UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA
Dipartimento di Ingegneria dell'Informazione
Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione

TESI DI LAUREA

Progettazione e implementazione delle attività di engineering in un sistema di data analytics per il calcolo dei dividendi di un importante gruppo di moda

Design and implementation of engineering activities in a data analytics system for computing dividends of an important fashion group

Relatore

Prof. Domenico Ursino

Candidato

Giovanni Baldascino

ANNO ACCADEMICO 2021-2022

Sommario

La Data Science, negli ultimi anni, ha scatenato una rivoluzione senza precedenti perchè, grazie alla sua capacità di analizzare grandi quantità di dati, ha permesso a molte aziende di anticipare le tendenze di mercato, migliorare la produttività e offrire prodotti personalizzati ai clienti. La Data Science è, dunque, diventata la parola d'ordine per tutti coloro che cercano di rimanere competitivi nel mercato globale, ma non solo, in quanto la stessa ha anche trovato applicazioni in molteplici settori rivoluzionando il modo in cui questi operano. È essenziale sottolineare che senza un processo di ETL ben pianificato le aziende rischiano di avere dati sporchi, incompleti o incoerenti, e ciò può portare a decisioni sbagliate e perdite finanziarie. Con l'aiuto dell'ETL, le aziende possono garantire che i loro dati siano affidabili e utilizzabili per l'analisi. Il fine della presente tesi è analizzare diverse declinazioni della Data Science esponendo e valutando il percorso progettuale che ha portato alla progettazione e implementazione di un sistema per il calcolo dei dividendi di un importante gruppo di moda quotato in borsa. Il lavoro sviluppato mostra come i leader di un determinato business, come, appunto, quello della moda, si appoggino su tecnologie che, nei loro ambiti d'applicazione, sono altrettanto leader, quali Google BigQuery e Microsoft Power BI.

Keywords: Data Science, Data Analytics, Data Visualization, Data Engineering, Business Intelligence, ETL, Borsa, Dividendi.

Indice

Elenco delle figure

Elenco dei listati

Introduzione

L'affiorare di una nuova ondata di dati provenienti da diverse fonti, insieme alla naturale crescita dei dataset all'interno delle organizzazioni, crea la necessità di una nuova gestione dei dati. I Big Data sono un campo emergente poiché possono far fronte a nuove "scale" di dati; inoltre, sono un campo in cui la tecnologia è innovativa ed offre nuovi modi per riutilizzare i dati ed estrarre valore da essi.

La capacità di gestire efficacemente le informazioni e di estrarre conoscenza è oggi considerata un vantaggio competitivo fondamentale, e molte organizzazioni stanno costruendo il loro core business sulla capacità di raccogliere e analizzare informazioni per raggiungere tale scopo. Oggigiorno, l'adozione della tecnologia dei Big Data nei settori industriali non è più un lusso, ma un'esigenza imprescindibile per la maggior parte delle organizzazioni al fine di guadagnare un vantaggio competitivo sul mercato.

L'analisi dei Big Data può fornire alle aziende una maggiore conoscenza dei loro clienti, dei loro processi interni e dei loro mercati di riferimento e ciò significa prendere decisioni più informate e strategiche. Inoltre, l'analisi dei Big Data può anche aiutare le aziende a migliorare la loro efficienza e ridurre i costi, identificando aree in cui possono essere effettuate ottimizzazioni. Per consentire alle aziende di sfruttare il proprio patrimonio informativo, in vista di decisioni tattico-strategiche, è necessaria la presenza di un insieme di procedure dedite all'estrazione dei dati da una o più fonti, alla loro trasformazione in un formato coerente e al loro caricamento in un sistema di destinazione; ciò prende il nome di ETL.

L'ETL svolge un ruolo importante nella gestione dei dati, in quanto consente alle aziende di integrare dati provenienti da diverse sorgenti, come database, file, servizi web, applicazioni, sensori IoT e molto altro ancora, in un'unica fonte. In questo modo, i dati possono essere analizzati, interpretati e utilizzati per prendere decisioni aziendali informate.

La presente tesi nasce dall'esigenza di un grande gruppo internazionale, quotato in borsa, che opera nel settore del lusso, di intraprendere un programma di digitalizzazione e automazione di tutti i processi finanziari, che in origine erano svolti manualmente. In dettaglio, il cliente ha commissionato a Gruppo Filippetti, società che eroga servizi e sviluppa soluzioni smart negli ambiti della Digital Transformation, delle IoT Technologies & dell'Industry, la progettazione e l'implementazione delle attività di Data Engineering in un sistema di Data Analytics per il calcolo dei dividendi.

La tesi in oggetto è composta da otto capitoli, strutturati come di seguito specificato:

- Nel Capitolo 1 introdurremo il lettore alla Data Science, alla Business Intelligence e al valore dei dati al giorno d'oggi. Successivamente, presenteremo i sistemi utilizzati nella Data Science e i software leader di mercato per i contesti applicativi di riferimento.

- Nel Capitolo 2 parleremo in dettaglio dei tool a supporto del progetto di tesi svolto, ovvero Google BigQuery e Microsoft Power BI.
- Nel Capitolo 3 descriveremo il contesto aziendale di riferimento e il contesto del sistema che si vuole realizzare, evidenziando gli obiettivi da raggiungere. In seguito presenteremo le sorgenti d'alimentazione del processo.
- Nel Capitolo 4 esporremo la progettazione logica del sistema, presentando i sotto-processi. Nell'ultima parte introdurremo la metodologia di project managment utilizzata, cioè quella Agile.
- Nel Capitolo 5 proporremo una panoramica generale sull'implementazione, evidenziando gli errori incontrati e la gestione di questi ultimi. Successivamente verranno presentati l'esecuzione dell'intero sistema e il setup necessario.
- Nel Capitolo 6 descriveremo e effettueremo la SWOT Analysis, con i suoi quattro step, ed elencheremo le lezioni apprese.

CAPITOLO 1

Introduzione alla Data Analytics e Business Intelligence

In questo primo capitolo daremo un'occhiata da vicino allo scenario tecnologico di riferimento; in particolare, parleremo della storia della Business Intelligence, dell'importanza dell'utilizzo dei dati analizzando chi sono i protagonisti in gioco.

1.1 Big Data

I big data sono un termine utilizzato per descrivere un'enorme mole di dati, sia strutturati che non, i quali vengono generati e raccolti in modo continuo e veloce da varie fonti, come sensori, social media, dispositivi mobili, transazioni finanziarie, e molto altro. Questi dati sono caratterizzati dalle famose "5 V" dei big data (Figura 1.1):

- *Volume.*La quantità di dati presenti nel mondo digitale è esorbitante. Gli esperti del settore hanno stimato che nel 2020 c'erano 44 zettabyte di dati nel mondo e prevedono che questo numero quasi quadruplicherà, fino ad arrivare a 175 zettabyte nel 2025. (per intenderci un singolo zettabyte equivale a un miliardo di miliardi di gigabyte). Anche se una singola azienda avrà molti meno dati, potrebbero essere, comunque, molti da gestire. Per un solo terabyte di dati, si ha bisogno di un modo automatizzato per analizzarli in modo efficiente. Con questo volume, non c'è da stupirsi che i Big Data siano un settore così importante e che le aziende di tutte le dimensioni si stiano impegnando per utilizzarli a proprio favore.

- *Velocità.* La velocità di lettura e scrittura dei dati non è applicabile a tutti i settori, e nemmeno a tutte le aziende, ed è, per questo, una V critica nelle 5 V dei Big Data. Con alta velocità dei dati si intende la necessità di lavorare con dati in tempo reale, aggiornati al minuto, che mostrano gli approfondimenti più rilevanti. Alcuni esempi di dati ad alta velocità sono: un negozio online che tiene traccia dei click su una schermata per ogni cliente che naviga sul sito, un'applicazione di navigazione che raccoglie e invia dati a ogni viaggiatore, una rete di social media che tiene traccia di quanto tempo un utente guarda un post e dei tipi di post con cui interagisce.

- *Varietà.* Un'elevata varietà di dati aiuta a scoprire sottili sfumature all'interno del database. Grazie alla varietà dei dati, è possibile mantenere un'esperienza personalizzata per i clienti. I dati raccolti possono provenire da diverse fonti, possono essere sia strutturati che non strutturati, interni o esterni all'azienda, in modo che il loro contenuto

informativo sia in grado di rappresentare la complessità della realtà in esame, e non "una sola dimensione" della stessa.

- *Veridicità.* I dati raccolti devono essere affidabili e "integri" nella loro potenzialità di descrivere la realtà. Dati falsati da errori di rilevazione, da distorsioni dovute a malintesi oppure a inganni costituiscono un grave danno, come osservato da chi opera nel settore: "Bad data is worse than no data": meglio non avere informazioni, che utilizzare dati errati.
- *Valore.* Questa variabile fondamentale indica che l'importanza dei Big Data stessi sta nella possibilità di introdurre informazioni utili per le aziende, e quindi di apportare dei benefici. I dati fine a se stessi, infatti, non hanno alcun valore. Per essere davvero utili devono poter essere convertiti in informazioni preziose, che permetteranno all'organizzazione di verificare, ed eventualmente modificare, la scelta delle sue mosse strategiche.

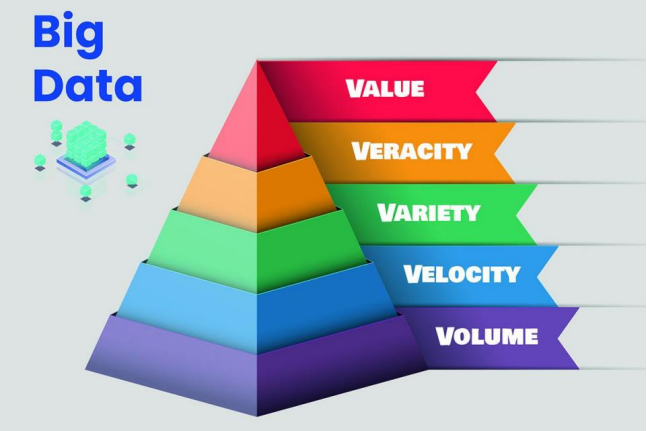

Figura 1.1: Le 5V dei Big Data

I big data sono considerati preziosi perché contengono informazioni che possono essere utilizzate per ottenere insight significativi, identificare tendenze, modelli e correlazioni che possono portare a decisioni informate e a una migliore comprensione del comportamento degli utenti, dei mercati e di altri fenomeni complessi. Tuttavia, l'analisi dei big data presenta anche sfide significative, come la necessità di archiviare, elaborare e analizzare grandi quantità di dati in modo efficiente e accurato, garantire la privacy e la sicurezza dei dati e affrontare questioni etiche legate all'uso dei dati. Nonostante le sfide, i big data hanno un'enorme potenzialità e sono ampiamente utilizzati in molte aree, come il marketing, la finanza, la sanità, la scienza, la logistica, l'energia, e molto altro ancora. L'analisi dei big data ha permesso di sviluppare nuovi modelli di business, migliorare l'efficienza operativa, creare prodotti e servizi innovativi e risolvere problemi complessi in molte industrie.

1.2 Data Analytics

L'analisi dei dati è il processo di esaminare, pulire, trasformare e interpretare i dati al fine di ottenere informazioni significative e informate. Questo processo coinvolge l'applicazione di metodi statistici, matematici e informatici per analizzare grandi quantità di dati, spesso raccolti da diverse fonti, al fine di identificare pattern, tendenze, correlazioni e insight nascosti. L'obiettivo della data analytics è quello di estrarre informazioni utili e prendere decisioni informate basate sui dati, sia per fini commerciali che scientifici. La data analytics è

ampiamente utilizzata in diversi settori, come il business, la sanità, la finanza, il marketing, la scienza dei dati, la ricerca scientifica e molti altri. Esistono quattro tipi principali di analisi dei dati che possono essere utili per le aziende quando si parla di Big Data: descrittiva, diagnostica, predittiva e prescrittiva (Figura 1.2). Vediamoli più in dettaglio:

- *Analisi Descrittiva*: mostra ciò che sta accadendo in un determinato momento in base ai dati in arrivo, spesso tramite una dashboard in tempo reale o mediante l'invio di report.
- *Analisi Diagnostica*: esamina delle informazioni passate per determinare cosa è successo e le cause, spesso sotto forma di una dashboard analitica.
- *Analisi Predittiva*: prevede gli scenari più probabili di ciò che potrebbe accadere sulla base dei dati in possesso.
- *Analisi Prescrittiva*: rivela quali azioni devono essere intraprese dopo aver eseguito un'analisi predittiva e si traduce in regole e passi raccomandati.

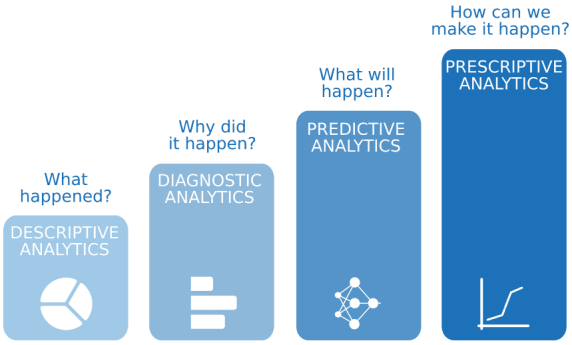

Figura 1.2: Le quattro principali tipologie di analisi

1.2.1 Benefici

Una scelta di strategia efficace è un fattore chiave per il successo aziendale. Esistono diversi casi d'uso dei Big Data, uno di quelli di maggiore impatto è l'utilizzo di questi per il miglioramento aziendale. Ma perché proprio i Big Data sono vantaggiosi per le aziende? I vantaggi dei Big Data per le aziende sono principalmente quattro:

- *Migliorare l'esperienza dei clienti*. Quando si raccolgono dati sui clienti, si possono tracciare le tendenze per poter offrire loro il miglior servizio durante tutto il percorso con l'azienda.
- *Problem Solving*. I Big Data possono essere utilizzati per identificare i processi che generano dei colli di bottiglia e per porvi rimedio con l'analisi diagnostica dei dati.
- *Aumento dei ricavi*. I Big Data consentono alle aziende di aumentare i ricavi anche iterando i prodotti esistenti e progettando nuovi lanci in base alle esigenze comprovate dei clienti.
- *Riduzione dei costi*. Per loro natura, i Big Data funzionano su scala; essi eliminano la necessità di raccogliere e analizzare manualmente i dati, il che, insieme alla riduzione del personale adibito, comporta un notevole risparmio sui costi.

1.3 Business Intelligence

Note, quindi, le grandi potenzialità dei Big Data c'è bisogno di un insieme di metodologie, processi, architetture e tecnologie che da un input di dati strutturati o non, restituiscano in output informazioni utili all'azienda in termini di Decision Making, ovvero un insieme di decisioni strategiche che ne migliorino il business, e quindi i profitti. È proprio a questo scopo che nasce la Business Intelligence, un importante strumento che si pone come obiettivo quello di proporre un cambiamento di approccio e cultura aziendale per poterla valorizzare al meglio. Una volta in possesso dei dati ed effettuate le analisi descrittive e diagnostiche, per la BI sarà possibile avviare anche analisi predittive e prescrittive, cercando, quindi, di prevedere l'andamento dell'azienda, con dei processi conosciuti come Business Analytics Process. In breve, la Business Intelligence è essenziale per interpretare i dati e da questi estrapolare informazioni per capire come sta andando la nostra azienda e quali scelte prendere per un futuro più competitivo sul mercato. Si potrebbe pensare che il termine "Business Intelligence" sia un neologismo; in realtà non è così. Infatti, questo veniva utilizzato già nel 1800; comparve per la prima volta nell'opera pubblicata da Richard M. Devens nel 1865: "Cyclopaedia of Commercial and Business Anec- dotes" nella quale si utilizzava il termine "Business Intelligence" per descrivere il segreto del banchiere Sir Henry Furnese, il quale, raccogliendo informazioni politiche e di mercato al fine di proporre delle scelte strategiche prima dei suoi concorrenti, riusciva ad aumentare i suoi guadagni. È nel 1958 che, però, verranno riconosciuti i benefici della Business Intelligence, grazie alla pubblicazione dell'articolo "A Business Intellicence System" opera di Hans Peter Luhn (Figura 1.3) che, in quel tempo, lavorava per IBM. Egli espose, nel suo articolo, un sistema automatico in grado di trasmettere informazioni in diverse aree di organizzazioni industriali, scientifiche e governative. Egli si concentrò su questi particolari organismi poiché erano quelli maggiormente coinvolti nella terza rivoluzione industriale. Si necessitava, quindi, di affiancare alla crescente innovazione tecnologica un sistema di organizzazione di dati che li rendeva più facilmente fruibili.

Figura 1.3: Hans Peter Luhn, il padre della Business Intelligence

Nei primi anni '70 i vendor cominciarono a proporre alle aziende dei sistemi di supporto alle decisioni (DSS, da Decision Support System) alle aziende. Nacque sin da subito una certa competizione che portò a migliorare man mano questi software e alla creazione dei primi Data Wharehouse, che permisero il raggruppamento di dati prima ripartiti in diversi database. Tutto ciò diede il via all'ideazione di diversi tool e software utilizzati nella BI.

Nell'ultimo ventennio del secolo scorso la Business Intelligence veniva sempre più adottata dalle aziende, tuttavia, erano due i principali problemi riscontrati:

- i software e i tool, per via della loro difficoltà di utilizzo da parte di utenti meno esperti, non erano fruibili al massimo;

- le tempistiche necessarie ad elaborare i dati e generare i report risultavano proibitive.

Per tali ragioni iniziarono a nascere tool di BI facilmente utilizzabili dagli utenti meno esperti e venne migliorata la velocità di elaborazione dei dati. Le aziende utilizzavano tantissimo la Business Intelligence costringendo, così, le aziende che non la utilizzavano a decidere se investire su questa o essere surclassate dai competitor più innovativi. Allo stato dell'arte i tool e i software di BI sono mirati a soddisfare le esigenze di qualsiasi azienda facente parte di un settore o mercato specifico sia per questioni di sicurezza aziendale che per ragioni di crescita economica.

1.3.1 Piramide DIKW

Sono conoscenza e saggezza ciò che i soggetti della Business Intelligence vogliono ottenere partendo da dati e informazioni. Questi quattro elementi fanno parte della piramide DIKW che è un modello concettuale che rappresenta una gerarchia di livelli di conoscenza: Dati, Informazioni, Conoscenza e Saggezza. Questa piramide è stata sviluppata come strumento per descrivere la trasformazione dei dati in conoscenza e saggezza attraverso una serie di processi di elaborazione e interpretazione.

Ecco una spiegazione dettagliata dei diversi livelli della piramide di DIKW:

- *Dati.* Il livello più basso della piramide DIKW (Figura 1.4) è rappresentato dai dati, che sono fatti grezzi, elementi di informazione privi di significato intrinseco. I dati possono essere numeri, parole, immagini o qualsiasi altra forma di informazione che possa essere raccolta o registrata.

- *Informazioni* Il livello successivo è rappresentato dalle informazioni, che sono dati organizzati e contestualizzati in modo da avere un significato comprensibile. Le informazioni sono il risultato di elaborazioni e interpretazioni dei dati.

- *Conoscenza.* Il terzo livello è rappresentato dalla conoscenza, che rappresenta una comprensione più profonda e significativa delle informazioni. La conoscenza è il risultato dell'analisi, dell'interpretazione e della comprensione delle informazioni alla luce di esperienze, competenze, intuizioni e contesti. Essa permette di estrarre pattern, relazioni e significati dalle informazioni e di applicarli a situazioni specifiche.

- *Saggezza.* Il livello più alto della piramide DIKW è rappresentato dalla saggezza, che è la capacità di applicare la conoscenza in modo appropriato per prendere decisioni sagge, affrontare problemi complessi e agire in modo etico e responsabile. La saggezza implica la comprensione profonda dei contesti, delle implicazioni e delle conseguenze delle decisioni, nonché la capacità di apprendere e adattarsi alle nuove situazioni. La saggezza si basa sulla combinazione di conoscenza, esperienza e intuizione ed è spesso considerata un risultato di un lungo processo di apprendimento e maturazione.

In sintesi, la piramide DIKW rappresenta una gerarchia di livelli di conoscenza, partendo dai dati grezzi fino ad arrivare alla saggezza, attraverso la trasformazione dei dati in informazioni, conoscenza e, infine, saggezza. Questo modello è spesso utilizzato nel contesto della Business Intelligence e della gestione delle informazioni per comprendere come i dati possano essere utilizzati in modo significativo per prendere decisioni informate e guidare l'azione aziendale.

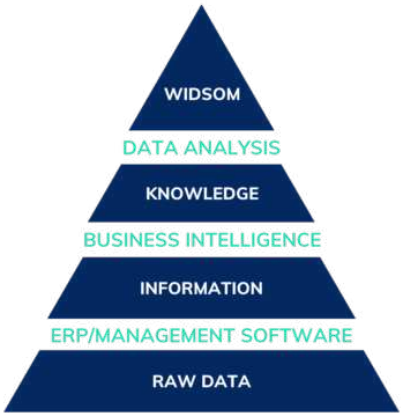

Figura 1.4: Piramide DIKW: dai dati grezzi alla saggezza

1.3.2 Data Engineer

L'analisi dei dati è diventata una pratica comune per prendere decisioni informate e sviluppare strategie aziendali basate su dati concreti. Ma prima di poter analizzare i dati, è necessario un processo di preparazione, trasformazione e gestione degli stessi; è qui che entra in gioco la figura del data engineer o ingegnere dei dati. L'ingegnere dei dati è un professionista specializzato nell'estrazione, trasformazione e caricamento dei dati (ETL) all'interno di un'organizzazione. Il suo ruolo è cruciale per garantire che i dati siano raccolti, memorizzati e resi disponibili in modo efficiente e accurato per l'analisi e l'utilizzo da parte degli analisti e degli scienziati dei dati. L'ingegnere dei dati lavora a stretto contatto con il team di sviluppo software, il team di analisi dei dati e altre figure professionali coinvolte nel processo di gestione dei dati. Una delle principali responsabilità dell'ingegnere dei dati è la progettazione e l'implementazione di soluzioni di data integration e data pipeline, che consentono di acquisire dati provenienti da diverse fonti, come database, file di testo, API, sensori e molto altro, e di trasformarli in un formato coerente e utilizzabile. Ciò richiede una profonda conoscenza dei diversi strumenti di ETL e delle tecniche di data modeling e data governance. Inoltre, l'ingegnere dei dati è responsabile della gestione del flusso dei dati, compresa la pulizia, l'elaborazione, la trasformazione, la validazione e l'archiviazione dei dati. Ciò può includere la normalizzazione dei dati, la gestione dei dati mancanti o errati, la creazione di metadati e la definizione di regole di business per garantire la qualità e l'integrità dei dati. L'ingegnere dei dati deve anche assicurarsi che i dati siano archiviati in modo sicuro e in conformità con le leggi sulla privacy e la protezione dei dati. Inoltre, egli svolge un ruolo chiave nella progettazione e nell'implementazione di soluzioni di scalabilità e performance per i sistemi di gestione dei dati. A mano a mano che le dimensioni dei dati crescono sempre di più, è fondamentale garantire che i sistemi siano in grado di gestire grandi volumi di dati in modo efficiente, evitando tempi di latenza e garantendo alte prestazioni nelle operazioni di interrogazione e analisi dei dati.

1.3.3 Sistemi utilizzati

Affinché le figure appena descritte svolgano il loro compito al meglio devono affidarsi a mezzi che permettano di estrarre, immagazzinare, manipolare e analizzare i dati. Questi riguardano principalmente i sistemi informativi, i database, i DBMS e le loro varie evoluzioni.

Database

Per quanto detto finora, risulta indispensabile un modo di memorizzare i dati che abbia determinate caratteristiche. In particolare, saranno necessarie le seguenti proprietà:

- *Consistenza:* i dati devono essere utilizzabili nelle applicazioni aziendali.
- *Sicurezza:* è necessario impedire danni irreversibili alle informazioni memorizzate.
- *Integrità:* è necessario garantire la conservazione del dato senza perdite.

Inoltre, avendo a che fare con i Big Data, saranno presenti le seguenti caratteristiche:

- *grandi volumi* di dati;
- *alte velocità* di memorizzazione;
- *varietà* dei dati;
- *veridicità* dei dati;
- *valore* da trarre dal dato.

Vengono a crearsi, quindi, delle collezioni di dati correlati dette dataset; ciascun membro all'interno di un dataset condivide gli stessi attributi le stesse proprietà. I dati elaborati in ambito Big Data sono di vario tipo (Figura 1.5) e sono suddivisibili in:

- *Dati Strutturati:* sono dati che si conformano a un modello o ad uno schema e sono memorizzati all'interno di tabelle relazionali.
- *Dati Non Strutturati:* non sono conformi ad uno schema e sono, quindi, di natura diversa. Essei non possono essere interrogati tramite SQL, ma vengono memorizzati in database No-SQL.
- *Dati Semi-Strutturati:* sono per loro natura non relazionali ma hanno, comunque, un certo livello di struttura definito.

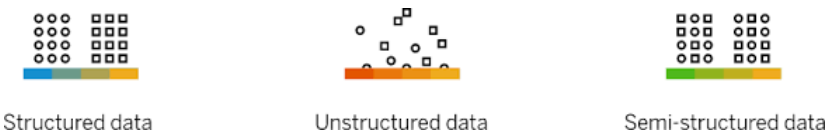

Figura 1.5: Possibili strutture dei dati

DBMS

I database appena descritti necessitano di essere organizzati mediante dei software, detti Database Management System, o DBMS, che consentono le operazioni CRUD (Create, Read, Update, Delete) ovvero operazioni di creazione, lettura,aggiornamento e cancellazione dei dati. Essi, inoltre, si occupano della sicurezza e dall'integrità dei database stessi. Nell'ambito della Business Intelligence i DBMS possono essere classificati in due differenti tipologie, ovvero:

- *OLTP*: sta per On-Line Transactional Process; è volto alla lettura e alla modifica parziale del dato. I dati sono aggiornati; non fanno riferimento, quindi, a dati storici, ponendo l'interesse sulle attività quotidiane. Inoltre, essendo il database utilizzato contemporaneamente da un certo numero di utenti, il DBMS deve gestire meccanismi di controllo, ripristino e concorrenza.
- *OLAP*: sta per On-Line Analytical Process. In questo caso i dati sono aggregati e storici in modo da coprire un arco temporale sufficiente a sottolineare delle variazioni. Il volume dei dati cresce maggiormente, e anche se le operazioni sono di sola lettura, le query risulteranno molto più complesse al punto da richiedere, per la loro gestione, la presenza di utenti specializzati e formati.

Nella Figura 1.6 viene visualizzata la differenza tra OLTP e OLAP.

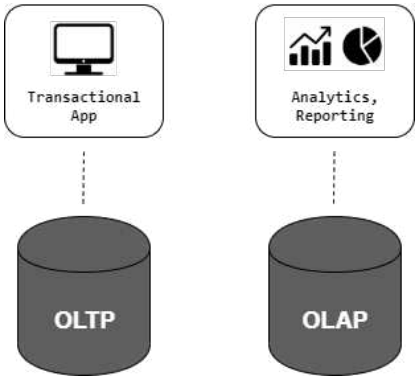

Figura 1.6: DBMS: OLTP e OLAP

Data Warehouse

I dati, a seguito di elaborazione da parte della Business Intelligence, confluiscono in collezioni di dati strutturati chiamati Data Warehouse (DW). Questi permetteranno ad analisti o gestori di vario tipo di eseguire query, effettuare analisi e generare report in modo che le aziende possano ricavare valore da tali informazioni. Ci sono delle caratteristiche che differenziano i DW dagli altri tipi di raccolte di dati. Infatti, i Data Warehouse sono:

- *Integrati:* essendo i dati di un Data Warehouse ottenuti come unione di dati provenienti da fonti esterne, c'è bisogno di creare coerenza; ciò può avvenire, ad esempio, mediante metodi di codifica uniforme od omogeneità semantica delle variabili.
- *Orientati al soggetto:* i DW sono orientati a specifiche aree aziendali, funzioni o applicazioni. I dati, quindi, sono archiviati al fine di facilitarne la lettura e l'elaborazione da parte degli utenti. Lo scopo non è più minimizzare le ridondanze ma fornire dati organizzati per meglio produrre informazioni utili. Si passa da una progettazione per funzione a una modellazione di dati che consente una visione multidimensionale del problema.
- *Variabili nel tempo:* i dati memorizzati in un DW coprono un orizzonte temporale molto maggiore dei singoli sistemi operazionali. Quindi, i dati disponibili sono, di norma, antecedenti alla data in cui l'utente li interroga.
- *Non volatili:* i dati contenuti nel Data Wharehouse non sono modificabili; infatti, l'accesso è disponibile in sola lettura, semplificando di molto la progettazione.

Dovendo il DW compiere operazioni complesse e di pesante elaborazione, risulta chiaro che la sua architettura è di fondamentale importanza. Per questo i DW devono possedere caratteristiche essenziali, come:

- *Separazione:* occorre mantenere separate logicamente l'elaborazione analitica e quella operazionale.
- *Scalabilità:* i volumi crescono velocemente e, di conseguenza, l'architettura deve essere facilmente ridimensionabile in termini di utenti da soddisfare.
- *Estensibilità:* non ci deve essere bisogno di riprogettare il sistema nel caso di ingresso di nuove tecnologie o software.
- *Sicurezza:* per via della natura strategica dei dati contenuti, è essenziale mantenere un rigido controllo degli accessi.
- *Amministrabilità:* è utile che la complessità delle mansioni amministrative non sia eccessiva.

Nella FIgura 1.7 viene riportato uno schema di organizzazione di un Data Warehouse.

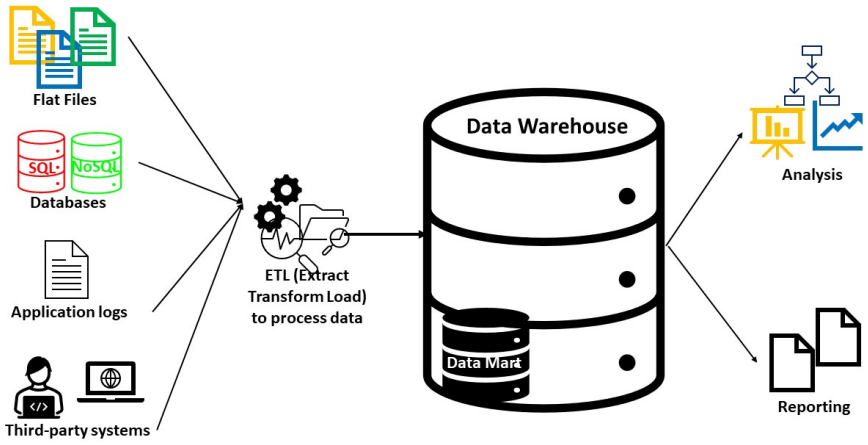

Figura 1.7: Schema di organizzazione di un Data Warehouse

Le architetture, quindi, sono di quattro tipologie principali:

- *Ad un livello:* tale architettura ha come obiettivo quello di minimizzare la memorizzazione di dati, eliminando le ridondanze. È una tipologia virtuale di DW, implementata come una vista multidimensionale dei dati. Essa non rispetta, però, la separazione tra l'elaborazione analitica e operazionale.
- *A due livelli:* si suddivide in quattro livelli distinti; avremo il livello sorgente con varie fonti di dati, il livello dell'alimentazione, con i processi classici di BI, il livello di DW, in cui vengono raccolte le informazioni; da quest'ultimo è possibile leggere direttamente i dati, oppure creare Data Mart, ovvero repliche parziali orientate verso specifiche aree del business. C'è, inoltre, accanto al DW, il repository dei meta-dati che mantiene informa- zioni sui meccanismi di accesso, sulle sorgenti, sugli utenti, etc. Infine troviamo il livello di analisi, che permette la consultazione efficiente dei dati integrati per simulazione, analisi e stesura di report.

- *A due livelli con Data Mart indipendenti:* in questo caso i Data Mart sono alimentati direttamente dalle sorgenti e, quindi, risultano indipendenti.
- *A tre livelli:* in questo caso viene introdotto un terzo livello a valle delle operazioni di Business Intelligence. Quest'ultimo materializza i dati operazionali ottenuti da processi di integrazione e ripulitura.

Data Lake

Un Data Lake è un'architettura di gestione dei dati che consente di memorizzare, gestire e analizzare grandi quantità di dati eterogenei in modo flessibile e scalabile. È un approccio molto moderno alla gestione dei dati, che si è sviluppato con l'avvento del Big Data e delle sfide associate alla gestione di grandi volumi di dati provenienti da diverse fonti. In un Data Lake i dati vengono memorizzati in un repository centralizzato, spesso basato su tecnologie di archiviazione distribuita. Questo repository è progettato per memorizzare dati in modo flessibile, senza la necessità di definire uno schema rigido in anticipo. Ciò consente di immagazzinare dati strutturati, semi-strutturati e non strutturati, come, ad esempio, dati provenienti da sensori, log di server, social media, immagini, video e molto altro, senza doverli trasformare o normalizzare in modo rigido prima di memorizzarli. Un altro aspetto importante del Data Lake è la scalabilità. I Data Lake sono progettati per gestire grandi volumi di dati e sono in grado di espandersi in modo orizzontale, ovvero possono essere usati in modo dinamico in base alle necessità di archiviazione e di elaborazione dei dati. Ciò consente alle organizzazioni di adattarsi facilmente ai cambiamenti nel volume e nella varietà dei dati, senza dover investire in infrastrutture costose o complesse. Inoltre, i Data Lake offrono un'ampia gamma di strumenti e servizi per l'elaborazione e l'analisi dei dati. Questi possono includere motori di elaborazione distribuita, servizi di analisi dei dati e strumenti di visualizzazione dei dati. Ciò consente agli utenti di eseguire una vasta gamma di operazioni sui dati all'interno del Data Lake stesso, senza dover spostare i dati in sistemi separati per l'elaborazione o l'analisi. Fondamentale per il Data Lake è però la gestione dei metadati, che se non è svolta in modo adeguato potrebbe trasformare il Data Lake in un Data Swamp, rendendone difficile, se non impossibile, l'utilizzo (Figura 1.8).

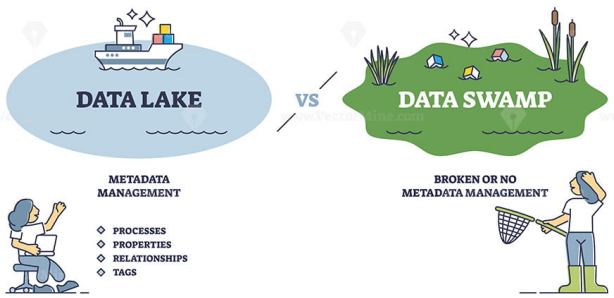

Figura 1.8: Data Lake e Data Swamp

1.4 Fasi della Business Intelligence

Come spiegato in precedenza, nella Business Intelligence sono impiegate figure specifiche e mezzi specifici. Questi, però, per poter essere utilizzati al meglio, hanno bisogno di regole,

o meglio, passi da seguire per poter arrivare, da un insieme di dati apparentemente non correlati, a delle informazioni che diano vantaggio competitivo. Esistono, quindi, delle fasi ben definite nei processi di BI. Queste sono:

- *Fase Zero*: comunicare con l'utente. In BI la figura dell'analista, deve essere in grado di interagire con le figure di tutti i dipartimenti aziendali, nonchè comprendere ed omologare l'uso di una terminologia che sia univoca e condivisibile con il resto dell'azienda, in modo che lo stesso concetto non sia espresso con termini diversi, evitando, così, di creare ambiguità.

- *Fase Uno*: comprendere la richiesta. Se il vocabolario dei termini aziendali non è ben definito, potrebbe essere molto complicato comprendere il bisogno del cliente che, di solito, conoscendo bene i propri processi, omette dei dettagli di fondamentale importanza poiché per lui risultano ovvi. In questo caso il lavoro potrebbe essere ostacolato da incomprensioni e potrebbe generare errori che allungheranno i tempi, e quindi i costi. Ecco perché risulta fondamentale sviluppare una terminologia chiara e condivisa tra i membri dell'associazione.

- *Fase Due*: identificare l'origine dei dati. Le fonti dei dati nelle aziende sono di vario genere; queste possono arrivare da basi di dati, fogli di calcolo, documenti di testo, e tante altre fonti. È di fondamentale importanza che l'analista BI conosca la provenienza dei dati che dovrà, poi, rielaborare.

- *Fase Tre*: elaborazione dei dati. Spesso il lavoro degli analisti BI viene riassunto con la parola "rielaborazione", termine troppo riduttivo per spiegare, in realtà, a cosa vengono sottoposti i dati estratti dal sistema informativo. Come spiegato in precedenza, i dati vengono memorizzati in basi di dati specifiche, dette Data Warehouse; quest'ultimo è il grande protagonista dietro ad ogni progetto di Business Intelligence. Infatti è in base alla sua qualità che dipende la buona riuscita di un progetto. È, quindi, questa la fase in cui si concentra la maggior parte del lavoro, attraverso quelli che sono i passi più famosi della BI, ovvero i passi di ETL, dall'acronimo inglese Extract, Transform and Load (Figura 1.9).

Figura 1.9: Processo di ETL

Obiettivo dell'estrazione è quello di unire tra loro dati provenienti da varie fonti, come, ad esempio, database, registri, report e altre attività transazionali. La fase di trasformazione è la più critica; infatti, in questa fase ai dati si applicano le regole aziendali necessarie a soddisfare i requisiti. Gli standard che guidano la trasformazione, al fine di garantire una certa qualità del dato, sono:

  – *standardizzazione*, ovvero definizione dei dati, della modalità di memorizzazione e altri fattori che definiranno le fasi successive del processo;

  - *deduplicazione*, che esclude o elimina dati ridondanti;
  - *verifica*, che mette a confronto informazioni simili al fine di eliminare ulteriori ridondanze;
  - *ordinamento*, che utilizza le regole di trasformazione per determinare come ogni singolo dato viene classificato e dove sarà collocato successivamente.

  Spesso, infatti, i processi di ETL hanno come output tabelle di aggregazione per report riepilogativi, lavoro che richiede un certo ordinamento e una successiva aggregazione dei dati. L'ultima fase del processo di ETL prevede il *caricamento* dei dati estratti e trasformati in quella che sarà una nuova destinazione, di solito il DW, tramite il caricamento completo o incrementale.

- *Fase Quattro:* visualizzare l'informazione. Gli utenti della BI hanno accesso solo ed esclusivamente al prodotto finale scaturito da questa fase; esso è rappresentato da un insieme di documenti che visualizzano a livello grafico, in diverse forme e modalità di accesso o interazione, dati da esso generati. Tale insieme di dati è chiamato, anche, reportistica. Da non sottovalutare l'importanza di presentare dei dati il più corretti possibile, nella maniera più intuitiva ed appetibile possibile, in modo tale che tutti possano comprendere i vantaggi legati all'adozione della BI. Se l'utente non capisse i dati e facesse delle scelte sbagliate, ciò andrebbe a discapito della propria azienda. La BI fornisce come output dati pronti per essere analizzati. I report servono solo a presentare i dati in un certo modo che risulti facile da comprendere. Il reporting può essere di 3 livelli, ovvero:

  - *Il livello operativo*; si tratta del livello più lontano dai vertici dell'azienda dove, solitamente, il dettaglio è molto alto e i grafici non sono strettamente necessari; i dati, quindi, vengono presentati sotto forma di tabelle o matrici.
  - *Il livello direzionale*; in questo caso c'è meno dettaglio di informazioni, ma la visione è più ampia; i grafici cominciano ad avere un ruolo importante e c'è la possibilità di mostrare previsioni sul futuro imminente.
  - *Il livello strategico*; il dettaglio è minimo. In questo caso viene utilizzato un diverso strumento della BI, ovvero il dashboarding. I dashboard (o cruscotti, in italiano) perdono livelli di dettaglio e presentano misure che riassumono, attraverso degli indicatori, l'andamento dell'intera azienda. Queste visualizzazioni possono essere statiche o dinamiche. Nel primo caso avremo uno shapshot non navigabile; di solito si trova in messaggi di e-mail inviati agli utenti finali; nel secondo caso, invece, c'è un certo livello di interattività con le informazioni, applicando filtri e agendo sul dettaglio di visualizzazione.

1.5 Stato dell'arte della BI con Gartner

Ad oggi sono molteplici gli strumenti che permettono alle aziende di praticare la Business Intelligence. Non tutti, però, hanno le stesse caratteristiche e permettono di prendere le stesse decisioni. Gartner è una multinazionale che si occupa di consulenza strategica, ricerca e analisi nel campo della tecnologia dell'informazione (Figura 1.10).

Ogni anno questa società produce i cosiddetti Magic Quadrant, o MQ, che hanno il compito di analizzare i principali player su mercato riguardanti un determinato settore o servizio, utilizzando dei metodi proprietari di analisi qualitativa, studiando le tendenze di mercato, come direzione, maturità e partecipanti. Graficamente un Magic Quadrant si presenta come un piano dove vengono collocate le principali aziende appartenenti al settore

Figura 1.10: Logo della Gartner S.p.A

che si sta esaminando. Un Magic Quadrant viene suddiviso in quattro categorie: Leader, Visionari, Giocatori di nicchia e Sfidanti. Un fornitore viene inserito in una delle 4 categorie. La semantica associata a questi ultimi è la seguente:

- *Leaders*: qui troviamo i competitor che hanno i punteggi più alti per quanto riguarda completezza di visione e capacità di esecuzione. Qui i fornitori dimostrano un'adeguata comprensione delle esigenze del mercato e hanno piani ben articolati che i clienti già possono utilizzare quando progettano le loro strategie. Inoltre, sono presenti nelle cinque principali aree geografiche del mondo e hanno capacità finanziarie ottime; pertanto, promettono un ottimo supporto della piattaforma.
- *Challengers o Sfidanti*: in quest'area troviamo i competitor che possono rappresentare una minaccia per i leader, grazie ai loro prodotti forti e alla loro continua crescita.
- *Visionari*: sono i competitor che forniscono i prodotti più innovativi, ma che non hanno ancora dimostrato di poter acquisire una fetta di mercato sostenibile poiché sono, spesso, società private con minori obiettivi di acquisizione.
- *Players di nicchia*: qui, i fornitori sono spesso concentrati strettamente su specifici segmenti di mercato, effettuando un'economia verticale. Il quadrante comprende fornitori che stanno adottando i loro prodotti per entrare nel mercato o fornitori maggiori che hanno difficoltà a sviluppare ed eseguire la loro visione.

1.5.1 I Magic Quadrant di BI e Cloud DBMS

Il contesto di sviluppo del progetto nella seguente tesi sarà prevalentemente su Google BigQuery, soluzione per la Business Intelligence associata ai Cloud DBMS. Tuttavia essendo il nostro progetto parte di un progetto più ampio avente anche una successiva componente su Microsoft Power BI, nel seguito esamineremo i Magic Quadrant relativi ai Cloud DBMS e alla Business Intelligence. Il primo di questi Magic Quadrant viene riportato nella Figura 1.11 mentre il secondo nella figura 1.12. Come si evince dalla figura 1.11, i leader di mercato nel settore di cloud DBMS sono: Amazon Web Services, Microsoft, Oracle e Google (BigQuery).

Invece, nella Figura 1.12 possiamo osservare che i leader di mercato per la Business Intelligence sono: Microsoft (Power BI), Salesforce (Tableau) e Qlik.

1.5.2 Vantaggi Business Intelligence

Conoscere tutti i benefici della BI non è possibile dato che il beneficio che possono trarre diverse aziende che operano in settori diversi non è lo stesso. Ci sono, però, dei vantaggi diretti di cui tutti possono godere. Questi sono:

- *Automatizzazione*. Non c'è più necessità, da parte del business, di prelevare e integrare manualmente i dati, operazione che, tra l'altro, è soggetta ad errori umani. Grazie agli strumenti della BI, il business è sollevato da queste scomode attività.

Figura 1.11: Magic Quadrant di Gartner per i Cloud DBMS

Figura 1.12: Magic Quadrant di Gartner per la Business Intelligence e Analytics

- *Flessibilità*. I Data Warehouse permettono di rimediare alle restrizioni imposte. Con gli strumenti odierni di visualizzazione ed analisi l'attività del business è semplificata e potenziata.

- *Puntualità*. Il business conduce ogni giorno le analisi prendendo decisioni su un dato che è costantemente aggiornato.
- *Produttività*. L'IT è sollevato da alcune attività potendosi, così, concentrare su attività produttive di creazione di contenuti.
- *Qualità dei dati*. Le informazioni usate dall'utente finale sono generate da procedure validate.
- *Indipendenza*. Con la BI a regime, il reparto IT e il business sono indipendenti, e ciascuno può concentrarsi sulle proprie attività per la maggior parte del tempo.
- *Sguardo al futuro*. Un progetto di BI forma i propri dipendenti e li rende più liberi di ricercare nuovi aspetti e problematiche prima ignorate, generando, così, miglioramenti nel business aziendale.

CAPITOLO 2

Introduzione a BigQuery

In questo capitolo verrà presentato lo strumento utilizzato per lo svolgimento del progetto relativo alla presente tesi, ovvero BigQuery di Google. In seguito verrà, anche, presentato Microsoft Power BI.

2.1 Google BigQuery

Google BigQuery (Figura 2.1) è un servizio di cloud computing offerto da Google Cloud Platform che consente di analizzare e gestire grandi quantità di dati in modo veloce ed efficiente. Si tratta di un sistema di archiviazione e analisi di dati completamente gestito, basato sul concetto di "data warehouse as a service", che permette di elaborare grandi quantità di dati senza doversi preoccupare della gestione dell'infrastruttura sottostante.

Figura 2.1: Il logo di Google BigQuery

Una delle caratteristiche distintive di Google BigQuery è la sua scalabilità;esso può gestire grandi quantità di dati, dall'ordine dei terabyte fino ai petabyte, consentendo di analizzare dati provenienti da diverse fonti in modo rapido e parallelo. Questa capacità di scalabilità permette alle aziende di gestire e analizzare grandi volumi di dati senza dover investire in costose infrastrutture hardware e di elaborazione dati.

Un'altra caratteristica di Google BigQuery è la sua velocità di elaborazione; infatti, grazie all'architettura distribuita e all'utilizzo di processori paralleli, BigQuery è in grado di eseguire query complesse su grandi quantità di dati in pochi secondi o minuti, a seconda delle dimensioni del dataset. Ciò consente agli utenti di ottenere risultati rapidi e immediati dalle loro query, permettendo loro di prendere decisioni informate in tempo reale.

Inoltre, Google BigQuey offre un'ampia gamma di strumenti per l'analisi dei dati. Supporta il linguaggio di query SQL standard, il che lo rende facile da usare per gli sviluppatori e gli analisti di dati, che sono già familiari con SQL. In aggiunta, BigQuery offre anche l'integrazione con altre tecnologie di analisi dei dati di Google, come Google Data Studio, per la creazione di report, e Google AI Platform per l'apprendimento automatico, consentendo di costruire soluzioni di analisi dei dati complete e personalizzate.

Un'altra caratteristica importante è la sua sicurezza; i dati archiviati in BigQuery sono crittografati in transito e a riposo, e BigQuery offre anche funzionalità avanzate di gestione degli accessi, come l'autenticazione basata su OAuth e l'accesso granulare ai dataset. BigQuery offre, inoltre, la possibilità di controllare gli accessi ai dati con regole di autorizzazione basate sui ruoli, consentendo di gestire in modo preciso chi ha accesso a quali dati. Infine, BigQuery è un servizio completamente gestito, il che significa che Google si occupa della gestione dell'infrastruttura sottostante, come la scalabilità, la disponibilità e la manutenzione del sistema. Ciò permette agli utenti di concentrarsi sull'analisi dei dati e sulla scoperta di informazioni significative senza dover dedicare tempo ed energie alla gestione dell'infrastruttura.

2.1.1 Interfaccia e Componenti

L'interfaccia di Google BigQuery è la piattaforma web che consente agli utenti di interagire con il servizio e di eseguire varie operazioni di gestione e analisi dei dati. Ecco una panoramica dei principali componenti dell'interfaccia di Google BigQuery:

- *Console di BigQuery:* la console di BigQuery è l'interfaccia principale che offre un'ampia gamma di funzionalità per la gestione dei dati. Qui è possibile creare e gestire dataset, tabelle e visualizzazioni, eseguire query SQL, caricare e scaricare dati, configurare autorizzazioni e monitorare le attività.
- *Editor di query:* l'editor di query è uno strumento potente per scrivere ed eseguire query SQL su dataset di BigQuery. Esso offre un'interfaccia utente intuitiva con evidenziazione della sintassi SQL, suggerimenti automatici, cronologia delle query e possibilità di visualizzare i risultati in diversi formati, come tabelle o grafici.
- *Explorer:* l'Explorer è una funzionalità che permette di esplorare il proprio dataset in modo interattivo. Si possono visualizzare le tabelle, i campi e i dati nel proprio dataset tramite una visualizzazione ad albero, consentendo di esplorare la struttura e il contenuto dei dati in modo intuitivo.
- *Monitoraggio e logging:* BigQuery offre, anche, funzionalità di monitoraggio e logging per consentire agli utenti di tenere traccia delle attività di query, dei tempi di esecuzione, dei costi e di altri dettagli delle attività di analisi dei dati. Ciò consente di monitorare le prestazioni e il costo delle query eseguite nel sistema.
- *Caricamento e scaricamento dei dati:* l'interfaccia di BigQuery permette di caricare e scaricare dati in vari formati, come CSV, JSON, Avro e altri, utilizzando strumenti di caricamento e scaricamento di dati incorporati. È, inoltre, possibile caricare dati da origini locali o da altre origini di dati di Google Cloud, come Google Cloud Storage e Google Sheets.

Nella Figura 2.2 troviamo la console di BigQuery; sulla sinistra vi è l'Explorer, nella zona centrale alta l'Editor e nella zona centrale bassa il risultato della query.

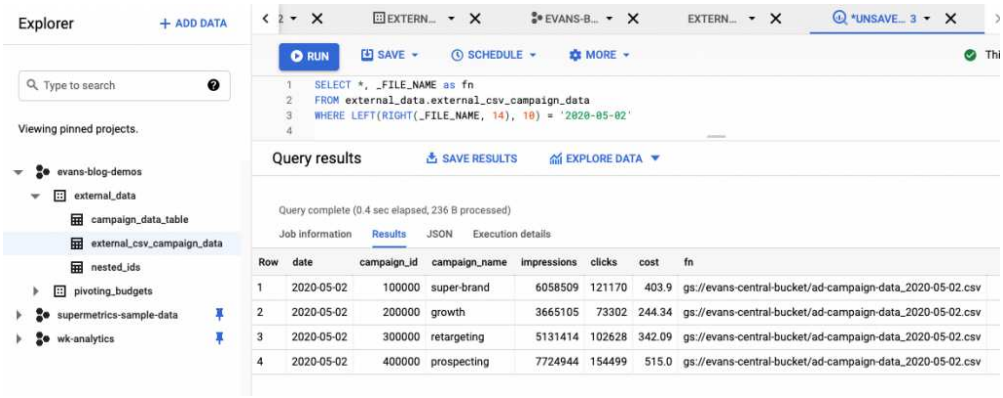

Figura 2.2: La console di Google BigQuery

2.1.2 Vantaggi e Tool

BigQuery offre diversi vantaggi che si possono riassumere in:

- *Importazione dati e condivisone:* possiamo trasmettere a BigQuery migliaia di righe al secondo, in modo da poter offrire un'analisi real time, condividendo istantaneamente con gli utenti.
- *Serverless:* non ci sono costi di manutenzione poiché la gestione dell'infrastruttura e la sicurezza dei dati sono garantite dall'azienda che fornisce il servizio, ovvero da Google.
- *SQL Standard:* la standardizzazione del linguaggio SQL evita alle aziende di dover riscrivere codice da zero. Queste aziende, già in possesso di database, scelgono di adottare un servizio cloud come Google BigQuery.
- *Backup e ripristino:* la posizione dei dati viene diversificata, salvando le informazioni in diversi data center, di cui esistono vari backup. In questo modo i dati sono al sicuro e il ripristino è semplice e veloce.
- *Data transfer service:* BigQuery offre la possibilità di trasferire automaticamente i dati da strumenti esterni, come Marketing Platform, Google Ads o YouTube.
- *Piattaforma per l'intelligenza artificiale:* BigQuery consente, grazie agli ultimi aggiornamenti, di testare algoritmi di machine learning.
- *Piattaforma per la Business Intelligence:* BigQuery permette, grazie alla trasformazione e all'analisi dei dati, di applicare la Business Intelligence.
- *Controllo dei costi:* per ogni azienda i costi sono personalizzabili in base alla tipologia di mercato in cui opera e al consumo che essa genera.

Direttamente dal sito ufficiale troviamo elencate le funzionalità di BigQuery; tra le più importanti troviamo i tool di:

- *Data QnA:* permette l'elaborazione del linguaggio naturale (Natural Language Processing, o NLP); utile per creare, ad esempio, chatbot.
- *BigQuery GIS:* combina l'architettura serverless con il supporto nativo per l'analisi geospaziale.
- *Integrazione programmatica:* offre un'API REST che facilita l'integrazione di applicazioni mettendo a disposizione diverse librerie, come Java, Python e Go.

2.1.3 SQL

SQL (Structured Query Language) è un linguaggio di programmazione utilizzato per gestire e interrogare database relazionali. È uno standard de facto per la gestione dei dati in diversi sistemi di gestione di database (DBMS), come, ad esempio, MySQL, PostgreSQL, SQLite, Oracle, Microsoft SQL Server, e molti altri.

SQL è progettato per consentire agli utenti di interagire con i dati all'interno di un database in modo efficiente ed efficace. Le sue principali funzionalità includono la capacità di creare, modificare ed eliminare tabelle, inserire, aggiornare ed eliminare dati, definire vincoli di integrità dei dati, creare viste per presentare i dati in modi diversi e interrogare i dati per ottenere informazioni specifiche.

Ecco alcuni concetti chiave di SQL:

- *Dichiarazione SQL*: le istruzioni SQL vengono scritte come dichiarazioni che vengono interpretate dal DBMS. Le dichiarazioni SQL più comuni includono `CREATE` (per creare tabelle, viste o indici), `INSERT` (per inserire dati in una tabella), `SELECT` (per interrogare i dati), `UPDATE` (per aggiornare i dati) e `DELETE` (per eliminare i dati).
- *Tabelle*: una tabella è una struttura di dati organizzata in righe e colonne, simile ad un foglio di calcolo. Ogni colonna corrisponde ad un attributo e ogni riga corrisponde ad un record. Le tabelle vengono utilizzate per organizzare e archiviare i dati in un database.
- *Interrogazioni*: le interrogazioni sono comandi SQL utilizzati per ottenere dati specifici da una o più tabelle. Esse possono includere condizioni, filtri, ordinamenti e altre operazioni per manipolare i dati.
- *Vincoli di integrità*: i vincoli di integrità vengono utilizzati per garantire la coerenza e la validità dei dati all'interno di una tabella. Ad esempio, i vincoli di chiave primaria assicurano che ogni riga in una tabella abbia un valore univoco per una colonna specifica, mentre i vincoli di chiave esterna stabiliscono relazioni tra tabelle.
- *Transazioni*: le transazioni sono utilizzate per garantire l'integrità dei dati in un database. Una transazione può includere una o più operazioni SQL, come inserimenti, aggiornamenti o eliminazioni, che vengono eseguite in modo atomico, cioè o vengono eseguite tutte o nessuna. In caso di errore o fallimento, una transazione può essere annullata (rollback) per ripristinare lo stato del database a quello precedente all'inizio della transazione, oppure può essere confermata (commit) per rendere permanenti le modifiche apportate.

2.2 Microsoft Power BI

Microsoft Power BI (Figura 2.3) è uno strumento di Business Intelligence e visualizzazione dei dati sviluppato da Microsoft. Power BI consente agli utenti di connettersi, trasformare, analizzare e visualizzare dati provenienti da diverse fonti, offrendo insight e informazioni utili per prendere decisioni basate sui dati.

Ecco alcuni dei principali elementi di Microsoft Power BI:

- *Connessione ai dati:* Power BI può connettersi ad una vasta gamma di fonti dati, tra cui database, servizi cloud, file locali, API. Ciò consente agli utenti di consolidare dati provenienti da diverse fonti in un unico dashboard.

Figura 2.3: Il logo di Microsoft PowerBI

- *Trasformazione dei dati.* Power BI offre strumenti intuitivi per la trasformazione dei dati che consentono di pulire, combinare, filtrare e trasformare i dati in modo da adattarli alle esigenze di analisi.
- *Creazione di report interattivi:* Power BI consente di creare report interattivi con una vasta gamma di visualizzazioni, come grafici, tabelle, mappe e altro ancora. Questi report possono essere personalizzati per soddisfare le esigenze specifiche di analisi e possono essere esplorati in modo interattivo per scoprire insight nascosti.
- *Creazione di dashboard:* Power BI permette di creare dashboard interattive (Figura 2.4) che offrono una panoramica visiva dei principali indicatori di performance (KPI) e delle metriche aziendali. Le dashboard possono essere condivise con i membri del team o con gli utenti esterni, consentendo una visualizzazione in tempo reale delle informazioni chiave.
- *Collaborazione e condivisione:* Power BI offre funzionalità di collaborazione che permettono di lavorare in team, condividere report e dashboard con altri utenti e collaborare nella creazione di analisi e report.
- *Integrazione con altre applicazioni Microsoft:* Power BI si integra con altre applicazioni Microsoft, tra cui Azure, Excel, Teams, consentendo una sinergia tra le diverse piattaforme e un flusso di lavoro senza soluzione di continuità.
- *Sicurezza:* Power BI offre funzionalità avanzate di sicurezza per proteggere i dati sensibili, tra cui l'autenticazione basata su ruoli, la crittografia dei dati e l'integrazione con le politiche di sicurezza aziendali.

2.2.1 I Componenti Chiave

I componenti chiave dell'ecosistema Power BI sono:

- *Desktop Power BI:* si tratta di un' applicazione basata su desktop; Windows per PC e desktop, essa serve principalmente per la progettazione e la pubblicazione di report per il Servizio.
- *Servizio PowerBI:* si tratta di un servizio online basato su SaaS (Software as a Service). Questo, precedentemente noto come Power BI per Office 365, ora è denominato PowerBI.com o, semplicemente, Power BI.
- *App per dispositivi mobili Power BI:* le app Power BI sono disponibili per dispositivi mobili Android o iOS, nonché per telefoni e tablet Windows.

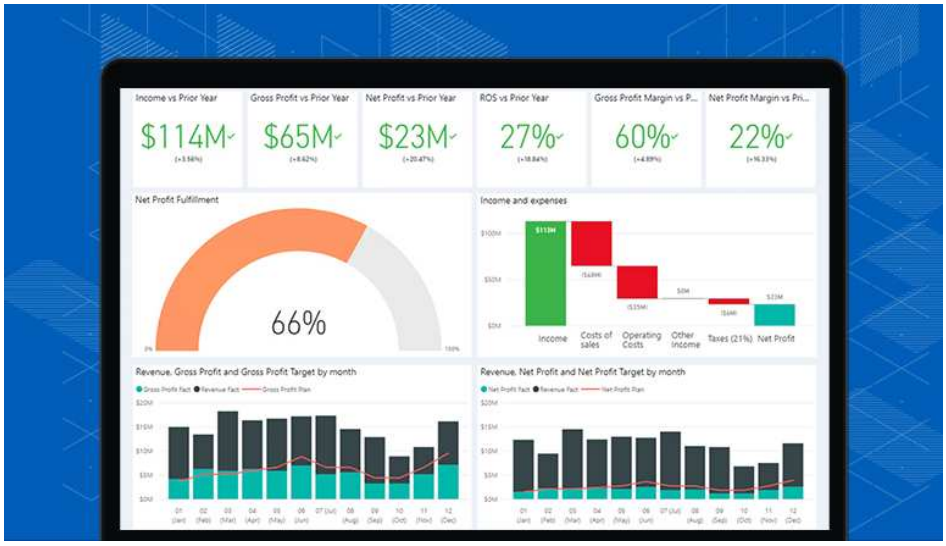

Figura 2.4: Dashboard creata con Power BI

- *Gateway di Power BI:* i gateway vengono usati per sincronizzare i dati esterni in entrata e in uscita da Power BI; essi sono necessari per gli aggiornamenti automatici. In modalità Enterprise possono essere utilizzati anche da Power Automate (precedentemente denominato Flows) e PowerApps in Office 365.
- *Power BI integrato:* l'API REST di Power BI può essere usata per creare dashboard e report nelle applicazioni personalizzate che servono sia gli utenti Power BI e sia gli utenti non Power BI.
- *Server di report di Power BI:* si tratta di una soluzione di reporting di Power BI locale per le aziende che non archiviano o non possono archiviare i dati nel servizio Power BI basato su cloud.
- *Power BI Premium:* si tratta di un'offerta basata sulla capacità che include la flessibilità di pubblicare report su vasta scala in tutta l'azienda, senza, inoltre, richiedere ai destinatari di ottenere licenze individuali per utente; fondamentalmente offre scalabilità e prestazioni maggiori rispetto alla capacità condivisa nel servizio Power BI.
- *Marketplace di elementi visivi di Power BI:* si tratta di un marketplace di elementi visivi personalizzati ed elementi visivi basati su R per dare vita ai dati aziendali ed estrapolare informazioni da questi.
- *Flusso di dati di Power BI:* si tratta di un'implementazione di Power Query nel cloud che può essere utilizzata per le trasformazioni dei dati volte a creare un set di dati Power BI comune; esso può essere reso disponibile per diversi sviluppatori di report tramite Common Data Service di Microsoft. Può essere utilizzato come alternativa, ad esempio, all'esecuzione di trasformazioni in SSAS e può garantire che diversi sviluppatori di report utilizzino dati che sono stati trasformati in modo simile.
- *Set di dati di Power BI:* un set di dati di Power BI può funzionare come una raccolta di dati da utilizzare nei report di Power BI e può essere connesso o importato in un report di Power BI. Un set di dati può essere connesso e ottenere i dati di origine tramite uno o più flussi di dati.
- *Datamart di Power BI:* all'interno di Power BI, Datamart è un contenitore che combina flussi di dati Power BI, set di dati e un tipo di data mart o data warehouse (sotto

forma di un database SQL di Azure) nella stessa interfaccia. L'interfaccia ha, quindi, la possibilità di essere un unico luogo per l'amministrazione sia del livello ETL (Dataflow), cioè un data mart intermediario (con ad esempio l'archiviazione di schemi a stella, tabelle dimensionali, tabelle dei fatti), sia del livello di modellazione (Dataset) .

- *Hub dati di Power BI:* si tratta di un hub di dati per l'individuazione dei set di dati di Power BI all'interno del servizio Power BI di un'organizzazione in modo tale che i set di dati possano essere riutilizzati da una posizione centrale.

CAPITOLO 3

Contesto di Riferimento e Analisi dei Requisiti

In questo capitolo analizzeremo il contesto di riferimento in cui la presente tesi è stata sviluppata. In particolare descriverà il progetto in generale, inquadrando il cliente, l'architettura e come questa viene alimentata.

3.1 Il Cliente

Al gruppo Filippetti, sede di svolgimento del tirocinio, è stato commissionato il progetto, oggetto della presente, da parte di un grande gruppo internazionale che opera nel settore della moda di lusso. Non possiamo riferire il nome di tale gruppo per ragioni commerciali; nel seguito lo chiameremo usando le pseudonimo "Society".

3.1.1 Identità dell'Azienda

Society è una S.A. (Società Anonima) ed è un tipo di società di capitali in cui il capitale sociale è diviso in azioni. A differenza di altre forme di società, come le società a responsabilità limitata (S.r.l.), le azioni di una S.A. possono essere acquistate e vendute liberamente sul mercato.

L'accezione "anonima" deriva dal fatto che i proprietari delle azioni non sono necessariamente noti al pubblico in generale. In altre parole, l'identità degli azionisti può essere tenuta anonima, anche se in molti casi è richiesto che venga mantenuto un registro degli azionisti.

Una S.A. ha una struttura organizzativa complessa, con un consiglio di amministrazione e un'assemblea degli azionisti che si riuniscono regolarmente per prendere decisioni importanti sulla gestione della società. Inoltre, la responsabilità dei soci è limitata al capitale investito, il che significa che gli azionisti non rispondono personalmente per le obbligazioni della società al di là dell'investimento iniziale.

Questo tipo di società è spesso utilizzato come forma di organizzazione per le grandi imprese, poiché consente di raccogliere grandi quantità di capitale e di attirare investimenti da parte di un gran numero di azionisti. Tuttavia, a causa della complessità della loro struttura organizzativa e delle regole che governano il loro funzionamento, le S.A. sono soggette a un rigoroso controllo e a una rigorosa regolamentazione da parte delle autorità competenti.

Society è quotata sul mercato Euronext (Figura 3.1), ossia il principale mercato finanziario e borsa valori pan-europeo nell'Eurozona, con più di 1.300 titoli quotati per un valore di circa 3.600 miliardi di euro di capitalizzazione di mercato a fine dicembre 2017, di cui un paniere

di blue chips senza eguali composto da 25 titoli nell'indice Euro Stoxx 50 e una forte base diversificata nazionale e internazionale di clienti.

Figura 3.1: Il logo di Euronext

3.1.2 I Dividendi

Society ha una politica di distribuzione dei dividendi stabilita dal suo consiglio di amministrazione, che può variare nel tempo a seconda delle condizioni finanziarie e delle strategie aziendali. Ecco come Society paga i dividendi:

- *Dichiarazione dei dividendi:* il consiglio di amministrazione di Society dichiara l'importo dei dividendi e la data di registrazione, che è la data in cui gli azionisti registrati saranno considerati idonei a ricevere i dividendi. La dichiarazione dei dividendi di Society viene generalmente effettuata durante l'assemblea generale annuale degli azionisti, nella quale gli stessi possono votare per l'approvazione della distribuzione dei dividendi proposta.
- *Calcolo dei dividendi:* l'importo dei dividendi di Society è determinato dal consiglio di amministrazione in base alle politiche di distribuzione dei profitti e alle esigenze finanziarie dell'azienda. L'importo dei dividendi per azione è calcolato dividendo l'importo totale dei dividendi per il numero di azioni in circolazione. Ad esempio, se Society dichiara un dividendo totale di 100 milioni di euro e ha 200 milioni di azioni in circolazione, l'importo del dividendo per azione sarà di 0,50 euro (100 milioni di euro / 200 milioni di azioni).
- *Registro dei dividendi:* Society stabilisce una data di registrazione, che è la data in cui viene stabilito il registro degli azionisti che saranno considerati idonei a ricevere i dividendi. Gli azionisti registrati sono coloro che detengono azioni della società prima della data di registrazione. Se si acquistano azioni di Society dopo la data di registrazione, non si avrà diritto a ricevere i dividendi dichiarati.
- *Data di pagamento dei dividendi:* Society stabilisce una data di pagamento, che è la data effettiva in cui i dividendi saranno distribuiti agli azionisti registrati. Questa è la data in cui Society effettua il pagamento dei dividendi in contanti o assegna le azioni aggiuntive ai beneficiari. La data di pagamento può essere diversa dalla data di registrazione e viene annunciata dalla società.
- *Modalità di pagamento:* Society è solita pagare i dividendi in contanti, addebitandoli sul conto corrente degli azionisti registrati, o attraverso altre modalità di pagamento come, ad esempio, l'emissione di assegni o bonifici bancari. In alcuni casi, se previsto dalla politica di distribuzione dei dividendi della società, Society potrebbe offrire anche l'opzione di ricevere i dividendi sotto forma di azioni aggiuntive invece di contanti.
- *Impatto sui prezzi delle azioni:* l'annuncio e il pagamento dei dividendi possono influenzare i prezzi delle azioni di Society, come accennato in precedenza. In genere, quando la società annuncia un dividendo, il prezzo delle azioni tende a diminuire di un importo

pari all'importo del dividendo dichiarato, poiché il valore del dividendo viene "estratto" dal prezzo delle azioni. Una volta che il dividendo è stato pagato, il prezzo delle azioni potrebbe subire variazioni in base alle dinamiche di mercato e alle performance finanziarie dell'azienda.

È importante considerare che la distribuzione dei dividendi dipende dalle politiche e dalle decisioni del consiglio di amministrazione di Society, che può decidere di modificare o sospendere la distribuzione dei dividendi in base alle condizioni finanziarie dell'azienda, alle esigenze di investimento o ad altre considerazioni strategiche.

Inoltre, è fondamentale sottolineare che la distribuzione dei dividendi è riservata agli azionisti registrati alla data di registrazione: se si acquistano azioni di Society dopo la data di registrazione, non si avrà diritto a ricevere i dividendi dichiarati; dunque, è importante fare attenzione alle date di registrazione e di pagamento dei dividendi se si è interessati a riceverli come azionista di Society o di qualsiasi altra società quotata in borsa.

Infine, è sempre consigliabile consultare le informazioni ufficiali fornite dalla società, come i comunicati stampa, i rapporti finanziari e i documenti presentati all'assemblea generale degli azionisti, per avere una comprensione accurata e aggiornata sulle politiche di distribuzione dei dividendi di Society o di qualsiasi altra società in cui si è interessati ad investire.

3.2 Analisi dei Requisiti

Il progetto in questione, facente parte di un progetto più ampio, verrà presentato prima nella totalità, entro i limiti imposti dalla privacy, per poi aumentare il grado di dettaglio nei successivi capitoli, relativamente al lavoro svolto nel tirocinio.

3.2.1 Contesto del Progetto

Negli ultimi anni, il dipartimento di controllo finanziario del gruppo ha lanciato un programma di digitalizzazione e automazione di tutti i processi finanziari che in origine erano svolti manualmente tramite Microsoft Excel; tra questi processi troviamo il calcolo dei dividendi. Il calcolo dei dividendi è un'operazione onerosa, non dal punto di vista dei calcoli o delle formule da applicare, ma a causa dell'enorme enorme quantità di dati in gioco e della loro etereogeneità. In origine il processo manuale (Figura 3.2) consisteva semplicemente, nei seguenti passi:

- *Ricezione dei dati.* Esportazione manuale dei dati dai team delle Risorse Umane.
- *Trattamento dei dati.* Applicazione dei calcoli all'interno di un modello Excel.
- *Condivisione dei risultati.* Trasmissione manuale (e-mail) dei risultati ai diversi soggetti interessati.

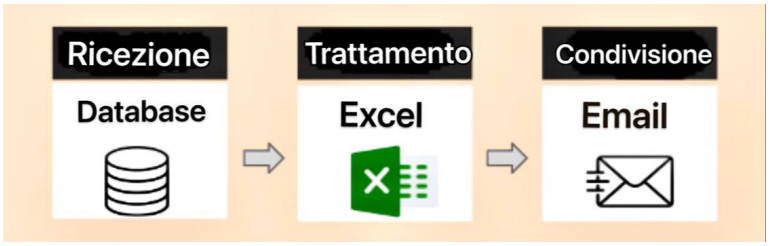

Figura 3.2: Schema del processo esistente manuale

3.2.2 Obiettivi del Progetto

L'obiettivo del macro progetto è l'ottenimento dell'automatizzazione del processo (Figura 3.3) come segue:

- *Ricezione e trattamento dei dati.* Export e calcoli automatizzati tramite BigQuery.
- *Analisi dei risultati.* Visualizzazioni su Power BI.
- *Export dei risultati.* Export dei dati ad-hoc tramite Excel.
- *Condivisione di risultati.* Condivisione tramite Oracle Hyperion Financial Management.

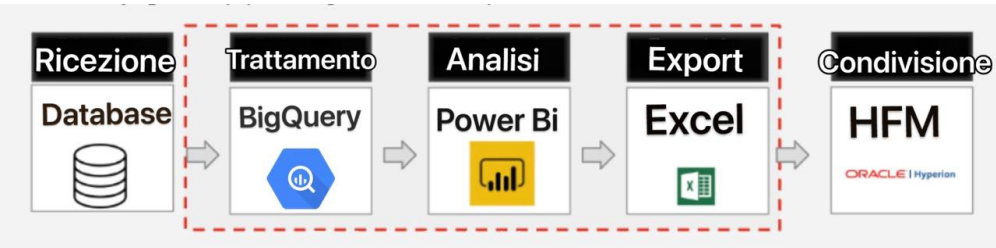

Figura 3.3: Schema del processo automatizzato

La parte di *Trattamento dei dati* è ulteriormente scissa su 4 livelli in sequenza, in cui i primi tre si occupano della fase di ETL, trasformando informazioni non strutturate e non organizzate in dati utili e significativi, mentre l'ultimo livello, cioè Usage, è quello nel quale avviene il calcolo dei dividendi e che verrà trattato in dettaglio nella presente tesi. Più specificamente i quattro livelli sono i seguenti:

- *Raw employ.* L'estrazione di dati grezzi, ossia la prima fase del processo di elaborazione dei dati, consiste nel raccogliere informazioni in formato digitale da diverse fonti, come ad esempio database, file CSV, siti web, sensori, etc. Durante questa fase, i dati sono raccolti nella loro forma originale e non sono ancora strutturati oppure organizzati in alcun modo.
- *Normalizated.* La normalizzazione dei dati è la fase successiva durante la quale i dati grezzi vengono trasformati in un formato standard, uniforme e coerente. Ciò significa che i dati vengono organizzati in tabelle con campi specifici e con valori che seguono regole ben definite; ad esempio, se si raccolgono dati da differenti fonti, potrebbe essere necessario normalizzarli per assicurarsi che i valori siano compatibili e comparabili; in questo modo si garantisce l'accuratezza e la coerenza dei dati stessi, che diventano facilmente accessibili e interpretabili.
- *Aggregated.* L'aggregazione dei dati è la fase finale del processo di elaborazione dei dati, durante la quale i dati stessi vengono combinati, filtrati e organizzati in modo da generare informazioni più significative e di valore. In questa fase, i dati vengono analizzati per individuare eventuali tendenze, pattern o relazioni, e vengono utilizzati per generare indicatori, statistiche, report e dashboard; ad esempio, i dati raccolti su una certa popolazione possono essere aggregati in modo da fornire informazioni sulla media di età, di reddito o di istruzione della popolazione stessa.
- *Usage.* La fase di usage prevede l'applicazione delle formule per il calcolo dei dividendi; si parte dai dati preprocessati nei livelli sottostanti, pronti all'utilizzo e si arriva all'output finale che poi verrà usato per alimentare PowerBI. Tale fase verrà affrontata in dettaglio nei prossimi capitoli.

3.2.3 Sorgenti d'Alimentazione

Nella trattazione delle sorgenti d'alimentazione, per motivi di privacy, non si può scendere troppo nel dettaglio, soprattutto per quanto riguarda i database; ci limiteremo nel seguito a un elenco generale. Il livello *usage* è alimentato da diverse sorgenti di due tipologie:

- *Database*
- *File Excel*

La principale differenza tra le due tipologie è la dimensione; i file excel sono di molti ordini di grandezza inferiori rispetto ai database.

Database

I dati provenienti dai database passano, prima di arrivare al livello usage, per i tre livelli precedentemente esposti (Raw employ, Normalized e Aggregated). I database in questione sono quattro:

- `MRP_History_new`. Contiene i dati storici delle aziende in possesso di Society.
- `Consolidation_Report_new`. Contiene tutte le informazioni utili per la costituzione delle disposizioni di remunerazione.
- `Workday_exception`. Contiene le aziende in possesso di Society le cui informazioni sulla retribuzione non devono comparire nelle analisi.
- `Cash_Out_new`. Contiene i dati relativi a tutti i pagamenti di Society alle sue aziende.

File Excel

I file in questione sono quattro:

- `Plan_Value`. Contiene dati relativi a diversi tipi di piano di investimento (Figura 3.4).
- `TCS`. Contiene dati relativi alla tassazione dei diversi paesi (Figura 3.5).
- `Table_Assumptions`. Contiene dati sull'evoluzione dei diversi piani di investimento negli anni (Figura 3.6).
- `HRA_Missing_Beneficiaries`. Contiene dati relativi agli investitori che devono essere retribuiti ma non lo sono ancora stati (Figura 3.7).

Field name	Type	Mode
Plan_Type	STRING	NULLABLE
Brand	STRING	NULLABLE
Valorization_Date	DATE	NULLABLE
Last_value	FLOAT	NULLABLE

Figura 3.4: I campi di Plan_Value

Field name	Type	Mode
Beneficiary_ID	STRING	NULLABLE
Country	STRING	NULLABLE
Social_Charges_for_KPS	FLOAT	NULLABLE
Social_Charges_for_KMU_BMU_CP	FLOAT	NULLABLE

Figura 3.5: I campi di TCS

Field name	Type	Mode
LTI_Plan	STRING	NULLABLE
LTI_Year	INTEGER	NULLABLE
Brand	STRING	NULLABLE
Annual_turnover_percentage	FLOAT	NULLABLE
Annual_growth_percentage_LTI_Year	FLOAT	NULLABLE
Annual_growth_percentage_LTI_Year_plus_1	FLOAT	NULLABLE
Annual_growth_percentage_LTI_Year_plus_2	FLOAT	NULLABLE
Annual_growth_percentage_LTI_Year_plus_3	FLOAT	NULLABLE

Figura 3.6: I campi di Table_Assumptions

Field name	Type	Mode
Beneficiary_ID	STRING	NULLABLE
Origin	STRING	NULLABLE
Year	INTEGER	NULLABLE
Month	STRING	NULLABLE
Date	DATE	NULLABLE
MRP_Code	STRING	NULLABLE
MRP_Description	STRING	NULLABLE
Country	STRING	NULLABLE
Brand	STRING	NULLABLE

Figura 3.7: I campi di HRA_Missing_Beneficiaries

CAPITOLO 4

Progettazione della Componente Applicativa

In questo capitolo descriveremo il processo di progettazione logica del sistema su Google BigQuery, illustrando l'intero processo di trasformazione dei dati e le tabelle associate; verranno successivamente presentate le viste che lo compongono. Nell'ultima parte verrà presentata la metodologia Agile.

4.1 Workflow

In informatica, il termine "workflow", ossia flusso di lavoro, si riferisce a una sequenza di attività o passaggi che vengono eseguiti per completare un processo o un compito. In altre parole, un workflow è un insieme di regole, procedure o azioni che definiscono come vengono gestiti i dati e i documenti all'interno di un'organizzazione o di un sistema informatico.

I workflow sono spesso utilizzati per automatizzare i processi aziendali e migliorare l'efficienza operativa, poiché permettono di standardizzare le attività e di gestire le interazioni tra i diversi sistemi, applicazioni e utenti coinvolti nel processo. Inoltre, i workflow possono essere utilizzati per monitorare e gestire le attività in tempo reale, fornendo una maggiore trasparenza e visibilità sulle operazioni aziendali.

La digitalizzazione è avvenuta scomponendo l'intero processo in sottoprocessi, ognuno dei quali realizzato tramite una query SQL; ad ogni query SQL è associata la relativa vista. Per comprendere meglio spieghiamo ora la differenza tra query e vista (Figura 4.1):

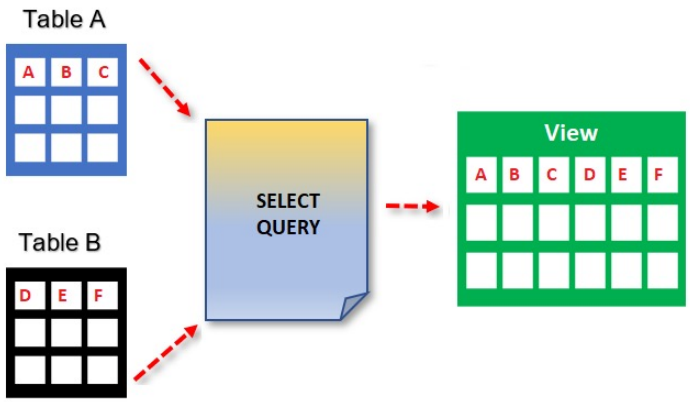

Figura 4.1: Schema della relazione tra query e vista

- Una *vista* (o "view" in inglese) è una rappresentazione virtuale di una tabella o di una combinazione di tabelle in un database. Una vista non contiene dati effettivi, ma fornisce un'interfaccia logica per accedere ai dati memorizzati nelle tabelle sottostanti. In altre parole, si tratta di una finestra attraverso la quale si può vedere una parte specifica dei dati in un database, senza dover accedere direttamente alle tabelle.

- Una *query*, d'altra parte, è una richiesta di informazioni effettuata su un database. Una query può essere utilizzata per selezionare, filtrare e ordinare i dati all'interno di una o più tabelle in base a determinati criteri. In pratica, una query è un modo per estrarre informazioni specifiche dal database.

In sintesi, la differenza principale tra vista e query è che la prima è una rappresentazione virtuale dei dati di una o più tabelle in un database, mentre la seconda è una richiesta specifica per estrarre dati da una o più tabelle. La vista è utilizzata per semplificare l'accesso ai dati e migliorare l'efficienza delle query, mentre la query è utilizzata per ottenere informazioni specifiche dal database.

Nel nostro progetto avremo principalmente due tipi di query: alcune che semplicemente uniscono dati provenienti da diverse fonti mentre altre, più elaborate, in cui avvengono i calcoli.

Nella Figura 4.2 troviamo il worflow, per come è stato impostato logicamente, vedremo nel capitolo successivo che ci sono state delle modifiche che verranno giustificate. Il workflow in Figura 4.2 è composto da una successione di viste, le quali sono alimentate dalle sorgenti dati già presentate o da altre viste.

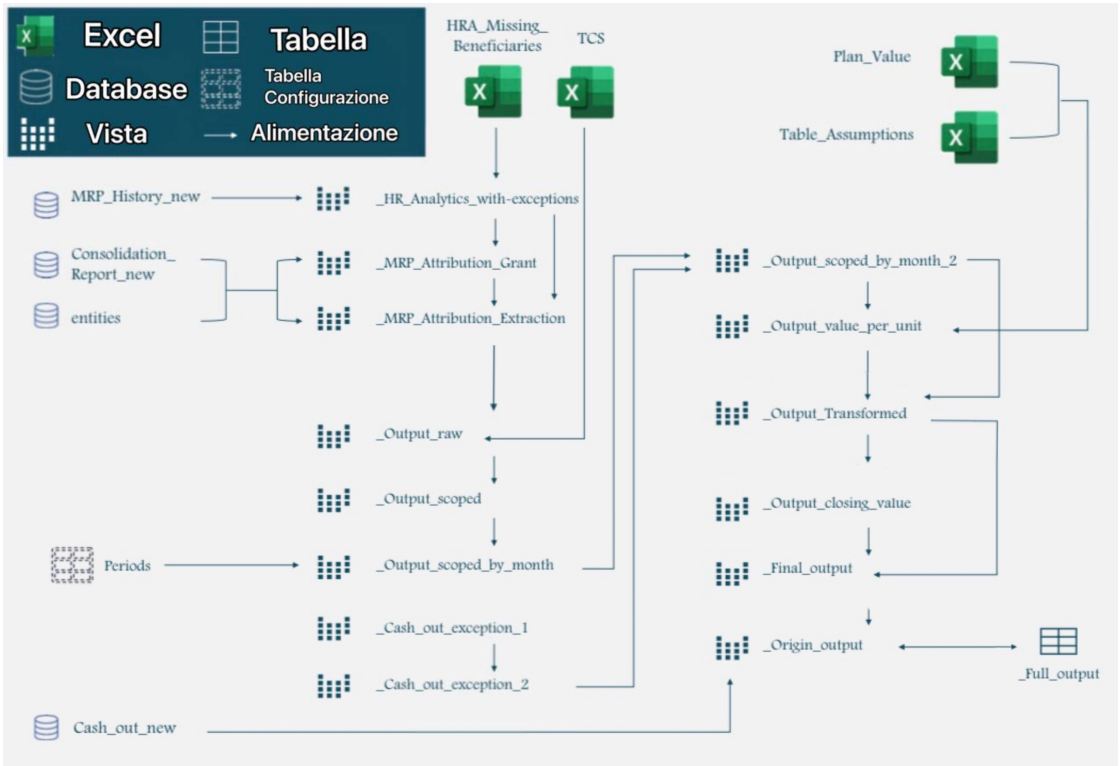

Figura 4.2: Il processo di trasformazione dei dati e delle tabelle associate

4.2 Viste

In questa sezione presentiamo le viste, per le quali non possiamo scendere nei dettagli o nelle effettive implementazioni a causa dei limiti imposti da ragioni commerciali:

- `_HR_Analytics_with-exceptions.`

  Questa vista unisce la tabella *MRP_History_new* presente nel dataflow di origine, cioè *"Aggregated"* dati, e il file *HRA_Missing_Beneficiaries*.

- `_MRP_Attribution_Extraction & _MRP_Attribution_Grant.`

  Queste due viste si occupano delle assegnazioni dei codici MRP, delle entità legali e dei codici dei relativi paesi.

- `_Output_raw.`

  Questa vista estrae le colonne utili dalla vista precedente e le associa con il file *TCS*.

- `_Output_scoped.`

  Questa vista genera, per ogni riga della vista precedente, la proiezione temporale nei successivi 6 anni.

- `_Output_scoped_by_month.`

  Questa vista scompone ogni linea di ogni anno generato precedentemente, in dodici linee corrispondenti ai dodici mesi dell'anno tramite l'utilizzo della tabella di configurazione *Periods*, che è semplicemente un elenco dei dodici mesi.

- `_Cash_out_exception_1 & _Cash_out_exception_2.`

  Queste due viste intercettano i dati che formano le eccezioni in *Consolidation_Report_new* per elaborare i vecchi piani e reinserirli.

- `_Output_scoped_by_month_2.`

  Questa vista unisce le eccezioni individuate sopra al sistema.

- `_Output_value_per_unit.`

  Questa vista unisce i due file *Plan_Value & Table_Assumptions* per aggiungere informazioni relative ai piani d'investimento, come, ad esempio, il tasso di crescita.

- `_Output_transformed.`

  Questa vista calcola una metà degli indicatori che serviranno per i calcoli finali.

- `_Output_closing_value.`

  Questa vista calcola gli stessi indicatori, ma dopo un anno esatto.

- `_Final_output.`

  Questa vista calcola i restanti indicatori, quindi conterrà tutti gli indicatori.

- `_Origin_output.`

  Questa è la vista finale dove avvengono i calcoli di P&L (Profit & Loss, cioè profitto e perdita). I dati finali, generati da questa vista ogni 6 mesi, vengono poi storicizzati nella tabella *_Full_output*: quest'ultima quindi conterrà i calcoli applicati all'intero database; essa ha la stessa struttura di *_Origin_output* ma con due colonne aggiuntive che rappresentano il tag temporale. Da questa tabella vengono anche ripresi iterativamente i dati per il calcolo dei 6 mesi successivi.

4.3 Metodologia Agile

L'intero progetto è stato portato avanti utilizzando un approccio flessibile e collaborativo per lo sviluppo del software e la gestione dei progetti, chiamato Agile. Agile project management (gestione di progetto Agile) è un termine generico per i metodi di sviluppo che adottano un approccio incrementale e iterativo; anche se ha avuto origine nello sviluppo del software, e si trova ancora principalmente in questo ambiente, i suoi principi possono essere applicati ad altre discipline.

Agile è progettato per adattarsi ai cambiamenti rapidi, favorire la comunicazione e promuovere la consegna di valore in modo incrementale. La metodologia Agile è stata introdotta per superare le limitazioni delle metodologie di sviluppo tradizionali, come il modello a cascata, che spesso si concentrano su una pianificazione dettagliata e su una consegna finale (Figura 4.3).

Ecco i principi chiave della metodologia Agile:

- *Individui e interazioni al di sopra ai processi e agli strumenti.* L'Agile mette l'accento sulla comunicazione e sulla collaborazione tra i membri del team. Le interazioni faccia a faccia sono considerate più efficaci rispetto alla documentazione eccessiva o alle procedure rigide.
- *Software funzionante al di sopra alla documentazione esaustiva.* Invece di concentrarsi sulla creazione di documenti dettagliati, l'Agile favorisce la consegna di software funzionante in tempi brevi. Ciò consente al team di ricevere feedback tempestivi e di apportare modifiche in modo iterativo.
- *Collaborazione con il cliente al di sopra alla negoziazione dei contratti.* La metodologia Agile pone un'importanza significativa sulla collaborazione con il cliente. Il coinvolgimento attivo del cliente durante tutto il processo di sviluppo aiuta il team a comprendere meglio le esigenze e a soddisfare le aspettative in modo più accurato.
- *Rispondere al cambiamento al di sopra al seguire un piano.* L'Agile è flessibile e si adatta ai cambiamenti. Riconosce che i requisiti e le priorità possono cambiare nel corso del progetto e promuove la capacità di adattamento rapido del team.

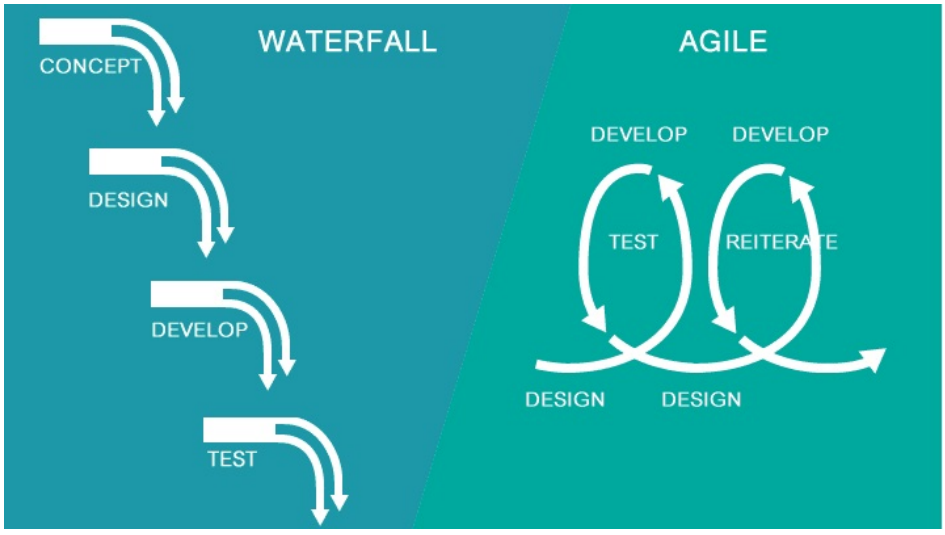

Figura 4.3: I due tipi di project managment: a cascata e Agile

La metodologia Agile utilizza cicli di sviluppo iterativi ed incrementali noti come "sprint" (Figura 4.4). Ecco i passaggi chiave:

- *Pianificazione.* Il team si incontra per definire gli obiettivi del prossimo sprint e pianificare le attività necessarie per raggiungerli.
- *Analisi dei requisiti.* Il team collabora con il cliente per comprendere i requisiti specifici delle funzionalità da sviluppare nel prossimo sprint.
- *Sviluppo.* Il team di sviluppo lavora sulle attività pianificate per lo sprint, seguendo cicli di lavoro di solito di 1-4 settimane.
- *Revisione e feedback.* Alla fine di ogni sprint, viene organizzata una revisione in cui il team presenta al cliente il lavoro completato. Questo momento è importante per ricevere feedback, valutare i risultati e apportare modifiche o miglioramenti.
- *Retrospettiva.* Dopo la revisione, il team riflette sul processo di sviluppo dello sprint e identifica punti di forza e di debolezza. Inoltre, engono discussi i miglioramenti da apportare per i prossimi sprint.

Questi passaggi si ripetono iterativamente per tutta la durata del progetto, consentendo al team di adattarsi ai cambiamenti, di focalizzarsi sul valore e di consegnare software funzionante ad intervalli regolari.

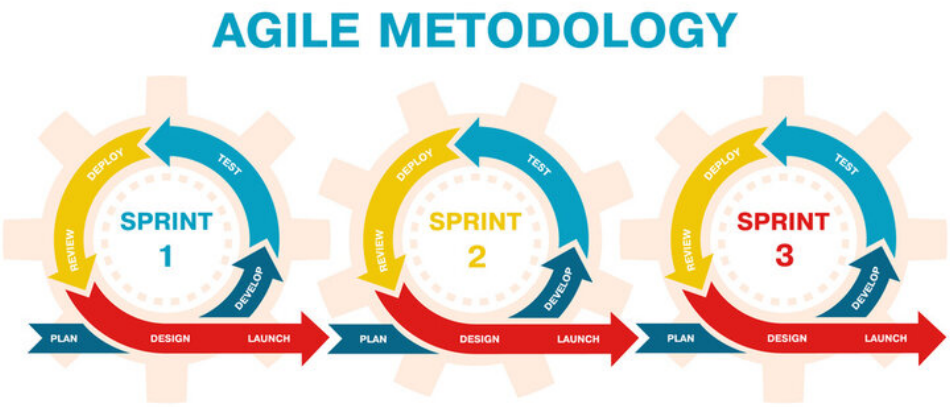

Figura 4.4: La metodologia Agile

4.3.1 Vantaggi

Presentiamo ora alcuni vantaggi della metodologia Agile:

- *Adattabilità.* L'Agile permette di rispondere in modo rapido e flessibile ai cambiamenti. Poiché i requisiti e le priorità possono evolvere nel corso del progetto, l'Agile consente di apportare modifiche durante il processo senza dover riprogettare completamente tutto il progetto.
- *Consegna rapida di valore.* L'Agile suddivide il lavoro in iterazioni chiamate "sprint". Ogni sprint ha una durata specifica, solitamente di una o due settimane, durante la quale il team si impegna a consegnare un insieme di funzionalità ben definite.

Questo permette di ottenere risultati concreti in tempi brevi, consentendo di rilasciare valore agli stakeholder in modo più rapido rispetto ai tradizionali approcci di project management.

- *Coinvolgimento degli stakeholder.* L'Agile promuove una stretta collaborazione con gli stakeholder, come il cliente o il committente del progetto. Essi sono coinvolti attivamente nel processo decisionale e nel feedback durante tutto lo sviluppo. Ciò consente di mantenere una comunicazione costante e di ottenere una maggiore comprensione delle aspettative degli stakeholder.
- *Miglioramento continuo.* L'Agile adotta un approccio iterativo che consente al team di apprendere e migliorare costantemente. Attraverso la riflessione regolare sugli sprint precedenti, il team identifica gli aspetti da migliorare e implementa le correzioni necessarie nel prossimo sprint. Questo ciclo di miglioramento continuo favorisce l'adattamento e l'evoluzione del progetto.
- *Riduzione dei rischi.* L'Agile riduce il rischio di fallimento del progetto. Poiché il team effettua consegne frequenti e regolari, gli eventuali problemi o errori vengono identificati precocemente. Ciò permette al team di apportare le correzioni necessarie in modo tempestivo, evitando ritardi o costi elevati associati a correzioni tardive.

4.3.2 Applicazione di Agile al progetto relativo alla presente tesi

L'Agile è stato applicato in tutti i suoi passaggi chiave lungo lo sviluppo del progetto, nel dettaglio: gli sprint hanno avuto la durata di una o, al massimo, due settimane, in base alle esigenze di sviluppo. La consegna, alla fine di ogni sprint, è una documentazione in cui viene presentato l'elenco delle query modificate, relativamente all'ultima revisione svolta, insieme alla tabella esportata da BigQuery, contenente i valori della vista `_Origin_output`.

La fase di testing è stata delegata al cliente; questo perchè, essendo gestito l'intero processo manualmente tramite Excel, il feedback era semplicemente un confronto di tutti i valori, i quali non dovevano presentare nessuno scostamento.

Riassumendo le fasi di progetto, così come avvenute in ogni sprint, sono state le seguenti:

- *Call.* In questa fase c'è la chiamata con il cliente in cui si valuta lo stato del progetto e i prossimi step da svolgere, gerarchizzandoli in base all'importanza.
- *Analisi.* In questa fase si analizza cosa è stato richiesto dal cliente e si valuta cosa modificare e in che parte del codice.
- *Implementazione.* In questa fase si modifica il codice, appuntandosi cosa si modifica e dove.
- *Report.* In questa fase si prepara il report da consegnare al cliente (documentazione e tabella dei risultati annessa).
- *Feedback.* In questa fase si riceve un documento da parte del cliente che indica se quanto implementato è corretto o se sono presenti errori.

CAPITOLO 5

Implementazione & Esecuzione

In questo capitolo parliamo dell'ottimizzaizone del codice e dei limiti di BigQuery, successivamente, illustriamo le modifiche da necessarie prima di ogni esecuzione, presentando anche la query finale con relativo output. Si analizzeranno, quindi, i KO. Nell'ultima parte ci sarà un focus su come avvengono l'esecuzione e il setup necessario.

5.1 Ottimizzazione & Limiti di BigQuery

Il progetto in questione lavora con un enorme quantità di dati, per questo motivo, oltre alla suddivisione logica delle query, è stata necessaria una fase di ottimizzazione del codice SQL, anche grazie a BigQuery, che offre la possibilità di tenere traccia delle attività delle query e dei relativi tempi di esecuzione, direttamente nell'interfaccia.

Di seguito vengono descritte alcune delle tecniche utilizzate per ottimizzare il codice SQL:

- *Evitare l'utilizzo di "SELECT *".* Specificare solo le colonne necessarie nella clausola SELECT, anziché selezionare tutte le colonne disponibili. In questo modo si riduce il trasferimento di dati dal database al client e si ottimizzano le prestazioni.
- *Utilizzare le join appropriate.* Scegliere il tipo di join più adatto alla situazione. Ad esempio, se si ha bisogno solo di corrispondenze esistenti tra due tabelle, si utilizza INNER JOIN invece di OUTER JOIN, in quanto quest'ultimo può essere più oneroso in termini di prestazioni.
- *Evitare le sottoquery quando possibile.* Le sottoquery possono essere utili in alcune situazioni, ma possono rallentare le prestazioni se utilizzate in modo inefficiente. Riscrivere le query complesse, che utilizzano sottoquery, in join o query correlate, porta ad un miglioramento delle prestazioni.

BigQuery, inoltre, ha alcuni limiti da tenere in considerazione quando si utilizza la piattaforma. Questi possono influire sulla quantità di dati che è possibile archiviare, interrogare o elaborare. BigQuery, durante l'implementazione, ha, infatti, presentato il seguente errore: "Risorse superate durante l'esecuzione della query: risorse insufficienti per la pianificazione della query - troppe sottoquery o query troppo complessa".

Il problema è stato risolto tramite la creazione di due tabelle intermedie, ossia delle tabelle ausiliari, che dividevano il workflow in tre parti. Le due tabelle sono state le seguenti:

- `_Consolidation_Report.`

  Questa tabella contiene i risultati della vista `_MRP_Attribution_Extraction.`

- Output_Transformed_Table.

  Questa tabella contiene i risultati della vista `_Output_Transformed`.

Durante ogni esecuzione del processo, è necessario salvare i risultati di entrambe le viste nelle due tabelle (sovrascrivendo i contenuti esistenti).

Il workflow, comprensivo delle due tabelle ausiliari, è mostrato in Figura 5.1.

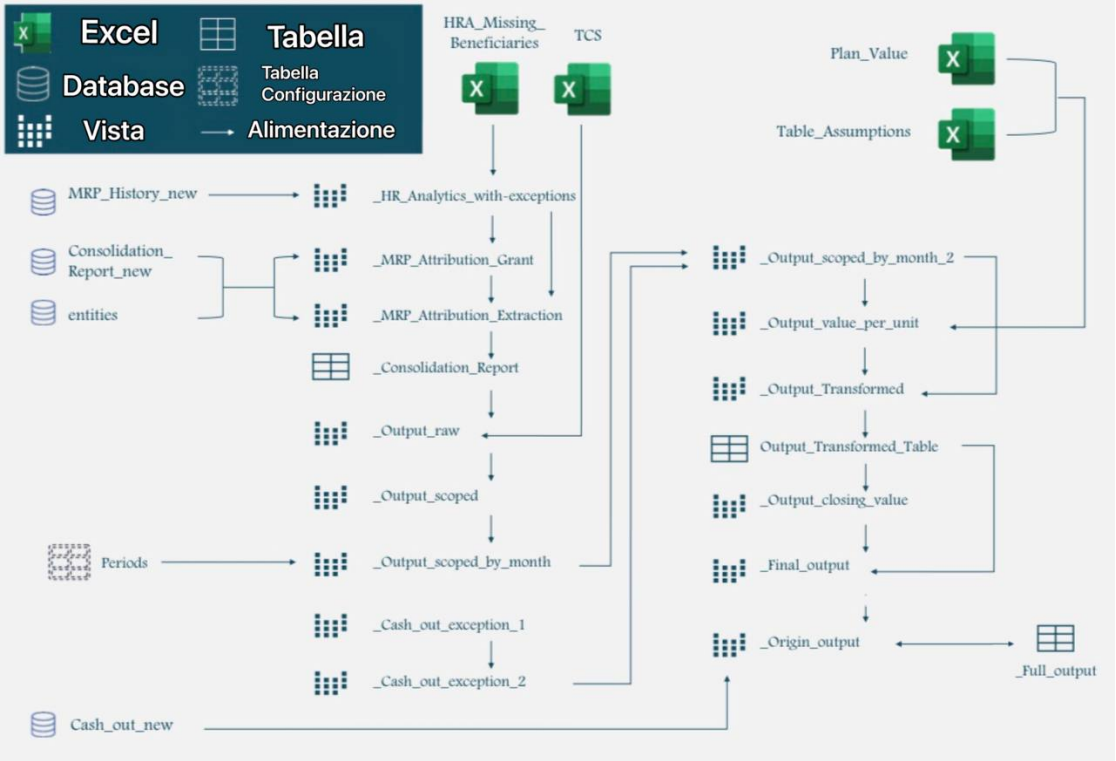

Figura 5.1: Workflow con le tabelle intermedie

5.2 Codice

Nei precedenti capitoli è stata fortemente ribadita l'impossibilità di aumentare eccessivamente il grado di dettaglio della descrizione del progetto per motivi commerciali; infatti non è possibile mostrare le query. Mostriamo, comunque, in questo paragrafo, per quanto possibile, le parti di codice relative a modifiche necessarie per l'esecuzione e, per completezza e visione d'insieme, anche la query finale (poichè contengono materiale divulgabile).

5.2.1 Modifiche manuali nel codice

Presentiamo ora le parti del codice che necessitano di cambi manuali prima di ogni esecuzione.

- La prima modifica da fare (Listato 5.1) è dovuta al fatto che l'Euro è la moneta di riferimento per il calcolo; affinchè vi sia una conversione corretta di tutte le valute degli investitori non appartenenti all'Unione Europea, nella query `_Output_Transformed`

è necessario aggiornare i valori.

```
CASE
    WHEN Currency = "EUR"
    THEN 1
    WHEN Currency = "USD"
    THEN 1.1363
    WHEN Currency = "JPY"
    THEN 128.2
    WHEN Currency = "CHF"
    THEN 1.043
    WHEN Currency = "CNY"
    THEN 7.2395
    WHEN Currency = "GBP"
    THEN 0.85173
    WHEN Currency = "AED"
    THEN 4.1731
    WHEN Currency = "HKD"
    THEN 8.8601
END
    AS Conversion_rate_EUR_to_local
```

Listato 5.1: Valute da aggiornare

- Prima di mostrare la query `_Origin_output`, presentiamo, nel Listato 5.2, la query aggiuntiva, non presente nel workflow, che serve a storicizzare i dati nella tabella `_Full_output`, in modo da facilitare la comprensione della relazione tra le due. La seconda modifica è nella vista `_Origin_output`, ossia quella che produce l'output dell'intero processo.

```
WITH CTE AS (
    SELECT *,
        "ACT Pmm yyyy" AS Tag_History_Name,
        yyyymm AS Tag_History_Code
    FROM `k-tps1-datalake-dev.usage_lti_DEV._Origin_Output`
)
SELECT * FROM CTE

UNION ALL

(
 SELECT * FROM `k-tps1-datalake-dev.usage_lti_DEV.Full_Output`
)
```

Listato 5.2: Query che storicizza l'output.

La query nel Listato 5.2 aggiunge semplicemente all'output due colonne:

  - `Tag_History_Name`: tag storico con il mese e l'anno di quando è avvenuta l'esecuzione.
  - `Tag_History_Code`: codice ricavato dal tag sopraindicato.

Nel Listato 5.3 troviamo la query di `_Origin_output`, dove nella riga 28, troviamo il tag storico, appartenente al secondo `join`, che occorre completare coerentemente con l'ultima esecuzione avvenuta, in modo da avere continuità tra la data di chiusura del periodo finanziario precedente e la data di apertura del periodo finanziario attuale.

```
WITH CTE_2 AS
 (WITH CTE_1 AS
  (WITH CTE AS
   (SELECT t1.*,
     T2.Opening_Period_Provision
     AS Opening_Period_Provision_History,
     CASE
       WHEN t1.Plan = "KPS-A" OR t1.Plan = "KPS-B"
       THEN -t3.Totaly_Payout_Value_in_Euros
         * (1 + t1.Social_Charges_for_KPS)
       ELSE -t3.Totaly_Payout_Value_in_Euros
         * (1 + t1.Social_Charges_for_KMU_BMU_CP)
     END AS Cash_Out_with_social_costs
    FROM `k-tps1-datalake-dev.usage_lti_DEV._final_output` t1

    LEFT JOIN

    `k-tps1-datalake-dev.normalized_lti.Cash_out_new` t3
       ON t1.Beneficiary_ID = t3.Beneficiary_ID
       AND t1.Grant_Name = t3.Grant
       AND DATE_TRUNC(t1.Financial_Period_Closing_Date, MONTH)
       = DATE_TRUNC(t3.Transaction_Date, MONTH)

    LEFT JOIN

    (SELECT *
    FROM `k-tps1-datalake-dev.usage_lti_DEV.Full_Output`
      WHERE Tag_History_Name = "ACT Pmm yyyy") T2
      ON T1.Beneficiary_ID = T2.Beneficiary_ID
      AND T1.Plan = T2.Plan
      AND T1.Period = T2.Period
      AND T1.Grant_Name = T2.Grant_Name
      AND DATE_TRUNC(t1.Financial_Period_Closing_Date, MONTH)
      DATE_ADD(DATE_TRUNC(T2.Financial_Period_Closing_Date,
      MONTH),INTERVAL 12 MONTH))

    SELECT * EXCEPT(Cash_Out_with_social_costs,
    Opening_Period_Provision_History),
      CASE
        WHEN Cash_Out_with_social_costs IS NULL THEN 0
        ELSE Cash_Out_with_social_costs
      END AS Cash_Out_with_social_costs,
      CASE
        WHEN Opening_Period_Provision_History IS NULL THEN 0
        ELSE Opening_Period_Provision_History
        END AS Opening_Period_Provision_History
      FROM CTE)

  SELECT * EXCEPT(Cash_Out_with_social_costs),
     SUM(Cash_Out_with_social_costs)
     OVER(PARTITION BY Beneficiary_ID, PLAN,
     Grant_Name, Unit_or_Share_Value
     ORDER BY Beneficiary_ID, PLAN, Grant_Date_for_finance
     ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)
     AS Cash_Out_with_social_costs
  FROM CTE_1)

SELECT *,
       Opening_Period_Provision
       - Opening_Period_Provision_History
       - Cash_Out_with_social_costs
```

```
62        AS PL_Impact_of_Assumptions_Adjustment,
63        PL_Expense_Period + Opening_Period_Provision
64        - Opening_Period_Provision_History
65        - Cash_Out_with_social_costs
66        AS PL_Expense_N
67  FROM CTE_2
68
69  ORDER BY
70    Beneficiary_ID,
71    Grant_Date_for_finance,
72    Grant_Name,
73    Plan_Year,
74    Period
```

Listato 5.3: La query `_Origin_output`

5.3 Output Finale

Nel Listato 5.4 è mostrata un record dell'output finale, cioè ciò che viene generato da `_Origin_output` in formato JSON, in cui le coppie chiave-valore presentano come chiavi i campi delle colonne e come valori i dati. I dati utilizzati, sono mock, cioè informazioni generate artificialmente che simulano dati reali, utilizzati quando i dati reali non sono adatti per l'uso desiderato.

Ogni beneficiario avrà ottantaquattro record, ricavati da sette anni moltiplicati per dodici mesi; nell'esempio presentato, i valori sono relativi al settimo mese del primo anno.


```
{
  "Beneficiary_ID": "giovanni",
  "Plan_Year": "N",
  "Financial_Period_Opening_Date": "2020-12-31",
  "Financial_Period_Closing_Date": "2021-01-31",
  "Brand_at_Grant": "giulia",
  "Period": "7",
  "Grant_Date_for_finance": "2021-10-01",
  "Acquisition_Date": "2024-03-31",
  "Cancelled": "0",
  "Quantity_Exercised_Released_Monetized": "0",
  "Total_vesting_days_number": "911",
  "Vesting_Period_Opening_Date": "2021-10-01",
  "Vesting_Period_Closing_Date": "2021-10-01",
  "Total_Vesting_Days_number_per_period": "0",
  "Vesting_pourcentage": "0.0",
  "Total_Vesting_Days_number_cumulative": "0",
  "Vesting_pourcentage_Cumulative": "0.0",
  "Unit_or_Share_Value": "69",
  "Social_Charges_for_KPS": "0.0",
  "Social_Charges_for_KMU_BMU_CP": "0.0",
  "Extraction_date": "2022-11-15",
  "Total_Theoretical_Turnover_Days_Nber": "42",
  "Prorata_Temporis_Theoretical_Turnover_pourcentage": "0.55043122806",
  "Total_Vesting_Period_in_years": "2.4013663013",
  "Initial_Plan_Name": "fano",
  "Grant_Name": "blitz - 2021",
  "Plan": "BMU",
  "Currency": "EUR",
  "Conversion_rate_EUR_to_local": "1.0",
  "MRP_Payroll_Vision_at_Grant": "gb1996",
  "MRP_Budget_Vision_at_Grant": "gb1996",
```

```
  "MRP_Payroll_Vision_at_Extraction_Date": "gb1996",
  "MRP_Budget_Vision_at_Extraction_Date": "gb1996",
  "Legal_entity_description_payroll_at_grant": "lol (ITALY) TRADING LTD",
  "Legal_entity_description_Budget_at_grant": "lol (ITALY) TRADING LTD",
  "Legal_entity_description_payroll_at_extraction": "lol (ITALY) TRADING LTD",
  "Legal_entity_description_Budget_at_extraction": "lol (ITALY) TRADING LTD",
  "Code_country_payroll_vision_at_grant": "IT",
  "Code_country_budget_vision_at_grant": "IT",
  "Code_country_payroll_vision_at_extraction": "IT",
  "Code_country_budget_vision_at_extraction": "IT",
  "Outstanding_Quantity_Unvested_and_Vested": "420",
  "Exercisable_Quantity_Vested": "0",
  "Annual_turnover_percentage": "0.17",
  "Last_Value": "888",
  "Annual_growth_percentage_LTI_Year": "0.15",
  "Annual_growth_percentage_LTI_Year_plus_1": "0.14",
  "Annual_growth_percentage_LTI_Year_plus_2": "0.14",
  "Annual_growth_percentage_LTI_Year_plus_3": "0.0",
  "Instrument_value": "1303.9800002",
  "Theoretical_Turnover_percentage": "0.27484849314",
  "Prorated_Theoretical_Turnover_pourcentage": "0.15712328767",
  "Total_Projected_Value": "62591.335200000009",
  "Total_Provision_excl_Turnover": "62591.330009",
  "Total_Provision_incl_Turnover": "5122.0825628",
  "Total_Vesting_excl_Turnover": "0.0",
  "Total_Vesting_incl_Turnover": "0.0",
  "PL_Expense_Period": "0.0",
  "Opening_Period_Provision": "0.0",
  "Closing_Period_Provision": "0.0",
  "Opening_Period_Provision_History": "0.0",
  "Cash_Out_with_social_costs": "0.0",
  "PL_Impact_of_Assumptions_Adjustment": "0.0",
  "PL_Expense_N": "0.0"
}
```


Listato 5.4: JSON di un record in output dal processo

5.4 KO

In quest'ultima parte analizziamo i KO emersi durante lo sviluppo del progetto, definendo, innanzitutto, cosa si intende con questo termine.

In ogni sprint, dopo aver consegnato al cliente la documentazione e la tabella contenente i risultati, si è ricevuto un feedback contenete eventuali correzioni da effettuare, gerarchizzate per priorità (dove KO1 è la più importante e urgente); l'elenco dei KO è sempre accompagnato da un file Excel che mostra dove visualizzare, nei dati, il suddetto problema.

Le possibili tipologie di KO sono le seguenti:

- *I bug.* Un bug è un errore, o un difetto nel software, che causa un comportamento indesiderato o imprevisto. I bug possono manifestarsi in vari modi: potrebbero causare il blocco del programma, produrre risultati errati o provocare malfunzionamenti in generale.

  I bug possono essere causati da errori di programmazione, problemi di progettazione o interazioni complesse tra diverse parti del software. Possono essere presenti in qualsiasi tipo di software.

  Quando viene rilevato un bug, gli sviluppatori cercano di identificarne la causa e di correggerlo mediante la scrittura di nuovi codici o l'aggiornamento del software. Questi

aggiornamenti, noti come "patch" o "fix", vengono quindi distribuiti all'utente o al cliente.

La scoperta e la risoluzione dei bug rappresentano un aspetto fondamentale nello sviluppo del software, in quanto consentono di migliorare la qualità e l'affidabilità dei programmi.

La maggiorparte dei KO sono stati di questo tipo; essi sono, tendenzialmente, KO a bassa priorià; nel seguito troviamo degli esempi che si sono avuti realmente:

  - "La colonna `Opening_Provision_History` non è presa correttamente dalla tabella `Full_output`.
  - "KO di date. La colonna `Acquisition_Date` di `Origin_output` ha origine dalla colonna `Vesting_Date_1` della tabella `Consolidation_Report_New` quando quella corretta sarebbe `Vesting_Date`.
  - "KO dei costi sociali per i piani KPS-A e KPS-B: per i piani KPS-A e KPS-B i costi sociali da considerare per calcolare la colonna `PL_Expense_Period` sono quelli della colonna `Social_Charges_for_KPS`. Ad oggi il calcolo è fatto sulla colonna `Social_Charges_for_KMU_BMU_CP` il che non è corretto.

- *Le implementazioni aggiuntive.* Con implementazioni aggiuntive, si intendono funzionalità o modifiche personalizzate che vengono richieste espressamente dal cliente, per adattare un prodotto o un servizio alle esigenze specifiche. Tali richieste possono variare notevolmente a seconda del settore e del tipo di prodotto o servizio coinvolto. Un KO di questo tipo è ad alta priorità; un esempio è riportato di seguito:

  1. "Problema di generazioni di righe per gli strumenti non pagati (`Outstanding instruments` nella tabella `Consolidation_Report`) o per gli strumenti pagati durante l'anno (`Number of units paid out` nella tabella `Cash_out_new`).
     => Le righe non sono generate correttamente dal sistema.

Implementazione aggiuntiva

Nel dettaglio, l'implementazione aggiuntiva presentata precedentemente, fa riferimento all'implementazione di una nuova funzionalità. Essa ha richiesto fondamentalmente una variazione di come funzionava il processo originario, il quale, proiettava l'investimento lungo i successivi tre anni. Invece la richiesta era di proiettarlo e farlo evolvere nei successivi sei anni; per esempio, se un piano iniziava nel 2018, prima sarebbe finito nel 2021 mentre ora doveva finire nel 2024.

5.4.1 Issue Managment

La risoluzione dei KO di entrambi i tipi è avvenuta tramite l'utilizzo dell'issue management, o gestione delle problematiche. Questa è un processo che consiste nell'identificazione, registrazione, monitoraggio e risoluzione delle problematiche o dei problemi che sorgono durante lo sviluppo o l'erogazione di un prodotto o servizio. Questo processo è fondamentale per garantire che le problematiche vengano gestite in modo efficace, minimizzando l'impatto negativo sul progetto nel suo complesso.

Ecco alcuni punti chiave relativi all'issue management:

- *Identificazione delle problematiche.* L'identificazione delle problematiche avviene attraverso il monitoraggio continuo del progetto o del servizio. Le problematiche possono essere individuate da vari canali, come segnalazioni dei clienti, feedback degli utenti,

test di qualità o monitoraggio delle prestazioni. È importante avere una comunicazione aperta e un meccanismo per segnalare le problematiche in modo tempestivo.

- *Registrazione delle problematiche.* Le problematiche identificate devono essere registrate in un sistema di issue tracking; questo può essere un software dedicato o uno strumento più semplice come un foglio di calcolo. Ogni problematica dovrebbe essere documentata accuratamente, includendo una descrizione dettagliata, il contesto, le informazioni sui tempi e le priorità assegnate.

- *Prioritizzazione e assegnazione delle problematiche.* Le problematiche registrate devono essere valutate e classificate in base alla loro importanza e al loro impatto sul progetto. Una valutazione accurata della priorità aiuta a gestire le problematiche in ordine di importanza. Inoltre, le problematiche dovrebbero essere assegnate a membri specifici del team o ai responsabili delle risoluzioni per garantire che siano trattate in modo tempestivo.

- *Monitoraggio delle problematiche.* Le problematiche registrate devono essere monitorate nel tempo per tenere traccia dei progressi nella loro risoluzione. È importante aggiornare regolarmente lo stato delle problematiche nel sistema di issue tracking, tenere traccia delle attività svolte, delle discussioni o delle soluzioni proposte. Ciò aiuta a mantenere una visione chiara dello stato di ogni problematica e ad agire di conseguenza.

- *Risoluzione delle problematiche.* Una volta che una problematica è stata assegnata e monitorata, è necessario prendere le misure appropriate per risolverla. Questo può richiedere la collaborazione tra team, lo sviluppo di soluzioni, il testing e la verifica dei risultati. È importante mantenere una comunicazione efficace durante il processo di risoluzione delle problematiche e assicurarsi che le soluzioni siano documentate per riferimenti futuri.

- *Valutazione post-risoluzione*. Dopo la risoluzione di una problematica, è utile valutare il corrispettivo processo di gestione. Questo può comportare l'analisi delle cause delle problematiche, l'identificazione di aree di miglioramento e l'implementazione di azioni correttive per prevenire problemi simili in futuro.

5.5 Esecuzione del processo

Ricapitoliamo ora, i passaggi che Society deve fare, ogni sei mesi, per il calcolo dei dividendi e la relativa analisi:

1. Verifica della disponibilità di `Consolidation_Report_new`, `MRP_History_new` e `Cash_out_new`, i quali sono presenti nel livello *Normalized*.

2. Aggiornamento dei file Excel `Plan_Value`, `Table_Assumptions` e `TCS` secondo gli schemi definiti.

3. Aggiornamento dei valori di cambio delle valute e salvataggio dei risultati delle viste `_MRP_Attribution_Extraction` e `_Output_Transformed`, rispettivamente nelle tabelle ausiliari `Consolidation_Report` e `Output_Transformed_Table`, sovrascrivendo quelle già esistenti

4. Aggiornamento del tag storico e run di `_Origin_output` .

5. Alimentazione diretta di Power BI con la tabella `_Full_output` .

I dati in output sono visualizzati direttamente in dashboard Power BI (Figura 5.2) per svolgere task di data visualization e data analytics, già spiegati all'inizio.

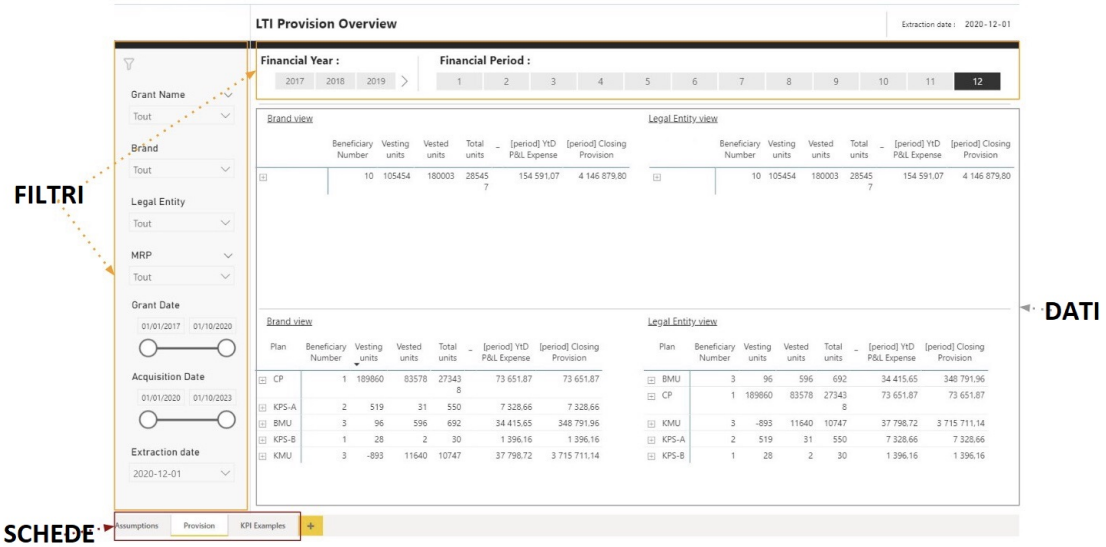

Figura 5.2: Dashboard su Power BI

In particolare, è possibile effettuare le seguenti attività:

- *Valutazione delle performance.* Analizzare i dati delle perdite e dei guadagni degli azionisti consente all'azienda di valutare le proprie performance finanziarie. Questa analisi può rivelare le aree in cui essa ha avuto successo e quelle in cui ha registrato perdite. Queste informazioni possono essere utilizzate per prendere decisioni strategiche e pianificare azioni future.

- *Rendicontazione finanziaria.* Le aziende sono tenute a fornire informazioni finanziarie accurate e trasparenti agli azionisti e ad altre parti interessate. L'analisi dei dati relativi alle perdite e ai guadagni degli azionisti consente all'azienda di preparare i report finanziari necessari, come il bilancio annuale, i rendiconti trimestrali o i rapporti agli azionisti.

- *Identificazione dei trend di mercato.* L'analisi dei dati finanziari può aiutare l'azienda a identificare i trend di mercato e a comprendere come gli azionisti stanno rispondendo alle performance dell'azienda. Questo può fornire importanti indizi sulle preferenze degli investitori, sulle aspettative di mercato e sulle opportunità di crescita.

- *Identificazione delle cause delle perdite.* L'analisi dei dati può aiutare l'azienda a individuare le cause delle perdite finanziarie degli azionisti. Queste possono includere fattori interni, come problemi operativi o inefficienze, e/o fattori esterni, come cambiamenti nel mercato o nella concorrenza. Identificare le cause delle perdite può consentire all'azienda di apportare modifiche strategiche per mitigare i rischi futuri.

- *Monitoraggio delle performance degli investimenti.* Le aziende che gestiscono investimenti per conto dei propri azionisti possono utilizzare l'analisi dei dati finanziari per monitorare le performance di tali investimenti. Ciò può aiutare l'azienda a valutare la redditività dei propri portafogli di investimento e a prendere decisioni informate sulla gestione degli investimenti futuri.

- *Comunicazione con gli azionisti.* L'analisi dei dati finanziari può essere utilizzata per comunicare con gli azionisti e fornire loro una chiara comprensione delle performance finanziarie dell'azienda. Ciò può contribuire a mantenere un rapporto fiduciario con gli azionisti e a fornire loro informazioni trasparenti per prendere decisioni di investimento o di mantenimento delle azioni.

CAPITOLO 6

Discussione in merito al lavoro svolto

In questo capitolo viene presentata e applicata la SWOT Analysis, successivamente, viene presentata una panoramica sulle lezioni apprese durante lo sviluppo del progetto.

6.1 SWOT Analysis

L'analisi SWOT è una tecnica utilizzata per identificare i punti di forza, di debolezza, le opportunità e le minacce della propria azienda o di un progetto nello specifico. Ampiamente usata da molte organizzazioni, dalle piccole imprese agli enti no-profit fino alle grandi imprese, l'analisi SWOT può essere utilizzata sia per scopi personali che professionali. Come si può vedere in Figura 6.1, l'acronimo SWOT sta per Strengths, Weaknesses, Opportunities e Threats.

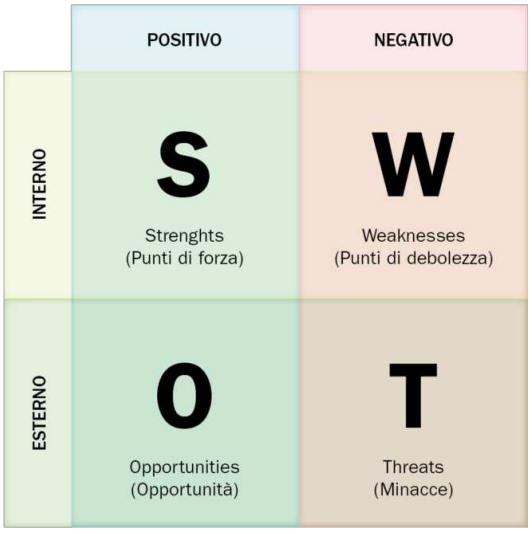

Figura 6.1: Grafico dell'analisi SWOT

Gli elementi che costituiscono l'analisi SWOT sono:

- *Strenghts:* sono gli aspetti in cui il progetto eccelle e indicano cosa lo contraddistinguono dalla concorrenza. Si tratta, quindi, di quegli elementi che facilitano il conseguimento

dell'obiettivo di progetto. I punti di forza, ad esempio, possono essere le risorse a disposizione o, in generale, tutto ciò che costituisce il vantaggio competitivo.

- *Weaknesses:* sono gli aspetti che non permettono al progetto di avere performance ottime: sono, cioè, aree in cui possono essere fatti miglioramenti. Le debolezze, ad esempio, possono essere date da una mancanza di risorse, da ciò che è ancora migliorabile o da qualche componente che sta attualmente "sottoperformando". In generale, ci si riferisce a tutto ciò che può essere considerato dannoso rispetto al raggiungimento dell'obiettivo.
- *Opportunities:* sono i fattori esterni favorevoli che potrebbero fornire un vantaggio competitivo. Per esempio, il rilascio di nuove funzionalità di un framework o l'apertura di un nuovo mercato in cui è possibile espandersi possono dare vita a opportunità.
- *Threats:* sono fattori esterni che potrebbero danneggiare il progetto o l'organizzazione. Ad esempio, le minacce possono essere costituite da nuove regolamentazioni o leggi che limitano l'uso del prodotto frutto del progetto, oppure da una crescente competività dovuta a un interessamento delle tematiche affrontate nel progetto da parte di aziende concorrenti.

6.1.1 Punti di forza

I principali punti di forza del progetto sono l'ordine e la chiarezza con cui è stato realizzato; questi permettono una gestione adeguata dell'intero processo, sia grazie alla suddivisione logica delle query di ogni sottoprocesso, sia per la pulizia del codice, il quale è stato ottimizzazo e, successivamente, commentato ad hoc.

6.1.2 Punti di debolezza

Il principale punto di debolezza è dato dalla doverosa presenza delle tabelle ausiliari, sia perchè appesantiscono l'intero processo, dovendo sempre essere aggiornate prima di ogni esecuzione, sia perchè hanno evidenziato come modifiche al codice possono richiedere l'implementazione di ulteriori tabelle intermedie, obbligando a ridisegnaree il workflow e appesantendo ulteriormente il tutto.

6.1.3 Opportunità

La principale opportunità di questo progetto è stata la creazione di rapporti lavorativi con un importante cliente a livello mondiale, che permette a Gruppo Filippetti sia di avere un ottimo ritorno d'immagine sia di poter mantenere rapporti con Society per altri progetti o per il mantenimento o la gestione di quanto presentato nella suddetta tesi.

6.1.4 Minacce

La principale minaccia è dettata dal fatto che l'intero processo mette in relazione un Cloud DBMS di proprietà di Google, cioè BigQuery, con un software per la Business Intelligence di proprietà di Microsoft, cioè Power BI. Essendo entrambe le aziende leader nel settore informatico, Microsoft potrebbe decidere di non permettere più questa compatibilità di Power BI con BigQuery, a favore di Microsoft Azure, che è la piattaforma di cloud computing offerta da Microsoft.

6.2 Lezioni apprese

Ogni progetto offre un'opportunità di apprendimento unica. Le lezioni apprese possono contribuire a migliorare le pratiche aziendali, guidare il miglioramento continuo e favorire il successo dei futuri progetti. Le lezioni apprese nel corso di questo progetto sono state molteplici; queste sono riassumibili nel seguente elenco:

- *Relazione con il cliente.* Abbiamo capito che il rispetto da parte del cliente è fondamentale affinchè vi siano un clima sereno e di reciproca collaborazione; quindi, l'educazione e la puntualità sono sempre da mettere al primo posto.
- *Comunicazione efficace.* Abbiamo capito l'importanza della comunicazione per comprendere le richieste e le spiegazioni del cliente, soprattuto quando si presentano barriere linguistiche che la rendono più complicata. In questo caso specifico, il cliente non era italiano, pertanto, la conoscenza dell'inglese parlato e scritto, si è rivelata fondamentale.
- *La pulizia del codice.* Abbiamo capito l'importanza di un codice scritto in maniera chiara e leggibile, per facilitarne la comprensione a chiunque debba leggere direttamente il codice.
- *Documentazione adeguata.* Ho capito l'importanza di creare una documentazione adeguata per mostrare il lavoro svolto e i risultati ottenuti al cliente, sia perchè è necessario riassumere una gran mole di lavoro in poche informazioni, sia perchè quanto si mostra deve essere efficace, nel momento in cui ci si relaziona con figure, che possono non essere tecniche.
- *Lavoro di squadra.* Abbiamo capito che un team sano rappresenta la base per lavorare efficacemente e efficientemente, sia quando le cose vanno bene sia quando le cose vanno meno bene; pertanto è necessario fare del proprio meglio affinchè non si ledano i rapporti tra i vari membri.
- *Know-how & best practices.* Abbiamo capito l'importanza del know-how, cioè le conoscenze e le abilità operative necessarie per svolgere una determinata attività lavorativa, e delle best practices, cioè le azioni più significative, o comunque quelle che hanno permesso di ottenere i migliori risultati, relativamente a svariati contesti e obiettivi preposti. Soprattutto, abbiamo capito come tutto questo si possa ottenere, sia facendo esperienza, sia tramite passaggio di know-how da parte di chi, quell'esperienza già la possiede.

Conclusioni

In questa tesi sono state illustrate le attività di Data Engineering volte all'implementazione di un sistema di Data Analytics per un processo finanziario quale il calcolo dei dividendi.

Il percorso che ci ha portato alla realizzazione di questo sistema parte da uno studio preliminare su quanto sia impattante l'utilizzo della Data Science e di alcune sue declinazioni come la Data Analytics, la Data Visualization e la Business Intelligence, all'interno dei processi aziendali.

Successivamente, si passa all'analisi approfondita delle tecnologie hardware e software utilizzate. Dopo la presentazione di tutti gli elementi coinvolti, viene presentata l'analisi dei requisiti, contestualizzando il cliente e il progetto, per rendere chiari gli obiettivi di quest'ultimo. Il progetto prevedeva l'utilizzo di Google BigQuery, leader del mercato per i DBMS basati su cloud, per svolgere la fase di ETL e i calcoli necessari per ottenere i dividendi; i dati prodotti in output vengono poi utilizzati per alimentare Microsoft Power BI, leader di mercato nella Business Intelligence e Analytics. In seguito, si procede con la trattazione delle sorgenti d'alimentazione, spiegando sia la provenienza dei dati, sia le operazioni necessarie per preprocessarli.

Per quanto riguarda la fase di progettazione, viene esposta la suddivisione logica dell'intero processo in sotto-processi, seguendo il principio del divide et impera; vengono, quindi, presentate le diverse query che compongono il workflow. La trattazione del progetto svolto è stata accompagnata dai principi di management utilizzati, sia per il project management, attraverso Agile, sia per l'issue management. Dopo aver chiarito la progettazione, si passa alla implementazione in SQL, sottolineando i principali problemi incontrati e la loro risoluzione; successivamente viene presentato il recap di come avviene il processo e il setup necessario. In ultima istanza viene svolta la SWOT Analysis sull'intero progetto.

Il risultato ottenuto ha pienamente soddisfatto il committente poichè sono stati raggiunti tutti gli obiettivi prefissati. Nonostante ciò, sono previsti sviluppi futuri per l'integrazione di funzionalità e miglioramenti del sistema complessivo. In particolare, il miglioramento dell'automatizzazione del processo, sia poichè nel setup necessario per l'esecuzione è obbligatorio modificare dei parametri direttamente nel codice SQL, sia per la presenza di tabelle intermedie in determinati punti del workflow, che non permettono l'esecuzione in un singolo step.

Bibliografia

BERGAMASCHI, M. (2021), *Dalla Business Intelligence al Data Warehouse*, Youcanprint.

BERTHOLD, M. e HAND, D. (2007), *Intelligent Data Analysis: An Introduction*, Springer Berlin Heidelberg.

BLOKDYK, G. (2018), *DIKW Pyramid: Complete Self-Assessment Guide*, CreateSpace Independent Publishing Platform.

CADY, F. (2017), *The Data Science Handbook*, Wiley.

CHAUDHURI, S. e DAYAL, U. (1997), «An overview of data warehousing and OLAP technology», *ACM Sigmod record*, vol. 26 (1), p. 65–74.

FRIENDLY, M. (2008), «A brief history of data visualization», in «Handbook of data visualization», p. 15–56, Springer.

INMON, W. e LINSTEDT, D. (2014), *Data architecture: a primer for the data scientist: big data, data warehouse and data vault*, Morgan Kaufmann.

KENNY, D., KASHY, D. e BOLGER, N. (1998), «Data analysis», in «The handbook of social psychology: Vols. 1 and 2,», p. 233–265, McGraw-Hill New York.

LAKSHMANAN, V. e TIGANI, J. (2019), *Google BigQuery: The Definitive Guide: Data Warehousing, Analytics, and Machine Learning at Scale*, O'Reilly Media.

LIM, E.-P., CHEN, H. e CHEN, G. (2013), «Business intelligence and analytics: Research directions», *ACM Transactions on Management Information Systems (TMIS)*, vol. 3 (4), p. 1–10.

NAVA, O. (2021), *Le SGR. Società di gestione del risparmio*, Minerva Bancaria.

NEGASH, S. e GRAY, P. (2008), «Business intelligence», in «Handbook on decision support systems 2», p. 175–193, Springer.

REZZANI, A. (2012), *Business intelligence*, PerCorsi di studio, Apogeo Education.

SAGIROGLU, S. e SINANC, D. (2013), «Big data: A review», in «2013 international conference on collaboration technologies and systems (CTS)», p. 42–47, IEEE.

SCHEPS, S. (2011), *Business Intelligence For Dummies*, For dummies, Wiley.

SCHWABISH, J. (2021), *Better Data Visualizations: A Guide for Scholars, Researchers, and Wonks*, Columbia University Press.

SULLIVAN, D. (2020), *Official Google Cloud Certified Professional Data Engineer Study Guide*, Wiley.

TSAI, C.-W., LAI, C.-F., CHAO, H.-C. e VASILAKOS, A. V. (2015), «Big data analytics: a survey», *Journal of Big data*, vol. 2 (1), p. 1–32.

Siti consultati

- Magic Quadrant di Gartner – `https://coresistemi.it`
- Piramide Dikw Business – `https://premoneo.com`
- What's The Real Value of Big Data For Business? – `https://blog.hubspot.com`
- Le 5V dei Big Data – `https://www.flyip.it`
- La Business Intelligence... dalla BI alla Z! – `https://www.datamaze.it`
- Che cos'è un processo ETL – `https://www.talend.com`
- Il valore dei Big Data per la performance aziendale – `https://iris.luiss.it`
- Cos'è Google BigQuery – `https://www.tagmanageritalia.it`

Ringraziamenti

Vorrei ringraziare, innanzitutto, il mio relatore, Domenico Ursino, per la sua preziosa guida e supporto, sia durante il percorso universitario, sia durante l'intera stesura della tesi; senza il suo prezioso contributo non avrei mai potuto raggiungere questo traguardo. Desidero esprimere la mia gratitudine a tutti coloro che mi hanno sostenuto e incoraggiato durante questo percorso: la mia famiglia, i miei parenti, gli amici, i compagni di corso e Giulia. Voglio ringraziare, anche, i colleghi presso Gruppo Filippetti, Gilberto Girini e Andrea D'Angelo, senza i quali sarebbe stato impossibile affrontare questo progetto. Ognuno di loro ha contribuito, a proprio modo, a insegnarmi che è necessario essere una brava persona, prima ancora di essere un bravo ingegnere.