ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

FACOLTÀ DI INGEGNERIA

Corso di laurea magistrale in Ingegneria Gestionale

USO DI GENERATIVE AI NELL'ANALISI DEI DATI: IL CASO DI UN'AZIENDA LEADER NEL MERCATO EYEWEAR

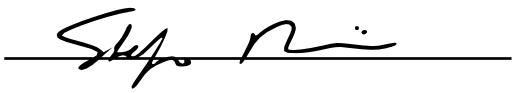
Tesi di laurea in Business Intelligence e Big Data

Tutor universitario

Stefano Rizzi

Presentato da

Simone Romagnoli



Sessione V

Anno accademico 2023/2024

Abstract

Il seguente studio analizza e comprende le metodologie e l'approccio usato per inserire l'intelligenza artificiale generativa all'interno di un processo di analisi e migrazioni dati di un caso reale.

L'elaborato comprende un'introduzione sulle tecnologie sviluppate nel processo, lo svolgimento delle attività e la spiegazione dei vari step in cui si è adoperata e integrata l'IA. Ci si è focalizzati sulla fattibilità dei processi e si sono analizzati gli output ottenuti. Sono presenti analisi future sull'utilizzo della tecnologia e sulla possibilità di espandere le tecniche di analisi adottate anche su fronti nuovi. Si è carcato, inoltre, di mostrare dove e come questi processi potranno esssere migliorati col passare del tempo. Tutte le varie fasi sono state documentate con molteplici grafici ed immagini per aumentare la comprensione della lettura, in particolar modo nella parte finale dello scritto sono riportati alcuni output prodotti.

La ricerca del tool specifico, la sperimentazione e la comprensione delle logiche di funzionamento e l'implementazione sono state tutte fasi preliminari del progetto che hanno richiesto del tempo dedicato e uno studio approfondito, al fine di rendere efficace a tutti gli effetti l'utilizzo di questo strumento all'interno dell'analisi proposta.

Parole chiave

generative AI, intelligenza artficiale, prompt, script, google cloud platform, dataform, excel, codice,sql, python, json, analisi dati, pipeline, flusso dati, bronze layer, silver layer

INDICE

INDICE DELLE FIGURE

1 Introduzione

Il presente studio verte sulla possibilità di utilizzare tecnologie di Generative AI nell'analisi dei dati all'interno di un case study specifico. Il progetto è stato sviluppato all'interno di NTT Data Italia S.p.a., una multinazionale di consulenza e servizi IT che fa parte del gruppo NTT Data, azienda leader nel settore dei servizi IT. NTT Data Italia fornisce soluzioni e servizi IT innovativi per una vasta gamma di settori, tra cui finanza, sanità, manifatturiero e pubblica amministrazione.

La società offre consulenza, sviluppo di software su misura, gestione dell'infrastruttura IT e servizi di outsourcing. Il case study specifico su cui si è portato avanti il progetto riguarda la creazione di una nuova Customer Data Platform per un'importante azienda leader nel mercato eyewear.

La tesi è suddivisa in diversi capitoli. Il primo riguarda una parte di introduzione in cui si illustrano la struttura complessiva e gli argomenti trattati. Il secondo capitolo è quello adibito ad elencare lo stack tecnologico utilizzato, ovvero tutti gli strumenti e le piattaforme digitali su cui si basa questo studio e che sono state sviluppate. Al suo interno, inoltre, è compresa una struttura generale ad alto livello di come sono state implementate e integrate le varie piattaforme e tool (AI e non) all'interno del processo di analisi dei dati. Proseguendo con i capitoli e addentrandosi nel corpo centrale dell'elaborato, ci si interfaccerà con la parte di sviluppo e prototipazione. Qui viene spiegato l'As-is e il To-be, nonchè l'obiettivo principale della tesi in sé. Ci si concentra nel dettaglio sullo spiegare come è stato possibile integrare l'intelligenza artificiale generativa all'interno del processo di analisi dei dati, in quali fasi del processo è stata inserita e soprattutto si analizzeranno i risultati ottenuti. Il quarto capitolo riguarda una panoramica delle logiche con cui si sono gestiti i dati in ingresso, come si sono integrati con la generative AI e dove sono stati poi caricati dopo essere stati processati o trasformati.

Successivamente è illustrato un piccolo questionario in cui si è chiesto, a cinque membri dell'azienda all'interno della quale è stato svolto questo progetto, di esprimere un giudizio riguardante l'utilizzo e i risultati ottenuti mediante il tool AI. Inoltre, la sezione dedicata al feedback raccolto dai membri dell'azienda viene arricchita con commenti dettagliati, suggerimenti e riflessioni che emergono dal loro utilizzo quotidiano del tool AI. Queste

testimonianze forniscono una prospettiva pratica e diretta sull'impatto reale che la tecnologia ha avuto all'interno dell'organizzazione.

In ultima analisi si trovano le conclusioni tratte a fine progetto e i futuri scenari di applicazione di questa tecnologia. Inoltre è presente anche un'appendice in cui sono riportati tutti i codici generati e che sono stati oggetto di analisi in quanto output finale prodotto e validato. In aggiunta ai capitoli precedentemente delineati, la sezione conclusiva della tesi si arricchisce di un approfondimento sulle riflessioni personali maturate nel corso dello sviluppo del progetto. Questa parte intende offrire uno sguardo introspettivo sulle sfide, i traguardi raggiunti e sulle lezioni apprese, delineando come l'esperienza accumulata abbia influito sull'approccio personale verso l'implementazione dell'intelligenza artificiale in ambito aziendale. Vengono quindi esaminate le peculiarità del progetto, evidenziando come ogni fase abbia contribuito alla formazione di una visione complessiva più matura e articolata. Segue un'analisi critica dei risultati ottenuti, dove oltre ai successi, si pone l'accento sulle limitazioni incontrate. Questa sezione è volta a fornire una visione equilibrata, mettendo in luce come determinate aspettative iniziali si siano confrontate con la realtà operativa, guidando verso una riflessione su come migliorare e ottimizzare future implementazioni di sistemi basati su intelligenza artificiale. La tesi si proietta poi verso lo studio di futuri scenari applicativi.

L'appendice viene ampliata per includere una documentazione più dettagliata dei codici e degli algoritmi impiegati, che illustrano passo dopo passo le fasi di implementazione e integrazione. Questo materiale supplementare è pensato per offrire al lettore una comprensione più approfondita e tangibile delle metodologie adottate.

2 Stack tecnologico

Il capitolo che segue sarà un'introduzione agli strumenti e alle tecnologie utilizzate nell'ambito del progetto e dell'organizzazione.

2.1 Google Cloud Platform

Durante il periodo in azienda si è potuto sfruttare a pieno il potenziale della Google Cloud Platform. E' stato importante capire come funzionasse questa piattaforma poichè è risultata essere la base e il tool principale con cui ci si è interfacciati giornalmente. Di seguito si riportano alcuni pro e contro di questa tecnologia.

2.1.1 Vantaggi

Google è un cloud molto efficiente. Ogni servizio ha il proprio caso d'uso ed è stato progettato affinchè ciascun tool risulti perfettamente integrato col successivo.

Quando si tratta di punti di forza, la documentazione di Google Cloud Platform non è seconda a nessuno. Uno dei punti cardine della documentazione è il modo in cui Google incorpora le azioni nei documenti di GCP. Sono divisi in una sezione panoramica, seguita da una sezione pratica, che guida il lettore attraverso l'implementazione della funzionalità o del servizio.

Per ogni strumento o tool di GCP ci sono corsi veramente ben fatti e guidati che offrono una panoramica completa del tool e successivamente mettono alla prova sull'ambiente di training.

Sono presenti laboratori dedicati per apprendere i possibili use case del tool.

Scalabilità: uno dei principali vantaggi del cloud computing è la possibilità di scalare verso l'alto o verso il basso in base alle esigenze aziendali. Con un'infrastruttura basata su cloud, si possono aggiungere o rimuovere rapidamente le risorse in base alle necessità, consentendo di rispondere ai cambiamenti della domanda e di tenere sotto controllo i costi.

Risparmio sui costi: il cloud computing può offrire notevoli risparmi sui costi rispetto alla tradizionale infrastruttura IT on-premise. Con un modello di prezzi con pagamento in

base al consumo, si paga solo per le risorse che si utilizzano, invece di dover investire in hardware e software costosi in anticipo.

Migliore collaborazione: il cloud computing può anche migliorare la collaborazione all'interno dell'azienda. Memorizzando dati e applicazioni nel cloud, i membri del team possono accedervi da qualsiasi luogo, in qualsiasi momento e lavorare insieme in modo più efficace.

Ripristino di emergenza: con il cloud computing, i tuoi dati aziendali vengono sottoposti a backup e protetti, anche in caso di emergenza. I fornitori di servizi cloud dispongono spesso di robusti meccanismi di ripristino di emergenza, in modo da poter recuperare rapidamente i dati e rimettere in funzione la tua attività.

Accesso alla tecnologia più recente: i fornitori di servizi cloud aggiornano e migliorano costantemente i loro servizi, così si può essere sicuri di utilizzare sempre la tecnologia più recente. Ciò può aiutare l' azienda a rimanere competitiva, agile e garantire sempre le ultime tendenze e sviluppi.

2.1.2 Svantaggi

Affidamento alla connettività Internet: il cloud computing dipende dalla connettività Internet, quindi in caso di interruzioni o connessioni lente, sarebbe compromessa la capacità di accedere ai dati e alle applicazioni.

Controllo limitato: quando si utilizza il cloud computing, si sta essenzialmente affittando risorse da un provider di terze parti, il che significa che si ha un controllo limitato sull'infrastruttura. Questo può essere difficile per le aziende che richiedono un maggiore controllo sul proprio ambiente informatico o devono rispettare specifici requisiti normativi.

Integrazione con sistemi legacy**:** se l'azienda dispone di sistemi o applicazioni legacy esistenti, può essere difficile integrarli con l'infrastruttura basata su cloud. Ciò potrebbe richiedere risorse e competenze aggiuntive e potrebbe richiedere una significativa riprogettazione dei sistemi esistenti.

Quest'ultimo punto è stato l'ostacolo principale che si è dovuto affrontare nel progetto. Proprio perchè MapR è un database legacy e con logiche di immagazzinamento dei dati obsolete, si sono dovute realizzare pipeline ad hoc per l'instradamento dei dati, ma prima ancora sono state create pipeline degli ambienti specifici e rimappate secondo il nuovo database, che comunque doveva ricordare quello vecchio.

E' stato un percorso di apprendimento bottom-up. Il percorso è iniziato dalla piattaforma finale dove i dati saranno trasferiti e si è percorso a ritroso l'instradamento dei dati da MapR a GCP. Bisognava capire quali fossero gli step intermedi, i software e i tool che garantissero questo trasferimento nella maniera più efficiente possibile, ma sempre rispecchiando le richieste del cliente.

2.2 MapR

MapR Database è un sistema di gestione di database NoSQL ad alte prestazioni integrato nella piattaforma dati MapR. Si tratta di un database multimodello altamente scalabile che riunisce operazioni e analisi, nonché flussi di lavoro di database e streaming in tempo reale per consentire un insieme più ampio di applicazioni ad alta intensità di dati di prossima generazione nelle organizzazioni. Il Data Vault di MapR è un sistema di gestione dei dati progettato per garantire l'alta scalabilità, la flessibilità e la sicurezza nella gestione di grandi volumi di dati. La struttura del Data Vault può essere descritta attraverso diversi componenti chiave:

- **Hub Tables**: le tabelle hub sono al centro della struttura del Data Vault. Contengono le chiavi univoche degli elementi di business e rappresentano il nucleo intorno al quale si costruiscono gli altri elementi. Queste tabelle sono fondamentali per mantenere l'integrità dei dati e per garantire un rapido accesso alle informazioni.
- **Link Tables**: le tabelle di collegamento, o Link Tables, servono a creare relazioni tra i vari Hub Tables. Queste tabelle sono utilizzate per gestire le relazioni many-to-many.
- **Satellite Tables**: le Satellite Tables contengono i dati descrittivi associati agli elementi negli Hub Tables (o più raramente nei Link Tables). Questi dati possono includere attributi, cronologie temporali, e altre informazioni che cambiano nel tempo. Le Satellite Tables permettono di gestire la versione e la cronologia dei dati, consentendo una migliore tracciabilità e analisi temporale.
- **Load Date Timestamp**: ogni record in una Satellite Table è associato a un timestamp che indica quando il record è stato caricato nel sistema. Questo meccanismo supporta la tracciabilità storica dei dati e facilita l'analisi delle variazioni nel tempo.
- **Business Keys e Surrogate Keys**: il Data Vault utilizza una combinazione di chiavi di business (Business Keys), che sono significative nel contesto del business, e chiavi surrogate (Surrogate Keys), che sono utilizzate internamente per l'ottimizzazione e l'integrazione dei dati.

- **Architettura Scalabile e Flessibile**: il Data Vault di MapR è progettato per essere altamente scalabile e flessibile, supportando sia i carichi di lavoro batch che quelli in tempo reale, e consentendo l'integrazione con diverse fonti di dati e applicazioni.

L'azienda cliente ha mappato il suo database MapR su sorgenti json e ora quel database andava rielaborato tutto mantenendo dove possibile le logiche precedenti, ma rendendolo più scalabile e con meno computazioni, nel caso di implementazioni di dati da nuovi brand. Questo prima a livello teorico e poi pratico è stato fatto in due layer nuovi su ambiente GCP che verranno illustrati in seguito: il bronze e il silver.

In sintesi, il Data Vault di MapR fornisce una soluzione robusta e flessibile per la gestione di grandi volumi di dati, con una forte enfasi sulla sicurezza, la scalabilità e la governance dei dati.

Nelle figure 1,2,3 si riporta un esempio per le tre categorie principali di strutture presenti in MapR: hub, satellite e link.

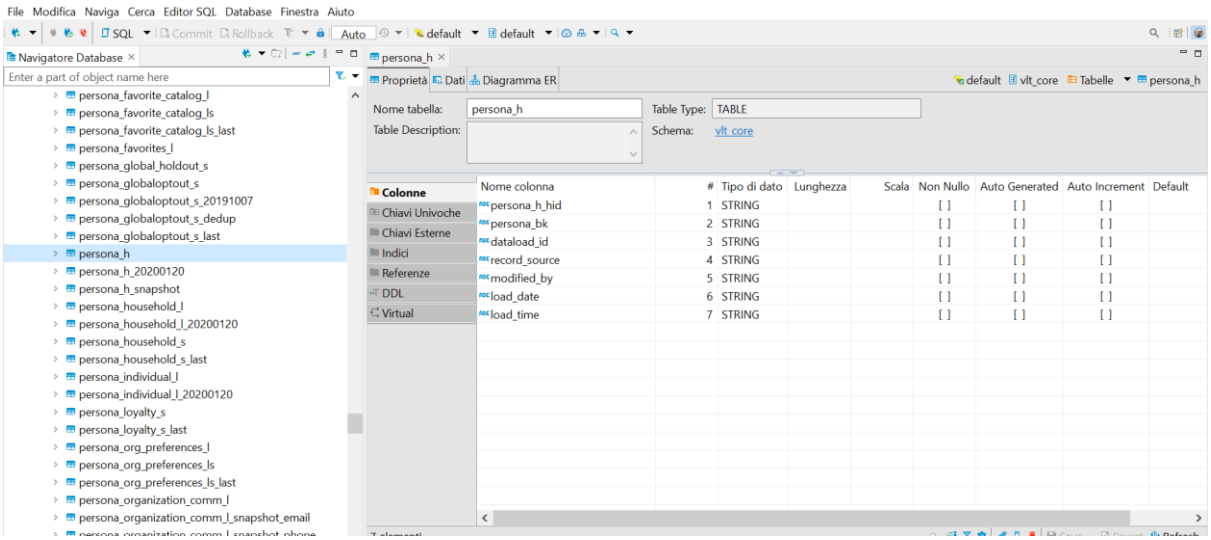


Figura 1-Persona_h (hub)

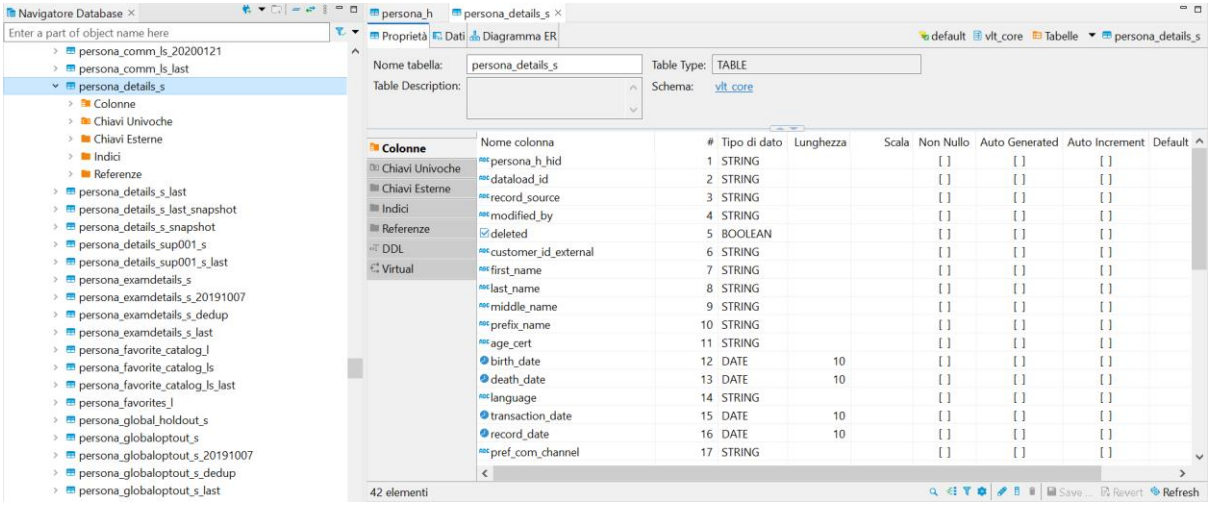


Figura 2-Persona_details_s(satellite

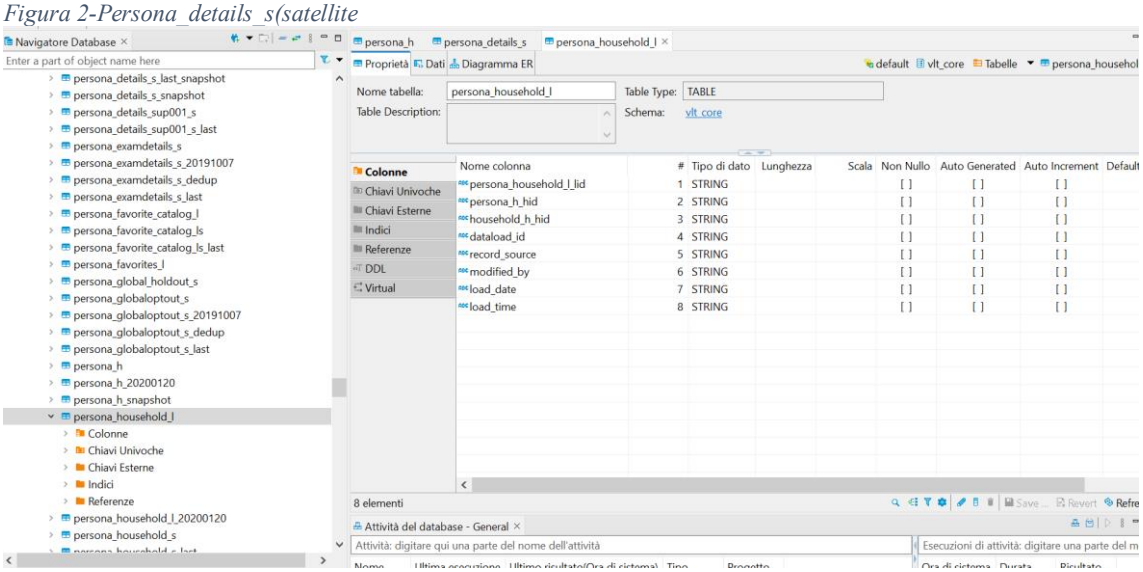


Figura 3-Persona_household_l (link)

2.3 Percorso dati

In questa sezione viene spiegata la struttura e i collegamenti che si sarebbero dovuti creare per instradare i dati da MapR a GCP.

Si illustra una panoramica generale dell'architettura di base che a livello teorico era stata ideata, ma nello specifico ci si concentrerà sull'analisi del Core layer e in particolare il Silver Layer.

Di seguito un grafico che chiarisce la struttura dell'architettura e il ruolo dei vari componenti.

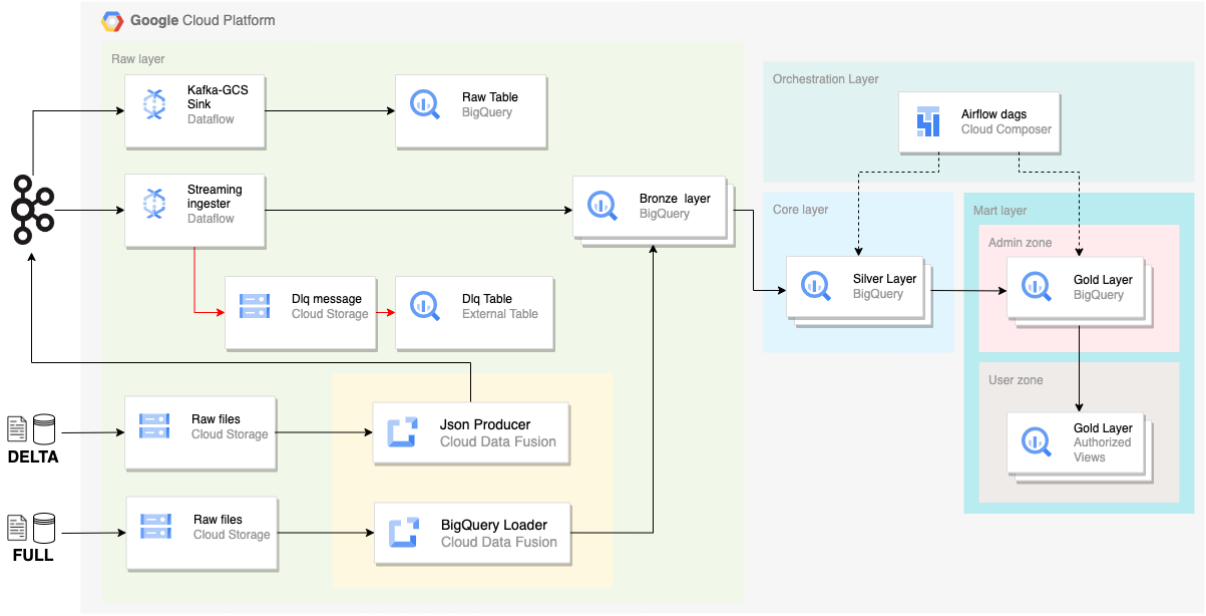


Figura 4- Architettura

2.4 Struttura e architettura progetto MapR to BigQuery

Questa parte analizza le diverse soluzioni per implementare l'ingestion dei dati di input, nonché il processo di trasformazione fino alla pubblicazione del data mart (a cui appartiene il Gold Layer finale a cui faranno riferimento gli utenti del business). In questa parte viene menzionato BigQuery. Si tratta di un servizio di data warehouse completamente gestito, offerto da Google come parte della Google Cloud Platform (ambiente di gestione query e tabelle dati finali). È progettato per l'elaborazione e l'analisi di grandi set di dati, utilizzando la potenza dell'infrastruttura cloud di Google. Ecco alcune caratteristiche chiave di BigQuery:

- **Prestazioni e Scalabilità**: BigQuery può gestire enormi quantità di dati in tempo quasi reale, grazie alla sua capacità di eseguire query su petabyte di dati in pochi secondi.
- **Completamente Gestito**: Come servizio cloud, BigQuery richiede una manutenzione minima. Google gestisce l'infrastruttura sottostante, comprese le attività come l'ottimizzazione del server e l'aggiornamento del software.
- **Modello di Prezzi Basato sull'Uso**: BigQuery adotta un modello di prezzi basato sull'utilizzo, dove si paga per la quantità di dati elaborati dalle query e per lo spazio di archiviazione dei dati.
- **Integrazione con Strumenti di Analisi e Machine Learning**: BigQuery si integra bene con altri strumenti e servizi di Google Cloud e diversi strumenti di machine learning, permettendo un'analisi dati avanzata.
- **Sicurezza e Conformità**: Offre robuste funzionalità di sicurezza, compreso il controllo degli accessi a livello di colonna e la crittografia dei dati in transito e a riposo, conformandosi a diversi standard di sicurezza e privacy.

Ora si analizzerà l'architettura.

Di seguito sono elencati i requisiti da rispettare nell'implementazione del sistema:

- **ingestion di messaggi JSON** da coda Kafka per acquisire i dati in input;
- gestione dei dati in modo **performante e conforme alle best practice** di BigQuery.

- **Salvataggio dei messaggi raw** nella forma originale ricevuta dalla sorgente, per permettere eventuali ricalcoli successivi.
- **Implementazione del masking** dei dati sensibili (PII - Personally Identifiable Information) per garantire la protezione della privacy.
- Fornire la possibilità alle applicazioni e agli amministratori del progetto di visualizzare tutti i dati, compresi i PII, e al contempo **limitare la visione dei PII solo agli utenti finali autorizzati**.
- **Configurazione flessibile e semplice per la schedulazione** del calcolo del core layer e del data mart, senza la necessità di apportare modifiche al processo sottostante

2.4.1 High Level Architecture

L'architettura è suddivisa in 3 layer: *raw*, *core* e *mart*. Si tratta dell'ambiente di sviluppo del progetto e da quali strumenti è composto. In questa tesi si tratterà del core layer, ma si citeranno anche il funzionamento e le logiche di comunicazione tra tutti gli altri tool della tabella sotto riportata.

2.4.2 Raw layer (Brone Layer)

È il primo livello dell'architettura ed è responsabile di ospitare gli applicativi e le strutture che elaborano e salvano i dati provenienti dai sistemi sorgente, senza l'applicazione di regole di business. È composto da tre componenti principali:

- **Kafka-BQ Sink**: Legge i messaggi dalla coda Kafka di input e li archivia in una tabella BigQuery per garantire il salvataggio permanente dei dati inviati dalla sorgente.
- **Streaming Ingester**: Legge i messaggi dalla coda Kafka e, dopo aver validato lo schema, inserisce i dati in una tabella di BigQuery. Poiché il messaggio in ingresso può essere composto da oggetti nidificati e array, la tabella di BigQuery sarà strutturata con strutture (struct) e array.
  In caso di errori durante l'elaborazione, i messaggi vengono inviati alla "Dead Letter Queue" (DLQ) salvata su una tabella BigQuery. Per facilitare l'analisi degli errori, i

messaggi nella DLQ sono resi accessibili tramite una tabella esterna in BigQuery, semplificando la consultazione e la risoluzione dei problemi.

- **BigQuery Loader:** questo componente gestisce l'ingestion dei dati in modalità batch. È una pipeline Data Fusion che viene attivata quando un file viene scritto nel bucket designato per il caricamento dei dati. La pipeline ha il compito di avviare un job di caricamento di BigQuery per inserire i dati contenuti nel file nella tabella del layer raw di BigQuery. Questa soluzione verrà adottata per le sorgenti da cui recuperiamo i dati in full.

Il vantaggio di questa soluzione è l'utilizzo dell'evento di scrittura del file per avviare l'intero processo. Inoltre, tutti gli strumenti utilizzati sono completamente fully managed.

2.4.3 Core layer (Silver-Layer)

Il core layer ospita le tabelle contenenti i dati trasformati secondo le regole di business. Questo layer è accessibile sia dalle applicazioni interne che dagli utenti finali. È quindi necessario limitare l'accesso ai dati sensibili solo agli utenti autorizzati.

Per soddisfare questa esigenza, si sono identificate due soluzioni native di BigQuery:

- **Authorized Views:** Queste viste consentono agli utenti di accedere ai dati esposti dalla vista senza la necessità di concedere loro il permesso di lettura sulla tabella sottostante.

- **Column-level Access Control:** Questa soluzione permette di limitare la visibilità di specifiche colonne all'interno di una tabella solo a determinati utenti o gruppi autorizzati. In questo modo, evitiamo di creare strutture aggiuntive e semplifichiamo la gestione delle autorizzazioni, garantendo che solo gli utenti autorizzati possano accedere alle informazioni sensibili.

Il modello dati da utilizzare nel core layer, è stato studiato in modo tale da garantire che la struttura delle tabelle sia adeguata per soddisfare i requisiti di business e allo stesso tempo garantire buone performance durante le interrogazioni.

In particolare il compito che si è dovuto affrontare in questo livello era mantenere le logiche di strutturazione dei campi delle nuove tabelle il più simile possibile a quelle

originali. Questo poiché dal lato del business, l'azienda cliente voleva che rimanessero intatti i processi con cui oggi si interroga il database.

2.5 Dataform

Dataform di Google Cloud Platform è uno strumento progettato per aiutare i teams di sviluppo a gestire la trasformazione e l'organizzazione dei loro dati all'interno dei data warehouse di Google Cloud. È particolarmente utile nel contesto del moderno data engineering e business intelligence. Questo ambiente su GCP garantisce, ad esempio, uno spazio dove scrivere le query di creazione delle nuove tabelle e le query di popolamento dei campi delle tabelle create.

Ecco alcune caratteristiche chiave di Dataform:

- **Gestione dei Pipeline di Dati**: Dataform permette agli utenti di creare e gestire pipeline di trasformazione dei dati. Questo è essenziale per preparare i dati per analisi approfondite o per alimentare dashboard e report (nel nostro caso il nuovo ambiente su BigQuery).
- **SQLX-based**: Dataform utilizza SQLX per definire trasformazioni e logiche di business. E' un linguaggio molto simile all'SQL classico con l'aggiunta di alcune funzioni specifiche per configuarare lo script in cui si lavora o referenziare le tabelle a cui si fa riferimento durante la scrituura codice. In particolare le referenziazioni possono essere fatte a tabelle create in ambiente Dataform, ma anche tabelle già presenti su ambiente BigQuery. Questo rende lo strumento accessibile a molti data analysts e ingegneri che sono già familiari con SQL.

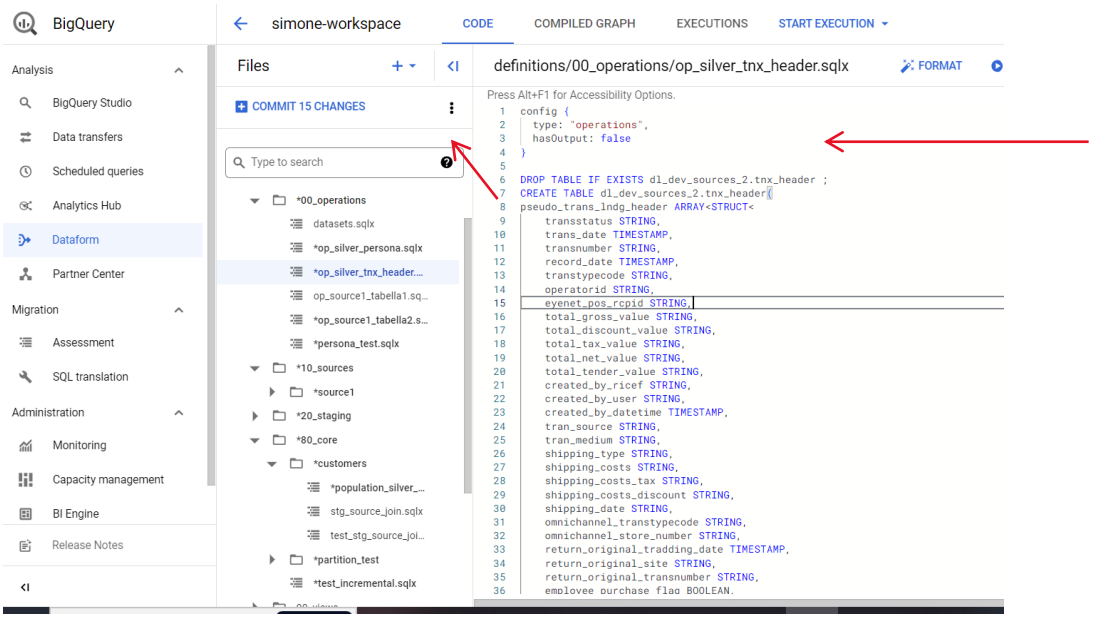


Figura 5- schermata Dataform

Esempio di referenziazione alla tabella che possiede lo stesso nome del file in cui si esegue lo script.

```
FROM ${self()} )`) }
```

- **Version Control e Collaborazione**: Dataform integra il controllo di versione, simile a Git, che consente una collaborazione efficiente tra membri del team e una migliore gestione del codice. Dataform infatti è collegato ad un repository GitHub dove viene salvato tutto il codice che è stato scritto fino a quel momento. Una volta che si esegue il commit dei vari script creati si aggiorna il repository (vedi immagine sopra, sezione commit). Questo offre la possibilità di visionare il lavoro svolto anche a persone esterne al progetto.
- **Automazione e Orchestrazione**: Dataform consente di automatizzare le routine di trasformazione dei dati e di pianificare l'esecuzione di script SQL. Ogni volta che si manda in esecuzione la pipeline creata, questo aggiornerà tutti i processi ad essa collegata (come la creazione o il popolamento dati in ambiente BigQuery)

- **Testing e Validazione dei Dati**: Offre la capacità di implementare test per garantire che le trasformazioni dei dati siano corrette e che i dati soddisfino determinate condizioni di qualità.
- **Integrazione con Google Cloud Platform**: Dataform è progettato per integrarsi perfettamente con altri servizi di Google Cloud, come BigQuery, facilitando la gestione dei dati in un ecosistema cloud.
- **Gestione degli Ambienti di Sviluppo e Produzione**: Permette di gestire facilmente ambienti multipli, come sviluppo e produzione, assicurando che le modifiche possano essere testate in modo sicuro prima di essere applicate alla base dati principale.

2.6 Generative AI tool

Grazie ad alcune settimane di ricerca sono stati testati molti tool di intelligenza artificiale. Tra i più avanzati e capaci c'è sicuramente il ramo di Open AI: ChatGPT. In particolare, grazie all'account a pagamento (20$/mese), offerto da NTT, ci si è potuti interfacciare con un plug in di ultima generazione rilasciato a luglio 2023 da Open AI : DATA ANALYSIS.

Data Analysis può ricevere file in input (excel,csv,immagini...) e rispondere alle domande in linguaggio naturale, riguardanti l'analisi del file. Può modificare il file e restituirne una copia modificata, generare script sql o nel codice richiesto e farlo scaricare in modo da essere già pronto all'uso. Questo tool permette il caricamento di file excel anche con più fogli compilati e offre la possibilità di immettere un prompt adeguato che ricavi un'analisi specifica del file caricato.

Questa AI lavora cercando di capire, eseguire, comprendere eventuali problemi di generazione, in base a codice python autogenerato. In pratica traduce in codice quello che l'utente offre sotto forma di testo e si pone interrogativi su come risolvere il lavoro in continuazione fino alla generazione di un output. La cosa intelligente è che è un processo molto guidato in cui lui stesso spiega quello che sta facendo durante l'analisi.

Nel caso avesse dei dubbi sul procedimento o se quello che sta compilando rispecchia le richieste te lo chiede direttamente tramite interfaccia e tu puoi indicargli se procedere o intraprendere altre strade a percorso iniziato. In base al prompt possiamo ricavare come output diagrammi, analisi di mercato, script di codice e tanto altro. Proprio sullo script di codice ho basato la mia attenzione perchè volevo quello come output del mio lavoro.

Infatti, essendo il database di MapR basato su una grafica molto riconducibile al foglio excel (o per lo meno il contenuto degli hub satelliti o link), sarebbe stato di grande aiuto riuscire a mappare i campi su un file xlsx e poi caricarlo sul tool per ricavare il codice sql adatto ad una ddl su GCP.

In ambiente GCP le query vanno scritte sul tab di BigQuery, che è uno spazio che mette a disposizione Google nella sua piattaforma, dove si possono creare tabelle da un dataset predefinito su un database colonnare. Nel nostro progetto invece è stato usato Dataform come tramite per generare tabelle e archiviare dati su BigQuery poichè garantisce maggiori controlli sul processo.

In pratica l'ambiente cloud in esame, una volta creato un progetto, permette di creare dataset con sede in un server google in una parte del mondo a tua scelta. Dal dataset creato e secondo lo script sql che gli viene fornito su ambiente Dataform, verrà creata la tabella corrispondente e collegata al dataset di riferimento su BigQuery. La tabella cosi generata avrà i campi mappati secondo la nostra logica, ma con i nomi coincidenti ad esempio con quelli che trovo anche su Mapr.

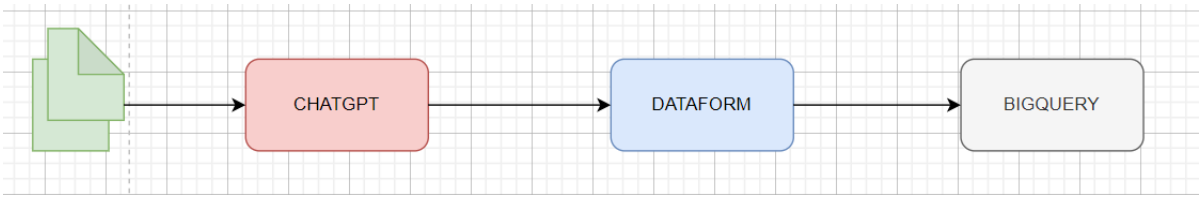


Figura 6- Diagramma flusso dati chatgpt-bigquery

Questo è stata la primissima idea di base che è stata vagliata un volta compreso il tool e seguendo le indicazioni per l'output atteso che venivano fornite dal team di progetto.

Partendo da MapR si è cercato di caricare in un file excel tutti gli hub (foglio con desinenza _h) e i satelliti (fogli con desinenza _s). Non sono stati inseriti i link poichè sono tabelle che non contengono informazioni rilevanti, ma consentono di capire le logiche di collegamento tra i vari hub nel data vault. In pratica al suo interno sono presenti le due chiavi delle entità collegate e una chiave di identificativo per l'entità link.

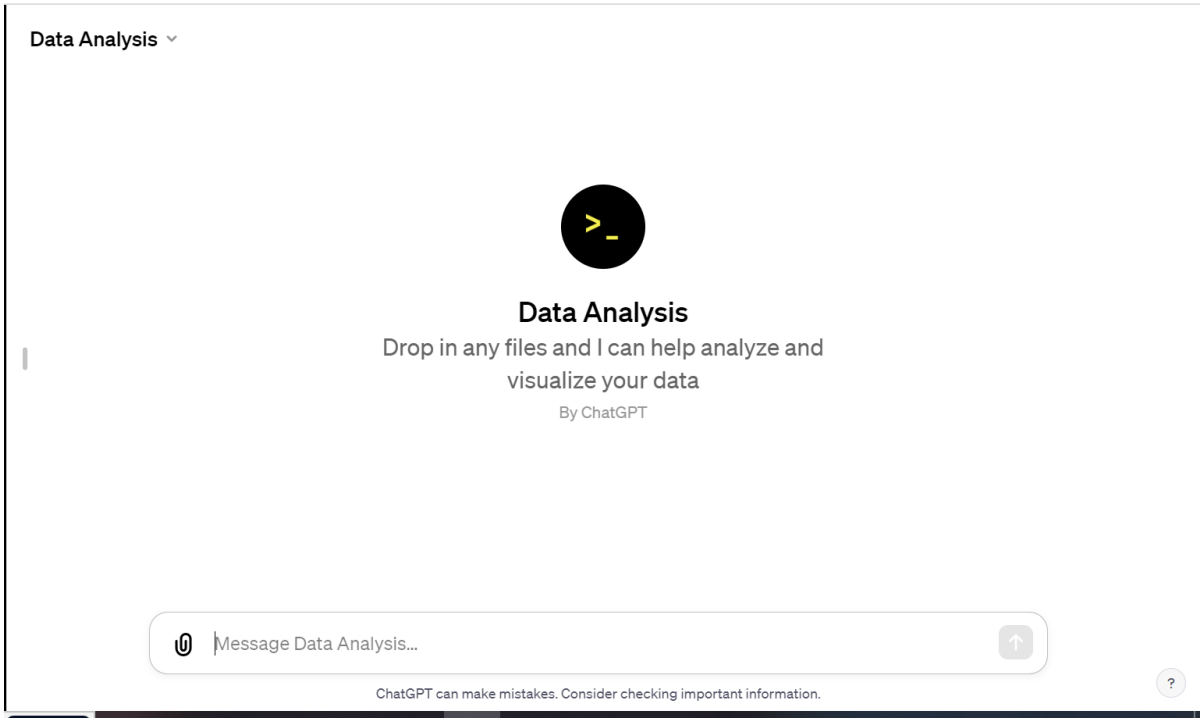


Figura 7-Interfaccia tool Data Analysis

2.7 Jira

In questa ultima parte dello stack tecnologico verranno menzionate le due tecnologie di gestione delle task e della documentazione. L'uso principale è quello di poter interfacciare tutti i membri del team reciprocamente ed essere allineati tutti col progetto. In secondo luogo garantisce la condivisione della documentazione con il cliente.

Lo strumento su cui si basava l'organizzazione è stato Jira. Quest'ultimo è un popolare strumento di gestione di progetti sviluppato da Atlassian, ampiamente utilizzato per il tracciamento di bug, la gestione delle attività e la pianificazione di progetti, specialmente in contesti di sviluppo software. Questo software consente alle squadre di organizzare compiti, tracciare il loro avanzamento e collaborare efficacemente.

In questo tool ad ogni persona venivano assegnati dei task o ticket da svolgere all'interno di uno sprint. Lo sprint è in pratica associabile a due settimane di lavoro.

Alla fine degli sprint (che coincidevano di solito con il venerdi), c'era un momento di incontro chiamato sprint-review dove si andava a vedere lo stato di avanzamento dei lavori sul progetto e si stabilivano le nuove linee guida e i nuovi task da svogere nello sprint sucessivo.

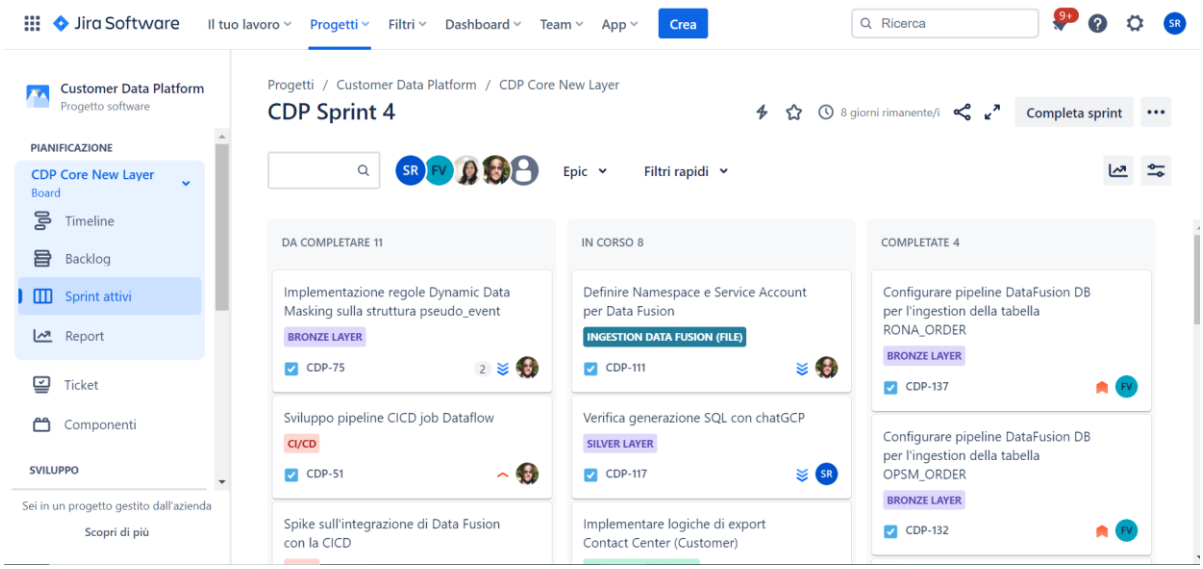


Figura 8- Jira

2.8 Confluence

Tutto il resoconto sullo stato di avanzamento del progetto e tutta la documentazione che veniva prodotta di settimana in settimana e che teneva traccia del lavoro svolto, veniva caricata su Confluence.

Confluence è un software di collaborazione sviluppato da Atlassian, pensato per aiutare le squadre a creare, condividere e organizzare il lavoro e la documentazione in un unico luogo centralizzato. È particolarmente popolare tra le squadre di sviluppo software ma è ampiamente utilizzato anche in altri contesti aziendali per la gestione della conoscenza e la collaborazione di progetto.

Questo strumento permette agli utenti di creare pagine e spazi di lavoro personalizzati dove possono documentare progetti, idee, riunioni e altro ancora. Confluence supporta una vasta gamma di contenuti, inclusi testo, tabelle, script di codice, immagini e video, e offre funzionalità avanzate di ricerca per trovare facilmente le informazioni. La sua integrazione con altri strumenti di Atlassian, come Jira, ne amplifica l'efficacia, permettendo una collaborazione fluida e un allineamento tra i team (in questo caso tra NTT e il cliente).

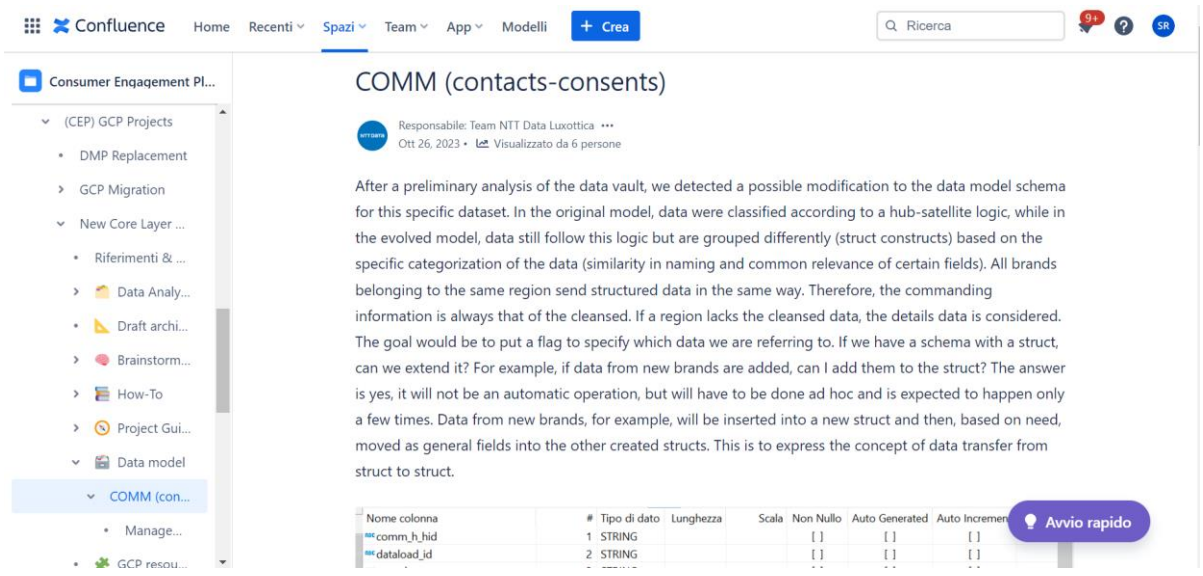


Figura 9- Confluence

3 Sviluppo e Prototipazione

La parte seguente comprende tutti i processi che erano presenti, che sono stati implementati e che saranno usati in seguito durante tutto il processo di mappatura e analisi dati.

3.1 As is

L'approccio all'analisi dati che al momento si stava utilizzando all'interno di NTT è stato quello di utilizzare script python che ricavassero da file excel strutturati, il corrispettivo codice customizzato secondo le linee guida fornite. Inoltre, per ricavare il file excel strutturato che rappresentasse tutta la parte dei file streaming provenienti dal bronze-layer, si doveva scaricare il json completo e poi, per utilizzare solamente il "name" dei vari campi del json, bisognava costruire un codice python ad hoc. Questo codice doveva essere in grado di ricavare anche i "name" all'interno delle struct annidate. Questo processo richiede molto molto tempo: si parla di settimane di lavoro.

L'obiettivo principale era capire in quale fase di questo processo potesse intervenire la generative AI e quale aiuto potesse fornire, sia in quantità di tempo risparmiato che velocizzazione del processo.

3.2 Obiettivo

Per riuscire a creare una metodologia nel lavoro e nell'utilizzo di questa AI si è dovuto capire quale erano le esigenze del tool e su cosa performasse meglio. Infatti, nel lavoro di analisi sono state identificate tre possibili sorgenti dati che avrebbero dovuto alimentare il silver-layer. La prima riguarda i dati provenienti da MapR, la seconda dai dati in streaming del bronze layer e la terza dai dati batch. Concentrandosi sulla prima risulta che il data vault costruito su MapR si basi su un file json unico che è anche lo stesso che alimenta il bronze layer nuovo. C'è una differenza però: su Mapr il json è stata la base di costruzione del data vault. A questo sono stati aggiunti molti campi e anche cambiati i nomi di quelli esistenti una volta che venivano aggiunte nuove informazioni o nell'azienda venivano mappati i dati da nuovi brand. Nel bronze layer invece i campi del json sono immutati e quindi riportano la nomenclatura originale.

Questo rappresenta un primo ostacolo, poichè, come si è detto in precedenza, l'obiettivo è quello di lasciare il più possibile la struttura dati simile a MapR. La soluzione quindi è collegare i campi del json che alimenta il bronze con i campi specifici su MapR.

Per questo ci serve una struttura dati completa e chiara dei dati su entrambi gli ambienti.

Questa fase di collegamento richiede un'analisi accurata e un approccio meticoloso per garantire che tutti i dati siano correttamente mappati e integrati. È un compito che richiede una profonda comprensione sia della struttura dati esistente che delle esigenze aziendali, oltre a un'attenta pianificazione per evitare eventuali disallineamenti o perdite di dati critici.

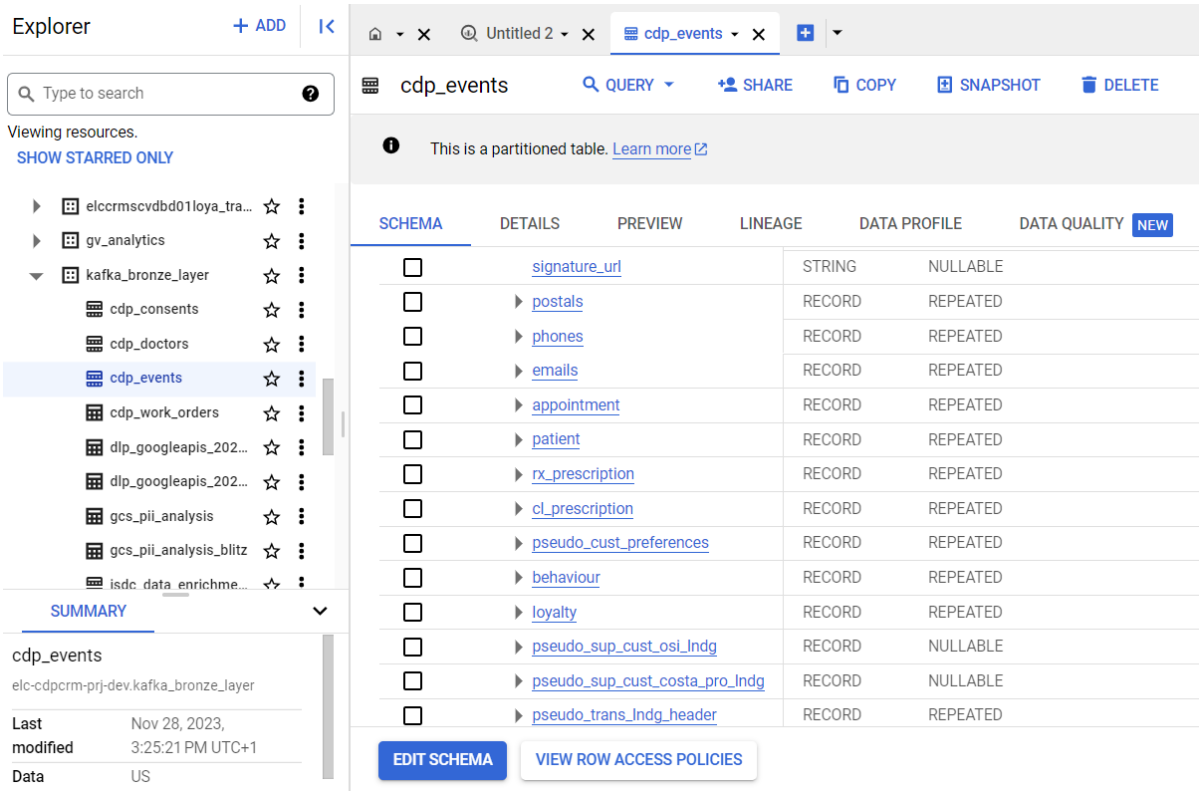


Figura 10-bronze-layer gcp

La figura riporta una parte del bronze layer su BigQuery. É un database colonnare dove sono presenti array strutturati con più livelli di annidamento. É un database abbastanza articolato ed è per questo che per semplificare molta parte computazionale si è cercato di fare uso della generative ai dove possibile.

Per ritrovare le logiche di strutturazione dei dati su MapR si possono riguardare le figure 1,2,3.

3.3 To-be

Dopo varie prove di tool di generative AI (chat-gpt 3.5, Bard, Duet AI, chat-gpt 4.0) i primi risultati concreti si sono ottenuti utilizzando il plug in di Chat-GPT 4.0: Data Analysis.

Non è un tool gratuito. Per poterlo utilizzare bisogna sottoscrivere un abbonamento mensile alla versione plus di ChatGPT dal costo di 20$ al mese.

Il tool è stato applicato in vari punti del processo.

Il primo ha consentito di trasformare il file json della parte streaming in un file excel strutturato secondo le linee guida che gli erano state fornite. La parte di scrittura del codice python non sparisce, ma è svolta in autonomia da chat gpt per analizzare il file json in input e produrre un file excel.

Il secondo punto in cui è intervenuta l'AI è stato nel trasformare parti dell'excel in vere e proprie query sql adatte a BigQuery secondo prompt specifici.

Oltre a query di creazione di tabelle, la parte più corposa e importante in cui ha operato è stata la generazione di query di popolamento dei campi. Dal dal bronze-layer era importante che generasse l'sql corretto per popolare i campi delle tabelle del silver-layer mantenendo la nomenclatura presente su mapr.

I risultati ottenuti sono stati soddisfacenti e hanno dato un forte sviluppo a questa parte del progetto, che nel vecchio procedimento avrebbero richiesto sicuramente molto più tempo.

Più tempo non nella creazione vera e propria del codice, ma nella creazione del python che avrebbe poi fornito l'sql.

3.4 Diagramma flusso dati

Il diagramma in questione offre una rappresentazione dettagliata e strutturata degli elementi chiave e dei processi coinvolti nella creazione del Silver-layer all'interno del progetto. Questo strato rappresenta una fase cruciale nel flusso di lavoro dei dati, dove l'informazione viene raffinata e preparata per analisi più avanzate nel layer successivo.

Il punto di partenza del diagramma è rappresentato da un'icona arancione, simbolo delle sorgenti dati. Queste sorgenti includono dati provenienti da MapR, dal bronze-layer e dalle sorgenti batch. Il bronze-layer funge da strato iniziale di raccolta dati, dove le informazioni grezze vengono immagazzinate. Importante in questa fase è la mappatura dei campi dati in un modello entity/relationship, che stabilisce le relazioni fondamentali tra i diversi tipi di dati e facilita la loro manipolazione nelle fasi successive.

Procedendo nel flusso, il diagramma mostra come i dati vengono poi trasferiti in strumenti come Excel e come si generi codice in linguaggio di programmazione SQL. In questa fase entra in gioco la generative AI. L'intelligenza artificiale generativa qui ha il ruolo di assistere nell'elaborazione e nella trasformazione dei dati, rendendo il processo più efficiente e meno soggetto a errori umani.

Le query SQL create vengono quindi inserite in Dataform: un orchestratore che si occupa di gestire e coordinare le varie fasi di elaborazione e integrazione dei dati. Dataform gioca un ruolo fondamentale nell'assicurare che il processo di trasformazione dei dati sia fluido e ben organizzato, facilitando la transizione dei dati dal bronze-layer al silver-layer.

Infine, il diagramma conclude con l'icona di BigQuery, indicando che è in questa piattaforma che si trova la sede finale delle tabelle create per il Silver-layer. BigQuery risulta essere una soluzione di data warehouse di Google Cloud che fornisce un ambiente potente e scalabile per l'analisi e l'archiviazione dei dati. Qui, i dati del silver-layer vengono conservati, pronti per essere utilizzati in analisi complesse e per supportare la presa di decisioni basate sui dati.

In sintesi, questo diagramma illustra un flusso di lavoro ben definito e integrato, che va dalla raccolta iniziale dei dati fino alla loro elaborazione e archiviazione nel silver-layer.

Ogni passaggio è fondamentale per garantire che i dati siano non solo accurati e ben organizzati, ma anche prontamente disponibili per analisi e applicazioni successive.

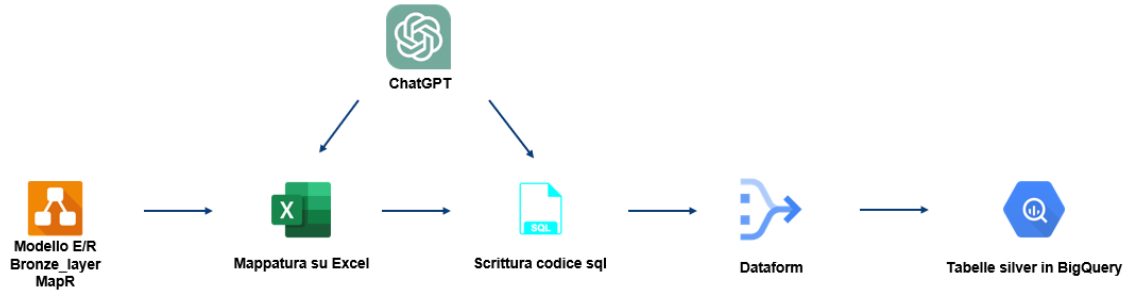


Figura 11-diagramma flusso dati

3.5 Prompt engineering

Per comprendere quali siano stati l'approccio e la metodologia utilizzati nel progetto, bisogna illustrare quali siano le caratteristiche di questa parte della scienza che sta alla base dell'intelligenza artificiale utilizzata. Proprio perché ogni input fornito all'IA possiede una parte scritta sottoforma di testo, di seguito verrà analizzato questo approccio.

Il *prompt engineering* è il processo di creazione e messa a punto del testo di input (o "prompt") fornito a uno strumento di intelligenza artificiale come ChatGPT per ottenere risultati molto migliori e per risolvere meglio i problemi.

Il prompt si occupa di dare le migliori indicazioni possibili in modo da poter ottenere esattamente quello che si sta cercando. L'obiettivo del prompt engineering è controllare l'output del modello linguistico (strumento AI) fornendogli un contesto specifico, vincoli, regole, ecc. Questo può essere fatto creando manualmente e in modo preciso il testo di input o fornendo informazioni aggiuntive come parole chiave o etichette insieme al testo di input per ChatGPT. Nel caso di studio di cui si occupa questo elaborato sono stati testati sia i soli input di linguaggio naturale, sia gli input di testo e file e in ultimo anche gli input visivi uniti ai primi due. Il processo di prompt engineering può essere iterativo. Inoltre, l'output del modello analizzato e il prompt possono essere adattati in base ai risultati. Ormai si riesce anche a rigenerare domande e risposte imparando molto anche attraverso quest'ultime.

Un *prompt engineer* è una persona che si concentra sull'inserimento di prompt, cioé regole e indicazioni ben congegnati per ottenere risultati specifici dagli strumenti di intelligenza artificiale. Ciò può comportare testo di input avanzato, etichette specifiche o strategie di input e molto altro. Implica anche la comprensione delle capacità e dei limiti del modello o dello strumento di intelligenza artificiale che si sta utilizzando e la creazione del testo di input in modo da guidare l'output del modello verso un obiettivo molto specifico. I prompt engineer lavorano su progetti di elaborazione del linguaggio naturale (NLP) e sono responsabili della progettazione e della creazione dei prompt a cui i modelli risponderanno, della messa a punto dei modelli in base all'output e dell'analisi regolare delle prestazioni dei modelli per migliorare la loro prestazione È proprio questo ruolo che si è cercato di emulare e prendere in consideranzione nel processo di "trial and error" che si è seguito interfacciandosi con questa nuova tecnologia.

Il prompt engineering è l'aspetto più cruciale dell'utilizzo efficace degli LLM (large language model, ovvero i linguaggi di addestramento dei vari algoritmi di intelligenza artificiale) ed è un potente strumento per personalizzare le interazioni con ChatGPT. Si tratta di creare istruzioni o domande chiare e specifiche per ottenere le risposte desiderate dal modello linguistico. Costruendo attentamente le richieste, gli utenti possono guidare l'output di ChatGPT verso gli obiettivi previsti e garantire risposte più accurate e utili.

I prompt sono strumenti essenziali per facilitare la comunicazione senza soluzione di continuità con i modelli linguistici dell'intelligenza artificiale. Per creare prompt di alta qualità, bisogna prima capire come vengono classificati. Ciò consente di strutturarli in modo efficace concentrandosi su una risposta target specifica. Le principali categorie di prompt includono:

a. **Richieste di ricerca di informazioni**:
   questi suggerimenti sono realizzati per raccogliere informazioni ponendo domande "Cosa" e "Come". Sono ideali per estrarre dettagli o fatti specifici dal modello AI.
b. **Prompt basati su istruzioni**:
   i prompt basati su istruzioni indirizzano il modello AI a eseguire un'attività specifica (sulla falsa riga di come funzionano Alexa e Siri).
c. **Suggerimenti che forniscono contesto**:
   questi suggerimenti forniscono informazioni contestuali al modello AI, consentendogli di comprendere meglio la risposta desiderata dall'utente.
d. **Suggerimenti comparativi**:
   i suggerimenti comparativi vengono utilizzati per valutare o confrontare diverse opzioni, aiutando gli utenti a prendere decisioni informate. Sono particolarmente utili quando si valutano i pro e i contro delle varie alternative.
e. **Spunti per la ricerca di opinioni**:
   questi suggerimenti suscitano l'opinione o il punto di vista dell'IA su un determinato argomento. Possono aiutare a generare idee creative.

Nell'ambito dell'intelligenza artificiale, la progettazione di prompt assume un ruolo cruciale, specialmente quando si utilizzano modelli linguistici avanzati come ChatGPT. L'efficacia e l'accuratezza delle risposte fornite da questi modelli sono notevolmente influenzate dalla qualità dei prompt inseriti. Ciò richiede una profonda comprensione delle capacità e delle limitazioni del modello in questione. Questi modelli, nonostante la loro avanzata tecnologia, possono presentare lacune in specifici compiti o generare informazioni non corrette. La conoscenza di questi aspetti consente agli utenti di formulare prompt che massimizzino le potenzialità del modello, riducendo al contempo il rischio di errori. Un altro fattore determinante è l'intento dell'utente. La capacità di interpretare e rispondere in maniera pertinente a quest'ultimo fattore è una qualità fondamentale di questi modelli. I prompt dovrebbero quindi riflettere chiaramente le aspettative dell'utilizzatore, permettendo al modello di fornire risposte pertinenti e corrette.

La chiarezza e la specificità del prompt sono essenziali per minimizzare ambiguità e incertezze, che possono altrimenti portare a risposte inadeguate. Inoltre, quando si tratta di domini altamente specializzati, è importante utilizzare un vocabolario o un contesto specifici, per guidare il modello verso risposte più accurate e rilevanti. L'aggiunta di contesto o di esempi specifici può migliorare significativamente la pertinenza e l'accuratezza delle risposte. In questo elaborato infatti sono presenti molti esempi sia grafici che di prompt proprio per illustrare al meglio come sia avvenuto questo processo di utilizzo della tecnologia.

Infine, la definizione di eventuali limitazioni, come la lunghezza della risposta o un formato strutturato, può essere fondamentale per assicurare che le risposte del modello soddisfino esigenze specifiche. Queste limitazioni possono essere stabilite attraverso specifiche direttive inserite nel prompt.

In conclusione, la progettazione di prompt efficaci per modelli come ChatGPT richiede una comprensione approfondita delle sue capacità e limitazioni, oltre alla capacità di interpretare accuratamente l'intento dell'utente. La chiarezza, la specificità e la considerazione del dominio specifico sono elementi cruciali per garantire risposte accurate e pertinenti. La continua evoluzione e il perfezionamento nel campo del prompt

engineering sottolineano l'importanza di questa pratica per massimizzare l'utilità e l'interattività dei modelli linguistici avanzati.

Continuando nell'analisi della tesi ci si focalizzerà sulla struttura dei prompt usati e su come sono stati forniti gli input per l'analisi e il loro contenuto strutturato. Successivamente verranno presentati i risultati con le considerazioni pertinenti scaturite.

3.6 Bronze Layer to Silver Layer

Come scritto nei capitoli precedenti, era necessario capire le logiche di match tra i dati del bronze e MapR per riuscire a costruire un silver layer ottimale. In questo caso è stato estratto il json delle tabelle caratterizzanti il bronze layer (si riporta come esempio solo la tabella della cdp_events sotto il dataset kafka_bronze_layer- figura 10).

Il file json è stato poi caricato in un file di testo con estensione .txt e caricato su chatgpt scrivendo il seguente prompt.

3.6.1 Istruzioni da json a xlsx per tabella con livelli di annidamento

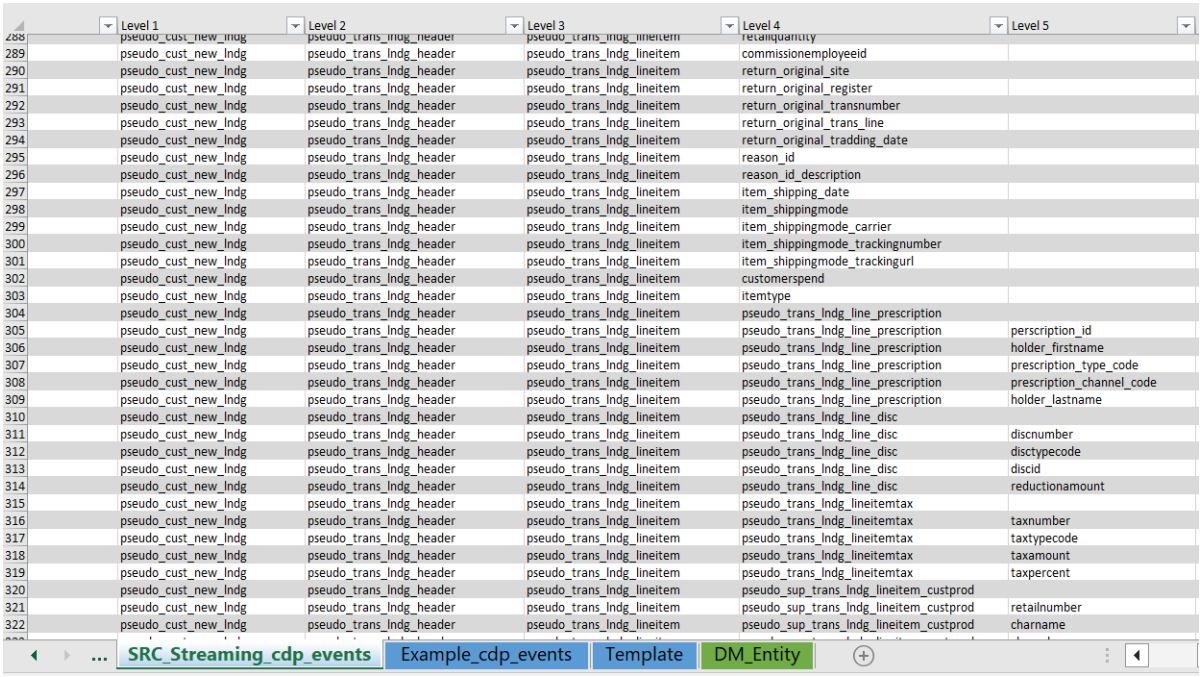
"Considera solo gli attributi name da riportare nel file excel e poi ogni colonna del file deve avere un nome progressivo per ogni livello di annidamento che trovi. Ad esempio trovato il primo annidamento metti tutti i valori contenuti nel primo annidamento sotto la colonna di nome level 1, poi il secondo annidamento che trovi lo metti nel level 2 se è contenuto nel primo e cosi via

Ogni volta che incontri nel json una tabulazione in avanti signifca che c'è stato un annidamento dei campi successivi in quello precedente, mentre se incontri una tabulazione all'indietro vuol dire che l'annidamento è finito e ne può stare iniziando un altro oppure no....sistema l'excel unendo queste informazioni a quelle di prima."

3.7 Output

Lo scopo è quello di riportare in un file excel solo i nomi dei campi riportati nel json e con i livelli di annidamento ben evidenziati.

Ci si aspetta un excel ben strutturato che riporti il lineage del dato nei diversi campi annidati in modo da avere una mappatura completa del dato nel bronze e dove trovarlo.



	Level 1	Level 2	Level 3	Level 4	Level 5
289	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	commissionemployeeid	
290	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	return_original_site	
291	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	return_original_register	
292	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	return_original_transnumber	
293	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	return_original_trans_line	
294	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	return_original_tradding_date	
295	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	reason_id	
296	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	reason_id_description	
297	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	item_shipping_date	
298	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	item_shippingmode	
299	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	item_shippingmode_carrier	
300	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	item_shippingmode_trackingnumber	
301	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	item_shippingmode_trackingurl	
302	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	customerspend	
303	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	itemtype	
304	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	pseudo_trans_lndg_line_prescription	
305	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	pseudo_trans_lndg_line_prescription	perscription_id
306	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	pseudo_trans_lndg_line_prescription	holder_firstname
307	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	pseudo_trans_lndg_line_prescription	prescription_type_code
308	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	pseudo_trans_lndg_line_prescription	prescription_channel_code
309	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	pseudo_trans_lndg_line_prescription	holder_lastname
310	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	pseudo_trans_lndg_line_disc	
311	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	pseudo_trans_lndg_line_disc	discnumber
312	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	pseudo_trans_lndg_line_disc	disctypecode
313	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	pseudo_trans_lndg_line_disc	discid
314	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	pseudo_trans_lndg_line_disc	reductionamount
315	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	pseudo_trans_lndg_lineitemtax	
316	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	pseudo_trans_lndg_lineitemtax	taxnumber
317	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	pseudo_trans_lndg_lineitemtax	taxtypecode
318	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	pseudo_trans_lndg_lineitemtax	taxamount
319	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	pseudo_trans_lndg_lineitemtax	taxpercent
320	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	pseudo_sup_trans_lndg_lineitem_custprod	
321	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	pseudo_sup_trans_lndg_lineitem_custprod	retailnumber
322	pseudo_cust_new_lndg	pseudo_trans_lndg_header	pseudo_trans_lndg_lineitem	pseudo_sup_trans_lndg_lineitem_custprod	charname

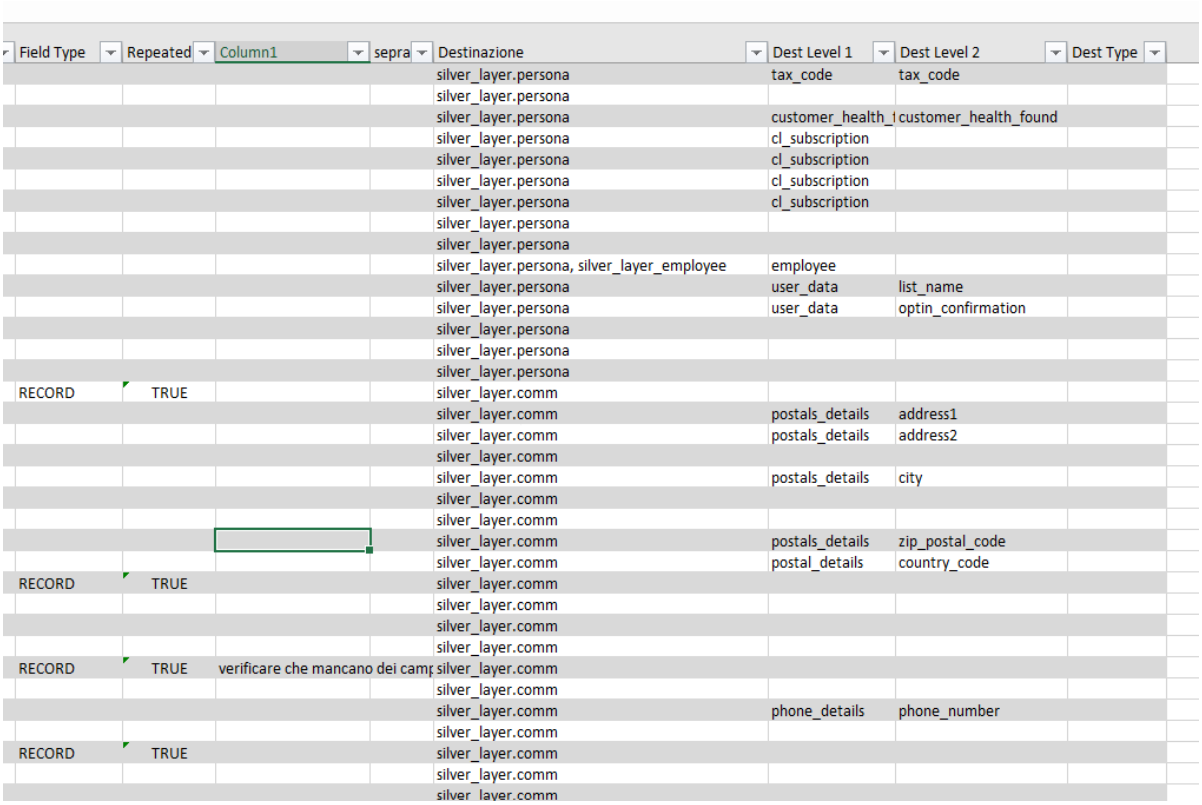
SRC_Streaming_cdp_events | Example_cdp_events | Template | DM_Entity

Figura 12- schermata excel ChatGPT4

Il risultato è in parte questo. Non si riporta l'excel completo poiché sono più di quattrocento campi mappati e non serve elencarli tutti allo scopo di illustrare il lavoro della generative AI.

Rifacendosi sempre alla figura 10 e confrontandola con questa immagine si nota che tutti i campi del record pseudo_trans_lndg_header sono stati spacchettati e posizionati su righe diverse dell'excel, pur mantenendo tutto il lineage del dato (cioé da quale altro record provengono). Ogni annidamento è riportato nella colonna con nome uguale al livello di annidamento (Level 1, Level 2....).

In questo excel sono poi state implementate altre colonne che identificavano se il nome della riga appartenesse ad un record repeated o meno, una colonna di commenti (column 1), una colonna rappresentante la destinazione del dato nel silver-layer e tre colonne ulteriori per identificare dove lo stesso dato fosse presente su MapR (Dest Level 1, Dest Level 2 e Dest Type).



Field Type	Repeated	Column1	sepra	Destinazione	Dest Level 1	Dest Level 2	Dest Type
				silver_layer.persona	tax_code	tax_code	
				silver_layer.persona			
				silver_layer.persona	customer_health_f	customer_health_found	
				silver_layer.persona	cl_subscription		
				silver_layer.persona	cl_subscription		
				silver_layer.persona	cl_subscription		
				silver_layer.persona	cl_subscription		
				silver_layer.persona			
				silver_layer.persona			
				silver_layer.persona, silver_layer_employee	employee		
				silver_layer.persona	user_data	list_name	
				silver_layer.persona	user_data	optin_confirmation	
				silver_layer.persona			
				silver_layer.persona			
				silver_layer.persona			
RECORD	TRUE			silver_layer.comm			
				silver_layer.comm	postals_details	address1	
				silver_layer.comm	postals_details	address2	
				silver_layer.comm			
				silver_layer.comm	postals_details	city	
				silver_layer.comm			
				silver_layer.comm			
				silver_layer.comm	postals_details	zip_postal_code	
				silver_layer.comm	postal_details	country_code	
RECORD	TRUE			silver_layer.comm			
				silver_layer.comm			
				silver_layer.comm			
				silver_layer.comm			
RECORD	TRUE	verificare che mancano dei camp		silver_layer.comm			
				silver_layer.comm			
				silver_layer.comm	phone_details	phone_number	
				silver_layer.comm			
RECORD	TRUE			silver_layer.comm			
				silver_layer.comm			
				silver_layer.comm			

Figura 13- schermata excel ChatGPT4

3.8 Generazione script python per standardizzare le procedure di generazione query sql

Una volta validato il prompt, ottenendo quindi l'output sperato si è chiesto all'IA di riportare il codice python che era stato autogenerato per poter produrre il file excel richiesto. Lo scopo era riuscire a ricavare un codice standard per questa procedura che risulterà nel corso del progetto molto ripetitiva e ricorrente. In questo modo si è creato uno script python in grado di generare un excel da un file json in meno di un minuto. L'alternativa era quella di parsare il json con python e poi usare il metodo .keys che restituisce il nome. Si otterrebbero tutti i nomi dei campi. L'altra sfida sarebbe stata quella di capire come estrarre i nomi dei campi annidati, ma con la procedura tramite ChatGPT si è evitato a monte questo eventuale problema.

Il processo "old style" richiederebbe sicuramente una quantità di tempo aggiuntivo molto elevata rispetto al modello con l'IA.

Nell'appendice nella sezione 8.5 si trova il codice python generato e che permette di realizzare file xlsx da file json.

3.9 Dall’excel allo scipt sql

Avendo struttuato un file excel complesso e ordinato, il passo successivo sarebbe stato quello di verificare se e come l’IA avesse potuto generare un codice sql preciso secondo un prompt adeguato avendo come input il file excel precedente. L’idea principale è stata quella di chiedere di inserire nel codice solo l’ultimo livello popolato di ogni riga e in base al tipo della colonna Field Type e Repeated, seguire delle best practices specifiche di scrittura del codice. In base alla colonna Destinazione invece si selezionavano solo le righe che si volevano mettere nella tabella relativa del silver-layer.

3.9.1 Prompt

“Considera il file excel mapping, in particolare il foglio SRC_Streaming_cdp_events.

Ignora le colonne : "Entità Sorgente", "Repeated", "Column1", "separator", "Destinazione",

"Dest Level 1", "Dest Level 2" e "Dest Type".

Le colonne "Level 1", "Level 2", "Level 3" etc... come livelli di annidamento di uno schema bigquery.

Ogni livello può rappresentare o un campo finale o una struct che contiene n campi.

Se la colonna "Field Type" è uguale a "Record" il livello rappresenta una struct.

Se la colonna "Field type" è vuota o "ANY" il livello rappresenta un campo della struct di tipo string.

Esempio di strutturazione della query per BigQuery:

Considera il foglio Example_cdp_events.

Quindi se ad esempio nella riga 2 dell'Example_cdp_events ho al Level 1 pseudo_cust_new_lndg e al Level 2

pseudo_trans_lndg_header, al Level 3 ho vuoto dovrò mettere nell'sql solo l'ultimo

campo trovato che equivale a pseudo_trans_lndg_header.

Vedo che però pseudo_trans_lndg_header ha un Field Type di tipo "RECORD" e la colonna "Repeated" true, allora faccio un array struct(pseudo_work_order ARRAY<STRUCT<>>).

Nel caso avessi trovato la colonna "Repeated"con scritto false o vuoto avrei messo

solo il costrutto STRUCT<> e non anche l'ARRAY.

Quindi all'interno della struct anniderò i campi successivi.

Vado alla riga 3 e trovo al Level 1 pseudo_cust_new_lndg, al Level 2 pseudo_trans_lndg_header, al Level 3 pseudo_work_order e al Level 4

trovo transstatus e al Level 4 trovo vuoto quindi in questa riga di sql dovrò inserire solo il transstatus con tipologia string nella STRUCT.

Questo esempio deve servirti per ripetere le stesse operazioni per tutte le altre righe dell'excel

per generare l'sql.Generami quindi lo script sql.

Nello script sql devi mettere sempre il nome del campo della riga dove ha il livello più alto

(se i livelli vanno da 1 a 6 uno è il più basso e 6 il più alto).

Tenendo la scrittura delle struct come nell'esempio fornito per l'annidamento dei campi.

Ora prendi in analisi anche la colonna "Destinazione" e crea una tabella in uno script sql scaricabile in formato colonnare,

ma con le stesse regole precedenti per la destinazione silver_layer.tnx_header.

Se la riga ha come destinazione silver_layer.tnx_header allora metti il campo in questa nuova tabella.

Generami l'sql della tabella adatta a bigquery generata e tramite link fammi scaricare lo script."

3.10 Considerazioni risultati

L'output prodotto è stato validato per due tabelle specifiche: la silver-layer.persona e la silver-layer.comm.

Infatti l'output prodotto è stato adattato alle configurazioni di dataform per quanto riguarda la prima riga di create della tabella e la referenziazione del dataset, ma tutto il corpo della query è stato prodotto dall'IA.

Ora ci si è dovuti confrontare con la generazione di due tabelle più complesse: la tnx-header e la tnx-detail.

Alcuni campi relativi a queste tabelle, nel json si trovavano anche al quinto livello di annidamento e l'IA con l prompt che è stato mostrato in precedenza al capitolo 3.9.1 non era in grado di restituire un output corretto.

Dopo alcune ulteriori prove si è ottenuto il risultato sperato a cui sono state fatte solo poche modifche a livello di indentazioni o parentesi.

4 Instradamento dati Bronze-layer – Silver-layer

4.1 Mappatura dati MapR

Per poter avere un quadro completo dei dati che dovevano essere collegati tra il bronze e MapR proprio come si è fatto col bronze, bisognava rimappare in modo più strutturato e chiaro i dati del vault.

L'idea è stata quella di strutturare questi dati proprio come sarebbero apparsi nel silver-layer e poi cercare di riadattare i dati del bronze sulle logiche nuove del silver, ma con la nomenclatura di MapR.

Questo procedimento non si è potuto fare con la generative AI, ma serve illustrarlo per capire poi come è stato implementato il silver-layer finale.

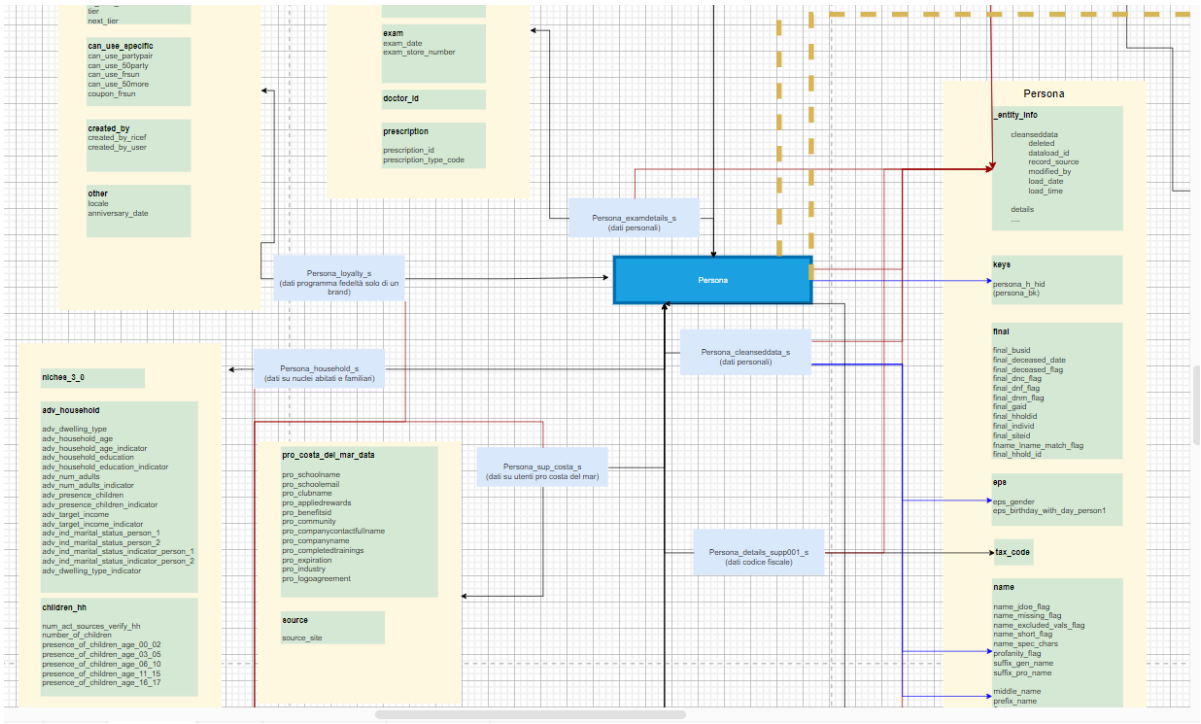


Figura 14- schermata draw.io

A livello centrale in blu è rappresentato l'entità hub di MapR, ma che sarà anche il nome della tabella del silver.

In azzurro chiaro sono riportati i nomi dei satelliti che però non compariranno nel nuovo silver layer.

In verde invece sono riportati i campi che apparterrano al silver layer dopo gli opportuni match con quelli del bronze.

Il titolo in nero sarà il nome della struct a cui apparterranno i campi raggruppati corrispettivi e sarà anche il nome che nel file excel comparirà sotto la colonna Dest Level 1.

I campi non in grassetto infine sono i campi di MapR riorganizzati e che nell'excel compariranno sotto la colonna Dest Level 2.

4.2 Logiche di instradamento dati

Nell'appendice nella sezione dei codici (8.1-8.2-8.3-8.4) sono riportati solamente i codici di creazioni delle tabelle realizzati tramite AI. Sono riportati solamente 4 esempi di codice poichè il procedimento per la creazione delle restanti segue sempre lo stesso schema riportato in precedenza, mentre le query di popolamento dei campi delle suddette tabelle viene fatto in un altro spazio. Il tutto viene prodotto in ambiente Dataform.

In questo ambiente confluiranno i dati dei tre flussi e ne usciranno tutti e tre in una tabella specifica del silver_layer. Si tenga conto ad esempio dei tre flussi dei dati principali: streaming, batch e quelli provenienti da mapr. Tutti questi tre devono confluire nel silver_layer e infatti bisogna implementare un sistema di pipeline su Dataform che permetta di raccogliere i dati dalle tre sorgenti e confluirli tutti nelle tabelle specifiche del silver_layer. Prendendo ad esempio in analisi l'hub persona presente su mapr, avrò dei dati streaming che vengono forniti dal json che alimenta mapr e il bronze layer. In aggiunta avrò dei dati batch sempre su persona (nel caso dell'azienda i dati relativi al satellite persona_cleanseddata_s) cioè campi di dati che vengono inviati ad enti terzi esterni che li popolano o li validano. Questi dati arrivano con cadenza periodica definita dall'azienda con l'ente (solitamente giornaliera, ma potrebbe essere anche mensile o settimanale). In ultimo c'è il flusso di dati presente su mapr (sempre riguardante l'hub persona) che in parte è diverso da quello streaming, poichè sono stati aggiunti negli anni dati nuovi su brand acquisiti. Tutti questi flussi dovranno poi essere riuniti nel silver_layer_persona.

La parte complessa sarà quella di quadrare tutti e tre i flussi cercando di mantenere la dicitura dei campi uguale a mapr considerando che già nel json ci sono molte differenze.

In sintesi nel silver_layer si adotterà la scrittura dei campi di mapr. I dati del json o del flusso batch se presentano divergenze andranno rinominati secondo la dicitura di mapr.

4.3 Matching Bronze-layer-MapR-Silver-layer

L'ultimo step per l'instradamento corretto dei dati dal bronze-layer al silver-layer è stato quello di creare delle queries di popolamento che prendessero i dati da una sorgente e li posizionassero nei campi corrispondenti delle varie struct create nel silver-layer.

Anche in questa fase del processo la generative Ai ha fornito una grossa mano al lavoro completo. Infatti, si sono prese porzioni dell'excel globale e si è scomposto in tanti pezzetti, ciascuno corrispondente ai campi da portare in una tabella specifica del silver.

A questo punto si sono prese in considerazione solo le colonne riportanti i campi del bronze-layer e la colonna rappresentati i nomi delle struct e i nomi dei campi delle struct corrispondenti nel silver-layer.

Si è quindi chiesto a chat gpt di creare una query sql che collegasse i campi del layer bronze (Level 4, Level 5) nel silver (Dest Level 1, Dest Level 2) (Appendice 8.6 e 8.7).

Si sono presi da esempio due hub più ridotti e semplici da visualizzare come cl-prescription e rx-prescription, ma il lavoro e il risultato ottenuto è stato lo stesso anche per gli hub persona, comm, tnx-header e tnx-detail.

Level 4	Level 5	Dest Level 1	Dest Level 2
cl_left_lenscode		left_lens	left_lens_code
cl_left_pricetype		left_lens	left_lens_price_type
cl_left_quantity		left_lens	left_lens_quantity
cl_left_tintcode		left_lens	left_lens_tint_code
cl_prescription_date		prescriptioncl	prescription_date
cl_prescription_issue_date		prescriptioncl	prescription_issue_date
cl_prescription_number		prescriptioncl	prescription_number
cl_prescription_origin		prescriptioncl	prescription_origin
cl_provider_number		other_cl	required_date
cl_required_date		other_cl	provider_number
cl_right_lenscode		right_lens	right_lens_code
cl_right_pricetype		right_lens	right_lens_price_type
cl_right_quantity		right_lens	right_lens_quantity
cl_right_tintcode		right_lens	right_lens_tint_code
cl_contactlens_details			
cl_contactlens_details	cl_eye	cl	cl_eye
cl_contactlens_details	cl_eyenumber	cl	cl_eyenumber
cl_contactlens_details	cl_despar_code	cl	cl_despar_code
cl_contactlens_details	cl_despar_value	cl	cl_despar_value

Figura 15-excel cl-prescription

Level 4	Dest Level 1	Dest Level 2
rx_prescription_type	prescription	prescription_type
rx_prescription_date	prescription	prescription_date
rx_prescription_number	prescription	prescription_number
rx_prescription_issue_date	prescription2	prescription_issue_date
rx_prescription_expiration_date	prescription2	prescription_expiry
rx_comment	prescription	prescription_comment
rx_job_number	prescription2	prescription_job
rx_doctor_id	doctor_id	doctor_id
rx_firsttime_wear_flag	flag	first_time_wearer_flag
rx_medical_cond_flag	flag	medical_condition_flag
rx_left_lens_add	left_lens	left_lens_add
rx_left_lens_axis	left_lens	left_lens_axis
rx_left_lens_cylinder	left_lens	left_lens_cylinder
rx_left_lens_io_prism	left_lens	left_lens_prism
rx_left_lens_sphere	left_lens	left_lens_sphere
rx_left_lens_ud_prism	left_lens	left_lens_udprism
rx_left_lens_pd_far	left_lens	left_lens_pd_far
rx_left_lens_pd_near	left_lens	left_lens_pd_near
rx_right_lens_add	right_lens	right_lens_add
rx_right_lens_axis	right_lens	right_lens_axis
rx_right_lens_cyl	right_lens	right_lens_cylinder
rx_right_lens_io_prism	right_lens	right_lens_prism
rx_right_lens_sphere	right_lens	right_lens_sphere
rx_right_lens_ud_prism	right_lens	right_lens_udprism
rx_right_lens_pd_far	right_lens	right_lens_pd_far
rx_right_lens_pd_near	right_lens	right_lens_pd_near
rx_optom_script_flag	flag	optom_script_flag
rx_previous_dispenser		
rx_provider_number	other	provider_number
rx_scriptvalid_flag		
rx_prescription_origin	prescription	prescription_origin
rx_prism_usage	other	rx_prism_usage

Figura 16-excel rx-prescription

Nei risultati ottenuti le parti che si sono inserite a mano sono quelle di referenziazione alle tabelle già presenti in Dataform e la parte di unnesting con i rispettivi alias, necessari per riuscire ad estrarre i campi dal bronze-layer di BigQuery e rinominarli secondo la nomenclatura di MapR.

5 Survey

Per ottenere una comprensione dettagliata delle capacità e delle applicazioni pratiche di questo strumento nell'ambito dell'analisi specifica in corso, è stato implementato un approccio metodologico incentrato sull'interazione diretta con gli utenti coinvolti nel progetto. Questo approccio ha incluso l'elaborazione e la somministrazione di un questionario dettagliato, mirato a raccogliere feedback e percezioni da coloro per cui lo strumento di intelligenza artificiale poteva rappresentare un valore aggiunto significativo.

Il questionario è articolato in una serie di domande, che variavano da quelle di carattere generale a quelle più specifiche, al fine di acquisire una visione generale dell'esperienza dell'utente.

L'analisi dei dati raccolti attraverso il questionario ha permesso di identificare diversi aspetti chiave: le modalità di utilizzo dello strumento nell'ambito del progetto, la facilità di integrazione nelle attività quotidiane, l'impatto sulla produttività e l'efficienza, e le percezioni degli utenti riguardo l'accuratezza e l'affidabilità delle funzionalità offerte dall'intelligenza artificiale. Inoltre, sono stati esplorati gli aspetti relativi all'usabilità e all'interfaccia utente, al fine di comprendere come queste dimensioni influenzassero l'esperienza complessiva.

Complessivamente, questo approccio ha fornito una base solida per valutare in modo critico le potenzialità e le sfide associate all'utilizzo di questo strumento di intelligenza artificiale, nonché per delineare possibili percorsi di miglioramento e ottimizzazione per un utilizzo più efficace e efficiente in contesti simili in futuro.

5.1 Questionario sull'utilizzo del tool di AI per l'analisi dei dati

Ho posto le seguenti sette domande.

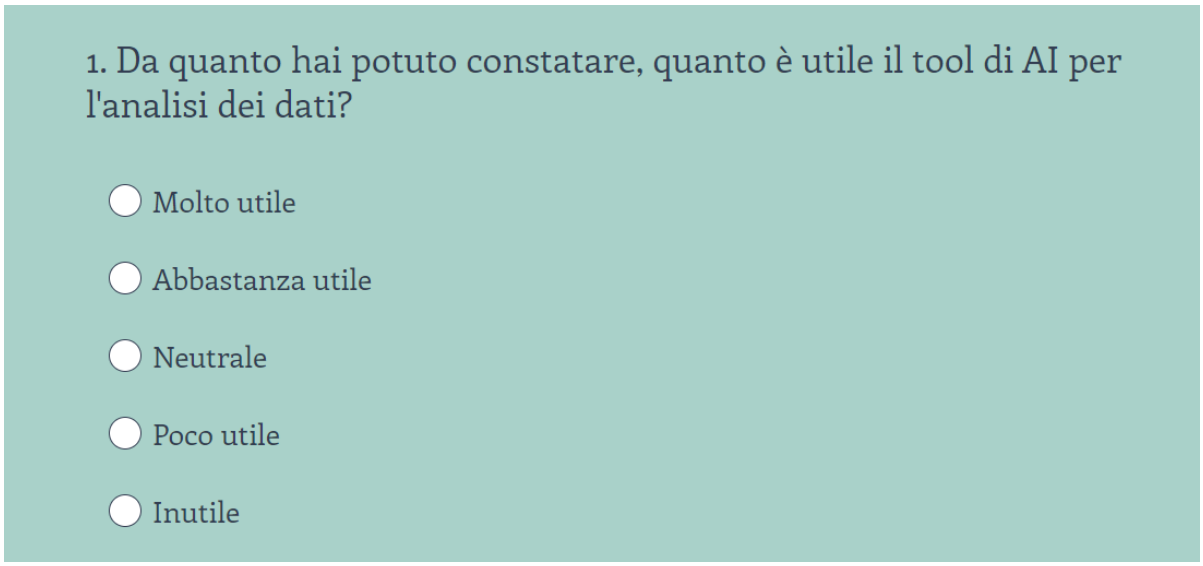


Figura 17- prima domanda questionario

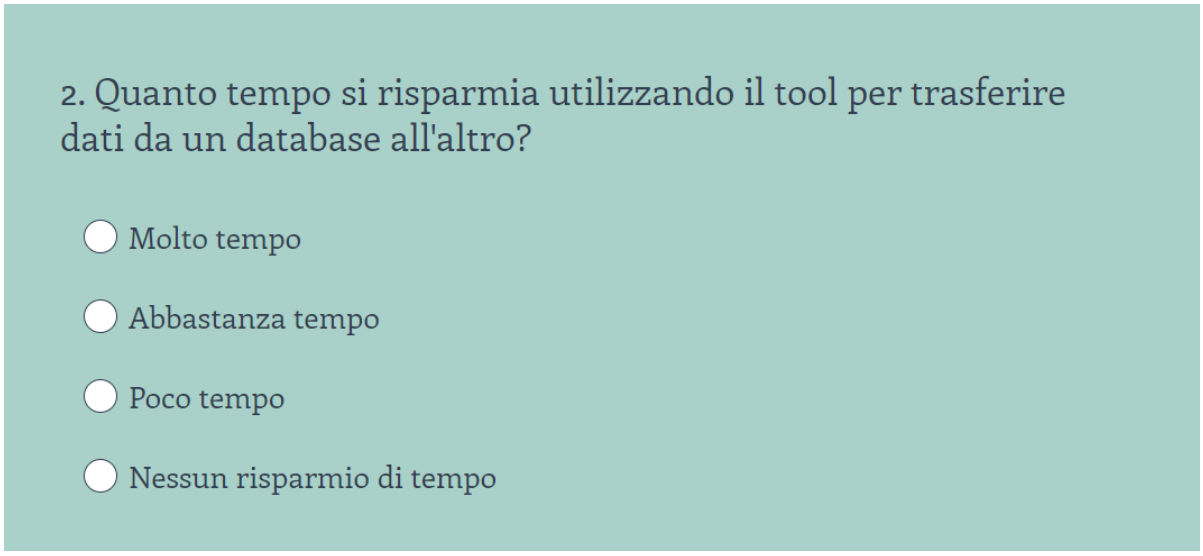


Figura 18- seconda domanda questionario

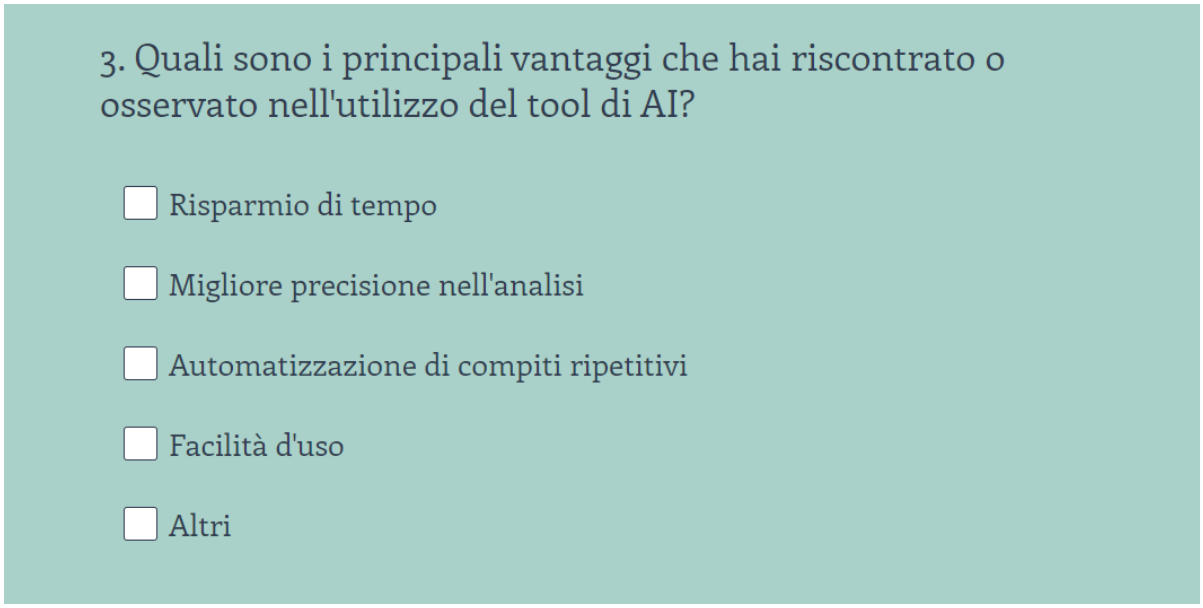


Figura 19- terza domanda questionario

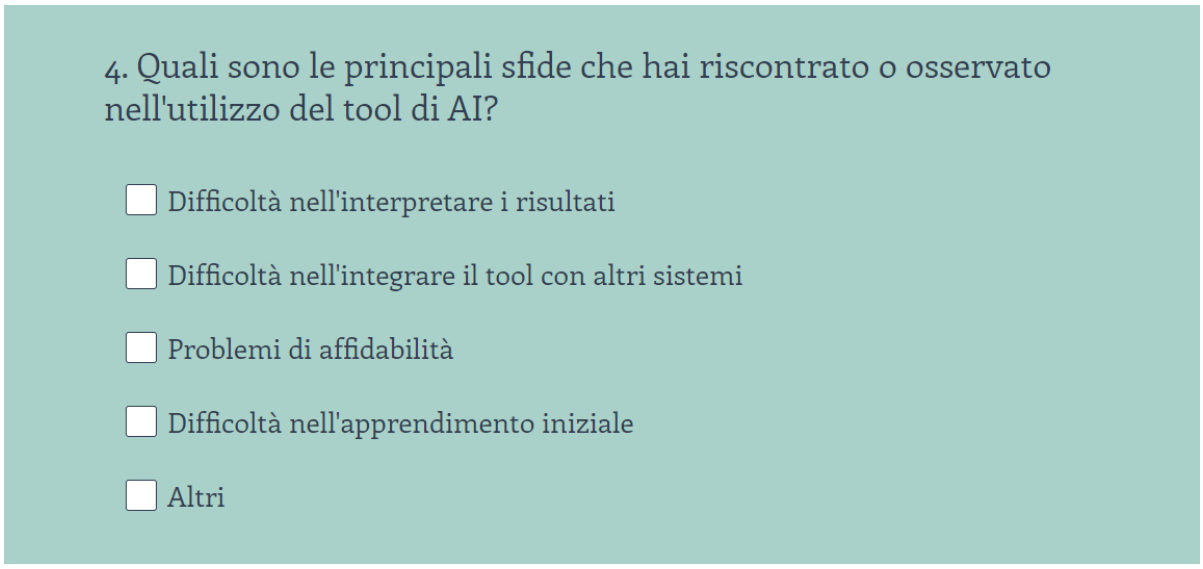


Figura 20- quarta domanda questionario

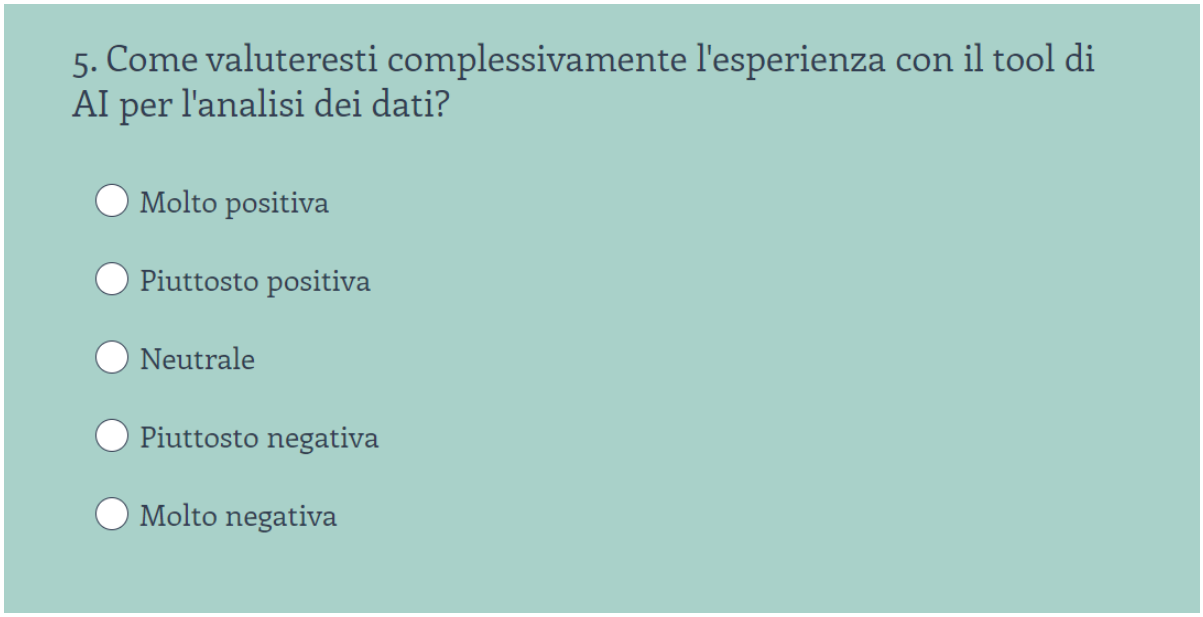


Figura 21- quinta domanda questionario

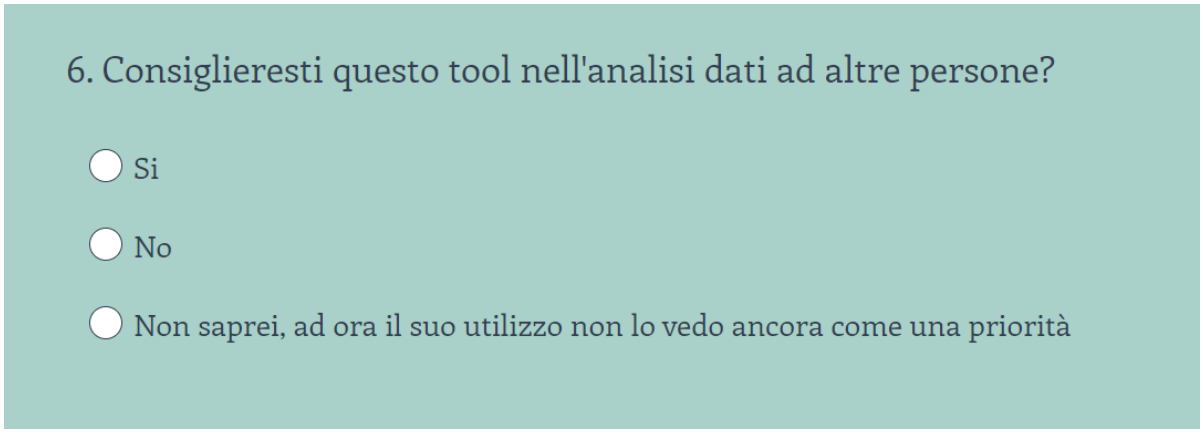


Figura 22- sesta domanda questionario

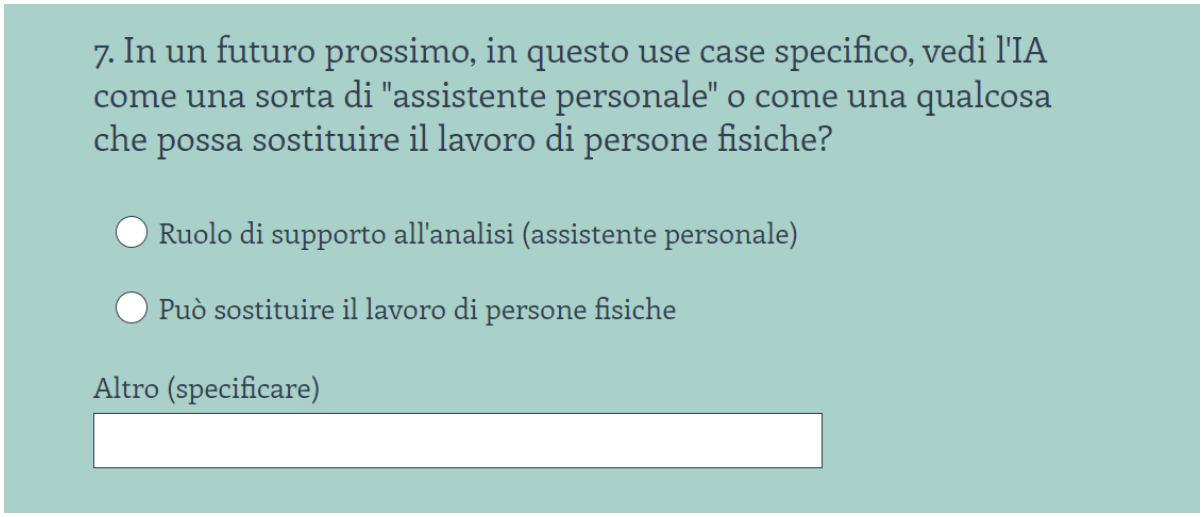


Figura 23- settima domanda questionario

5.2 Analisi risultati

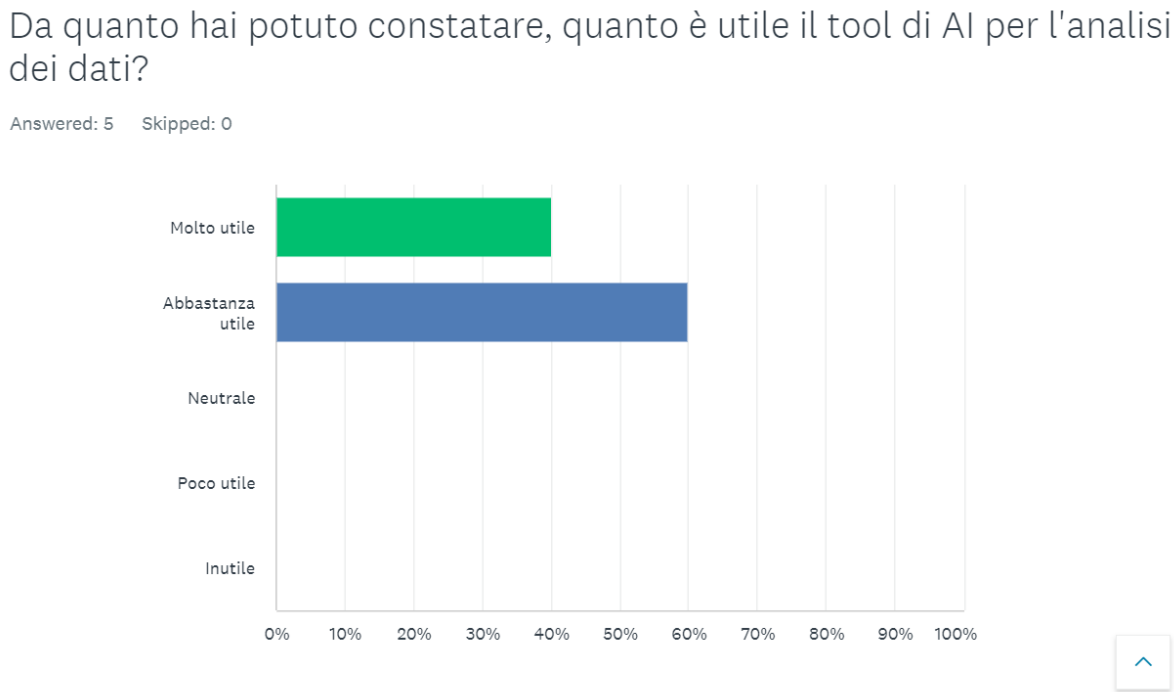


Figura 24- prima risposta questionario

Come si evince dal grafico dei cinque utilizzatori a cui è stato proposto il breve sondaggio c'è una completa propensione nel dire che il tool risulta utile nel processo di analisi.

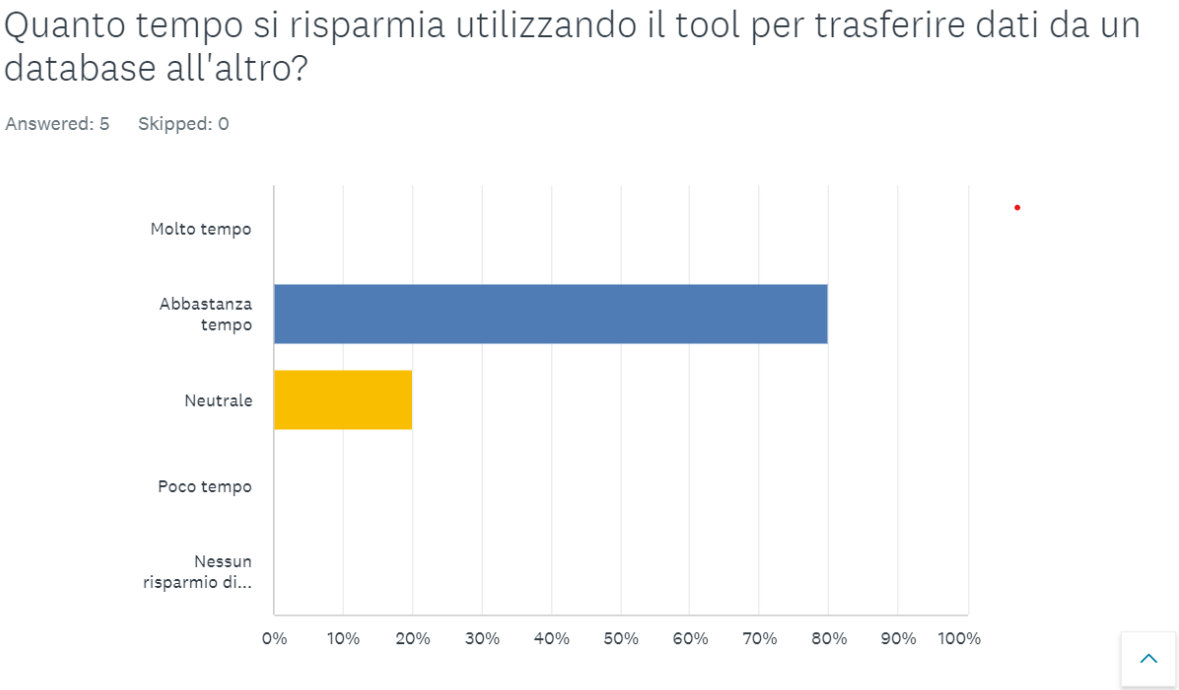


Figura 25- seconda risposta questionario

Alla seconda domanda circa l'80% ha risposto che c'è un effettivo risparmio di tempo nell'utilizzo dello strumento.

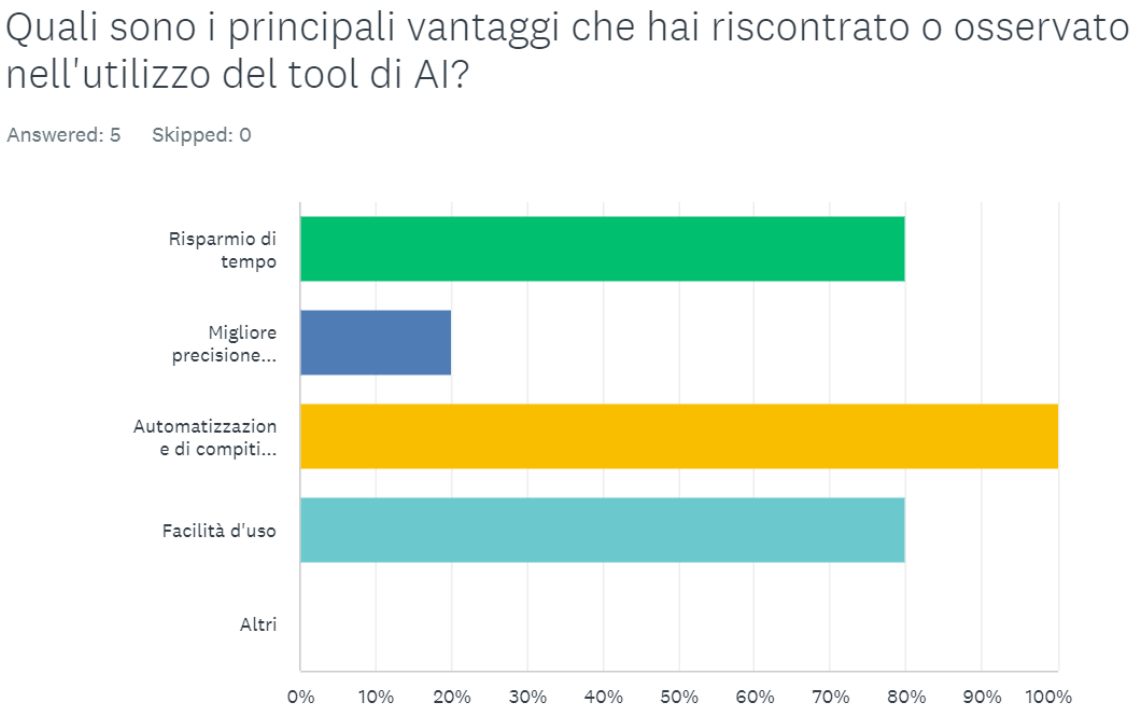


Figura 26- terza risposta questionario

Gli utilizzatori hanno individuato come vantaggi principali il risparmio di tempo, la facilità d'uso e l'automatizzazione di compiti ripetitivi.

C'è però una piccola digressione da fare per quanto riguarda la sezione "facilità d'uso". Riguarda infatti proprio la parte di utilizzo effettivo del tool e non la parte di configurazione e implementazione degli schemi che fungono da background (addestramento tool, prove di input e output, ricerca prompt corretto,...). Proprio questa parte potrebbe risultare complicata e portare via molto tempo.

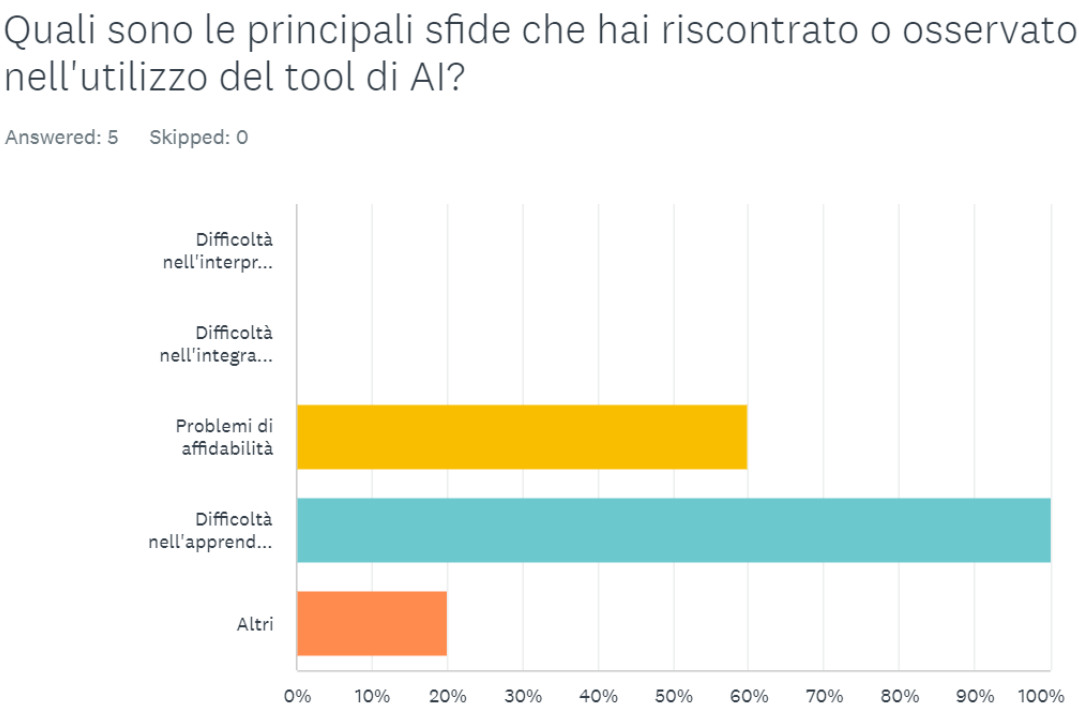


Figura 27- quarta risposta questionario

Risultano esserci anche degli svantaggi, poichè non sempre l'output è completamente corretto e va revisionato. Proprio come detto in precedenza si può occupare gran parte del tempo a disposizione a trovare le configurazioni giuste per il tool e per ricevere l'output desiderato.

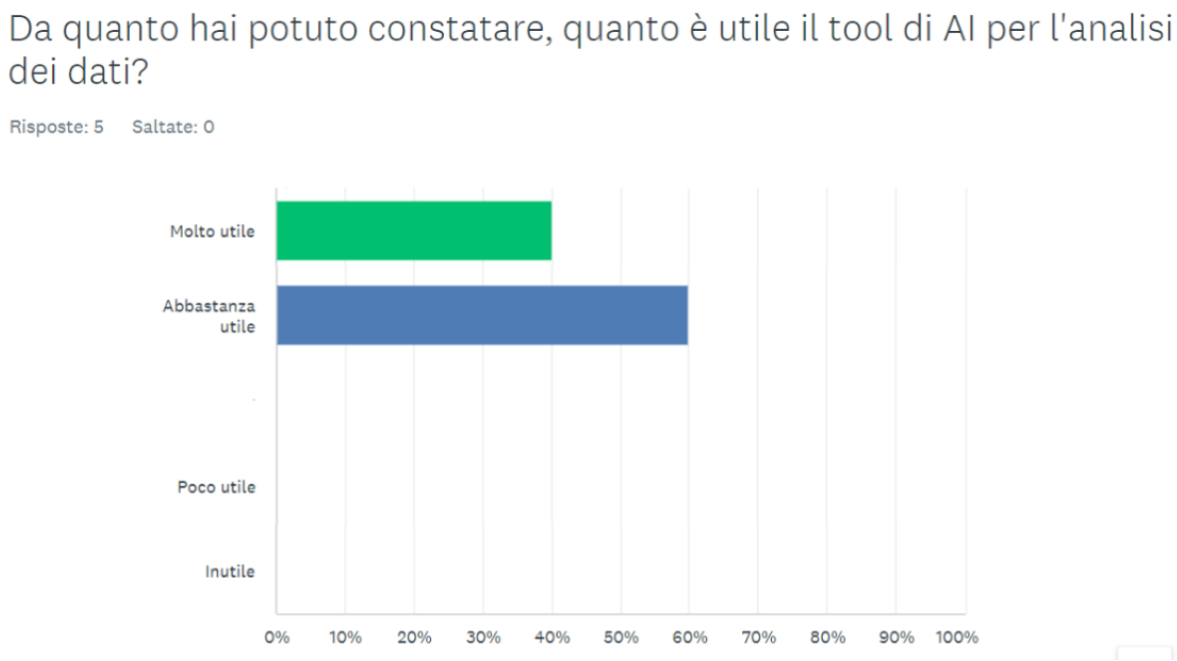


Figura 28- quinta risposta questionario

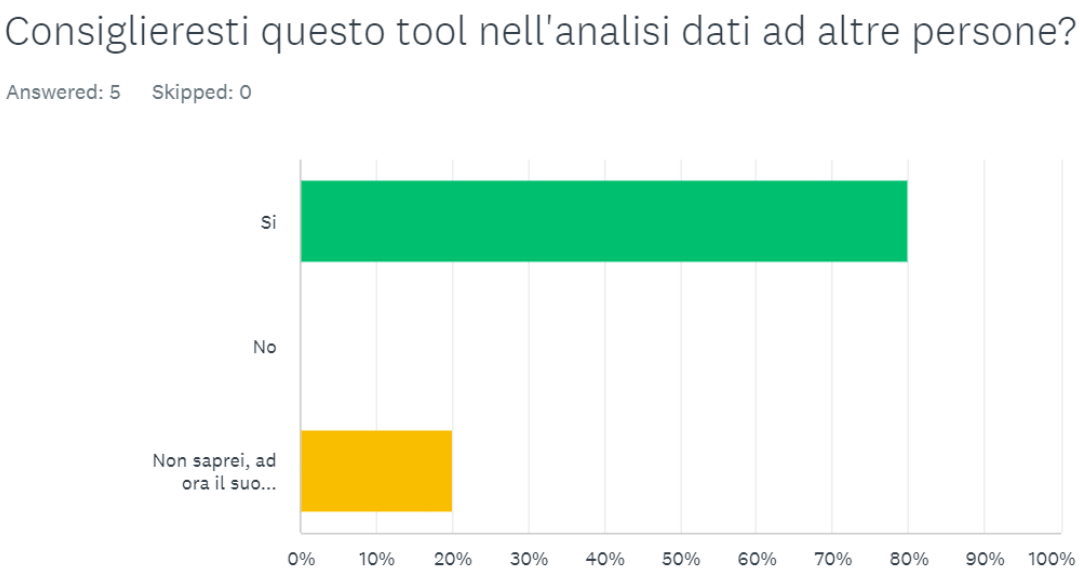


Figura 29- sesta risposta questionario

Nonostante gli aspetti negativi, da come si evince nel grafico sovrastante, nel risultato finale pesano più gli aspetti positivi, in quanto la valutazione delle prestazioni del tool nel complesso è favorevole.

Solo una persona ha risposto che non vede ancora una reale potenzialità dell'IA in questo campo.

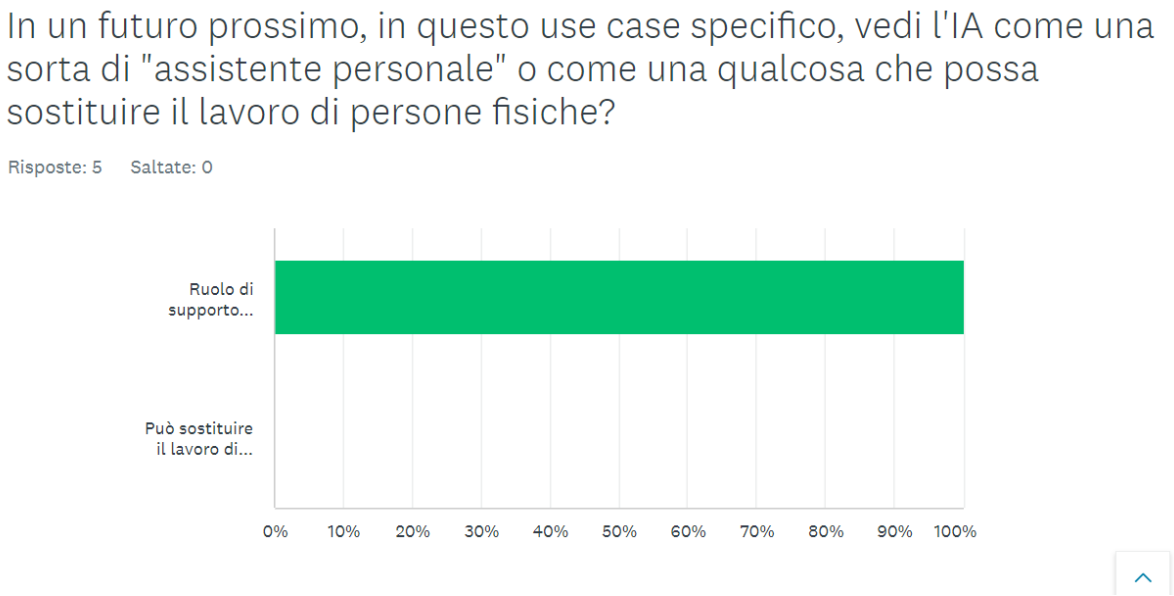


Figura 30- settima risposta questionario

Nell'ultima domanda si poneva l'utente di fronte ad un bivio: rispondere se l'IA nel complesso, ad ora, potesse rimpiazzare la persona fisica oppure fungere da "assistente personale". La totalità degli utenti ha scelto per quest'ultima.

Sono stati inseriti due commenti che specificano in modo più chiaro questa scelta.

- Bisogna effettuare tanto lavoro di setting iniziale e standardizzare procedure specifiche per fare lavorare il tool al massimo delle capacità in un particolare use case
- Rimane sempre la supervisione di una persona fisica perché bisogna sempre verificare l'output finale e non prendere i risultati finali come "oro colato"

6 Osservazioni

In questa analisi approfondita del processo, si sono identificati numerosi spunti e approcci interessanti che potrebbero fungere da base per future elaborazioni e ricerche nel campo dell'analisi dei dati. Uno degli aspetti più rilevanti è la capacità di decomporre un problema complesso in una serie di sottoproblemi più gestibili. Nel caso di questo studio infatti si è partiti dal capire il flusso dati globale e poi si è cercato di comprendere quali fossero le parti più corpose e ripetitive in cui l'IA avrebbe potuto fornire il suo apporto. Questo approccio si è dimostrato efficace e ha prodotto risultati soddisfacenti, evidenziando il potenziale dell'intelligenza artificiale come assistente nella soluzione di compiti ripetitivi e nella facilitazione di processi complessi.

Un elemento chiave di questo metodo è comprendere chiaramente l'output desiderato e, partendo da questo, formulare le domande più pertinenti da porre all'IA. Ciò implica un'interazione continua e dinamica con la tecnologia, dove l'intelligenza artificiale funge da supporto nel processo decisionale e operativo, ma sempre sotto la supervisione umana. All'interno del processo, infatti, è sempre stato supervisionato il tutto e si è analizzato in modo critico quello che veniva prodotto. Questo aspetto sottolinea l'importanza di mantenere un controllo manuale sui risultati prodotti, almeno per il momento, a causa dei margini di errore ancora presenti e della necessità di affinare ulteriormente le capacità dell'intelligenza artificiale.

Un altro punto di rilievo è l'utilizzo di immagini specifiche per generare codice corretto attraverso strumenti di intelligenza artificiale generativa. Questo rappresenta un significativo progresso tecnologico: partire da un input visivo per produrre codice scritto può aprire nuove frontiere nell'automazione dei processi. Più dettagliate sono le istruzioni fornite tramite l'immagine, minore è il margine di errore. In teoria, quasi ogni processo iterativo potrebbe essere automatizzato attraverso questa tecnologia. Sono state effettuate alcune prove per verificare questa funzionalità ed è risultato che per immagini chiare e più o meno schematiche riesce a performare bene.

Si è inoltre osservato che processi lunghi e complessi possono essere integrati e gestiti efficacemente attraverso soluzioni di intelligenza artificiale generativa. Questo amplia notevolmente il campo di applicazione dell'IA, dimostrando la sua versatilità e la sua capacità di adattarsi a una vasta gamma di esigenze e contesti operativi.

Inoltre, anche negli altri layer del progetto probabilmente si sarebbe potuti intervenire con best practice basate sull'intelligenza artificiale. Questo va a supporto delle potenzialità che potrebbe avere nel lavoro di tutti i giorni.

Queste osservazioni aprono la strada a nuove e entusiasmanti opportunità nel campo dell'automazione e dell'analisi dei dati, con un potenziale ancora in gran parte da esplorare.

Al momento avere un supporto da questo tipo di tool potrebbe essere molto utile sia per i vari motivi elencati precedentemente, sia perchè al momento il prezzo mensile è abbordabile per molte aziende.

Se si dovesse sottoscrivere un abbonamento da privato sicuramente la scelta più intelligente sarebbe il piano mensile da 20$ che permette di avere accesso all'ultimo modello di linguaggio di ChatGPT ed a tutti i plug in e funzioni personalizzabili che la piattaforma offre.

Diverso è invece il discorso se si parla di grandi aziende o multinazionali. Per queste organizzazioni ChatGPT offre la possibilità di sottoscrivere un abbonamento enterprise che consente una maggior protezione dei dati personali, l'accesso illimitato a tutta la potenzialità dell'IA e dei plug in e un minimo di 150 account attivi per avviare il programma. Il costo complessivo mensile di tutti questi account è di circa 9000$ (60$ al mese ad account). Per aziende molto grandi sono spese molto convenienti da sostenere soprattuto se si guarda alla parte di benefici e vantaggi che possono apportare.

L'aspetto sicuramente più incredibile di questa tecnologia è come riesca a facilitare il lavoro anche di persone che non hanno conoscenze mirate o specifiche in ambito ad esempio informatico. Avendo a disposizione questo tool come "assistente personale" molti dubbi o molti problemi vengono risolti in breve e spiegati al meglio. Ogni output prodotto, infatti, è spiegato passo passo e questo garantisce una panoramica completa su cosa si sta chiedendo e che cosa si vuole ottenere. Nel caso, ad esempio, del progetto in questione anche chi non avesse mai avuto modo di cimentarsi nella programmazione sql sarebbe stato in grado, dopo una spiegazione su tutte le funzionalità e gli step da implementare, di produrre query sql corrette e che rispecchiassero le richieste.

Bisogna capire le logiche generali, le metodologie corrette e studiare le potenzialità della nuova tecnologia in opera per poi iniziare a cimentarsi in ambiti che non si sono mai approfonditi prima.

7 Conclusioni

La presente tesi si è posta l'intento di testare, sperimentare e dimostrare come le tecnologie di intelligenza artificiale generativa possano essere utili all'analisi dei dati. Il tutto è stato affrontato in un case study specifico lavorando all'interno di una società di consulenza e servizi IT: NTT DATA ITALIA S.p.a. La presente ricerca è stata implementata cercando di analizzare criticamente tutto il processo di instradamento dei dati e si sono individuati i punti critici in cui l'IA poteva ricoprire un ruolo fondamentale di semplificazione e velocizzazione del processo. All'inzio della ricerca non era stato evidenziato un punto preciso in cui questa tecnologia potesse intervenire ed essere d'aiuto. Il primo scopo quindi è stato comprendere il processo globale di analisi dei dati e individuare le possibili opportunità in cui inserire l'intelligenza artificiale generativa e capire come sviluppare un processo coerente e snello. Questo poichè l'intento era quello di rendere più immediato la scrittura di codice, di modellazione dei dati e la generazione di tabelle del case study specifico. Analizzando la tecnologia in uso si sono delineati diversi pregi e limiti. In prima analisi è risultato chiaro come questo "assistente personale" potesse aiutare in molti punti del processo di instradamento dati, tra cui nei principali si possono citare la generazione di codice e la realizzazione di file in diversi fomati. Risulta molto importante capire che si tratta di codice ad hoc customizzato, realizzato a partire da file strutturati altamente specifici, anch'essi prodotti dall'intelligenza artificiale.

Un vantaggio significativo di questo strumento è la sua capacità di fornire una sintesi dettagliata e trasparente di ciò che sta generando in tempo reale. Ogni passaggio del processo viene tracciato e documentato, permettendo agli utenti di seguire con facilità l'avanzamento del lavoro. Questa caratteristica è particolarmente utile in caso di errori o problemi, poiché consente di individuare rapidamente il punto esatto in cui si è verificato. Grazie a questa trasparenza, gli utenti possono non solo comprendere meglio il funzionamento dello strumento, ma anche intervenire in modo più efficace per correggere eventuali problemi, ottimizzando così l'esperienza complessiva di utilizzo. Questo in parte può rappresentare un limite e un punto di partenza per possibili evoluzioni del modello in futuro. Infatti, l'intervento umano in questo tipo di processo è fondamentale al momento. Non deve essere presa come scienza certa, anzi deve essere la base su cui costruire ed implementare nuove funzionalità o strumenti altamente specifici. Le

tempistiche limitate dello studio non hanno permesso la piena contestualizzazine, sperimentazione e ricerca delle potenzialità dello strumento, ma sicuramente sarebbe possibile creare procedimenti che aiutano in altre sezioni o layer del progetto. Il metodo più corretto per utilizzare questa tecnologia sarebbe quello di farsi creare schemi adeguati all'output che si vuole ottenere e customizzare lo schema (che ad esempio può essere uno script python) in modo da poter condividere con altre persone il metodo creato per velocizzare o semplificare un procedimento complesso.

Questa è l'idea che è stata portata avanti durante tutto il lavoro. Bisogna essere consapevoli che potrebbero esserci infiniti studi o spunti paralleli ancora non analizzati, o magari risorse accademiche che non sono state vagliate per mancanza di tempo, che potrebbero offrire spunti per ricerche future o migliorare questa proposta.

La riflessione che scaturisce dallo studio di questo elaborato mette in risalto la capacità di questi tool di generare conoscenza e modelli dal nulla con il solo utilizzo del linguaggio normale o input visivi. Questo è uno dei punti forti della generative AI: se, come in questo caso, essa viene affiancata ai processi di analisi esistenti, non può che migliorarne le prestazioni e il lavoro di tutti.

8 Appendice

8.1 Query di output xlsx to BigQuery-Persona

```
CREATE TABLE IF NOT EXISTS
  ${ref('persona')}( source_customer_id STRING,
    record_date TIMESTAMP,
    sales_organization STRING,
    distribution_channel STRING,
    company_code STRING,
    customer_firstname STRING,
    customer_middlename STRING,
    customer_lastname STRING,
    customer_prefix_name STRING,
    customer_suffix_gen_name STRING,
    customer_suffix_pro_name STRING,
    industry STRING,
    company_name STRING,
    customer_sex STRING,
    customer_birthdate STRING,
    age_cert STRING,
    customer_deathdate DATE,
    language_code STRING,
    global_opt_out STRING,
    perscription_id STRING,
```

```
    doctor_id STRING,
    perscription_type_code STRING,
    exam_date DATE,
    expires_date DATE,
    created_by_ricef STRING,
    created_by_user STRING,
    created_by_datetime TIMESTAMP,
    pref_com_channel STRING,
    signature STRING,
    signup_store STRING,
    assigned_store STRING,
    cust_source STRING,
    cust_medium STRING,
    cust_campaign STRING,
    tax_code STRING,
    user_type STRING,
    event_info STRUCT< source_ip_address STRING,
    source_ricef STRING,
    source_site STRING,
    event_date TIMESTAMP,
    event_type STRING>,
    pseudo_sup_cust_osi_lndg STRUCT< organization_enum
STRING,
    scope_enum STRING,
```

```
    customer_status STRING,
    group_ STRING,
    service_status STRING,
    department STRING,
    rank STRING,
    expiration_date TIMESTAMP,
    primary_email STRING,
    ORGANIZATION STRING >,
    pseudo_sup_cust_costa_pro_lndg STRUCT< pro_schoolName
STRING,
    pro_schoolEmail STRING,
    pro_clubName STRING,
    pro_BenefitsID STRING,
    pro_Community STRING,
    pro_CompanyContactFullName STRING,
    pro_CompanyName STRING,
    pro_CompletedTrainings STRING,
    pro_Expiration STRING,
    pro_Industry STRING,
    pro_LogoAgreement BOOLEAN > )
```

8.2 Query di output xlsx to BigQuery: Comm

```
CREATE TABLE IF NOT EXISTS
  ${ref('comm')} ( source_customer_id STRING,
    postals ARRAY<STRUCT< postal_type STRING,
    postal_address STRING,
    address2 STRING,
    neighborhood STRING,
    city STRING,
    county STRING,
    state_code STRING,
    zip_code STRING,
    country_code STRING,
    consents ARRAY<STRUCT< consent_type STRING,
    consent_date TIMESTAMP,
    consent_value BOOLEAN >> >>,
    phones ARRAY<STRUCT< phone_type STRING,
    phone_number STRING,
    consents ARRAY<STRUCT< consent_type STRING,
    consent_date TIMESTAMP,
    consent_value BOOLEAN >> >>,
    emails ARRAY<STRUCT< email_type STRING,
    email_address STRING,
    consents ARRAY<STRUCT< consent_type STRING,
    consent_date TIMESTAMP,
```

8.3 Query di output xlsx to BiQuery: tnx-header

```
CREATE TABLE IF NOT EXISTS
  ${ref('tnx_header')}( source_customer_id STRING,
    TRANSACTION ARRAY<STRUCT< transnumber STRING,
    transaction_datetime TIMESTAMP,
    tran_source STRING,
    tran_medium STRING,
    transaction_type_cod STRING >>,
    total ARRAY<STRUCT< total_gross_value FLOAT64,
    total_discount_value FLOAT64,
    total_tax_value FLOAT64,
    total_net_value FLOAT64,
    total_tender_value FLOAT64 >>,
    eyenet_data ARRAY<STRUCT< eyenet_pos_receipt_id STRING,
    eyenet_pos_transaction_number STRING >>,
    sales_data ARRAY<STRUCT< operatorid STRING,
    sales_type STRING >>,
    shipping ARRAY<STRUCT< shippingcost FLOAT64,
    shippingtax FLOAT64,
    shippingdiscount FLOAT64 >>,
    return ARRAY<STRUCT< return_original_trading_date
TIMESTAMP,
    return_original_site STRING,
    return_original_transaction_number STRING >>,
```

```
    employee_data ARRAY<STRUCT< employee_purchase_flag
BOOLEAN >>,
    flight_data ARRAY<STRUCT< flight_code STRING,
    flight_destination STRING >>,
    omnichannel_data ARRAY<STRUCT< omnichannel_tran_type
STRING,
    omnichannel_store_identifier STRING >>,
    tax_data ARRAY<STRUCT< exemptioncode STRING,
    exemptionreason STRING,
    sales_application STRING,
    tax_exempt STRING >> )
```

8.4 Query di output xlsx to BiQuery: tnx-detail

```
CREATE TABLE IF NOT EXISTS
  ${ref('tnx_detail_1')}( source_customer_id STRING,
    retail ARRAY<STRUCT< retail_number STRING,
    retail_type_code STRING,
    retail_quantity INT64 >>,
    sales_object_detail ARRAY<STRUCT< material_number STRING,
    item_id STRING,
    merchandisecat STRING >>,
    eyenet_data ARRAY<STRUCT< eyenet_transaction_type_code STRING >>,
    sales_data ARRAY<STRUCT< customer_spend STRING >>,
    sales ARRAY<STRUCT< normal_sales_amount FLOAT64,
    sales_amount FLOAT64,
    commission_employeeid STRING >>,
    return ARRAY<STRUCT< return_original_site STRING,
    return_original_register STRING,
    return_original_transnumber STRING,
    return_original_trans_line STRING,
    return_original_trading_date TIMESTAMP >>,
    reason ARRAY<STRUCT< reason_id STRING,
    reason_description STRING >>,
    item_data ARRAY<STRUCT< item_shipping_date TIMESTAMP,
    item_shippingmode STRING,
    item_shippingmode_carrier STRING >>,
    workorder_data ARRAY<STRUCT< workorder_number STRING,
    workorder_date TIMESTAMP >>,
    other ARRAY<STRUCT< second_pair_code STRING,
    copay_amt STRING,
    plan_code STRING,
    benefit_level_code STRING,
    member_id STRING >>,
    doctor_data ARRAY<STRUCT< doctor_id STRING,
```

```
doctor_type_code STRING >>,
exam_data ARRAY<STRUCT< exam_date DATE,
expires_date DATE >>,
)
```

8.5 Da file json a file excel (python)

```
import json
import pandas as pd


def extract_names(json_element, level=1):
    """
    Recursive function to extract names from the JSON
structure.
    """
    data = []


    # Base case: If the element is a list, iterate over its
items
    if isinstance(json_element, list):
        for item in json_element:
            data.extend(extract_names(item, level))
    elif isinstance(json_element, dict):
        # Extract the name attribute
        name = json_element.get('name', None)
        if name:
            data.append((name, level))


        # Recurse into 'fields' if it exists, indicating a
nested structure
        if 'fields' in json_element:

data.extend(extract_names(json_element['fields'], level +
1))


    return data
```

```
# Load the JSON file
file_path = '/mnt/data/schema.json'
with open(file_path, 'r') as file:
    json_data = json.load(file)

# Extract names from the JSON data
extracted_data = extract_names(json_data)

# Convert to DataFrame
df = pd.DataFrame(extracted_data, columns=['Name',
'Level'])

# Pivot the data to get names in separate columns per level
df_pivoted = df.pivot(columns='Level', values='Name')

# Rename columns to "Level X"
level_columns = {i: f'Level {i}' for i in
df_pivoted.columns}
df_pivoted.rename(columns=level_columns, inplace=True)

# Export DataFrame to Excel
excel_file_path = '/mnt/data/transformed_data.xlsx'
df_pivoted.to_excel(excel_file_path, index=False)

# Print the path to the exported file
print(f"Data exported to Excel file at: {excel_file_path}")
```

8.6 Query popolamento hub persona bronze to silver

```
SELECT
  source_customer_id,
  pseudo_cust_new_lndg.record_date,
  pseudo_cust_new_lndg.sales_organization,
  pseudo_cust_new_lndg.distribution_channel,
  pseudo_cust_new_lndg.company_code,
  pseudo_cust_new_lndg.customer_firstname,
  pseudo_cust_new_lndg.customer_middlename,
  pseudo_cust_new_lndg.customer_lastname,
  pseudo_cust_new_lndg.customer_prefix_name,
  pseudo_cust_new_lndg.customer_suffix_gen_name,
  pseudo_cust_new_lndg.customer_suffix_pro_name,
  pseudo_cust_new_lndg.industry,
  pseudo_cust_new_lndg.company_name,
  pseudo_cust_new_lndg.customer_sex,
  pseudo_cust_new_lndg.customer_birthdate,
  pseudo_cust_new_lndg.age_cert,
  pseudo_cust_new_lndg.customer_deathdate,
  pseudo_cust_new_lndg.language_code,
  CAST(pseudo_cust_new_lndg.global_opt_out AS STRING) AS
global_opt_out,
  pseudo_cust_new_lndg.perscription_id,
  pseudo_cust_new_lndg.doctor_id,
  pseudo_cust_new_lndg.perscription_type_code,
  pseudo_cust_new_lndg.exam_date,
  pseudo_cust_new_lndg.expires_date,
  pseudo_cust_new_lndg.created_by_ricef,
  pseudo_cust_new_lndg.created_by_user,
  pseudo_cust_new_lndg.created_by_datetime,
  pseudo_cust_new_lndg.pref_com_channel,
  pseudo_cust_new_lndg.signature,
  pseudo_cust_new_lndg.signup_store,
```

```
  pseudo_cust_new_lndg.assigned_store,
  pseudo_cust_new_lndg.cust_source,
  pseudo_cust_new_lndg.cust_medium,
  pseudo_cust_new_lndg.cust_campaign,
  pseudo_cust_new_lndg.tax_code,
  pseudo_cust_new_lndg.user_type,
  pseudo_cust_new_lndg.customer_health_fund,
  pseudo_cust_new_lndg.cl_subscription_active,
  pseudo_cust_new_lndg.cl_subscription_inactive,
  pseudo_cust_new_lndg.cl_subscription_cancelled,
  pseudo_cust_new_lndg.cl_subscription_paused,
  pseudo_cust_new_lndg.consent_method,
  pseudo_cust_new_lndg.created_by_user_key,
  pseudo_cust_new_lndg.employeesale_id,
  pseudo_cust_new_lndg.list_name,
  pseudo_cust_new_lndg.optin_confirmation,
  pseudo_cust_new_lndg.profiling_date,
  pseudo_cust_new_lndg.profiling_value,
  pseudo_cust_new_lndg.signature_url,
  STRUCT( source_ip_address,
    source_ricef,
    source_site,
    CAST (event_date AS TIMESTAMP) AS event_date,
    event_type ) AS event_info,
  STRUCT(
pseudo_cust_new_lndg.pseudo_sup_cust_osi_lndg.organization_enum,
    pseudo_cust_new_lndg.pseudo_sup_cust_osi_lndg.scope_enum,
    pseudo_cust_new_lndg.pseudo_sup_cust_osi_lndg.customer_status,
    pseudo_cust_new_lndg.pseudo_sup_cust_osi_lndg.group AS group_,
    pseudo_cust_new_lndg.pseudo_sup_cust_osi_lndg.service_status,
    pseudo_cust_new_lndg.pseudo_sup_cust_osi_lndg.department,
    pseudo_cust_new_lndg.pseudo_sup_cust_osi_lndg.rank,
    pseudo_cust_new_lndg.pseudo_sup_cust_osi_lndg.expiration_date,
    pseudo_cust_new_lndg.pseudo_sup_cust_osi_lndg.primary_email,
```

```
    pseudo_cust_new_lndg.pseudo_sup_cust_osi_lndg.organization ) AS
pseudo_sup_cust_osi_lndg,

  STRUCT(
pseudo_cust_new_lndg.pseudo_sup_cust_costa_pro_lndg.pro_schoolName,

pseudo_cust_new_lndg.pseudo_sup_cust_costa_pro_lndg.pro_schoolEmail,

    pseudo_cust_new_lndg.pseudo_sup_cust_costa_pro_lndg.pro_clubName,

pseudo_cust_new_lndg.pseudo_sup_cust_costa_pro_lndg.pro_BenefitsID,

    pseudo_cust_new_lndg.pseudo_sup_cust_costa_pro_lndg.pro_Community,

pseudo_cust_new_lndg.pseudo_sup_cust_costa_pro_lndg.pro_CompanyContact
FullName,

pseudo_cust_new_lndg.pseudo_sup_cust_costa_pro_lndg.pro_CompanyName,

pseudo_cust_new_lndg.pseudo_sup_cust_costa_pro_lndg.pro_CompletedTrain
ings,

pseudo_cust_new_lndg.pseudo_sup_cust_costa_pro_lndg.pro_Expiration,

    pseudo_cust_new_lndg.pseudo_sup_cust_costa_pro_lndg.pro_Industry,

pseudo_cust_new_lndg.pseudo_sup_cust_costa_pro_lndg.pro_LogoAgreement
) AS pseudo_sup_cust_costa_pro_lndg


FROM

  ${ref("cdp_events")}
```

8.7 Query popolamento hub comm bronze to silver

```
SELECT

  source_customer_id,

  pseudo_cust_new_lndg.postals,

  pseudo_cust_new_lndg.phones,

  pseudo_cust_new_lndg.emails,

FROM

  ${ref('cdp_events')}
```

8.8 Query popolamento hub cl-prescription bronze to silver

```
SELECT
    source_customer_id,
    ARRAY_AGG(STRUCT(
        a.cl_left_lenscode AS left_lens_code,
        a.cl_left_pricetype AS left_lens_price_type,
        a.cl_left_quantity AS left_lens_quantity,
        a.cl_left_tintcode AS left_lens_tint_code
    ))  AS left_lens,
    ARRAY_AGG(STRUCT(
        a.cl_prescription_date AS prescription_date,
        a.cl_prescription_issue_date AS prescription_issue_date,
        a.cl_prescription_number AS prescription_number,
        a.cl_prescription_origin AS prescription_origin
    ))  AS prescriptioncl,
    ARRAY_AGG(STRUCT(
        a.cl_provider_number AS required_date,
        a.cl_required_date AS provider_number
    ))  AS other_cl,
    ARRAY_AGG(STRUCT(
        a.cl_right_lenscode AS right_lens_code,
        a.cl_right_pricetype AS right_lens_price_type,
        a.cl_right_quantity AS right_lens_quantity,
        a.cl_right_tintcode AS right_lens_tint_code
    )) AS right_lens,
    ARRAY_AGG(STRUCT(
        lens.cl_eye,
        lens.cl_eyenumber,
        lens.cl_despar_code,
        lens.cl_despar_value
    )) as cl
    FROM
  ${ref('cdp_events')},
```

```
UNNEST (pseudo_cust_new_lndg.cl_prescription) as a,
UNNEST (a.cl_contactlens_details) as lens
```

8.9 Query popolamento hub rx-prescription bronze to silver

```
SELECT
  source_customer_id,
  ARRAY_AGG(STRUCT(
    rx.rx_prescription_type AS prescription_type,
    rx.rx_prescription_date AS prescription_date,
    rx.rx_prescription_number AS prescription_number,
    rx.rx_comment AS prescription_comment,
    rx.rx_prescription_origin AS prescription_origin
  )) AS prescription,
  ARRAY_AGG(STRUCT(
    rx.rx_prescription_issue_date AS prescription_issue_date,
    rx.rx_prescription_expiration_date AS prescription_expiry,
    rx.rx_job_number AS prescription_job
  )) AS prescription2,
  ARRAY_AGG(STRUCT(
    rx.rx_doctor_id AS doctor_id
  )) AS doctor_id,
  ARRAY_AGG(STRUCT(
    rx.rx_firsttime_wear_flag AS first_time_wearer_flag,
    rx.rx_medical_cond_flag AS medical_condition_flag
  )) AS flag,
  ARRAY_AGG(STRUCT(
    rx.rx_optom_script_flag AS optom_script_flag
  )) as optom_script_flag,
  ARRAY_AGG(STRUCT(
    rx.rx_left_lens_add AS left_lens_add,
    rx.rx_left_lens_axis AS left_lens_axis,
    rx.rx_left_lens_cylinder AS left_lens_cylinder,
    rx.rx_left_lens_io_prism AS left_lens_prism,
    rx.rx_left_lens_sphere AS left_lens_sphere,
    rx.rx_left_lens_ud_prism AS left_lens_udprism,
    rx.rx_left_lens_pd_far AS left_lens_pd_far,
```

```
    rx.rx_left_lens_pd_near AS left_lens_pd_near
  )) AS left_lens,
  ARRAY_AGG(STRUCT(
    rx.rx_right_lens_add AS right_lens_add,
    rx.rx_right_lens_axis AS right_lens_axis,
    rx.rx_right_lens_cyl AS right_lens_cylinder,
    rx.rx_right_lens_io_prism AS right_lens_prism,
    rx.rx_right_lens_sphere AS right_lens_sphere,
    rx.rx_right_lens_ud_prism AS right_lens_udprism,
    rx.rx_right_lens_pd_far AS right_lens_pd_far,
    rx.rx_right_lens_pd_near AS right_lens_pd_near
  )) AS right_lens,
  ARRAY_AGG(STRUCT(
    rx.rx_provider_number AS provider_number,
    rx.rx_prism_usage AS rx_prism_usage
  )) AS other
FROM
${ref('cdp_events')},
UNNEST (pseudo_cust_new_lndg.rx_prescription) as rx
```