionali, allo studio dei manuali e della documentazione infrastrutturale rilasciata nella fase precedente. Viene effettuata una valutazione dei processi relativi al ciclo di vita del software e dell'architettura HW e SW della soluzione. Viene inoltre quantificata la portata e la complessità delle questioni tecniche e commerciali e potrà essere finalizzato il processo di stima dell'impatto delle evolutive richieste.
   Il Team di *Application Development* dovrà dettagliare i requisiti di qualità nonché i requisiti non funzionali, procedere alla redazione o all'aggiornamento dei piani di sicurezza, collaborando con i dipartimenti preposti per garantire che siano rispettate le norme di governance sulla protezione dei dati.

3) **Design**: Il gruppo di lavoro, avendo acquisito una conoscenza completa dell'ambiente di esecuzione delle applicazioni, delle conoscenze operative e delle logiche di business strutturali e progettuali procede al disegno della soluzione tecnica e alla redazione delle specifiche funzionali che descrivono il tipo di servizio richiesto con le relative soluzioni applicative e architetturali. Si procede quindi alla pianificazione degli unit test e dei system test e relativi use case per assicurare che la soluzione implementata continui a soddisfare i requisiti delineati durante la fase di progettazione e si integri in maniera corretta con l'ambiente consolidato già in essere. Il team dovrà inoltre gestire attività critiche come la prioritizzazione delle modifiche e il controllo delle versioni.

4) **Integrazione e testing**: A questo punto si intraprende la realizzazione effettiva dello sviluppo del codice, il bug fixing, si eseguono i test di sistema, unitari e funzionali sul codice sviluppato prima del rilascio a collaudo; si realizza il controllo delle richieste di modifica e viene aggiornata la documentazione. Il team dovrà inoltre realizzare il controllo di qualità del servizio secondo gli standard previsti in pianificazione.

5) **Delivery**: Terminata la fase di design lo staff dedicato agli sviluppi è pronto

al rilascio del software corredato dai test interni e della documentazione funzionale al gruppo di collaudo. Parallelamente agli sviluppatori, esso realizza attività di allineamento sulle funzionalità sviluppate e procede al rilascio del manuale utente al gruppo di *Application Management* per favorire la transizione del servizio. Non è da escludere però che il team di sviluppo e team di collaudo siano composti dalle medesime figure.
Contestualmente alla consegna del software sarà prodotto un documento di sintesi nel quale saranno elencati tutti i deliverables prodotti.

6) **Collaudo**: Le attività di collaudo sono indirizzate a verificare che il software sia esente da malfunzionamenti e che possa essere rilasciato all'esercizio senza impatti negativi sull'operatività e sull'erogazione del servizio.

Per garantire la continuità del business e favorire la minimizzazione del rischio legato al passaggio dalla fase di implementazione alla fase di manutenzione, si attua una sinergia tra i team di **Application Management**(AM), ovvero colore che si occupano del monitoraggio dei componenti attivi a regime, ed **Application Development** fin dall'inizio della transizione.
Già durante la fase di implementazione, si prevede di inserire key people di AM in affiancamento al team di sviluppo per effettuare training on the job prima del passaggio della release software in esercizio. In questo modo le attività di trasferimento delle conoscenze tra un team e l'altro risulteranno agevolate e faciliteranno la formazione delle risorse di sviluppo che non hanno partecipato all'implementazione della release. Questa sinergia consente di snellire il processo di passaggio delle nozioni applicative e tecnologiche e di atterrare su un modello di consegna efficace.

## 4.1 Gestione dei rilasci

Durante l'intero arco temporale interessante gli sviluppi si assume che sia sempre garantito il forte impegno dei fornitori delle tecnologie in ambito al progetto con l'obiettivo di risolvere tempestivamente anomalie di prodotto, suggerire best practices su eventuali soluzioni alternative e su anomalie bloccanti.
Le attività previste ricalcano l'approccio **Agile Ibrido** e prevedono quindi una gestione *Agile* di quota parte del ciclo di vita del software, dalla modellazione al rilascio in esercizio.
In particolare, l'approccio Agile potrà essere applicato nel caso fossero già disponibili dei framework di sviluppo consolidati per la realizzazione delle varie componenti software.

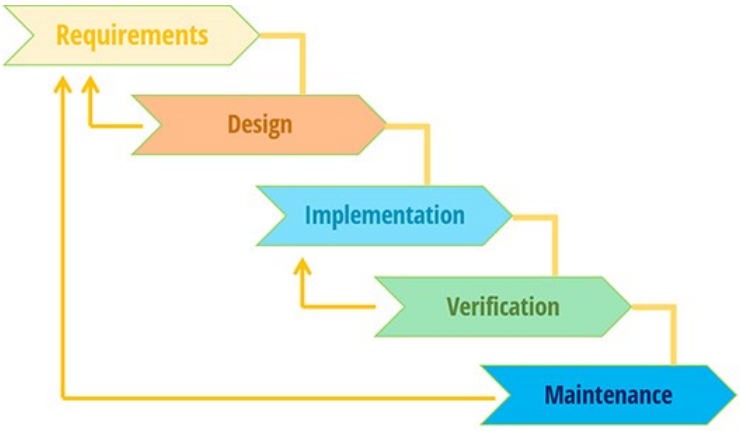



Figura 4.2: Interazione seguenziale e/o ciclica delle fasi tipica di un approccio Agile Ibrido

Tutelare la netta separazione delle diverse fasi e assicurare la possibilità di parallelizzare le attività di sviluppo richiede un'implementazione strutturata creando ambienti differenti posizionati sul server distinti al fine di garantire un distacco netto tra gli step: ovviamente il passaggio di un componente da un ambiente al successivo o al precedente sarà governato dalle metodologie imposte dalle linee guida interne alla banca. Tuttavia, nel caso in cui non fossero già presenti i framework di sviluppo consolidati o dove si prevede la realizzazione di funzionalità tutte fortemente interconnesse e dipendenti in termini di modello dati e processi di elaborazione, potrà essere adottato un approccio *Waterfall* più tradizionale.

# Capitolo 5

# Logiche di processo

## 5.1 Analisi dei componenti

### 5.1.1 Flussi dati

I flussi dati sono il contenuto informativo creato ed inviato in maniera costante all'interno del sistema fisico descritto è rappresentano l'unità fondamentale trattata. I singoli record contenuti al loro interno vengono formati da combinazioni di campi, raffigurabili come colonne di una tabella, più un ultimo campo precariamente vuoto per dare la possibilità in futuro di aggiungere nuovi campi al flusso senza doverne cambiarne la dimensione e mantenere quest'ultima costante. L'interruzione di un campo e l'inizio del seguente può avvenire secondo uno schema *delimitato*, in cui un separatore predefinito delinea la fine di un campo e l'inizio del successivo, oppure secondo uno schema *posizionale*, dove viene definita cardinalmente la lunghezza del campo con le rispettive posizioni iniziali e finali. Ogni flusso ha origine con un header e si conclude con un footer, due righe standard che fungono da informatori anagrafici dei flussi che li contiene per quelli che sono il codice identificativo, la banca mittente, il numero di record presenti nel flusso e l'orario di atterraggio nel sistema. Ogni tupla modellata dovrà infine contenere una serie di campi tecnici imposti dal framework come il timestamp di aggiornamento ed il timestamp di creazione del record, il primo indispensabile per le estrazioni a batch per cogliere i record modificati dall'ultimo approvvigionamento delle entità ed il secondo utile per sancire il momento in cui il record viene scritto in tabella. Le informazioni trasmesse nei flussi sono riportate in relazione ad una specifica data di riferimento, la quale precisa la giornata alla quale si imputa in contenuto informativo stesso.

I flussi bancari vengono suddivisi in Business Area secondo la natura intrinseca

dell'informazione contenuta nel pacchetto dati, dimodoché una volta catalogati seguano direttive imposte specificatamente per l'area di appartenenza.

### 5.1.2 Tabelle e Viste

Le tabelle e le viste rappresentano i contenitori dell'informazione presente nella base dati del modello. Per progetti di questa natura vengono costruite diverse tipologie di tabelle e viste con lo scopo di custodire differenti conoscenze sul dato e sulle esecuzioni attorno lui. Tra esse possiamo individuare:

| **Famiglia** | **Descrizione** |
|---|---|
| *Eventi* | Racchiudono il contenuto informativo dei record |
| *Relazione* | Definiscono le relazioni tra i vari componenti del framework |
| *Anagrafica* | Registrano i movimenti che avvengono all'interno del sistema |
| *Work* | Elementi di appoggio non visibile all'utente |
| *Eccezioni* | Contengono i record che hanno fallito i controlli |

Tabella 5.1: Elenco delle diverse tabelle componenti il sistema.

Per ridurre le tempistiche di impianto delle nuove tabelle nel sistema, specialmente in situazioni richiedenti una creazione massiva, si è deciso di ricorre ad un codice sviluppato in linguaggio Python per comporre automaticamente le DDL (Data Definition Language) desiderate.

### 5.1.3 Controlli sul dato

Durante le attività di elaborazione sui dati è necessario introdurre alcuni controlli al fine di verificare che il dato attualmente processato corrisponda effettivamente alle caratteristiche attese, in quanto esso potrebbe essere affetto da errori di vario tipo. I controlli effettuati si dividono in due grandi famiglie che sono quella dei **controlli tecnici**, che vengono eseguiti in fase di alimentazione del lake, e **controlli di integrità referenziale**, eseguiti questi in fase di alimentazione dello strato modellato.

I primi a loro volta si suddividono in **controlli formali** e **controlli di trasporto**. I controlli in atto sono elencabili con:

- **Controlli di Formato**
  - *Famiglia:* Tecnico - Formali
  - *Criterio di qualità:* Coerenza
  - *Descrizione:* Verifica che il formato sia in linea con il datatype dichiarato
- **Controlli di Nullability**
  - *Famiglia:* Tecnico - Formali
  - *Criterio di qualità:* Esistenza
  - *Descrizione:* Verifica che il campo sia valorizzato come atteso
- **Controlli di Chiave Duplicata**
  - *Famiglia:* Tecnico - Formali
  - *Criterio di qualità:* Consistenza
  - *Descrizione:* Verifica la presenza di chiave primaria duplicata
- **Controlli Andamentali**
  - *Famiglia:* Tecnico - Formali
  - *Criterio di qualità:* Coerenza
  - *Descrizione:* Verifica che, a parità di flusso, sul numero di record presenti non vi siano variazioni significative nel tempo
- **Controlli di Trasporto**
  - *Famiglia:* Tecnico - Trasporto
  - *Criterio di qualità:* Copertura
  - *Descrizione:* Verifica tra il numero di righe dichiarato sul record di coda del file ed il numero effettivo di righe presenti nel file sia il medesimo.
- **Controlli di Integrità Referenziale**

- *Famiglia:* Integrità Referenziale
- *Criterio di qualità:* Integrità
- *Descrizione:* Verifica della corretta referenzialità sulla chiave esterna di una tabella padre all'interno di una tabella figlia

Da un controllo avente esito negativo si può ricavare uno *scarto*, che rappresenta un dato che non ha superato un controllo di validità a fronte di impostazioni ben precisa e viene quindi eliminato definitivamente, un *warning*, che rappresenta un dato che non ha superato un controllo di validità ma non viene eliminato definitivamente poiché potrebbe divenire valido in futuro oppure potrebbero essere modificate le impostazioni che lo rendevano non valido ed un *riciclo* il quale rappresenta un dato precedentemente scartato con warning, differente dai dati appartenenti ad una nuova elaborazione, che ha la possibilità di rientrare nel flusso principale affinché sia nuovamente verificata la sua validità.

Un esito negativo dei controlli di trasporto o formali volti a verificare la bontà del flusso di input genera uno scarto, precisamente in presenza di errore di trasporto l'intero file viene scartato mentre in presenza di errore formale viene scartato il corrispondente record. La violazione dei controlli di integrità referenziale prevede invece due comportamenti distinti quali lo scarto del record oppure la segnalazione senza scarto; sarà l'utente ad indicare il comportamento da intraprendere.

Gli esiti di tutti i controlli eseguiti in fase di alimentazione devono essere opportunamente segnalati secondo le direttive dell'ufficio Data Quality che prevedono la realizzazione di un modello di gestione degli esiti che riporti, per ogni controllo implementato e dizionarizzato sul sistema, l'elenco delle informazioni che hanno violato un controllo in formato sintetico e dettagliato.

## 5.2 Fase di alimentazione

Il processo complessivo di invio dalla sorgente al sistema target non è rappresentato da un canale diretto tra le due figure bensì è suddiviso in due fasi principali, consentendo così il passaggio dei flussi attraverso il framework e la conseguente modellazione al suo interno. L'intera trasmissione dati inizia con la **Fase di Alimentazione** del modello da parte delle banche; il timing con cui vengono spediti i dati viene pattuito tra i due attori cooperanti in fase di analisi e va a definire quello che è il *cut-off* di invio, ossia l'istante atteso di pervenimento del flusso nel sistema, dipendente anch'esso dalla natura del dato stesso; mentre alcune famiglie di dato necessitano di essere ricevute con una cadenza giornaliera, altre vengono spedite in modalità settimanale, decadale (ogni 10 giorni), mensile, trimestrale, semestrale o annuale, in flussi che racchiudono l'informazione dell'intero arco di tempo trascorso dall'ultimo invio.

L'elemento contenente le informazioni richieste dalla capogruppo prodotto ed inviato dalle sorgenti, ovvero le banche controllate, viaggia verso il sistema centrale attraverso un canale basato sul protocollo File Transfer Protocol, tipico protocollo per la trasmissione di dati tra host fondato su un architettura di tipo client-server.

Terminato il percorso lungo il canale i flussi dati atterrano nel framework agganciando automaticamente gli schedulatori di sistema, i quali accodano l'informazione ed innescano il processo di caricamento.

L'operatività delle azioni viene affidata ai job Datastage in modalità automatica, andando a ribaltare il contenuto informativo presente nei flussi in entrata sulle tabelle di work, identificabili come staging area del modello, svuotandole prima del loro contenuto. Eseguita questa operazione le righe delle tabelle di staging vengono trasferite sulle tabelle degli eventi aventi il compito di storicizzare il patrimonio informativo, ed in ultima battuta popolare di conseguenza le tabelle degli scarti con il record che non hanno superato i controlli.

Tutte le esecuzioni avvenute fino a questo punto, come quelle che avverranno nelle fasi successive, saranno assoggettate ad una registrazione nei file di log presenti nel file system del modello in cui viene tracciata l'intera storia di ciò che è stato compiuto e grazie al quale, in uno scenario di malfunzionamento, permette di capire il punto esatto in cui si è verificato l'errore.

## 5.3 Fase di estrazione

In contrapposizione alla fase di alimentazione vi è la **Fase di Estrazione**, in cui il framework modella i flussi acquisiti dalle sorgenti plasmandoli in flussi TO-BE secondo le logiche ETL intrinseche del Modello Logico Applicativo. La modellazione viene stabilita con la collaborazione del Data Office che trasmette le specifiche per mezzo di allegati rappresentanti le trasposizioni tecniche delle analisi funzionali da loro condotte, ed una volta prese in carico vengono tradotte in termini tecnici applicabili direttamente ai meccanismi di modellazione del software.

Le logiche applicate tipicamente consistono nel raggruppare più tabelle secondo uno schema Master-Slave, in cui la tabella master è colei che in base alle chiavi primarie definisce il perimetro di dominio entro cui operare e le tabelle slave forniscono i campi scelti grazie ad operazioni di join. Non tutti i campi necessari vengono estratti 1:1 con i campi delle tabelle derivanti ma necessitano di essere calcolati, oppure alcuni attributi impongo aggregazioni per permettere di giungere ad entità significative per una interpretazione fluida. Dopodiché i valori dei campi vengono sottoposti a funzioni di Trim per privarli di spaziature in modo che non contengano caratteri vuoti ininfluenti per le analisi.

Terminata l'elaborazione dei componenti Datastage preposti alla creazione dei nuovi flussi, gli elementi generati andranno ad agganciare gli schedulatori di estrazione che procederanno al loro invio verso la filiera. La trasmissione delle informazioni attraverso i canali, ugualmente nello scenario di alimentazione ed in quello di estrazione, viene diretta da file Shell, eseguibili Unix equivalenti a file Bat di Windows contenenti istruzioni sequenziali che vengono interpretate dalla console ed eseguite localmente; saranno proprio loro ad automatizzare l'intero processo e notificare automaticamente tramite mail l'intero gruppo di figure coinvolte.

## 5.4 Parallel Running

L'intero processo appena descritto avviene in uno scenario in cui ognuno dei componenti citati è presente nell'ambiente a regime e gli sviluppi apportati ai moduli software producono effetti operativi non solo in una panoramica di test ma all'interno del sistema reale. Prima che tutto ciò risulti in auge dopo la pianificazione e che le banche e le filiere siano pronte al consolidamento dell'invio e della ricezione dei flussi possono passare numerosi mesi in cui il team di sviluppo e di collaudo, assieme alle figure appena citate stanti agli estremi del canale, simula l'esecuzione delle fasi di alimentazione ed estrazione dei flussi con operazioni svolte in modalità manuale/semi-automatica, operazioni queste che saranno in seguito automatizzate completamente; nel complesso, queste operazioni compongono la fase di **Parallel Running**.

Per quanto riguarda la fase di alimentazione al team sarà in capo il monitoraggio dei flussi provenienti dalle banche in ambiente di system test cosicché le sussidiarie possano adeguarsi ai tracciati tecnici che definiscono le specifiche censite ed i job incaricati del caricamento del framework vengano sottoposti a "test sotto sforzo". Esiti negativi nel monitoraggio portano a correzioni realizzabili con ampi gradi di libertà in questa fase, mentre qualora dovessero sorgere in ambiente collaudato le modifiche dovrebbero rispettare vincoli burocratici assai restrittivi.

Nel processo di estrazione invece, essendo il framework il mittente della fase, il team dovrà compiere le funzioni che svolgerebbe autonomamente il sistema secondo la schedulazione, quali l'avvio dell'elaborazione tramite il "run" manuale del job datastage ed effettuo dell'invio tramite il lancio della shell dedicata attraverso comandi da terminale.

In alternativa, per semplificare la procedura e minimizzare le operazioni svolte manualmente cosicché si riducano tempistiche ed errori legati al comportamento umano, si è deciso di creare un meccanismo automatizzato da una shell incaricata di leggere da una tabella elencante la lista di comandi impartiti dall'operatore ed eseguire quali di questi non siano ancora stati compiuti. All'operatore basterà semplicemente inserire un nuovo record nella tabella mediante script SQL per impartire alla procedura l'esecuzione dell'azione voluta; sarà poi la shell stessa ad individuare la tipologia di lavorazione impartita in base a quale dei flag presenti viene valorizzato

dal richiedente.

Oltre alle informazioni necessarie per un corretto funzionamento da parte dei meccanismi di estrazione viene aggiunto un campo rappresentante il timestamp della richiesta valorizzato al secondo corrente nel momento in cui viene scritto il comando in tabella per permettere alla proceduta, pianificata all'avvio automatico su un arco temporale predefinito, di andare a recuperare ogni richiesta risultante anche in attesa di esecuzione; grazie ad un ulteriore campo booleano la proceduta è in grado di distinguere le operazioni attuate da quelle ancora da attuare.

Per verificare la bontà del file generato viene infine lanciata una comparazione incrociata confrontante il flusso TO-BE in uscita con quello AS-IS inviato dalla banca verso il sistema target con l'ausilio di un job Datastage incaricato di scrive in tabella il rapporto del paragone. Il risultato viene successivamente estratto generando un report numerico, leggibile dal Data Office, con il quale compiere le proprie verifiche.

# Capitolo 6

# Strumenti a supporto

A seguito del crescente numero di operazioni raggiunte nel sistema e dalla numerosità degli operatori coinvolti si è reso fondamentale creare uno strumento di raccordo che garantisse l'adozione del modello di data governance vigente e fornisse uno metodo di monitoraggio e gestione di tutti i flussi alimentanti il sistema centrale. Per questo motivo è stato deciso di sviluppare una nuova dashboard che fornisse al Data Office sia gli strumenti necessari per un presidio diretto sulle attività lato capogruppo nonché lato controllate per tenere traccia, attraverso una singola interfaccia, le informazioni riguardanti l'alimentazione, il caricamento e l'elaborazione ed avvisasse gli utenti di business in caso di interruzioni sistemistiche, notificando inoltre l'evento scatenante.

Oltre a fungere da repository centralizzato per il framework, la dashboard in esame detiene una componente dispositiva per permettere all'utente di interagire con il modello documentale impiantato alla base in cui vengono racchiusi la totalità dei report scambiati tra gli utenti del Data Office ed il team di sviluppo, concedendo i privilegi di creazione, eliminazione ed aggiornamento delle versioni della documentazione.

Con questa nuova impostazione risulta possibile arricchire le informazioni giungenti all'utente dando una nuova visione globale delle anomalie, riuscendo di conseguenza a reagire prontamente attraverso richieste di correzione delle problematiche, rispedizione dei flussi e delucidazioni sull'incoerenza tra dati.

Per soddisfare al meglio i requisiti imposti al nuovo strumento è stata predisposta un'infrastruttura composta da 3 livelli principali:

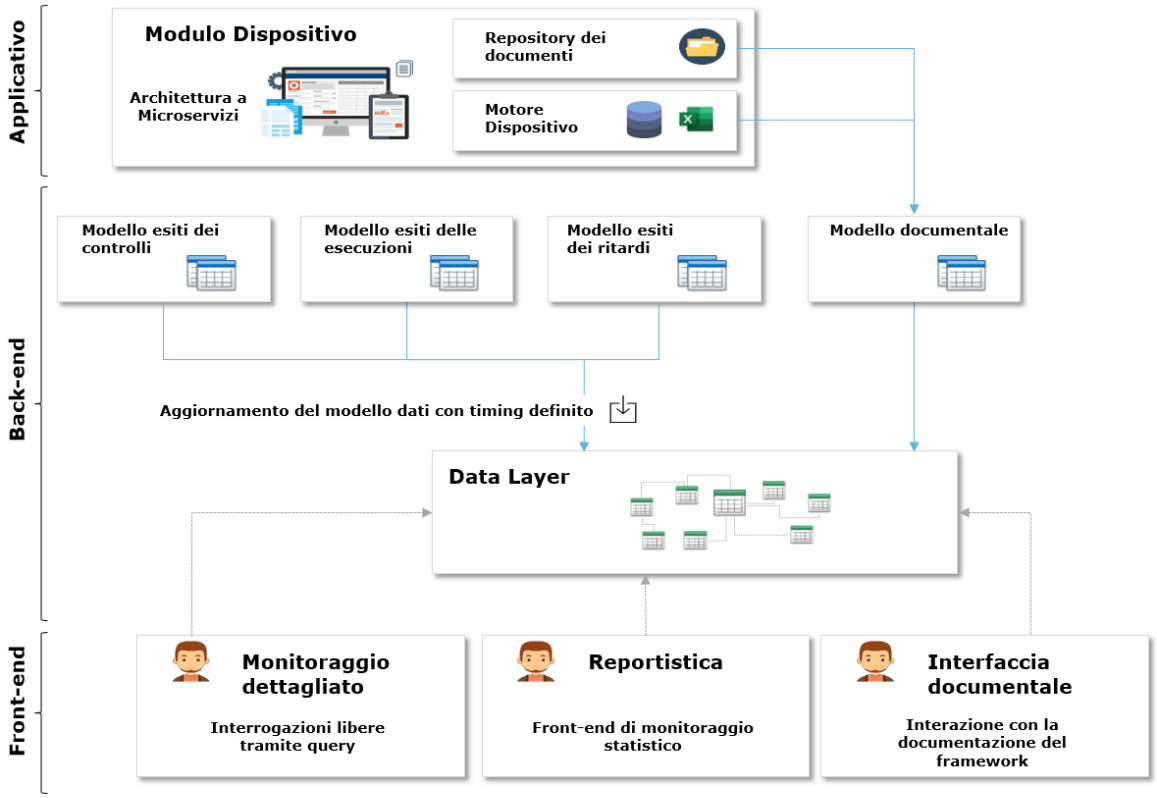



Figura 6.1: Rappresentazione concettuale dei tre strati architetturali formanti la dashboard.

1) **Livello Applicativo**: il primo strato architetturale prende il ruolo di repository centralizzato e motore di generazione di documenti con standard propri del framework, offrendo all'utente la possibilità di modificare le configurazioni direttamente sul front-end. Tutti i documenti basati su tali configurazioni specifiche vengono inviati automaticamente alle persone responsabili dell'attivazione del sistema di destinazione, alimentando il back-end. La realizzazione di questo modello si basa sul concetto di Microservizi.

2) **Livello Back-end**: questo livello è un modello dati con informazioni relative ad alimentazioni e caricamenti di dati, ritardi nelle spedizioni e risultati dei controlli oltre alle informazioni inserite direttamente dal dispositivo; ogni singolo elemento utile al monitoraggio viene estrapolato direttamente dalle tabelle registranti l'esecuzione delle elaborazioni così da avere sempre una fotografia non distorta di ciò che è accaduto. I dati verranno aggiornati con la massima frequenza consentita, in modo che la dashboard sia in grado di mostrare le informazioni nel minor tempo possibile.

3) **Livello Front-end**: gli utenti saranno in grado di eseguire query per recuperare informazioni dal modello dati implementato. Vengono creati report operativi e linee guida al fine di includere tutte le informazioni sul modello dei dati e su KPI imposti dalla Data Quality.

## 6.1 Microservizi

Realizzare la componente dispositiva del sistema analizzato ha richiesto l'implementazione di un'architettura di sviluppo basata su *Microservizi*, scelta maggiormente appropriata per la composizione di soluzioni di questo tipo, nonché imposta dalle linee guida interne dello sviluppo.

Un approccio allo sviluppo fondato sui microservizi rappresenta una via per far nascere architetture software in cui i componenti creati sono composti da servizi indipendenti di volumi ridotti che interagiscono grazie ad API specifiche, ovvero librerie di funzioni che permettono di effettuare chiamate a parti di un programma per abbreviare il lavoro dello sviluppatore.

Le architetture a microservizi danno la possibilità di raggiungere una scalabilità tale da sviluppare le applicazioni in modo molto più rapido delle metodologie tradizionali, consentendo quindi di accellerare il deploy di nuovi componenti.

Essa si differenzia dalle architetture monolitiche utilizzate nella programmazione comune degli anni precedenti all'avvento del cloud computing, in cui tutti i processi sono connessi tra loro, prendendo le sembianze di un unico servizio. Quest'ultima configurazione presenta inevitabilmente l'esigenza di essere interamente ridimensionata nel momento in cui viene sottoposta a picchi nelle richieste. Operando inoltre in questa modalità risulterà complessa l'aggiunta o il miglioramento delle funzionalità sui componenti, in quanto sarà necessario interrompere l'erogazione di tutti i servizi presenti nel programma per portare a termine eventuali modifiche. Per questo motivo saranno limitati numerosi tentanti di cambiamento nella struttura del codice, rendendo più difficoltoso implementare nuove idee. Le vecchie architetture rappresentano un ulteriore rischio per l'affidabilità dei programmi, poiché la presenza di numerosi processi dipendenti e strettamente collegati aumenta vertiginosamente il danno provocato da un possibile malfunzionamento su un singolo processo.

Con un'architettura costruita su microservizi invece gli elementi vengono sviluppati in modo che siano indipendenti, rendendoli ovvero singoli flussi elaborativi che eseguono ciascuno un processo applicativo a sé stante. La comunicazione tra questi componenti avviene attraverso API snelle che fungono da interfacce utili ai vari servizi per dialogare tra loro. I servizi sono realizzati da piccoli team di lavoro separati ed autonomi per le funzioni aziendali e ogni servizio esegue una sola ed unica funzione. Essendo strutturalmente isolato ogni servizio può essere ammodernato o ridimensionato liberamente per colmare esigenze funzionali specifiche del programma.

Una volta completata, l'applicazione gestirà le richieste eseguendo la logica impartitagli, effettuando l'accesso ai database e restituendo infine risposte HTML, JSON o XML. Dovranno inoltre dare la possibilità di integrare in modo asincrono applicazioni e/o microservizi esterni.

La configurazione di un servizio si struttura secondo tre tipologie di componenti:

- Componenti della presentazione, responsabili della restituzione di un'interfaccia con cui l'utente si relaziona con i servizi.

- Logica di dominio, ossia la logica operativa intrinseca imposta all'applicazione.

- Logica di accesso, costituita da componenti di accesso ai dati presenti nei database (SQL o NoSQL).

Potrebbe sembrare naturale che i client, approcciandosi con una applicazione composta da una moltitudine di servizi, chiamino direttamente tutti i servizi utili alla realizzazione di una determinata funzionalità. Nella realtà ciò viene semplificato attraverso l'introduzione di un componente avente lo scopo di orchestrare le richieste e fare da tramite verso i servizi necessari: questo elemento prende il nome di *API Gateway*. Questo strumento nasce principalmente per esporre un'interfaccia verso i client e si occupa di realizzare la logica in maniera trasparente al soggetto richiedente: per fare un esempio un client richiama un unico servizio e l'API gateway, dopo aver preso in carico la richiesta, lo realizza chiamando gli n servizi necessari all'esecuzione completa ed invia l'output al client. Questo strumento, se sviluppato il più leggero possibile per risultare altamente disponibile e scalabile in relazione al

carico, ottimizza la comunicazione tra il client e l'applicazione limitando di fatto il numero di servizi chiamati direttamente dall'operatore.

# Capitolo 7

# Conclusioni

L'esperienza lavorativa nel complesso è stata particolarmente interessante in quanto è riuscita a mostrarmi come i concetti informatici e gestionali appresi nel persorso di studi interagiscano operativamente tra loro e vengano applicati in un contesto aziendale. Grazie ad essa sono riuscito ad accrescere le competenze tecniche acquisite didatticamente in materia Big Data, Business Intelligence e Data Warehousing nonchè ad ampliare le conoscenze su molteplici linguaggi di programmazione e diversi strumenti informatici usati nel mondo degli Analytics e non solo. Ho inoltre avuto la possibilità di vedere da vicino come i metodi di gestione dei progetti sul cliente vengano applicati in uno scenario reale come la pianificazione di tempo e risorse, la partecipazione ai SAL (Stato Avanzamento Lavori) e project meeting e la partecipazione a RFP (Request for proposal) per offerte progettuali.

Questo periodo, seppur breve, mi è stato estremamente utile per toccare con mano diversi aspetti del mondo del lavoro che spaziano per esempio dallo svolgere attività in gruppo al rispettare le scadenze, dall'essere indipendente all'assumersi le proprie responsabili per le mansioni eseguite, facendomi acquisire le basi di un buon metodo di lavoro e collaborazione con gli altri.

## 7.1 Implementazioni future

In un ipotesi di prolungamento della durata progettuale le attività di sviluppo interesserebbero sicuramente l'ampliamento del numero di sorgenti dato del sistema, andando ad inglobare la restante parte di istituti controllati non standardizzati secondo il modello concepito. Allo stesso modo si potrebbero inserire nuove filie-

re utilizzatrici dell'informazione, andando a replicare i processi di modellazione e rendendoli customizzati per i nuovi applicativi entranti.

In virtù di ciò sarebbe doveroso apportare modifiche ad ogni strato dell'architettura volte ad ottimizzare il già performante processo creato cosicchè, all'aumentare dei carichi computazionali impartiti, il sistema non si sovraccarichi e continui a rispondere in maniera reattiva.

Per quanto riguarda il fronte degli strumenti a corredo, si potrebbe sicuremante potenziare l'interfaccia grafica della dashboard sviluppata, aumentando il grado di interattività concesso all'utente attraverso l'aggiunta di nuovi servizi connessi ad ulteriori aspetti infrastrutturali.

# Elenco delle figure

# Elenco delle tabelle

# Capitolo 8

# Sitografia


- https://www.zerounoweb.it/analytics/big-data/come-diventare-la-formula-1-delle-imprese-con-big-data-analytics-fast-smart/
- https://www.zerounoweb.it/techtarget/searchsecurity/la-definizione-della-data-governance-il-primo-passo-per-la-visione-unica-della-realta/
- https://www.01net.it/che-cosa-si-intende-per-data-quality/
- https://vitolavecchia.altervista.org/data-quality-che-cose-e-come-si-misura-la-qualita-dei-dati/
- https://www.techopedia.com/definition/28050/enterprise-data-management-edm
- https://www.zerounoweb.it/analytics/business-intelligence/content-management-un-problema-esplosivo/
- https://it.talend.com/resources/what-is-etl/
- https://it.talend.com/resources/etl-tools/
- https://www.bucap.it/news/approfondimenti-tematici/gestione-del-magazzino/cosa-sono-procedure-etl-data-warehousing.htm
- https://www.bigdata4innovation.it/big-data/big-data-analytics-data-science-e-data-scientist-soluzioni-e-skill-della-data-driven-economy/
- https://www.bucap.it/news/approfondimenti-tematici/gestione-del-magazzino /database-data-warehouse-principali-differenze.htm
- http://databasemaster.it/datawarehousing-strumenti-etl/

- https://atlantic-technologies.com/it/blog/che-cose-un-data-lake/
- https://www.sas.com/it_it/insights/big-data/data-visualization.html
- https://docs.microsoft.com/it-it/power-bi/guidance/star-schema
- https://www.zerounoweb.it/analytics/business-intelligence/data-governance-un-approccio-olistico-in-dieci-punti/
- https://blog.osservatori.net/it_it/data-lake-significato-vantaggi
- https://www.artera.net/it/blog/software/che-cose-il-ciclo-di-vita-del-software/
- https://openskills.info/infobox.php?ID=303
- https://www.zerounoweb.it/analytics/analytics-cosa-significa-quando-e-come-si-usa/
- https://www.zerounoweb.it/analytics/big-data/come-fare-big-data-analysis-e-ottenere-valore-per-le-aziende/
- https://www.digital4.biz/executive/teradata-il-data-warehouse-chiave-di-volta-della-business-analysis/
- https://it.wikipedia.org/wiki/IBM_InfoSphere_DataStage
- https://www.ibm.com/products/infosphere-datastage
- https://docs.microsoft.com/it-it/dotnet/architecture/microservices/multi-container-microservice-net-applications/microservice-application-design
- https://aws.amazon.com/it/microservices/


## 8.1 Sitografia delle figure


- https://www.aleagostini.com/importanza-big-data-21042014.html (figura 1.1)
- https://community.qlik.com/t5/New-to-QlikView/Which-schema-is-best-Star-or-Snowflake/m-p/1099951M259764 (figura 1.4)
- https://www.silicon.it/data-storage/bigdata/idc-big-data-e-analytics-in-ritardo-in-europa-e-solo-per-le-grandi-imprese-105177 (figura 2.1)

- https://etl-tools.info/en/datastage-tutorial-L006_performing-lookups-datastage.html (figura 3.1)
- http://ds.iexpertify.com/2013/01/datastage-director.html (figura 3.2)
- https://www.thesqlreport.com/?p=1065 (figura 3.3)
- https://www.binarysemantics.com/software-development-process-sdlc.html (figura 4.2)