POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Ingegneria Matematica

Tesi di Laurea Magistrale

Progettazione e realizzazione di moduli per la generazione di pipeline ETL

Relatore
Prof. Tania Cerquitelli
Tutor Aziendale
Dott. Filippo Balla

Candidato
Irene Michelotti

Anno Accademico 2024-2025

Sommario

Negli ultimi anni, la crescente digitalizzazione ha comportato un aumento esponenziale del volume di dati generati dalle aziende, rendendo sempre più necessaria l'adozione di sistemi avanzati per l'analisi e la gestione delle informazioni strategiche. In questo contesto, i processi ETL (Extract, Transform, Load) assumono un ruolo cruciale nell'alimentazione dei Data Warehouse con dati integrati, affidabili e coerenti, ma la loro realizzazione manuale risulta spesso onerosa e soggetta a errori.

La presente tesi, sviluppata in collaborazione con l'azienda Mediamente Consulting Srl, si propone di automatizzare la creazione dei flussi ETL attraverso lo sviluppo di script dedicati, con l'obiettivo di ridurre tempi e risorse impiegate, migliorando l'efficienza e la riutilizzabilità del processo. Dopo un'introduzione ai concetti fondamentali di Business Intelligence e al funzionamento dei sistemi ETL e dei Data Warehouse, la tesi analizza in modo approfondito le metodologie adottate per l'automazione dei flussi ETL, descrivendo le diverse fasi di progettazione e implementazione degli script sviluppati. Viene inoltre proposta una valutazione critica dei risultati ottenuti, evidenziando le principali sfide affrontate durante il processo e individuando le potenzialità di miglioramento per il futuro.

Indice

Elenco delle tabelle

Elenco delle figure

Capitolo 1

Introduzione

Negli ultimi decenni, la crescente digitalizzazione dei processi aziendali ha portato a un notevole incremento nella quantità di dati prodotti e raccolti dalle imprese. Questo fenomeno ha determinato un cambiamento significativo nel modo in cui le aziende gestiscono le informazioni, rendendo sempre più centrale la necessità di estrarre da questi dati conoscenze utili e strategiche su cui basare le decisioni future. Per rispondere a tale esigenza, i tradizionali sistemi di produzione, come quelli dedicati alla gestione della contabilità e del magazzino, utilizzati dal personale operativo, sono stati affiancati da innovativi sistemi di Data Warehousing. Questi ultimi sono stati specificamente concepiti per agevolare e ottimizzare l'analisi dei Key Performance Indicators (KPI) aziendali, offrendo una visione integrata e coerente delle performance.

Per assicurare che i data warehouse siano popolati con dati non solo integrati e consolidati, ma anche puliti e affidabili, diventa cruciale la progettazione e la realizzazione di processi ETL (Extract, Transform, Load) ben strutturati. Questi processi sono fondamentali per la raccolta e l'elaborazione di dati provenienti da una moltitudine di fonti eterogenee. Tuttavia, a fronte del volume sempre crescente di dati da analizzare, la creazione manuale di tali flussi si rivela spesso onerosa sia in termini di tempo che di risorse aziendali impiegate. Un approccio manuale può, nel lungo termine, incrementare significativamente il rischio di errori e inefficienze.

È in questo contesto che si inserisce il presente progetto di tesi, frutto di una collaborazione con l'azienda Mediamente Consulting srl, una società di consulenza fondata nel 2012. Mediamente Consulting si distingue per l'offerta di un'ampia gamma di servizi finalizzati alla progettazione e implementazione di sistemi di supporto alle decisioni, tra cui spiccano Technological Infrastructure, Data Integration e Management, Corporate Performance Management, Business Intelligence, Advanced Analytics..

L'obiettivo primario di questa tesi è lo sviluppo di script volti a incrementare significativamente il grado di automazione nella creazione dei flussi ETL. Questo approccio mira a conseguire una notevole riduzione del tempo attualmente dedicato alla realizzazione manuale di tali processi da parte dei dipendenti, promuovendo al contempo un processo più veloce, efficiente e riutilizzabile.

Il presente studio si articola in diverse fasi. Inizialmente, verrà fornita una panoramica approfondita sullo stato dell'arte nel campo della Business Intelligence e dei processi ETL. Successivamente, la trattazione si concentrerà sull'implementazione pratica del progetto, dettagliando le metodologie e gli strumenti impiegati. Infine, verranno analizzati con attenzione i risultati ottenuti, con un focus specifico sui possibili sbocchi futuri e sulle implicazioni pratiche dell'automazione proposta.

La tesi è strutturata in sei capitoli, ciascuno dei quali affronta aspetti specifici del progetto. I primi due capitoli offrono un'introduzione ai concetti fondamentali della Business Intelligence, fornendo una base teorica solida. Verrà inclusa un'analisi approfondita del processo ETL in tutte le sue fasi e delle architetture dei data Warehouse, elementi cardine per la comprensione del contesto in cui si inserisce il progetto.

Nei capitoli centrali della tesi, verrà presentata la descrizione dettagliata del progetto, illustrando le modalità con cui è stata concepita e attuata l'automazione dei flussi del processo ETL nel contesto specifico dell'azienda. Si fornirà inoltre un resoconto approfondito dell'implementazione pratica, includendo gli strumenti necessari e gli step procedurali seguiti per la realizzazione degli script.

Il quinto capitolo è dedicato alla presentazione e all'analisi critica dei risultati ottenuti. L'obiettivo è valutare l'effettivo raggiungimento degli obiettivi iniziali e quantificare i benefici derivanti dall'automazione. In conclusione, verranno formulate proposte per eventuali miglioramenti e sviluppi futuri, basate sulle intuizioni acquisite e sui risultati conseguiti.

Capitolo 2

Stato dell'Arte

In questo capitolo vengono introdotti i concetti fondamentali necessari per affrontare lo studio dell'automazione del processo ETL (Extract, Transform, Load), con l'obiettivo di fornire un quadro teorico di riferimento chiaro.

Nella prima sezione si approfondisce il tema della modellazione dei dati, elemento essenziale per comprendere la struttura e l'organizzazione delle informazioni all'interno di un sistema informativo complesso. Viene quindi analizzato il concetto di data warehouse, di cui si delineano le caratteristiche principali, tra cui l'integrazione, la storicizzazione e la disponibilità dei dati a supporto delle decisioni aziendali.

Successivamente, l'attenzione si concentra sul processo ETL, ossia quell'insieme di operazioni che permette di estrarre dati da sorgenti eterogenee, trasformarli secondo specifiche regole aziendali e caricarli nelle tabelle finali del data warehouse. Questo flusso garantisce che i dati risultino coerenti, puliti, consolidati e pronti per essere analizzati da parte dei decision maker, contribuendo così in modo determinante alla qualità delle decisioni strategiche.

Infine, viene introdotto il ruolo crescente e strategico dell'Intelligenza Artificiale Generativa nel contesto dei Big Data, con particolare riferimento alla sua capacità di automatizzare processi tradizionalmente complessi e ad alto impatto in termini di risorse. Questa tecnologia si propone infatti come una leva innovativa per ottimizzare e semplificare le attività di gestione e trasformazione dei dati, riducendo tempi, costi e margini di errore.

2.1 Data Warehouse

La crescente necessità, da parte delle aziende, di ottenere valore strategico dai grandi volumi di dati generati quotidianamente ha incentivato lo sviluppo di sistemi dedicati alla loro gestione e analisi in modo efficiente. Tra questi, i sistemi di **Data Warehousing** (DWH) si sono affermati come strumenti centrali nel supportare i processi decisionali, grazie alla loro capacità di gestire interrogazioni complesse in modo dinamico e strutturato.

In questo contesto, verranno dapprima introdotti i principi fondamentali che caratterizzano un sistema di data warehousing, per poi analizzare alcuni aspetti specifici legati alla sua implementazione e al suo impiego operativo.

I **sistemi di Data Warehousing** si basano su alcuni requisiti essenziali, fondamentali per garantire la qualità dell'analisi e la validità delle informazioni fornite ai decision maker [14]:

- **Accessibilità delle informazioni**: i dati devono essere facilmente consultabili mediante strumenti intuitivi, accessibili anche a utenti non tecnici. Le informazioni restituite devono essere chiare, accurate e immediatamente interpretabili, al fine di supportare decisioni consapevoli.

- **Consistenza dei dati**: poiché le informazioni provengono da fonti eterogenee, il compito del sistema di data warehousing è quello di raccogliere, integrare e armonizzare tali dati, creando un insieme coerente e affidabile che garantisca uniformità e qualità.

- **Adattabilità al cambiamento**: i sistemi di DWH devono essere in grado di evolvere nel tempo, integrando nuove fonti e adattandosi a trasformazioni tecnologiche o organizzative, senza compromettere l'integrità dei dati già presenti.

Sebbene a una prima analisi possano sembrare simili, i sistemi di data warehousing si distinguono in modo sostanziale dai tradizionali sistemi informativi, sia per finalità che per modalità operative. Questi ultimi sono principalmente orientati alla gestione di attività operative e transazioni di routine, come la contabilità, la gestione del magazzino o le vendite.

Al contrario, i sistemi di DWH assumono una funzione strategica: sono progettati per fornire informazioni ad alto valore aggiunto, frutto di elaborazioni complesse e integrate, che permettono di analizzare l'andamento del business in maniera approfondita. In particolare, mentre i sistemi informazionali operano con una prospettiva operativa, i data warehouse rispondono a esigenze analitiche, sintetizzando grandi moli di dati in conoscenze utili e immediatamente applicabili alle decisioni aziendali.

Tra le varie definizioni proposte in letteratura per descrivere il concetto di data warehouse, una delle più autorevoli è quella formulata da William H. Inmon [9], considerato il padre fondatore del data warehousing. Secondo la sua definizione, il data warehouse (DWH) è una collezione di dati *subject-oriented*, *integrated*, *time-variant* e *non-volatile*, progettata per supportare il processo decisionale.

Più nel dettaglio, un sistema di data warehousing si caratterizza per le seguenti proprietà fondamentali:

- **Orientato al soggetto**: i dati vengono organizzati attorno ai principali temi di interesse per l'utente finale, in modo da fornire una visione coerente e strutturata delle informazioni rilevanti per ciascuna area aziendale;

- **Integrato**: i dati, provenienti da fonti differenti, sono armonizzati e resi omogenei tramite processi di normalizzazione e standardizzazione, così da garantire coerenza semantica e sintattica nell'intero sistema;

- **Variante nel tempo**: i dati contenuti nel DWH appartengono a un ampio orizzonte temporale permettendo un'analisi dei trend aziendali attraverso l'utilizzo di chiavi contenenti un elemento temporale;

- **Non volatile**: una volta che i dati sono stati caricati nel DWH, non possono essere né modificati né cancellati dagli utenti. Essi restano disponibili solo per consultazione e analisi, garantendo così la stabilità delle informazioni nel tempo.

Dalla definizione proposta emerge chiaramente uno degli obiettivi principali di un data warehouse: migliorare la disponibilità e la qualità dei dati aziendali, rendendoli facilmente accessibili in modo trasversale all'interno dell'organizzazione. Questo approccio consente non solo di rispondere efficacemente alle esigenze informative attuali, ma anche di adattarsi con flessibilità ai futuri cambiamenti, sia sul piano tecnologico che strategico.

La centralizzazione dei dati in un sistema strutturato e unificato permette di razionalizzare i flussi informativi e semplificare l'accesso ai dati da parte degli utenti. Ne derivano una riduzione significativa dei costi e delle complessità legate alla distribuzione, alla manutenzione e alla gestione delle informazioni lungo l'intero ciclo di vita del software aziendale, oltre a un sensibile miglioramento della coerenza e dell'affidabilità del patrimonio informativo disponibile.

Un ulteriore concetto fondamentale che caratterizza i sistemi di data warehousing è la **storicizzazione dei dati**. Questo principio implica la capacità del sistema di conservare le informazioni nel tempo, mantenendone traccia in modo strutturato e organizzato. In termini architetturali, ciò si traduce nella progettazione di una struttura multilivello, in cui i dati sono distribuiti secondo il grado di sintesi: i livelli superiori ospitano informazioni aggregate (come nei Data Mart), mentre i livelli inferiori contengono dati più dettagliati e granulari.

Considerato l'elevato volume generato dalla memorizzazione di dati a livello di dettaglio, questi ultimi vengono spesso archiviati in modalità off-line o su sistemi di storage secondari, al fine di ottimizzare le prestazioni delle interrogazioni analitiche e contenere i costi infrastrutturali. Una tale organizzazione consente agli utenti di accedere e navigare tra diversi livelli di aggregazione, passando agevolmente da viste sintetiche, utili per il supporto decisionale, a informazioni dettagliate, indispensabili per analisi approfondite.

Con l'espansione costante dei volumi informativi memorizzati nei data warehouse, diventa essenziale dotarsi di strumenti che facilitino l'accesso, l'interpretazione e la gestione dei dati. In questo contesto, i **metadati** assumono un ruolo fondamentale: essi forniscono una documentazione completa del sistema, descrivendo in modo dettagliato la struttura, il contenuto e la provenienza dei dati contenuti nel DWH.

Tuttavia, la gestione dei metadati può presentare alcune criticità. Una delle principali è rappresentata dall'assenza di uno standard univoco per la loro definizione e rappresentazione. In ambito aziendale, infatti, i metadati sono spesso distribuiti in formati eterogenei (come file di testo, tabelle Excel, o contenuti multimediali), il che può generare difficoltà nel loro utilizzo e nella loro integrazione all'interno di un sistema centralizzato [2].

2.1.1 Data Mart e Data Lake

Le risorse di un'azienda possono essere organizzate e centralizzate in maniera efficiente attraverso l'impiego di diversi sistemi di gestione dei dati. Tra le soluzioni più diffuse e consolidate si distinguono in particolare i Data Mart e i Data Lake, due tecnologie che, pur perseguendo obiettivi simili in termini di raccolta e accessibilità dei dati, si differenziano per struttura, finalità e modalità di utilizzo.

Di seguito verranno analizzate entrambe le soluzioni, evidenziandone le caratteristiche principali e le differenze più rilevanti rispetto ai tradizionali sistemi di data warehousing.

I **Data Lake** rappresentano sistemi di archiviazione centralizzati progettati per raccogliere e conservare grandi volumi di dati in formato grezzo, ossia nel loro stato originale, senza necessità di una strutturazione preventiva [12]. Questi sistemi sono particolarmente indicati per memorizzare dati di natura eterogenea, come dati strutturati, semi-strutturati e non strutturati, offrendo un'elevata flessibilità nella gestione e nell'analisi delle informazioni.

A differenza dei data warehouse, che organizzano i dati in tabelle relazionali strutturate e richiedono un'attenta progettazione a monte, i Data Lake consentono l'acquisizione di dati provenienti da numerose fonti, come file di log, dati generati da sensori IoT, documenti testuali o flussi multimediali, senza imporre un'immediata normalizzazione o integrazione. Questo approccio consente di posticipare la modellazione dei dati al momento della loro effettiva analisi, adattandola alle specifiche esigenze applicative.

Sebbene questa modalità richieda l'utilizzo di tecnologie analitiche più dinamiche e avanzate rispetto a quelle tradizionalmente impiegate nei data warehouse, essa garantisce al tempo stesso una maggiore scalabilità e un più ampio margine di flessibilità operativa. I dati possono essere selezionati, trasformati e analizzati in un secondo momento, solo quando effettivamente necessari, permettendo una gestione più agile e adattabile in contesti caratterizzati da elevata variabilità e volume informativo (Figura 2.1).

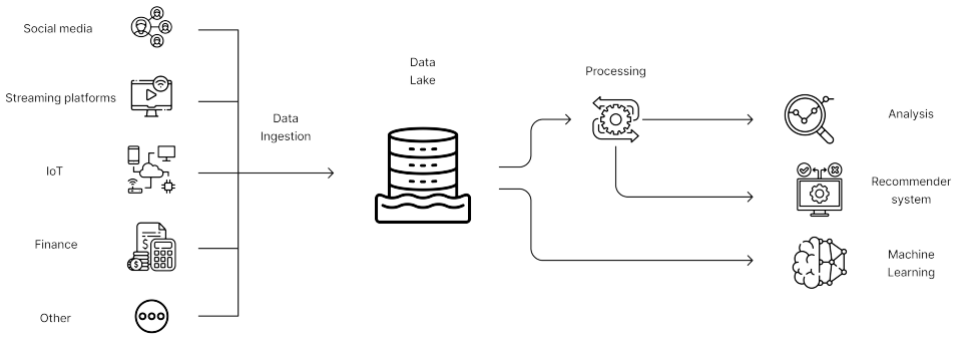

Figura 2.1. Organizzazione di un Data Lake

D'altra parte, i **Data Mart** (DM) si presentano come sistemi simili ai data warehouse, dai quali riprendono la struttura organizzativa e il principio di centralizzazione dei dati.

Tuttavia, a differenza dei DWH, i Data Mart sono progettati per rispondere a esigenze informative specifiche e circoscritte, e per questo motivo vengono spesso definiti come *data warehouse di tipo tematico* (Figura 2.2) [4]. In essi vengono archiviati dati consolidati e coerenti relativi a una singola funzione aziendale o a un'area di business ben definita, come ad esempio le vendite, il marketing o la gestione delle risorse umane.

Esistono due approcci principali per la costruzione di un Data Mart. Il primo prevede l'alimentazione diretta del DM a partire dalle sorgenti dati transazionali relative all'area di interesse, una modalità che consente una realizzazione più rapida ma che può risultare limitata in termini di integrazione e visione trasversale. Il secondo approccio, invece, prevede che i Data Mart vengano popolati a partire dal data warehouse centrale, garantendo così una maggiore coerenza complessiva tra i vari domini informativi, a fronte però di un maggiore sforzo progettuale iniziale.

L'adozione dei Data Mart porta con sé diversi vantaggi operativi. Innanzitutto, il volume di dati trattato è solitamente più contenuto rispetto a quello gestito da un DWH completo, con conseguenti benefici in termini di performance: le interrogazioni risultano più veloci e l'amministrazione del sistema risulta più semplice e meno onerosa. Inoltre, la segmentazione dei dati per area funzionale rappresenta un vantaggio anche sul piano della sicurezza, in quanto consente di limitare l'accesso alle informazioni solo ai reparti direttamente interessati, riducendo il rischio di esposizione non autorizzata.

Tuttavia, questa suddivisione tematica può comportare anche alcune criticità. In particolare, la replicazione di dati comuni tra più Data Mart può generare ridondanze e difficoltà nella gestione della consistenza informativa. Inoltre, la frammentazione dei dati può ostacolare una visione integrata e globale del business, aspetto fondamentale in fase di analisi strategica e di supporto alle decisioni aziendali.

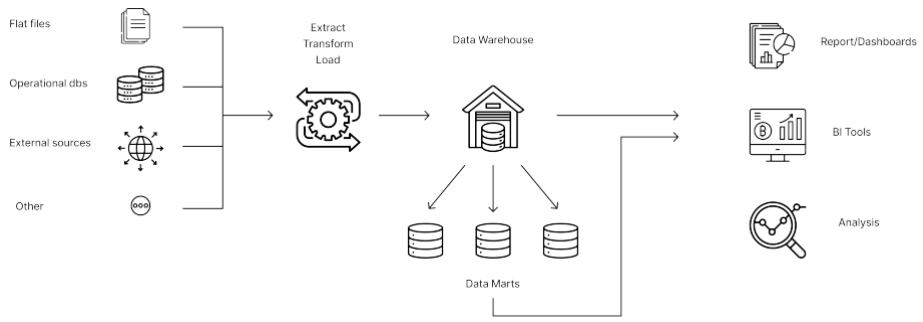

Figura 2.2. Organizzazione di un Data Warehouse

Negli ultimi anni è emersa una nuova architettura per la gestione dei dati, denominata **Data Lakehouse**, che si propone di integrare i punti di forza propri sia dei data warehouse sia dei Data Lake. Questo modello ibrido nasce con l'obiettivo di superare i limiti delle due soluzioni tradizionali, unificando in un'unica piattaforma le funzionalità di archiviazione strutturata e la flessibilità del trattamento di dati non strutturati.

In particolare, un Data Lakehouse è in grado di supportare la memorizzazione simultanea di dati strutturati, semi-strutturati e non strutturati in formato grezzo, offrendo

la possibilità di analizzarli successivamente in base alle esigenze. L'elaborazione dei dati può avvenire mediante tecnologie analitiche avanzate, adatte alla complessità e varietà dei dati non strutturati, oppure tramite strumenti e metodologie tradizionali per l'analisi dei dati strutturati.

Questa architettura consente quindi di combinare la scalabilità e la flessibilità dei Data Lake con l'affidabilità e le prestazioni analitiche tipiche dei data warehouse, rappresentando una soluzione moderna e versatile per l'analisi dei dati aziendali.

In Figura 2.3 è riportata una rappresentazione schematica dei principali passaggi e caratteristiche che distinguono l'organizzazione dei dati nei modelli di Data Warehouse, Data Lake e Data Lakehouse.

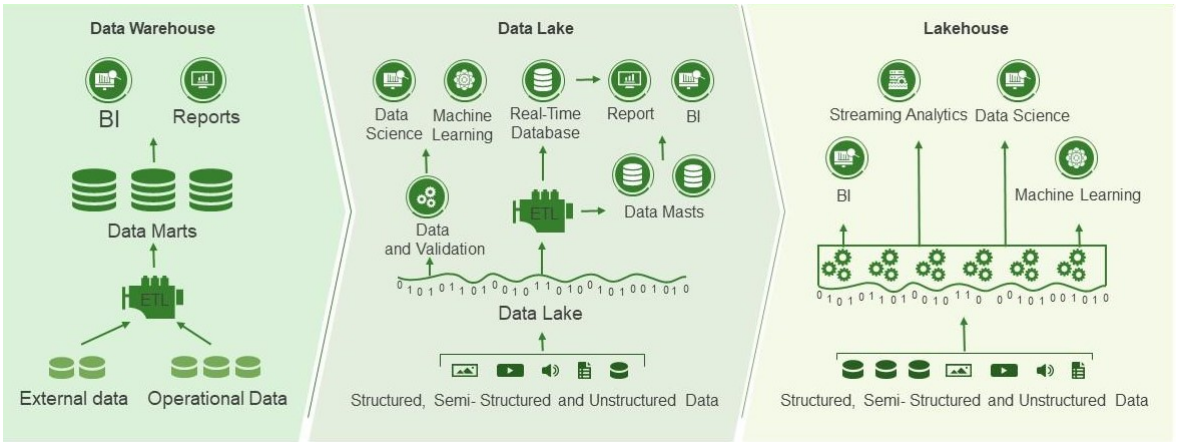

Figura 2.3. Data Mart - Data Lake - Data Lakehouse

2.1.2 Confronto tra Star Schema e Snowflake Schema

Dopo aver analizzato le diverse architetture attraverso cui le aziende possono raccogliere e organizzare i propri dati, risulta fondamentale definire l'approccio da adottare per la progettazione dei database all'interno dei sistemi di data warehousing.

A differenza dei sistemi informazionali tradizionali, che si basano prevalentemente su schemi entità-relazione (ER), i data warehouse fanno uso del cosiddetto **modello dimensionale**. Quest'ultimo è stato concepito con l'obiettivo di semplificare la struttura del database, agevolando il processo di estrazione, navigazione e analisi dei dati da parte degli utenti finali.

Gli schemi ER sono tipicamente caratterizzati da una struttura simmetrica e normalizzata, in cui le informazioni sono suddivise in entità, ossia oggetti o concetti rappresentati da tabelle, e relazioni, ovvero i legami logici tra le entità. Sebbene la normalizzazione consenta di ridurre la ridondanza e i duplicati, questa simmetria può risultare poco intuitiva per gli utenti non tecnici, poiché rende più complessa la comprensione della struttura dei

dati e la scrittura delle interrogazioni. Inoltre, la possibilità di collegare le tabelle seguendo percorsi alternativi può generare ambiguità, producendo risultati differenti a seconda del percorso scelto.

Per queste ragioni, mentre il modello entità-relazione risulta più adatto per i sistemi operativi, dove l'integrità e la coerenza dei dati sono prioritari; nei contesti di data warehousing si preferisce adottare il modello dimensionale. Questo modello si presta maggiormente all'analisi dei dati, in quanto consente un'interrogazione più rapida, semplice e intuitiva, rispondendo in modo efficace alle esigenze informative degli utenti aziendali.

Il modello dimensionale si distingue per una struttura asimmetrica, organizzata attorno a una tabella centrale di grandi dimensioni, detta **fact table** (tabella dei fatti), che raccoglie le misure quantitative rilevanti per il business. Questa tabella è collegata, tramite chiavi univoche, a un insieme di tabelle periferiche chiamate **dimensioni** (tabelle dimensionali), che rappresentano gli attributi descrittivi necessari per contestualizzare i fatti.

Il modello dimensionale può essere implementato secondo due principali schemi architetturali: lo schema a stella (star schema) e lo schema a fiocco di neve (snowflake schema).

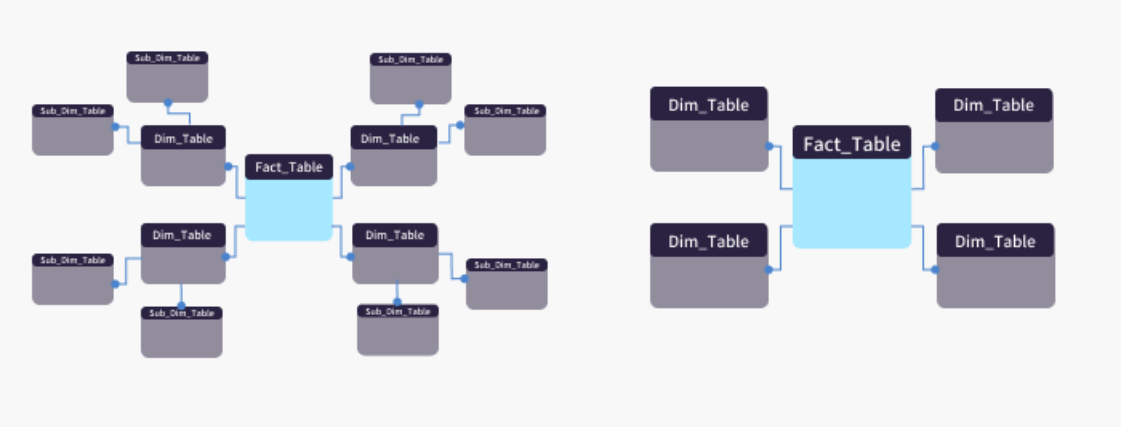

Figura 2.4. Snowflake schema vs Star schema

Lo **star schema** rappresenta la struttura più comunemente adottata nella progettazione dei data warehouse, in quanto garantisce una maggiore semplicità di interpretazione e navigazione. Come suggerisce il nome, questo schema assume la forma di una stella, con una fact table centrale circondata da tabelle dimensionali, le quali sono collegate esclusivamente alla tabella dei fatti tramite chiavi univoche, e non tra loro. In genere, solo la fact table è normalizzata: ciascuna riga corrisponde a una combinazione univoca delle chiavi delle dimensioni, associate a misure numeriche che descrivono fenomeni di business. Le dimensioni, al contrario, sono spesso denormalizzate per ridurre il numero di join necessari nelle interrogazioni, favorendo così le prestazioni, pur a fronte di una maggiore

occupazione di spazio.

Lo star schema e lo **snowflake schema** condividono una struttura simile, basata su una fact table centrale circondata da più tabelle dimensionali. La differenza principale risiede nel grado di normalizzazione delle dimensioni (Figura 2.4). Nello snowflake schema, infatti, le tabelle dimensionali sono ulteriormente normalizzate secondo una struttura gerarchica, in cui ogni livello è rappresentato da una tabella separata. Ad esempio, per modellare la dimensione *Luogo*, nello star schema si potrebbe utilizzare un'unica tabella contenente attributi come indirizzo, città e regione; nello snowflake schema, invece, questi attributi verrebbero suddivisi in più tabelle, come *Indirizzo*, *Città* e *Regione*, collegate tra loro mediante chiavi esterne che riflettono la gerarchia.

Questa organizzazione consente di ridurre la ridondanza dei dati, ma al tempo stesso aumenta il numero di join richieste per eseguire le query, con conseguenti impatti negativi sulle performance. Tuttavia, dal punto di vista della capacità di memoria, l'effetto della normalizzazione è generalmente trascurabile. La scelta tra i due schemi dipende quindi da fattori progettuali e pratici, tra cui il tipo di tool adottato per l'analisi, poiché alcuni strumenti risultano ottimizzati per specifiche architetture. Sarà pertanto compito del data warehouse architect valutare la soluzione più adatta in base al contesto applicativo e agli strumenti disponibili.

2.1.3 OLAP e OLTP

Una volta identificate le principali architetture alla base dei database, è fondamentale analizzare i sistemi destinati all'elaborazione e alla gestione dei dati. Questi sistemi possono essere classificati in due macro-categorie, in base alla tipologia di operazioni che sono progettati per supportare: i sistemi OLTP (Online Transaction Processing) e i sistemi OLAP (Online Analytical Processing).

I sistemi **OLTP (Online Transaction Processing)**, come suggerisce il nome stesso, sono progettati per gestire le transazioni quotidiane sui database operazionali. Tali transazioni sono generalmente di dimensioni contenute, coinvolgono un numero limitato di record e avvengono in tempo reale, con un'elevata frequenza. In questo contesto, l'obiettivo primario è garantire l'affidabilità e la rapidità nella gestione di ogni singola transazione. È quindi tollerata la presenza di campi non valorizzati, purché non compromettano l'integrità dell'operazione; al contrario, non sono ammesse ambiguità nei dati, che potrebbero compromettere la coerenza dell'intero sistema transazionale.

Al contrario, per l'analisi di grandi volumi di dati storici, come avviene nei contesti decisionali, è fondamentale disporre di informazioni complete e consolidate. In tale ambito si collocano i sistemi **OLAP (Online Analytical Processing)**, progettati per supportare attività di analisi multidimensionale su grandi quantità di dati memorizzati nei data warehouse [3]. Questi sistemi si concentrano sulla lettura e aggregazione dei dati, permettendo agli utenti aziendali, come manager, business analyst e decision maker, di esplorare i dati in modo interattivo, identificare pattern e tendenze e ottenere insight a supporto del processo decisionale. In un contesto OLAP, la presenza di valori nulli può compromettere

l'esito delle query aggregate, motivo per cui è richiesta una maggiore completezza e pulizia del dato.

Oltre alle differenze relative alla tipologia di operazioni supportate, OLTP e OLAP si distinguono per altre caratteristiche strutturali, come riassunto nella Tabella 2.1. I sistemi OLTP sono generalmente orientati all'applicazione, operano su dati recenti e si focalizzano sull'efficienza nell'esecuzione di un alto numero di operazioni brevi. I sistemi OLAP, invece, sono orientati all'analisi dei dati, tipicamente di tipo storico, e sono progettati per offrire una visione consolidata del business, utile a supportare valutazioni strategiche e tattiche [6].

Caratteristica	**OLTP**	**OLAP**
Tipo di operazioni	Transazioni quotidiane (inserimento, aggiornamento, cancellazione)	Analisi di dati storici e reportistica
Funzione	Gestione delle operazioni quotidiane	Supporto decisionale e analisi di grandi volumi di dati
Tipologia di dati	Dati strutturati, normalizzati, recenti	Dati denormalizzati, aggregati, storici
Volume delle operazioni	Alto numero di operazioni	Query complesse su grandi volumi di dati
Tipologia di dati	Dati transazionali (es. ordini, acquisti)	Dati aggregati (es. somme, medie, conteggi)
Integrazione dei dati	Operazioni in tempo reale, spesso aggiornamenti frequenti	Raramente modificati, dati storici aggregati
Tipo di utenti	Operatori	Decision-maker

Tabella 2.1. Confronto tra OLAP e OLTP

In fase di progettazione di un sistema di data warehousing, si pone quindi il problema se mantenere separati i due ambienti OLTP e OLAP oppure cercare soluzioni che ne permettano l'integrazione. La prassi prevede una separazione fisica e logica dei due sistemi, in quanto sono pensati per scopi profondamente diversi, fanno riferimento a utenti distinti, utilizzano organizzazioni dei dati differenti e presentano anche importanti divergenze tecniche. Tuttavia, negli ultimi anni, si è assistito a un crescente interesse verso soluzioni ibride che permettano di integrare capacità analitiche all'interno dei sistemi transazionali, con l'obiettivo di abilitare l'analisi in tempo reale e migliorare la competitività aziendale attraverso un processo decisionale più rapido e informato.

In [5] vengono presentati quattro possibili approcci per l'integrazione tra OLTP e OLAP. Tuttavia, al momento della pubblicazione dello studio, non erano ancora disponibili risultati sperimentali sufficienti per valutare con precisione le prestazioni di ciascun

approccio, rendendo prematura l'identificazione di una soluzione ottimale.

2.2 Processo ETL

Negli ultimi decenni, le aziende si sono trovate a dover affrontare un aumento esponenziale della quantità di dati raccolti, provenienti da fonti eterogenee e distribuite. Questa crescente complessità ha reso la gestione e l'analisi dei dati aziendali un'attività sempre più articolata e onerosa. Di conseguenza, è emersa la necessità di implementare processi di controllo della qualità dei dati che siano affidabili, rapidi e standardizzati [20].

In questo contesto si inseriscono i **processi ETL (Extract, Transform, Load)**, fondamentali per garantire che i dati inseriti all'interno dei data warehouse siano estratti correttamente, trasformati secondo criteri di qualità e coerenza, e infine caricati in modo strutturato e integrato. Tali processi costituiscono il cuore operativo dei sistemi di data warehousing, in quanto permettono di consolidare i dati provenienti da sorgenti diverse, assicurando la loro affidabilità e utilità a fini analitici.

Il processo ETL si articola in tre fasi principali [18]:

- **Estrazione** (Extract): in questa fase, i dati vengono prelevati da sorgenti eterogenee, con l'obiettivo di raccogliere tutte le informazioni rilevanti per l'analisi. L'estrazione può avvenire in due modalità: una estrazione iniziale (initial load), eseguita una sola volta per popolare il data warehouse vuoto, e delle estrazioni incrementali, utilizzate successivamente per aggiornare periodicamente il DWH con i dati modificati o aggiunti rispetto all'ultimo caricamento.

- **Trasformazione** (Transform): in questo step, i dati estratti vengono elaborati per renderli coerenti con la struttura e i requisiti del data warehouse aziendale. L'attività di trasformazione ha come obiettivo primario il miglioramento della qualità dei dati, attraverso operazioni di pulizia che rimuovono o correggono valori duplicati, mancanti, inconsistenti, errori di battitura, convenzioni diverse o identificatori non univoci [16].

- **Caricamento** (Load): la fase finale consiste nel trasferimento dei dati trasformati all'interno del data warehouse. Anche in questo caso è possibile scegliere tra due approcci: il refresh, che sovrascrive completamente i dati esistenti, e l'update, che aggiorna soltanto i record modificati o nuovi rispetto al caricamento precedente [5].

2.2.1 Confronto tra ETL e ELT

Negli ultimi anni si sono affermate due strategie principali per l'integrazione e la gestione dei dati: i processi ETL (Extract, Transform, Load) e ELT (Extract, Load, Transform), come illustrato in Figura 2.5. Sebbene entrambi seguano gli stessi passaggi fondamentali, questi due approcci presentano caratteristiche e vantaggi differenti, che ne influenzano l'applicazione in funzione delle specifiche esigenze di business e dell'infrastruttura tecnologica disponibile [17].

La differenza sostanziale tra ETL ed ELT riguarda l'ordine con cui viene eseguita la fase di trasformazione dei dati. In entrambi i casi, il processo inizia con l'estrazione dei

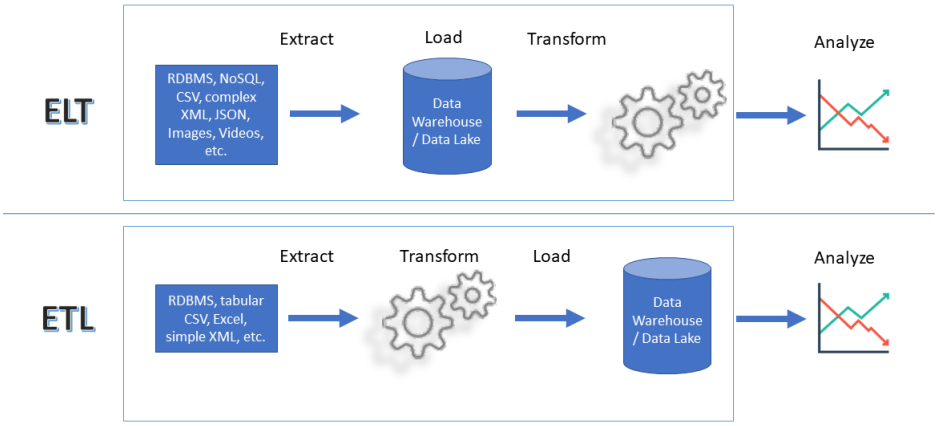

Figura 2.5. Processo ETL vs Processo ELT

dati dalle diverse sorgenti. Nel modello tradizionale **ETL**, la trasformazione dei dati avviene subito dopo l'estrazione, prima che i dati vengano caricati nel data warehouse di destinazione. Questo significa che i dati vengono elaborati, puliti e consolidati in un ambiente intermedio, esterno al DWH, per poi essere caricati in forma già pronta per l'analisi. Tale metodo garantisce che il data warehouse contenga esclusivamente dati accurati e coerenti, tuttavia richiede risorse di calcolo dedicate a supportare l'operazione di trasformazione, spesso complessa e dispendiosa in termini di tempo.

Al contrario, il processo **ELT** sfrutta la capacità dei sistemi moderni di gestire grandi quantità di dati e di effettuare trasformazioni direttamente all'interno del data warehouse stesso. In questo scenario, i dati vengono dapprima estratti e caricati nel DWH in forma grezza, per essere poi trasformati secondo le necessità, utilizzando le risorse del sistema target. Questo approccio elimina la necessità di un ambiente di trasformazione separato, ma impone al data warehouse di disporre di un ampio spazio di archiviazione e di elevate capacità computazionali per gestire i processi di trasformazione internamente.

Nella Tabella 2.2 vengono sintetizzate in modo schematico le principali differenze tra i processi ETL e ELT, evidenziandone vantaggi, svantaggi e scenari di utilizzo più adatti.

2.3 Generative AI

Una volta introdotti i principi fondamentali dell'integrazione dei dati, risulta essenziale, al fine di delineare il contesto in cui si colloca questo progetto, approfondire i concetti chiave legati alla Generative AI e alla Generative ETL, analizzandone i principali vantaggi, nonché i potenziali rischi associati all'impiego di tali tecnologie.

L'**Intelligenza Artificiale Generativa** (IA Generativa) rappresenta una delle evoluzioni più recenti e rilevanti nell'ambito dell'intelligenza artificiale. Essa si distingue per la

Caratteristica	ETL	ELT
Ordine delle operazioni	Dati trasformati prima del caricamento	Caricamento dei dati prima della trasformazione
Requisiti di sistema	Strumento ETL dedicato per la trasformazione dei dati	Utilizza la potenza dei DWH moderni o storage cloud
Elaborazione dei dati	Adatto a dati di piccoli o medie dimensioni	Ideale per grandi volumi e dati complessi
Prestazioni	Lento, con trasformazioni preliminari	Più rapido, trasformazioni post-caricamento
Qualità dei dati e governance	Maggiore controllo qualità prima del caricamento	Necessita di governance aggiuntiva per dati grezzi
Scalabilità	Limitata dalla capacità del motore di trasformazione	Scala con le capacità del data warehouse
Costo	Maggiori costi iniziali di configurazione e manutenzione	Minori costi iniziali, ma possibili costi operativi più alti

Tabella 2.2. Confronto tra i processi ETL e ELT

capacità di produrre contenuti nuovi e originali, come immagini, video, audio, testi e codice software, mediante l'utilizzo di modelli statistici complessi addestrati su vasti insiemi di dati. A differenza dei sistemi tradizionali, che si limitano a reagire a input predefiniti seguendo regole statiche, i modelli generativi sono in grado di apprendere le strutture sottostanti ai dati di input e generare in modo autonomo risultati che ne riproducono fedelmente le caratteristiche.

Questa tecnologia, in rapida espansione, sta attirando un interesse crescente per il suo potenziale trasformativo in numerosi ambiti, che spaziano dal settore artistico e creativo all'automazione di processi aziendali complessi. Un esempio concreto è l'impiego dell'IA Generativa nell'ambito dell'assistenza IT, dove può essere utilizzata per automatizzare attività ripetitive, rispondere a domande frequenti e migliorare l'esperienza utente. Tuttavia, il suo impatto si estende ben oltre questo contesto, offrendo soluzioni innovative in settori quali la sanità, l'istruzione, il marketing e la produzione industriale [19].

Dal punto di vista tecnico, i modelli generativi si basano su tecniche di machine learning, in particolare sull'utilizzo di reti neurali profonde (deep neural networks), che consentono di apprendere rappresentazioni complesse e stratificate dei dati. Dopo la fase di addestramento, questi modelli sono in grado di generare nuove istanze che seguono schemi e strutture simili a quelle dei dati originari. A differenza dei modelli discriminativi, il cui obiettivo è classificare o distinguere tra differenti categorie, i modelli generativi si concentrano sulla creazione di contenuti autonomi.

Tra gli esempi più evidenti vi è la generazione automatica di articoli, immagini o codice

sorgente, anche a partire da un numero limitato di input. Questo consente di adattare i risultati generati a specifici contesti o esigenze operative. Va sottolineato che la qualità e l'affidabilità del modello generativo dipendono fortemente dalla quantità e dalla qualità dei dati utilizzati durante la fase di addestramento: modelli alimentati con ampi e diversificati dataset tendono a produrre risultati più accurati e coerenti.

I modelli avanzati di IA Generativa, come i modelli di fondazione [7] (ad esempio GPT - Generative Pre-trained Transformer), sono in grado di svolgere una vasta gamma di compiti grazie alla loro potenza e complessità. Due delle loro caratteristiche principali sono l'emersione e l'omogeneizzazione. L'emersione riguarda il fatto che alcuni comportamenti si sviluppano spontaneamente nel modello, come la generazione di formati specifici senza essere stati addestrati a farlo direttamente. L'omogeneizzazione, invece, si riferisce alla capacità di un singolo modello di essere utilizzato per molteplici applicazioni, semplificando lo sviluppo e l'integrazione delle soluzioni tecnologiche.

L'intelligenza artificiale generativa non si limita alla creazione di testi, ma è in grado di comprendere e utilizzare diversi linguaggi, inclusi quelli di programmazione, oltre a generare immagini e sequenze audio. Questa versatilità la rende adatta a molteplici settori, dalla produzione artistica alla risoluzione di problemi complessi in ambito scientifico. Una delle sue caratteristiche più interessanti è la capacità di trasferire competenze acquisite in un ambito ad altri contesti, ampliando notevolmente le sue potenzialità applicative.

Le applicazioni dell'IA generativa sono già presenti in numerosi settori. Ad esempio, strumenti come ChatGPT vengono impiegati per migliorare l'esperienza utente, facilitare la creazione di contenuti e automatizzare diverse attività. In ambito aziendale, questa tecnologia può aumentare la produttività, supportando attività come la generazione automatica di report o la scrittura di codice. Non sorprende, quindi, che l'interesse verso l'IA generativa sia in rapida crescita, dato il suo potenziale nel trasformare profondamente le modalità di lavoro e di interazione con la tecnologia.

Tuttavia, l'adozione su larga scala di questi strumenti comporta anche sfide importanti. Oltre ai vantaggi, è fondamentale considerare aspetti etici e pratici, come l'affidabilità dei contenuti generati, la tutela della privacy e il rischio di un utilizzo improprio. Se affrontati in modo responsabile, questi temi possono contribuire a un impiego consapevole dell'IA generativa, valorizzandone il potenziale trasformativo nel mondo digitale e nel contesto lavorativo.

Nel panorama attuale, le applicazioni dell'intelligenza artificiale generativa sono estremamente diversificate, spaziando da utilizzi generici a implementazioni mirate in specifici settori e funzioni aziendali. Tra i principali casi d'uso si possono individuare i seguenti ambiti [15]:

- **Video**: generazione automatica di contenuti video e sistemi di previsione delle sequenze visive;

- **Immagini**: creazione di immagini a partire da descrizioni testuali, modifica di elementi specifici all'interno di un'immagine, miglioramento della qualità e incremento della risoluzione;

- **Audio**: produzione di voci realistiche, sistemi text-to-speech per la trasformazione del testo in audio;
- **Contenuti scritti**: generazione di testi coerenti, pertinenti e personalizzati in base al contesto e agli obiettivi
- **Codice**: sviluppo e ottimizzazione di codice software, identificazione e correzione automatica di bug;
- **Applicazioni industriali**: la Generative AI può essere applicata in numerosi settori industriali specifici, tra cui:
  - Sanità: supporto alla ricerca di nuove terapie, assistenti virtuali per il monitoraggio dei sintomi;
  - Istruzione: progettazione di materiali didattici personalizzati, supporto al tutoraggio e all'apprendimento individuale;
  - Assicurazioni: rilevamento delle frodi, analisi dei dati per il profiling dei clienti;
  - Retail: previsione dei trend di mercato, assistenza virtuale personalizzata durante il processo d'acquisto.

Tali esempi, unitamente a numerosi altri possibili scenari, mettono in evidenza i vantaggi concreti offerti dall'intelligenza artificiale generativa. Una sua implementazione efficace può rappresentare un importante vantaggio competitivo per le organizzazioni, in quanto consente di automatizzare attività ripetitive, accelerare i processi di sviluppo e favorire la creazione di prodotti e servizi innovativi. Inoltre, la capacità di generare contenuti altamente personalizzati permette alle aziende di distinguersi nel mercato, contribuendo alla fidelizzazione dei clienti.

I benefici derivanti dall'adozione di questa tecnologia includono un'accelerazione nell'innovazione, un miglioramento dell'esperienza utente e un aumento della produttività del personale. Tuttavia, è importante sottolineare che i risultati ottenibili variano in base al contesto di applicazione e alla qualità dell'integrazione nei processi aziendali.

Nonostante il notevole potenziale dell'intelligenza artificiale generativa, questa tecnologia presenta ancora alcune criticità che ne limitano l'adozione diffusa. Uno dei principali limiti è rappresentato dalla possibilità che i modelli generino contenuti imprecisi o incompleti, rendendo necessario l'intervento umano per la validazione dei risultati. Tale esigenza può ridurre significativamente i risparmi di tempo previsti e impattare sull'efficienza complessiva dei processi automatizzati.

Durante il processo di adozione è inoltre fondamentale affrontare le implicazioni etiche legate all'uso di tali sistemi. Risulta essenziale garantire il rispetto di principi fondamentali come la trasparenza, l'equità (fairness) e la responsabilità nell'uso e nella gestione dei dati. A ciò si aggiungono le preoccupazioni degli utenti riguardo alla fiducia nelle decisioni automatizzate, all'interpretabilità dei modelli generativi e alla possibilità di prevenire manipolazioni o abusi intenzionali.

Sebbene l'adozione degli algoritmi generativi sia in costante espansione, è necessario mantenere un approccio critico e consapevole, riconoscendo i principali **rischi e limiti** associati alla tecnologia. Tra i più rilevanti si possono individuare i seguenti aspetti [8]:

- **Scarsa trasparenza**: i modelli di IA generativa sono spesso caratterizzati da una struttura, che ne rende difficile l'interpretazione e la prevedibilità. Per tale ragione, vengono comunemente definiti black box, soprattutto nei contesti in cui manca una comprensione approfondita del loro funzionamento da parte delle organizzazioni che li adottano;
- **Precisione**: i risultati prodotti possono essere errati, incoerenti o decontestualizzati. Questo rende necessaria una verifica continua dei contenuti generati, al fine di garantirne l'affidabilità, soprattutto nei settori in cui l'accuratezza è cruciale;
- **Bias**: l'efficacia dei modelli dipende in larga parte dalla qualità e dalla rappresentatività dei dati di addestramento. L'assenza di controlli adeguati può portare alla riproduzione di distorsioni e pregiudizi, con conseguenze significative in termini di equità e inclusività. È quindi necessario implementare meccanismi di monitoraggio e mitigazione dei bias;
- **Copyright e proprietà intellettuale**: l'uso di modelli generativi solleva importanti interrogativi sulla protezione dei dati e dei contenuti. Attualmente, non vi sono garanzie certe sulla gestione della proprietà intellettuale né sulla riservatezza dei dati immessi nei sistemi. Inoltre, la capacità di generare contenuti simili o derivati da opere esistenti può violare i diritti degli autori, dando luogo a controversie legali ed etiche;
- **Frodi e cybersecurity**: i modelli di IA Generativa possono essere sfruttati da operatori fraudolenti per compiere attacchi informatici e frodi, come nel caso dei deep fake impiegati nell'ingegneria sociale per manipolare il personale. Questo scenario richiede l'adozione di adeguate misure di sicurezza e sistemi di rilevamento delle minacce;
- **Sostenibilità ambientale**: un ulteriore aspetto da considerare riguarda l'impatto ambientale. L'addestramento e il funzionamento di modelli generativi richiedono notevoli risorse computazionali ed energetiche, contribuendo in modo significativo alle emissioni di carbonio. Per favorire uno sviluppo sostenibile della tecnologia, è indispensabile migliorare l'efficienza degli algoritmi e incentivare l'utilizzo di fonti energetiche rinnovabili.

Negli ultimi anni, l'intelligenza artificiale generativa ha raggiunto una rilevanza mai vista prima, attirando l'attenzione non solo di personale del settore ma anche della popolazione intera per il suo straordinario potenziale innovativo. Il successo riscontrato dai modelli generativi è visibile nell'aumento esponenziale delle iscrizioni ai corsi di IA generativa nell'ultimo anno, andando a influenzare profondamente il futuro mercato lavorativo e a ridefinire le richieste dei datori di lavoro in fase di assunzione [10]. Nel panorama lavorativo la generative AI è una delle competenze più richieste; di pari passo la cybersecurity e la gestione del rischio stanno acquisendo sempre maggiore importanza.

In un'intervista rilasciata a Forbes [11], Jim Wilson, Direttore Generale Globale per la Leadership Intellettuale e la Tecnologia presso Accenture, analizza l'impatto dell'Intelligenza Artificiale Generativa sul mondo del lavoro, evidenziando l'emergere di nuove figure professionali all'interno delle aziende. Wilson individua tre principali categorie di ruoli tecnici specializzati che prenderanno piede con la diffusione di questa tecnologia: i trainers, responsabili della progettazione, dell'addestramento e dell'ottimizzazione dei modelli di IA; gli explainers, che si occupano di interpretare i risultati prodotti dai modelli e di sviluppare interfacce che ne facilitino la comprensione all'interno dell'organizzazione; infine, i sustainers, figure chiave per garantire che i sistemi di IA operino in modo affidabile, etico e sostenibile nel tempo.

Oltre alla creazione di nuovi profili professionali, Wilson evidenzia tre modalità principali attraverso cui l'IA generativa è destinata a trasformare i ruoli lavorativi esistenti. La prima è l'amplificazione, che consiste nel potenziamento delle capacità analitiche e creative delle persone grazie al supporto dell'IA. Segue l'interazione, che introduce nuove forme di collaborazione tra esseri umani e interfacce IA, modificando il modo in cui si accede e si utilizza l'informazione. Infine, vi è l'incarnazione, ovvero l'estensione delle capacità fisiche dell'essere umano attraverso l'integrazione dell'intelligenza artificiale con tecnologie robotiche, come nel caso della robotica collaborativa applicata ai contesti produttivi.

2.3.1 Generative ETL

Nel contesto dell'intelligenza artificiale generativa, il ruolo centrale dei dati nello sviluppo tecnologico del mondo moderno appare ancora più evidente. Oltre alla loro già riconosciuta importanza in ambito aziendale, dove supportano i decision maker nelle analisi dei KPI, i dati rappresentano un elemento essenziale per l'addestramento dei modelli generativi. Questi modelli, a loro volta, sono capaci di trasformare, pulire e strutturare i dati, rendendoli più facilmente utilizzabili per l'analisi.

Per ottenere modelli generativi di alta qualità, è fondamentale disporre di dataset altrettanto validi: ovvero grandi volumi di dati puliti, coerenti e bilanciati. Parallelamente, i nuovi strumenti basati su generative AI consentono di automatizzare molte operazioni di data preparation, riducendo il carico di lavoro manuale e liberando risorse all'interno dei team aziendali.

Sono numerosi gli esempi di applicazione dell'IA generativa nel trattamento dei dati, tra le quali si possono citare:

- **Generative SQL**: un nuovo strumento progettato per semplificare l'analisi statistica di dati strutturati. GenSQL [24] è un tool sviluppato dai ricercatori del MIT che consente di combinare dataset tabulari con un modello di intelligenza artificiale probabilistico generativo, migliorando la flessibilità del processo decisionale. Ad esempio, può generare dati sintetici, rilevare anomalie, correggere errori e prevedere tendenze future.

- **Generative ETL**: nell'analisi dei dati uno degli aspetti più impegnativi, in termini di risorse e tempo, è la fase di trasformazione dei dati. Tuttavia, grazie alla

generative AI, è possibile semplificare questo processo, riducendo il numero di operazioni manuali necessarie, come l'analisi delle tabelle di origine e di destinazione, la creazione dei mapping tra esse e l'integrazione di più fonti [22].

- **Generative BI**: nel campo della Business Intelligence generativa, l'obiettivo è rendere i dati accessibili anche a utenti privi di competenze tecniche. Gli strumenti di BI generativa consentono di interrogare i database aziendali utilizzando il linguaggio naturale. In questo modo si alleggerisce il lavoro dei data analyst, che possono concentrarsi su compiti più complessi, mentre gli utenti interni ottengono facilmente report e dashboard a supporto delle decisioni.

- **Generazione di dati sintetici**: la generazione di dati sintetici rappresenta una soluzione strategica in contesti in cui i dati reali sono limitati o soggetti a restrizioni, come nel settore sanitario o bancario. In questi casi, i modelli generativi permettono di creare dataset artificiali che mantengono le proprietà statistiche dei dati originali, rendendoli utilizzabili per l'addestramento e la validazione dei modelli, senza compromettere la privacy o la sicurezza.

L'emergere della Generative ETL (Extract, Transform, Load) rappresenta un'evoluzione significativa nell'ambito della gestione dei dati, grazie all'applicazione delle tecniche di intelligenza artificiale generativa ai tradizionali processi di integrazione dei dati. Questa nuova metodologia si basa sull'impiego di modelli generativi per automatizzare le fasi di estrazione, trasformazione e caricamento, permettendo la creazione di flussi di dati intelligenti, adattivi e ottimizzati.

A differenza dei processi ETL tradizionali, spesso rigidi e soggetti a una forte dipendenza da interventi manuali, la Generative ETL è in grado di adattarsi dinamicamente a modifiche nei dataset e a cambiamenti nei requisiti aziendali. Tale approccio ha il potenziale per rivoluzionare la gestione dei dati aziendali, migliorando l'efficienza operativa, riducendo il margine di errore e diminuendo la necessità di attività ripetitive da parte degli operatori.

Nell'ambito delle moderne strategie di gestione dei dati, l'integrazione rappresenta un elemento cruciale, poiché consente di costruire una visione completa, coerente e centralizzata del patrimonio informativo aziendale. Superare la frammentazione dei dati, spesso distribuiti tra database, applicazioni e sistemi differenti, è fondamentale per supportare decisioni strategiche basate su dati affidabili e aggiornati.

L'intelligenza artificiale sta rivoluzionando il modo in cui le organizzazioni affrontano l'integrazione dei dati. Grazie alla sua capacità di automatizzare processi complessi, aumentare l'accuratezza e gestire grandi volumi di informazioni eterogenee, l'IA si sta affermando come un alleato chiave nell'evoluzione delle architetture data-driven [23].

Tra le principali strategie di integrazione supportate dall'IA troviamo l'utilizzo di algoritmi di machine learning (ML) e di elaborazione del linguaggio naturale (NLP) per il riconoscimento, l'allineamento e la fusione automatica dei dati provenienti da fonti diverse. Questi sistemi permettono di ridurre sensibilmente l'intervento umano e di migliorare la qualità dell'integrazione.

Un'altra tendenza emergente riguarda l'integrazione dei dati in tempo reale. Tecnologie come lo stream processing e le architetture event-driven consentono di acquisire, elaborare e integrare i dati non appena vengono generati, offrendo così una visione costantemente aggiornata delle informazioni aziendali.

L'IA è inoltre in grado di gestire in modo intelligente la mappatura e la trasformazione degli schemi dei dati. Questo significa poter convertire e armonizzare strutture provenienti da database relazionali, sistemi NoSQL e altre tipologie di archivi, affrontando anche l'evoluzione degli schemi nel tempo in maniera automatizzata.

Infine, l'impiego dei knowledge graph consente di realizzare un'integrazione semantica dei dati, superando i limiti delle tradizionali tecniche di gestione informativa. Questi strumenti permettono di rappresentare in modo strutturato le relazioni tra entità e concetti, arricchendo il contesto dei dati disponibili e migliorandone la comprensione. Grazie a un approccio basato sul significato e non soltanto sulla struttura, i knowledge graph facilitano l'interrogazione e l'analisi di fonti eterogenee, offrendo una maggiore flessibilità e precisione nelle operazioni di ricerca e correlazione delle informazioni.

Se gestite in modo appropriato, l'intelligenza artificiale generativa e la Generative ETL offrono un potenziale straordinario per la trasformazione dei processi aziendali. Queste tecnologie, infatti, non si limitano a incrementare l'efficienza operativa, ma consentono anche l'emergere di nuovi modelli di business, promuovendo l'adozione di soluzioni tecnologiche più innovative, scalabili e sostenibili nel tempo.

Grazie alla capacità di automatizzare e ottimizzare l'intero ciclo di vita dei dati tali strumenti permettono alle organizzazioni di affrontare in modo più efficace la complessità crescente dei contesti informativi moderni. Ne deriva una gestione dei dati più dinamica, adattabile ai cambiamenti e maggiormente incentrata sul valore strategico dell'informazione. Questo approccio contribuisce a rendere i processi aziendali non solo più agili, ma anche orientati all'innovazione continua e alla competitività a lungo termine.

Capitolo 3

Progettazione e Realizzazione dei Moduli

La società Mediamente Consulting srl, presso la quale è stata svolta l'attività di tesi, ha espresso interesse nell'adozione e nell'ottimizzazione di un modello standardizzato per la creazione dei flussi ETL. L'obiettivo principale è l'automazione del codice impiegato per l'alimentazione dei data warehouse aziendali, al fine di ridurre tempi di sviluppo, costi operativi ed errori tipici dei processi manuali.

In questo contesto si inserisce il progetto della presente tesi, che rappresenta un primo passo concreto verso l'obiettivo di automatizzare i flussi ETL. A tal fine, sono stati sviluppati script in grado di automatizzare la generazione di alcune componenti del framework aziendale, con l'intento di ridurre sia il tempo impiegato dal personale tecnico nelle attività ripetitive, sia gli errori che possono derivare da un'esecuzione manuale.

La prima parte del capitolo è dedicata alla presentazione dello strumento selezionato, con un approfondimento sulle motivazioni che hanno guidato questa scelta e una descrizione della sua struttura fondamentale. A seguire, viene illustrato il framework ETL sviluppato dall'azienda, progettato per standardizzare la pipeline di caricamento dei dati, indipendentemente dallo strumento di integrazione utilizzato. Questo framework è stato concepito per garantire scalabilità, conformità e facilità di manutenzione nel tempo. Infine, il capitolo successivo è dedicato a un'analisi approfondita delle diverse fasi operative che hanno caratterizzato l'implementazione del progetto.

Nel panorama attuale sono presenti numerosi strumenti di data integration, ciascuno con caratteristiche specifiche progettate per soddisfare esigenze aziendali differenti. Questa ampia disponibilità rende particolarmente complessa l'individuazione della soluzione più adatta, soprattutto nella fase iniziale di analisi dei requisiti del cliente.

A supporto di questa valutazione, il **Magic Quadrant** elaborato da Gartner [21], società leader nel settore della ricerca e consulenza strategica, rappresenta uno strumento di riferimento. Il Magic Quadrant classifica i principali fornitori di una determinata tecnologia o mercato sulla base di due criteri fondamentali:

- **Completezza della visione (Completeness of Vision)**, rappresentata sull'asse orizzontale, che valuta la capacità del fornitore di innovare, anticipare i trend di mercato e definire una direzione strategica chiara.
- **Capacità di esecuzione (Ability to Execute)**, rappresentata sull'asse verticale, che misura l'efficacia con cui l'azienda realizza la propria visione, in termini di qualità del prodotto, servizi di supporto, presenza commerciale e solidità finanziaria.

L'intersezione tra questi due criteri genera una classificazione suddivisa in quattro quadranti, all'interno dei quali sono collocati i principali competitor di mercato [21]:

- **Leaders**: aziende caratterizzate da una solida capacità esecutiva e da una visione strategica ben definita, riconosciute come i principali punti di riferimento nel settore.
- **Visionaries**: realtà con un'elevata capacità di innovazione e una chiara visione strategica, ma che presentano ancora margini di miglioramento nell'esecuzione.
- **Challengers**: attori con una forte capacità esecutiva, ma una visione meno sviluppata o innovativa rispetto ai leader.
- **Niche Players**: fornitori specializzati in segmenti di mercato specifici o che devono ancora consolidare sia la strategia sia l'efficacia esecutiva.

Come mostrato nella Figura 3.1, relativa all'edizione di novembre 2024 del Magic Quadrant, tra i 20 fornitori analizzati si distinguono, nel quadrante dei Leaders, aziende di rilievo come Microsoft, Informatica, IBM e Oracle.

Nel dettaglio, i fornitori classificati nel quadrante dei Leaders si distinguono per la capacità di supportare un'ampia gamma di approcci all'integrazione dei dati, adattandosi ai diversi casi d'uso aziendali. Tra questi rientrano l'alternanza tra pipeline ETL/ELT, la replica e sincronizzazione dei dati, il data streaming e la virtualizzazione. Inoltre, questi vendor offrono funzionalità avanzate come la gestione attiva dei metadati, architetture data fabric, data mesh, lakehouse e la creazione di veri e propri data product. Un ulteriore elemento distintivo è l'integrazione con soluzioni di intelligenza artificiale e intelligenza artificiale generativa, tramite funzionalità incorporate o modelli come i LLM, abilitando pattern come il Retrieval-Augmented Generation (RAG).

Per rafforzare la propria competitività sul mercato, l'azienda Mediamente Consulting ha scelto di concentrare i propri investimenti su due di questi strumenti in particolare: Microsoft e Oracle.

Oracle si conferma anche quest'anno tra i Leaders del Magic Quadrant [21]. Con sede negli Stati Uniti, l'azienda offre come principale piattaforma dati l'Oracle Cloud Infrastructure (OCI), che include servizi avanzati di data integration. Tra gli strumenti più rilevanti vi sono Oracle GoldenGate e Oracle Data Integrator (ODI), apprezzati per la loro affidabilità e flessibilità. Grazie a una presenza globale e a un'offerta completa, Oracle rappresenta una scelta strategica per aziende che intendono investire in soluzioni robuste e scalabili per migliorare i propri processi di integrazione dati.

Anche **Microsoft** si conferma tra i Leaders del Magic Quadrant per il secondo anno consecutivo [21]. Con sede nello stato di Washington, l'azienda propone come principali

strumenti di data integration Azure Data Factory (ADF), SQL Server Integration Services (SSIS) e Power Query. La sua piattaforma dati principale è Microsoft Fabric, che integra al suo interno servizi di integrazione dati completi e avanzati.

Figura 3.1. Tool di Data Integration

3.1 Oracle Data Integrator (ODI)

Il progetto di tesi è stato sviluppato adottando **Oracle Data Integrator (ODI)** come piattaforma per la gestione e l'integrazione dei dati. ODI rappresenta una delle soluzioni di punta di Oracle per la gestione dei processi ETL (Extract, Transform, Load) e si distingue nel settore per la sua robustezza e ampia diffusione.

La decisione di impiegare questa specifica piattaforma è stata guidata da molteplici fattori: la sua elevata competitività sul mercato, il suo vasto impiego presso numerosi

clienti dell'azienda e, in particolare, la sua compatibilità con ODI Generator. Quest'ultimo è uno strumento che ne amplifica l'efficacia, permettendo la generazione automatica di componenti di Oracle Data Integrator, quali repository, mappature e scenari. Utilizzando un linguaggio dichiarativo ispirato alla sintassi SQL.

Oracle Data Integrator (ODI) è un software di integrazione dati ampiamente adottato, riconosciuto per il suo approccio dichiarativo nella definizione dei processi di trasformazione e integrazione. Questo modello consente uno sviluppo e una manutenzione dei flussi dati più rapidi ed efficienti [13]. La sua efficacia deriva principalmente dall'adozione di un'**architettura ELT (Extract, Load, Transform)**, che garantisce elevate prestazioni nelle fasi di trasformazione e validazione dei dati, configurandosi anche come una soluzione economicamente vantaggiosa.

L'architettura ELT di ODI si distingue significativamente dalle tradizionali architetture ETL (Extract, Transform, Load). Nei sistemi ETL classici, i dati vengono estratti dalle sorgenti, subiscono una trasformazione su un motore ETL dedicato, spesso residente su un server separato, e solo successivamente vengono caricati nel sistema di destinazione. Al contrario, l'approccio ELT elimina la necessità di un server ETL specifico e di un motore proprietario per la trasformazione. Questa architettura sfrutta le capacità computazionali native dei sistemi di gestione di database relazionali (RDBMS) già in uso. Questo non solo si traduce in una maggiore efficienza e scalabilità, ma riduce anche la complessità infrastrutturale, minimizzando il movimento dei dati sulla rete e diminuendo la necessità di aree di staging intermedie.

Tra i principali benefici che contribuiscono a rendere Oracle Data Integrator una soluzione competitiva si possono citare [13]:

- **Sviluppo e manutenzione più rapidi e semplici**: grazie a un approccio dichiarativo basato su regole, ODI riduce significativamente la curva di apprendimento e aumenta la produttività degli sviluppatori. Questo modello separa la logica del processo dalla sua implementazione tecnica, semplificando sia la progettazione che la manutenzione nel tempo.

- **Controllo della qualità dei dati**: ODI garantisce che eventuali dati non conformi vengano rilevati automaticamente e gestiti prima dell'inserimento nella tabella di destinazione, senza necessità di scrittura di codice. Offre funzionalità integrate per la convalida e il controllo della qualità dei dati, aiutando a garantire l'accuratezza e la coerenza delle informazioni

- **Prestazioni elevate in fase di esecuzione**: a differenza dei tradizionali strumenti ETL che elaborano i dati riga per riga tramite motori proprietari, ODI sfrutta l'architettura ELT, basata su SQL e su RDBMS, permettendo l'esecuzione di trasformazioni direttamente sul server di destinazione, con un notevole incremento delle prestazioni.

- **Architettura più semplice ed efficiente**: eliminando la necessità di un server ETL intermedio, ODI sfrutta direttamente i server di origine e destinazione per l'esecuzione delle trasformazioni, che avvengono in modalità batch.

- **Indipendenza dalla piattaforma**: ODI è compatibile con numerosi sistemi operativi, piattaforme hardware e ambienti software, garantendo ampia flessibilità d'uso.
- **Integrazione ibrida**: ODI è in grado di gestire sia sorgenti dati on-premise che cloud-based, facilitando la migrazione graduale dei processi di gestione dati al cloud.
- **Ampia connettività**: il software supporta vari RDBMS, incluse le principali piattaforme di data warehousing come Oracle, Exadata, Teradata, IBM DB2, Netezza, oltre a tecnologie come Big Data, file flat, ERP, LDAP e XML.
- **Riduzione dei costi**: l'eliminazione del server ETL e del relativo motore consente un risparmio sui costi di infrastruttura e manutenzione. Inoltre, la maggiore produttività degli sviluppatori e la semplicità d'uso contribuiscono a ridurre significativamente i costi complessivi del progetto, sia nella fase iniziale sia nelle evoluzioni successive.

La piattaforma ODI è composta da quattro componenti principali: il Repository, ODI Studio, l'Agent e la Console [1].

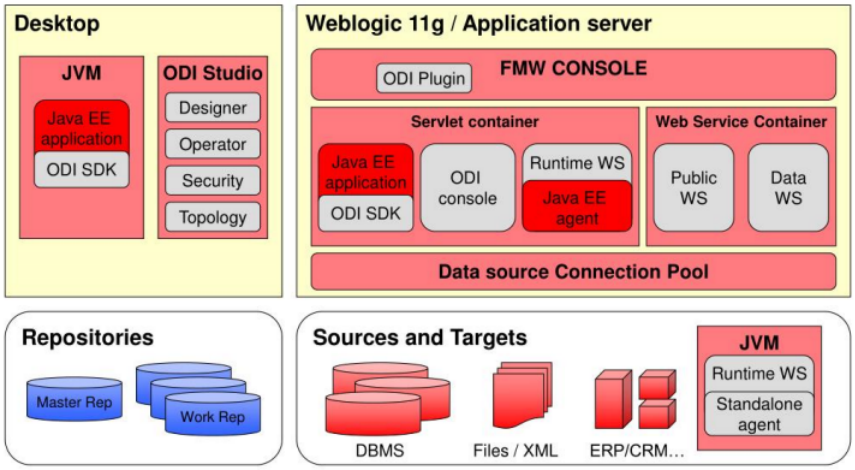

Figura 3.2. Architettura di ODI

3.1.1 Repository

Il Repository costituisce l'elemento centrale dell'architettura di ODI e rappresenta il principale spazio di archiviazione in cui vengono conservate tutte le informazioni gestite dalla piattaforma, come i dettagli di connettività, i metadati, le regole di trasformazione, gli scenari, il codice generato, i log di esecuzione e le statistiche [13]. L'architettura del repository è progettata per supportare più ambienti separati, tra loro interconnessi, che possono scambiarsi metadati e scenari, come per esempio ambienti di sviluppo, test e produzione. Inoltre, il repository svolge anche la funzione di sistema di controllo delle versioni: gli

oggetti al suo interno vengono archiviati e associati a un numero di versione, facilitando così la tracciabilità e la gestione delle modifiche nel tempo.

La struttura del repository si articola in due tipologie: un Master Repository, che contiene dati sensibili; e uno o più Work Repository, destinati a ospitare i dati relativi ai progetti.

Generalmente, è presente un solo **Master Repository**, che include:

- Informazioni relative alla sicurezza, come utenti, profili e diritti di accesso alla piattaforma ODI.
- Informazioni sulla topologia, che comprendono tecnologie, definizioni dei server, schemi e contesti.
- Oggetti versionati e archiviati, utili per il controllo delle modifiche e della loro evoluzione nel tempo.

Il **Work Repository**, invece, contiene gli oggetti effettivamente sviluppati. All'interno della stessa installazione di ODI possono coesistere più Work Repository, ad esempio per gestire ambienti separati. Il Work Repository conserva informazioni relative a:

- Modelli, tra cui definizioni degli schemi, strutture dei datastore e metadati, definizioni di campi e colonne, vincoli di qualità dei dati.
- Progetti, comprendenti regole di business, pacchetti, procedure, cartelle, Knowledge Module, variabili e altri elementi di configurazione.
- Esecuzione degli scenari, con relativi scenari, pianificazioni e log di esecuzione.

Quando un Work Repository viene configurato esclusivamente per contenere informazioni legate all'esecuzione, come accade tipicamente negli ambienti di produzione, prende il nome di Execution Repository.

3.1.2 ODI Studio

ODI Studio rappresenta l'interfaccia grafica della piattaforma e consente agli utenti, siano essi amministratori, sviluppatori o operatori, di accedere ai repository. ODI Studio può essere utilizzato per amministrare l'infrastruttura, effettuare il reverse-engineering dei metadati, sviluppare progetti, pianificare attività, eseguire e monitorare i processi.

ODI Studio è organizzato in quattro diversi moduli (Navigators), ciascuno dei quali è generalmente utilizzato da differenti categorie di utenti, in base al loro ruolo e ai profili di sicurezza assegnati [13].

Il **Security Navigator** è destinato agli amministratori di sistema e ai database administrator (DBA). Viene usato per gestire la sicurezza all'interno di ODI, permette la creazione di utenti, ruoli e profili, oltre a definire i diritti di accesso per vari oggetti all'interno del sistema.

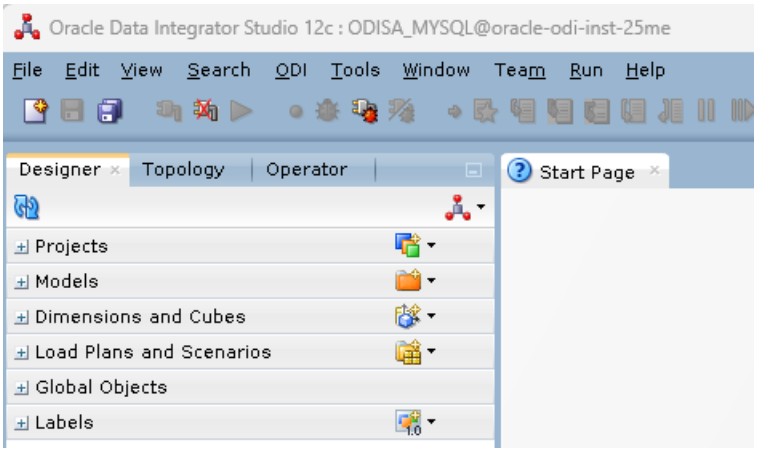

Figura 3.3. Moduli di ODI Studio

Il **Topology Navigator** è responsabile della gestione dell'architettura fisica e logica dei sistemi informativi coinvolti nei processi di integrazione. Attraverso di esso, è possibile definire e amministrare l'intera topologia del sistema, incluse le tecnologie utilizzate, i relativi tipi di dati, i server di dati, gli schemi, i contesti, i linguaggi, gli agenti e i repository. L'architettura logica rappresenta la struttura dei dati in modo indipendente dalle risorse fisiche sottostanti, ed è impiegata nella progettazione dei flussi di dati e delle trasformazioni senza riferimenti a specifiche risorse fisiche. In contrasto, l'architettura fisica descrive i dati in relazione al sistema di archiviazione effettivo, specificando le connessioni a database, tabelle, file e altri oggetti fisici. Ogni schema fisico è collegato a uno schema logico, e l'utilizzo dei contesti consente di associare a uno stesso schema logico molteplici schemi fisici. Questa funzionalità è cruciale per la gestione di ambienti differenti (quali sviluppo, test, produzione) o per l'impiego di risorse simili in contesti geografici o aziendali diversi, permettendo a ODI di eseguire le stesse mappature in ambienti fisici distinti. Il Topology Navigator permette inoltre di configurare le connessioni e le credenziali necessarie affinché ODI possa accedere ai sistemi sorgente e di destinazione. Sebbene queste informazioni siano frequentemente consultate dagli sviluppatori, la loro modifica è riservata a profili con privilegi superiori, garantendo così la sicurezza e la coerenza dell'ambiente di integrazione dati.

Il **Design Navigator** costituisce il cuore dell'ambiente di sviluppo di Oracle Data Integrator (ODI), essendo lo strumento principale per la definizione delle regole dichiarative di trasformazione e integrazione dei dati. Questo modulo offre una visione d'insieme del progetto, consentendo agli sviluppatori di creare e gestire oggetti fondamentali come mapping, pacchetti, piani di caricamento (load plans) e procedure, tutti essenziali per la costruzione dei flussi di integrazione. Oltre a supportare la progettazione e il controllo dell'integrità dei dati, anche attraverso funzionalità di reverse-engineering automatico, il Design Navigator facilita lo sviluppo grafico delle mappature di trasformazione, la visualizzazione intuitiva dei flussi di dati e la generazione automatica della documentazione.

Quest'ultima evidenzia in modo chiaro l'organizzazione logica e fisica delle sorgenti dati, delle destinazioni e delle trasformazioni applicate lungo il percorso dei dati, rendendo lo sviluppo e la manutenzione più rapidi ed efficienti.

L'**Operator Navigator** rappresenta il quarto componente fondamentale di ODI Studio, ed è preposto alla gestione e al monitoraggio delle attività di integrazione dati. Progettato specificamente per gli operatori IT, consente di supervisionare l'intero processo di integrazione, verificare l'esecuzione del codice e, se necessario, effettuare operazioni di debugging. Attraverso questo strumento, è possibile visualizzare registri di esecuzione dettagliati, che includono informazioni cruciali quali il numero di errori, la quantità di dati elaborati e le statistiche di performance, fornendo metriche essenziali per il controllo dell'efficienza e dell'affidabilità delle operazioni. L'Operator Navigator è organizzato in diverse sezioni per una consultazione mirata: la Session List mostra tutte le sessioni con criteri di filtro (data, agente, stato); le Hierarchical Sessions (sessioni gerarchiche) offrono una vista gerarchica delle esecuzioni, comprese le sessioni figlie; Load Plan elenca le esecuzioni dei piani di caricamento; Scheduling presenta gli agenti fisici e i relativi schedulatori configurati; Scenarios contiene l'elenco degli scenari disponibili; infine, Solutions include le soluzioni create nell'ambito della gestione delle versioni.

3.1.3 Run-Time Agent

Durante la fase di progettazione, gli sviluppatori definiscono le regole di business e, sulla base di queste, generano gli scenari che vengono memorizzati all'interno del repository. In fase di esecuzione, il Run-Time Agent recupera il codice degli scenari dal repository, si connette ai server di origine e destinazione dei dati e ne coordina l'esecuzione. Durante questo processo, l'Agent registra nel repository i codici di ritorno, i messaggi di esecuzione e ulteriori informazioni di log, come il numero di record elaborati e il tempo impiegato. Esistono tre tipologie di agenti [13]:

- gli agenti Java Enterprise Edition (Java EE), distribuiti su Oracle WebLogic Server e in grado di sfruttare le funzionalità del livello application server, inclusa l'alta disponibilità;

- gli Standalone Agents, installabili direttamente sui sistemi sorgente o di destinazione e che richiedono solo una Java Virtual Machine;

- i Colocated Standalone Agents, anch'essi installabili sui sistemi di origine o destinazione, ma eseguibili anche su macchine separate rispetto al server di amministrazione Oracle WebLogic, e gestibili tramite Oracle Enterprise Manager.

Gli agenti possono avere una propria pianificazione definita in Oracle Data Integrator, essere invocati da uno scheduler esterno, da un'API Java o tramite interfaccia web service.

3.1.4 ODI Console

Oracle Data Integrator (ODI) Console è un'interfaccia web, progettata per fornire agli utenti aziendali, così come a sviluppatori, amministratori e operatori, accesso in lettura ai repository. Permette inoltre di eseguire operazioni relative alla configurazione della topologia e alla gestione della produzione. Questa applicazione web può essere implementata su Oracle WebLogic Server, offrendo un punto di accesso centralizzato e intuitivo per monitorare e gestire i processi di integrazione dati.

3.2 ODI Generator

La creazione manuale degli artefatti ODI, come repository, datastores, mappature e scenari, attraverso l'interfaccia ODI Studio, può essere estremamente dispendiosa in termini di tempo e soggetta a errori, specialmente in presenza di progetti complessi o di grandi dimensioni. Questa esigenza di ottimizzazione ha portato allo sviluppo di soluzioni di automazione, tra cui spicca **ODI Generator**.

ODI Generator si propone come uno strumento innovativo che consente agli utenti di generare oggetti Oracle Data Integrator in modo semplice ed efficiente, utilizzando un linguaggio simile a SQL.

Tra le principali funzionalità di ODI Generator si possono citare:

- **Generazione automatica di oggetti ODI**: al posto della creazione manuale, ODI Generator permette di definire in un formato astratto o dichiarativo (principalmente tramite script simili a SQL) qualsiasi componente ODI, inclusi repository, modelli, datastores, mappature, procedure e scenari. Questo approccio automatizza la generazione del codice e degli oggetti corrispondenti, riducendo drasticamente il tempo di sviluppo.
- **Standardizzazione e consistenza**: in ambienti ETL complessi, mantenere la coerenza nella nomenclatura, nelle convenzioni di sviluppo e nell'applicazione delle best practice può essere una sfida. ODI Generator facilita l'imposizione di standard, minimizzando gli errori umani e garantendo che tutti gli oggetti generati seguano regole uniformi, migliorando così la qualità complessiva delle integrazioni.
- **Sviluppo accelerato**: per progetti di grandi dimensioni o con requisiti ripetitivi (ad esempio, la creazione di centinaia di mappature simili), la generazione automatica elimina gran parte del lavoro manuale e ripetitivo. Questo consente agli sviluppatori di focalizzarsi su logiche di business più complesse e strategiche, migliorando la produttività.

Un'ulteriore funzionalità di rilievo è la capacità di ODI Generator di leggere oggetti ODI esistenti ed esportarli in file nella loro rappresentazione SQL-like. Questa caratteristica è estremamente utile negli ambienti di sviluppo, poiché fornisce un meccanismo alternativo per il controllo di versione e permette modifiche rapide e tracciabili agli oggetti ODI.

Il software, sviluppato da Mediamente Consulting per automatizzare la generazione di mappature ODI, sfrutta la capacità di interpretare template di script in un linguaggio simile a SQL, arricchiti con informazioni provenienti da fogli Excel strutturati ad hoc. Questi fogli fungono da fonte di metadati, alimentando gli script per la creazione automatica delle mappature.

L'adozione di ODI Generator non si limita a facilitare la creazione delle mappature, ma apporta significativi miglioramenti in termini di precisione e qualità nell'integrazione dei dati. Un vantaggio chiave rispetto al processo manuale risiede nella possibilità di testare le mappature generate in un unico passaggio, permettendo una verifica rapida e completa del loro funzionamento. Questo riduce drasticamente il rischio di errori che, nel processo manuale, potrebbero emergere solo in fasi successive di test, comportando costi e tempi aggiuntivi per la correzione.

Il programma funziona sul server aziendale e, una volta completata la creazione delle mappature, queste vengono esportate in formato XML e caricate su ODI nel server del cliente. I passaggi di integrazione tramite ODI Generator si articolano nelle seguenti fasi principali:

- **Creazione e preparazione delle tabelle**: si procede con la creazione delle tabelle necessarie sia sul database del server aziendale sia sul database del server del cliente, garantendo la coerenza strutturale.

- **Configurazione della topology in ODI**: vengono configurate le impostazioni della Topology sia sul server ODI aziendale che su quello del cliente, definendo le connessioni ai database e agli agenti.

- **Preparazione dei dati di input**: vengono creati i file Excel che contengono le informazioni dettagliate sui tracciati e i metadati necessari per ogni mappatura. Questi file sono specificamente strutturati per essere letti dal codice di generazione.

- **Generazione automatica delle mappature**: ODI Generator si connette e, utilizzando i template di script SQL-like e i dati dai file Excel, genera le mappature ODI.

- **Esportazione e importazione delle mappature**: una volta generate sul server aziendale, le mappature vengono esportate in formato XML e successivamente importate su ODI nel server del cliente.

È cruciale sottolineare che, sebbene l'automazione snellisca enormemente il processo, alcune fasi preliminari rimangono essenziali per garantire un'integrazione dei dati accurata e sfruttare appieno le funzionalità del software. Tra queste, rivestono particolare importanza un'analisi precisa dei requisiti, la corretta definizione degli schemi e delle tabelle nei database coinvolti, e un'adeguata configurazione dell'ambiente ODI. Questi passaggi sono fondamentali per la buona riuscita del progetto, indipendentemente dal fatto che le mappature vengano create manualmente o generate automaticamente.

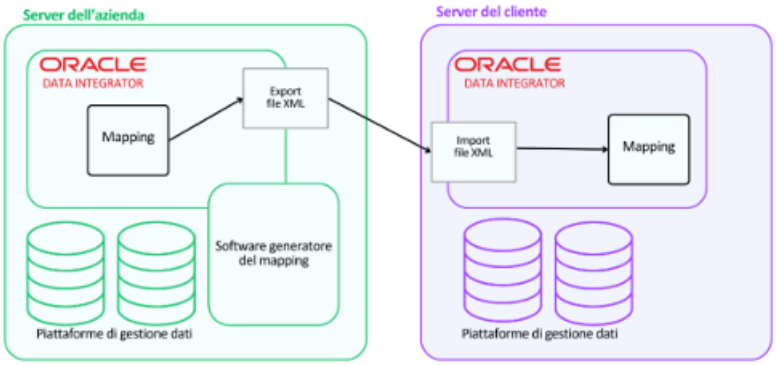

Figura 3.4. Esportazione mapping in ODI

3.3 Framework ETL

Un framework aziendale è un insieme di principi e linee guida che un'organizzazione adotta per gestire e migliorare i propri processi aziendali. Nel contesto dei processi ETL si può definire come framework aziendale un flusso di dati caratterizzato da un'insieme di operazioni che prelevano, trasformano e caricano dati da una o più fonti verso il database finale. In particolare, ogni componente del flusso processa e trasforma l'output restituito dall'operazione precedente definendo così l'input per la componente successiva; in questo modo si crea una pipeline standardizzata che offre diversi benefici, come la scalabilità, la sicurezza, la conformità e la facilità di manutenzione.

Di seguito verrà presentato il framework ETL per l'integrazione dei dati sviluppato dall'azienda Mediamente Consulting. Il flusso dei dati implementato è costituito da tre livelli sequenziali, come viene mostrato in Figura 3.5, che portano al popolamento del data warehouse finale con dati puliti e consolidati.

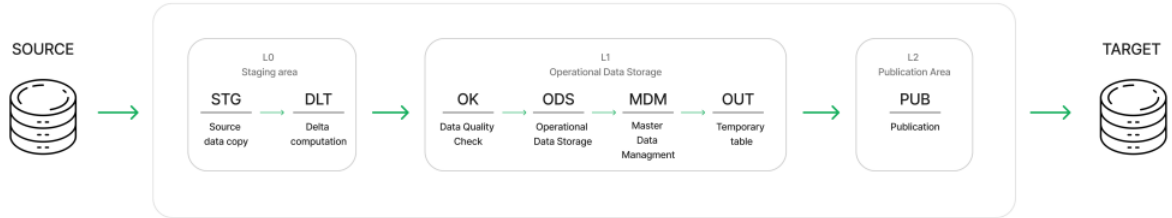

Figura 3.5. Framework aziendale

3.3.1 Livello L0

Il primo livello, chiamato Livello L0 o **Staging Area**, rappresenta la fase iniziale del processo di integrazione dei dati, in cui le informazioni grezze provenienti dalle diverse sorgenti vengono trasferite all'interno di apposite tabelle di staging. Questo livello si articola in due passaggi fondamentali: STG e DLT.

Nel passaggio **STG** viene creata una tabella contenente una copia fedele dei dati originari. A ogni record vengono associati due campi tecnici aggiuntivi, denominati JOBID e INS_TIME, entrambi utilizzati per tracciare il momento esatto del caricamento dei dati nella tabella di staging. L'obiettivo principale di questi campi è consentire una partizione temporale dei dati, facilitando così eventuali operazioni di filtraggio nei processi successivi.

Sebbene entrambi indichino la stessa informazione temporale, JOBID e INS_TIME si differenziano per il formato: il primo è una variabile numerica composta da dodici cifre, corrispondenti al formato *yyyymmddhhmmss*, mentre il secondo è registrato come valore di tipo data.

Nel secondo step, denominato **DLT**, viene calcolato il delta dei dati, ovvero vengono identificati e salvati nella tabella DLT solamente i dati nuovi o modificati rispetto al flusso di caricamento precedente. Questa operazione può essere eseguita in due modalità diverse in base alla presenza di un campo nella tabella sorgente che identifichi la data di aggiornamento o caricamento dei singoli record:

- nel caso in cui siano presenti tali informazioni, vengono caricati nella tabella DLT solamente i dati nuovi o aggiornati rispetto all'ultima lettura applicando un filtro in base al JOBID;

- in caso contrario, vengono effettuate due operazioni di *minus*: una tra i dati di 'ieri' e i dati di 'oggi' per ottenere i record cancellati in sorgente, mentre la seconda tra i dati di 'oggi' e quelli di 'ieri' per calcolare i dati aggiunti o aggiornati.

Per poter storicizzare i dati presenti nelle tabelle di DLT si possono prendere due strade diverse in base alla modalità di inserimento dei nuovi record nella tabella DTL, Se l'inserimento avviene tramite *truncate*, ovvero il contenuto della tabella viene eliminato prima del nuovo caricamento, allora viene creata un'ulteriore tabella, denominata DLT_HIS, contenente una copia dei dati della DLT; altrimenti la storicizzazione può essere effettuata direttamente nella tabella DLT con inserimento in *insert*, ovvero i nuovi record vengono aggiunti alla tabella man mano che avvengono i flussi senza cancellare o modificare i precedenti.

La storicizzazione della tabella DLT riveste un ruolo fondamentale in tutti quei casi in cui si renda necessario ricalcolare il Data Warehouse (DWH). Situazioni come la perdita di dati, il caricamento parziale delle informazioni o la presenza di discrepanze tra i dati effettivamente presenti nel DWH e quelli attesi dagli utenti possono compromettere l'affidabilità del sistema. In tali circostanze, disporre di una versione storicizzata della tabella DLT consente non solo di ricostruire correttamente i dati, ma anche di effettuare verifiche puntuali sull'esito dei flussi di caricamento, assicurandosi che l'intero processo sia stato eseguito correttamente.

A questo stadio, i record ottenuti presentano la stessa struttura e formato delle tabelle di sorgente, con l'aggiunta di due colonne contenenti la data di estrazione, espresse in formati differenti.

3.3.2 Livello L1

Nel livello L1, noto come **Operational Data Store**, vengono effettuate tutte quelle operazioni che permettono di ottenere dati consolidati, congruenti e uniformi. Inoltre la struttura delle tabelle viene modificata per preparare il dato al livello successivo, ovvero viene costruito il modello dati attraverso join tra più tabelle. Questo livello è caratterizzato da quattro step sequenziali e la creazione delle relative tabelle:

- **OK**: in questa fase vengono eseguite tutte le operazioni finalizzate al **controllo della qualità** dei dati. In un primo momento, i dati contenuti nella tabella DLT vengono filtrati in base al valore del campo JOBID, selezionando esclusivamente i record che non sono ancora stati sottoposti a verifica. Su questi dati vengono poi applicati diversi controlli, volti a garantire l'integrità e la coerenza delle informazioni rispetto alle regole del business. I controlli includono la validazione dell'integrità dei dati attraverso la rilevazione di valori nulli o non conformi, la verifica dell'unicità delle chiavi primarie per evitare duplicazioni, e il controllo referenziale delle chiavi esterne, con confronti puntuali rispetto alle tabelle di riferimento, per assicurare la consistenza dei dati tra le tabelle. I record che soddisfano i requisiti vengono inseriti nella tabella OK, mentre quelli scartati vengono salvati in tabelle ombra, denominate con il suffisso _ERR. A ognuno dei record scartati viene associato un campo che descrive il requisito non soddisfatto e in alcuni casi anche un codice di errore. Questo processo è essenziale per consentire il riciclo degli errori: le righe scartate vengono conservate per un eventuale reinserimento dei record, qualora nei flussi successivi vengano soddisfatti i requisiti non soddisfatti in precedenza.

- **ODS** (Operational Data Store): a questo punto del processo ha luogo la **storicizzazione** del dato consolidato e pulito. La tabella ODS (Operational Data Store) viene utilizzata per memorizzare l'intero storico dei dati provenienti dalla sorgente. Oltre ai campi tecnici già presenti, JOBID_INS e INS_TIME, vengono aggiunti altri due campi: JOBID_UPD e UPD_TIME, entrambi utilizzati per tracciare la data dell'ultimo aggiornamento del record, secondo i formati già descritti in precedenza. Il caricamento dei dati nella tabella ODS avviene tramite un'operazione di tipo *merge*, basata sulla chiave primaria definita a livello di database. Durante questa operazione si presta particolare attenzione a non modificare i campi chiave né quelli relativi all'inserimento iniziale (JOBID_INS e INS_TIME).

- **MDM** (Master Data Management): in questo step si svolgono tre operazioni fondamentali: unione dei dati provenienti da diverse sorgenti, **arricchimento dei dati** (data enrichment) e creazione delle **chiavi surrogate**. L'integrazione di più fonti permette di generare un'unica tabella che centralizza e unifica tutte le informazioni.

Quando uno stesso dato è presente in più sorgenti, viene stabilita una priorità tramite operazioni di left o right join. Il data enrichment consiste invece nell'aggiungere descrizioni e informazioni supplementari, ottenute attraverso join con tabelle che forniscono dettagli aggiuntivi. Una volta unificate le diverse sorgenti in un'unica tabella, denotata con MDM, si ottengono record univoci, ai quali viene assegnata una chiave surrogata (surrogate key): un identificativo numerico progressivo che facilita le interrogazioni nel modello finale.

- **OUT**: nell'ultima fase del livello L1 vengono eseguite le operazioni necessarie per preparare i dati in vista della loro pubblicazione. L'obiettivo principale è raccogliere tutte le informazioni richieste dallo strumento di data visualization adottato. A tal fine, vengono effettuate operazioni di join e union tra più tabelle, oltre alla definizione di funzioni utili a derivare informazioni aggiuntive. Un esempio può essere l'estrazione della stagione a partire dalla data di vendita di un prodotto. Inoltre, nel caso delle tabelle dei fatti, vengono calcolate le misure e associate le chiavi surrogate, ricavate dalle corrispondenti tabelle dimensionali.

Al termine del livello, vengono prodotti record puliti e consolidati, archiviati in tabelle temporanee, denominate tabelle di OUT, che possiedono già la struttura adeguata per essere utilizzate dallo strumento di visualizzazione.

3.3.3 Livello L2

L'ultimo livello del framework aziendale è il livello L2, noto anche come **Publication Area**. Questo livello prevede un unico step, denominato **PUB**, in cui i dati vengono caricati nelle tabelle finali destinate alla consultazione da parte degli utenti. Tali tabelle costituiscono la base per la generazione dei report e delle analisi tramite strumenti di data visualization. I dati, precedentemente preparati nello step OUT, vengono caricati nella tabella PUB dopo essere stati filtrati per JOBID, in modo da includere solo quelli non ancora letti. In seguito, vengono individuate le righe corrispondenti all'aggiornamento più recente, garantendo che solo le informazioni più attuali siano incluse nel processo di reporting. L'inserimento dei dati nella tabella target avviene tramite un'operazione di *merge*, utilizzando la chiave surrogata definita nel livello L1, assicurando così l'integrità e la coerenza dei dati durante l'aggiornamento delle tabelle finali.

Il presente studio si colloca all'interno di un più ampio percorso aziendale volto all'automazione progressiva dei processi ELT, con un focus specifico sulla progettazione e sull'implementazione di nuovi template di script per lo strumento ODI Generator, finalizzati alla generazione automatica dei mapping all'interno della piattaforma Oracle Data Integrator. L'obiettivo è stato estendere le capacità dello strumento su porzioni di flussi di integrazione non ancora automatizzate. Mentre le fasi iniziali (i mapping STG, DLT, OK e ODS) erano già state oggetto di sviluppo di template negli anni precedenti, questo progetto di tesi ha indirizzato l'attenzione sui blocchi successivi dei processi (MDM, OUT,

PUB). Questo approccio ha permesso di elevare il livello di automazione complessivo, apportando un'ulteriore ottimizzazione all'intero processo di integrazione dati. Il capitolo successivo descriverà nel dettaglio gli step seguiti per implementare questa automazione.

Capitolo 4

Implementazione del Progetto

Dopo aver analizzato nel capitolo precedente i principi fondamentali e le potenzialità della suite Oracle, con particolare attenzione agli strumenti ODI e ODI Generator, il presente capitolo è dedicato all'implementazione pratica del progetto. Verranno illustrati nel dettaglio gli aspetti metodologici e tecnici che hanno guidato lo sviluppo dei nuovi template di script e la loro integrazione all'interno del processo di generazione automatica delle mappature. Saranno inoltre descritte le principali sfide affrontate, gli strumenti utilizzati e le soluzioni adottate per estendere l'automazione anche alle fasi successive del flusso di integrazione, con l'obiettivo di migliorarne l'efficienza e garantire un maggiore livello di standardizzazione.

Il progetto descritto in questa tesi si inserisce in un percorso più ampio avviato dall'azienda negli anni precedenti, volto all'automatizzazione dei processi di integrazione dati mediante l'utilizzo della tecnologia Oracle Data Integrator. In particolare, erano già stati sviluppati e implementati script per le fasi iniziali del flusso ETL, fino al livello ODS, in cui i dati vengono copiati dalle fonti originarie senza modificarne la struttura, ma limitandosi a verificarne la qualità e la correttezza. Il presente lavoro si concentra invece sullo sviluppo dei template necessari per generare in modo automatico i mapping delle fasi successive, a partire dal livello MDM. In questa fase, l'integrazione dei dati provenienti da diverse sorgenti porta alla creazione di strutture informative più complesse, che rappresentano l'effettivo modello del data warehouse e rispondono alle esigenze analitiche dei clienti. Questo passaggio segna un'evoluzione significativa nella complessità dei flussi, richiedendo soluzioni tecniche più articolate e una maggiore attenzione alla standardizzazione delle trasformazioni.

All'inizio del capitolo vengono presentati gli strumenti utilizzati, accompagnati da una breve descrizione volta a chiarirne il ruolo all'interno del progetto. Successivamente, l'attenzione si concentra sugli step seguiti per la creazione degli script, a partire da una fase iniziale di sviluppo manuale dei mapping. Tale attività ha consentito di comprendere nel dettaglio la logica dei flussi e di individuare le componenti chiave, rendendo possibile la successiva progettazione e implementazione dei template.

4.1 Strumenti Usati

Per lo sviluppo del presente progetto sono stati impiegati diversi strumenti e tecnologie, ciascuno con un ruolo specifico all'interno del processo di integrazione e automazione dei flussi ETL. Di seguito viene fornita una panoramica dei principali strumenti utilizzati, accompagnata da una descrizione delle rispettive funzionalità e del loro impiego nel contesto progettuale.

DBeaver
DBeaver è uno strumento open-source per la gestione di database relazionali e non relazionali, compatibile con numerosi sistemi (Oracle, PostgreSQL, MySQL, ecc.). Nel contesto di questo progetto, è stato utilizzato principalmente per la consultazione e gestione delle tabelle sorgente e target presenti nel database. Il suo utilizzo ha facilitato le attività di verifica, esplorazione dei dati e supporto durante le fasi di sviluppo e test.

Oracle Data Integrator (ODI)
ODI è una piattaforma per l'integrazione dei dati basata su architettura ELT (Extract, Load, Transform). Progettato per orchestrare e gestire processi complessi di trasformazione, ODI consente di definire flussi di integrazione attraverso una combinazione di modelli, mapping e procedure. All'interno del progetto, ODI ha rappresentato lo strumento principale per la creazione dei mapping e per la definizione delle logiche di integrazione tra le varie componenti del sistema informativo.

ODI Generator
ODI Generator è uno strumento progettato per automatizzare la creazione di oggetti all'interno di Oracle Data Integrator, come repository, mapping e package. Si tratta di uno degli strumenti chiave del progetto, in quanto ha reso possibile l'automazione della generazione dei mapping a partire da dati strutturati. Il suo utilizzo ha contribuito in modo determinante alla riduzione dei tempi di sviluppo, oltre a garantire una maggiore standardizzazione e a ridurre il margine di errore tipico dei processi manuali.

Fogli Excel
I fogli Excel hanno rappresentato lo strumento scelto per fornire in input i metadati necessari alla generazione automatica dei mapping. In particolare, essi contengono informazioni come nomi delle tabelle, colonne da elaborare, trasformazioni da applicare e altre specifiche tecniche. Questi dati vengono letti e interpretati da ODI Generator per alimentare i template e produrre i corrispondenti oggetti ODI.

Apache Velocity
Apache Velocity è un linguaggio di template appartenente al progetto Apache Velocity Engine, concepito per la generazione dinamica di contenuti testuali (come XML, SQL o HTML) a partire da modelli predefiniti. All'interno del progetto, Velocity è stato impiegato per la scrittura dei template utilizzati da ODI Generator, rendendo possibile la creazione automatizzata e parametrica del codice ODI a partire da input esterni forniti nei fogli Excel. La flessibilità del linguaggio ha permesso di definire strutture generiche

riutilizzabili, adattabili ai diversi flussi da generare.

4.2 Sviluppo del Progetto

Lo sviluppo del progetto di questa tesi si è articolato in diverse fasi ben distinte, ognuna delle quali ha rappresentato un passaggio cruciale nel percorso verso il raggiungimento dell'obiettivo prefissato: automatizzare in maniera efficace, efficiente e affidabile l'intero processo di creazione dei mapping e dei flussi di integrazione dei dati. Questo approccio graduale ha permesso di affrontare con metodo le varie complessità del sistema, garantendo al contempo una solida base per la costruzione di una soluzione sostenibile e scalabile nel tempo.

La prima fase ha previsto la realizzazione manuale dei mapping utilizzando Oracle Data Integrator (ODI). Questo passaggio iniziale è stato fondamentale per acquisire una comprensione approfondita delle logiche sottostanti ai flussi di integrazione, nonché per familiarizzare con le funzionalità messe a disposizione dallo strumento. L'implementazione manuale rappresenta infatti il metodo tradizionale per progettare flussi ETL e costituisce la base teorica e pratica su cui si fonda l'intero processo di automazione.

Durante questa fase sono state eseguite diverse attività, in particolare:

- La definizione e configurazione delle tabelle sorgente e target, necessarie per strutturare il flusso di dati;
- La creazione delle regole di trasformazione, volte a manipolare i dati in base alle esigenze del sistema di destinazione;
- L'inserimento delle logiche aziendali, ossia condizioni e calcoli che rispondono ai requisiti specifici del cliente e garantiscono l'allineamento dei dati con i criteri di qualità richiesti.

Questa attività ha permesso di comprendere in modo puntuale come ODI gestisca il flusso di dati, quali siano i passaggi tecnici necessari per configurare un mapping efficace e quali siano gli elementi ricorrenti tra i vari flussi. Proprio l'individuazione di queste componenti ripetitive ha rappresentato il punto di partenza per ipotizzare un modello di automazione, in grado di replicare tali operazioni in maniera sistematica e con un margine di errore minimo.

La seconda fase ha riguardato la progettazione e la realizzazione dei template di automazione, fondamentali per la generazione automatica dei mapping. A differenza della configurazione manuale, in questa fase l'obiettivo è stato quello di astrarre e generalizzare la logica implementata nei mapping, rendendola replicabile attraverso codice parametrico.

I template sono stati sviluppati utilizzando il linguaggio Apache Velocity, scelto per la sua semplicità sintattica e per la capacità di generare file testuali dinamici a partire da dati esterni. In particolare, tali template vengono alimentati tramite fogli Excel, che rappresentano la struttura informativa del singolo mapping: essi contengono, ad esempio,

i nomi delle tabelle, le colonne da utilizzare, le chiavi di join, e le eventuali trasformazioni da applicare.

Il processo di generazione automatica avviene tramite uno strumento, denominato ODI Generator, il quale si occupa di:

- Leggere e interpretare i dati contenuti nei fogli Excel;
- Combinarli con le istruzioni generiche contenute nei template sviluppati con Apache Velocity;
- Generare automaticamente gli oggetti ODI, quali mapping, package o procedure, che vengono poi esportati in formato XML e successivamente importati nell'ambiente ODI del cliente, dove possono essere eseguiti.

Questo approccio ha portato numerosi vantaggi dal punto di vista operativo. In primo luogo, ha permesso una significativa riduzione dei tempi di configurazione, risultando particolarmente efficace soprattutto quando si tratta di gestire un elevato numero di flussi simili tra loro. Inoltre, ha favorito una maggiore standardizzazione dei processi, grazie all'applicazione coerente di regole comuni. Infine, ha contribuito a ridurre in modo significativo il rischio di errore umano, poiché la maggior parte delle operazioni viene affidata a un processo automatizzato e controllato.

4.2.1 Implementazione Manuale dei Mapping

La prima fase del progetto si concentra sulla creazione e integrazione manuale di un flusso di dati. Questa attività rappresenta un passaggio fondamentale per comprendere in maniera concreta e approfondita il funzionamento dei flussi di integrazione all'interno di un ambiente ELT. L'obiettivo non è solo quello di realizzare un flusso funzionante, ma anche di acquisire consapevolezza delle logiche operative e delle trasformazioni necessarie per rendere i dati pronti all'uso in sistemi informativi complessi.

Come già introdotto nei capitoli precedenti, nel contesto del progetto, viene adottato Oracle Data Integrator (ODI), uno strumento che si basa su un'architettura ELT (Extract, Load, Transform). In questo modello, i dati vengono inizialmente estratti e caricati nel sistema target, dove vengono poi trasformati secondo regole specifiche. Ciò consente di sfruttare appieno la capacità elaborativa del database, migliorando le performance complessive del processo.

Uno degli elementi centrali in ODI è il **mapping**, un oggetto che consente di definire il trasferimento dei dati da un datastore sorgente a uno di destinazione. Tuttavia, il mapping non si limita a spostare i dati: attraverso di esso è possibile applicare regole di trasformazione dichiarative, che permettono di modellare i dati in modo conforme alle esigenze del sistema di destinazione e alle logiche di business dell'azienda.

Le trasformazioni applicabili tramite mapping possono includere operazioni anche piuttosto complesse, come:

- Join tra tabelle differenti, necessari per aggregare dati distribuiti su più entità;
- Filtri, che consentono di selezionare soltanto i record rilevanti in base a criteri predefiniti;

- Vincoli di integrità e di coerenza, per assicurare che i dati integrati siano affidabili, completi e consistenti rispetto alle regole del sistema.

ODI consente di progettare questi mapping tramite un'interfaccia grafica intuitiva, in cui le trasformazioni vengono rappresentate visivamente. Questo approccio facilita sia la fase di sviluppo che quella di manutenzione, in quanto rende il flusso dei dati immediatamente leggibile.

L'utilizzo dei mapping apporta numerosi vantaggi, tra cui la possibilità di applicare automaticamente regole di trasformazione complesse a grandi volumi di dati, garantendo ripetibilità, coerenza e scalabilità. Anche se la creazione manuale richiede più tempo rispetto ad approcci automatizzati, essa risulta essenziale in una fase iniziale come questa, poiché permette di acquisire piena familiarità con le logiche implementative e con le criticità operative che possono emergere nei diversi scenari.

In particolare, l'attività si è focalizzata sulla progettazione e implementazione dei mapping successivi a ODS, ovvero quelli che, a partire da dati già validati e certificati nelle fasi precedenti, si occupano di aggregare, trasformare e strutturare le informazioni in modo funzionale alla costruzione del modello analitico finale. A differenza delle fasi iniziali del flusso, in cui i dati vengono semplicemente trasferiti e sottoposti a controlli di qualità, a partire dal livello MDM (Master Data Management) e oltre, le trasformazioni assumono una maggiore complessità: i dati provenienti da diverse tabelle vengono integrati, arricchiti e modellati secondo logiche di business specifiche, preparando il contenuto informativo necessario per il popolamento del data warehouse.

Sebbene il progetto si focalizzi su questa parte del processo, sono stati implementati anche i mapping relativi agli step precedenti, fino al livello ODS, al fine di disporre di dati corretti, normalizzati e pronti per l'elaborazione nei livelli successivi. Tuttavia, tali fasi non verranno descritte nel dettaglio, in quanto non rappresentano il fulcro del presente lavoro e, inoltre, l'automazione di questi primi passaggi era già stata sviluppata in anni precedenti all'interno dell'azienda.

Per supportare l'implementazione e il testing sia dei mapping manuali su ODI che di quelli generati tramite script automatici, sono state create due tabelle all'interno del database aziendale. Tali tabelle hanno fornito una base dati controllata e coerente, utile per verificare il corretto funzionamento dei flussi di integrazione sviluppati. Nel prosieguo della trattazione, verranno descritti nel dettaglio gli step realizzati per una di esse, ovvero la tabella denominata *NEGOZI*, scelta come caso rappresentativo per illustrare le logiche implementative e le soluzioni adottate nel corso del progetto. La tabella NEGOZI, nella versione presente nel sistema sorgente, è costituita da nove colonne che descrivono le principali caratteristiche dei punti vendita. Tra queste, è presente una colonna designata come chiave primaria, necessaria per identificare univocamente ciascun record e per mantenere la coerenza e l'affidabilità dei processi di integrazione che ne derivano.

Il primo mapping oggetto di analisi è quello relativo al livello **MDM (Master Data Management)**, costruito a partire dalla tabella NEGOZI, generata al termine del mapping ODS, con il nome ODS_NEGOZI. Questo rappresenta il primo passaggio in cui i dati

non si limitano più alla semplice validazione, ma iniziano a essere trasformati e aggregati secondo logiche specifiche.

Come accennato in precedenza, il mapping MDM prevede tre operazioni principali:

- Unione dei dati provenienti da più sorgenti: nel caso in cui i dati vengano caricati da fonti diverse, essi vengono integrati all'interno di un'unica tabella. Questo processo consente di centralizzare le informazioni, consolidandole in una vista unificata e coerente. Qualora lo stesso dato sia presente in più sorgenti, viene applicata una logica di priorità, definita in accordo con il business, tramite join di tipo LEFT o RIGHT, al fine di selezionare il valore ritenuto più rilevante.
- Arricchimento del dato (data enrichment): consiste nell'integrazione di informazioni aggiuntive, ottenute tramite join con tabelle esterne contenenti descrizioni o attributi supplementari. Questo passaggio ha lo scopo di migliorare la qualità e la completezza del dato.
- Creazione delle chiavi surrogate: una volta ottenuta la tabella unificata, denominata MDM, viene generata una chiave surrogata per ogni record. Tale chiave, rappresentata da un identificativo numerico progressivo, facilita le interrogazioni e migliora l'efficienza delle operazioni nel modello dati finale.

Concentrandosi ora sul caso specifico della tabella ODS_NEGOZI, essa rappresenta l'output del framework aziendale descritto nel capitolo precedente, fino al completamento del flusso ODS, e costituisce la sorgente principale del mapping MDM relativo ai negozi. Rispetto alla struttura originaria dei dati, questa tabella contiene cinque colonne aggiuntive, introdotte dal framework stesso, che svolgono un ruolo fondamentale nell'implementazione del flusso ETL:

- jobid_l1_ins: identifica l'esecuzione del job che ha effettuato l'inserimento del record;
- jobid_l1_upd: identifica il job che ha eseguito l'ultimo aggiornamento del record;
- ins_time: indica il timestamp di inserimento del record;
- upd_time: indica il timestamp dell'ultima modifica;
- flg_neg: flag che specifica lo stato del record.

Come visibile nella Figura 4.1, nel caso specifico della tabella ODS_NEGOZI non erano presenti più sorgenti da integrare. Di conseguenza, non è stato necessario implementare join tra tabelle differenti. Tuttavia, per garantire che l'elaborazione riguardasse esclusivamente i record nuovi o aggiornati, è stato applicato un filtro basato sul campo jobid_l1_upd, che consente di identificare le righe modificate o inserite più recentemente.

Una volta selezionati i soli record 'nuovi', viene generata una *surrogate key* per ciascuna di queste righe. In caso di aggiornamenti su record già esistenti, la chiave surrogata precedentemente assegnata non viene modificata; il sistema effettua invece un'operazione di *merge* basata sulla chiave primaria definita nel database, preservando l'univocità e la

coerenza dei dati nel modello finale. Questa chiave univoca rappresenta un elemento essenziale per garantire l'efficienza nelle interrogazioni del modello dati finale, in particolare nel contesto del data warehouse, dove l'uso di surrogate keys migliora le performance e la tracciabilità dei dati. I dati così elaborati vengono infine inseriti, tramite merge, all'interno della tabella MDM_NEGOZI, la quale rappresenta la sorgente per lo step successivo del processo di integrazione.

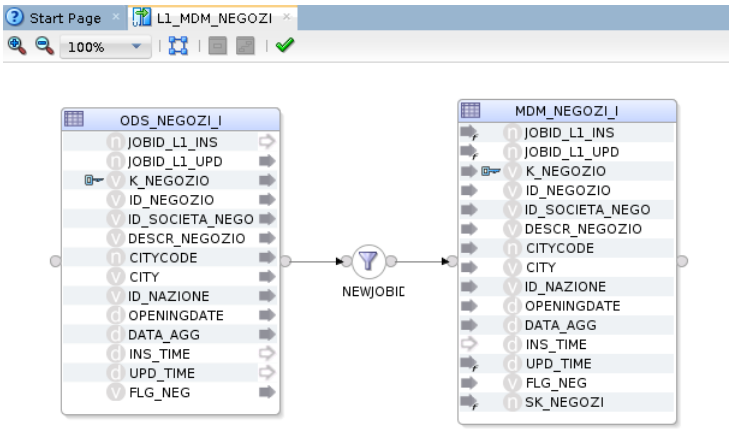

Figura 4.1. MDM della tabella NEGOZI

Come già introdotto nei capitoli precedenti, l'ultimo step del livello L1, noto come **mapping OUT**, ha l'obiettivo di preparare i dati per il caricamento nella tabella di pubblicazione. In questa fase vengono reperite tutte le informazioni richieste dallo strumento di data visualization utilizzato. A tal fine, si eseguono operazioni di join e union tra più tabelle, e si definiscono funzioni utili a derivare informazioni aggiuntive dai dati disponibili. Un esempio tipico è la classificazione di un cliente in una fascia d'età (es. 18–25, 26–35, ecc.) a partire dalla sua data di nascita. Inoltre, nel caso delle tabelle dei fatti, vengono calcolate le misure quantitative e recuperate le chiavi surrogate a partire dalle tabelle dimensionali.

Al termine di questo livello, il risultato consiste in record puliti, coerenti e già strutturati, i quali vengono archiviati in tabelle temporanee denominate tabelle OUT. Queste tabelle non contengono lo storico necessario per l'analisi tramite gli strumenti di visualizzazione, ma rappresentano un passaggio intermedio essenziale per la preparazione dei dati.

Nello specifico, il mapping di OUT utilizza come tabella sorgente MDM_NEGOZI, che raccoglie i dati provenienti dal livello di Master Data Management, e come tabella di destinazione OUT_NEGOZI, una tabella temporanea che possiede già la struttura richiesta per il successivo caricamento nella tabella di pubblicazione.

Nel caso del mapping in questione (Figura 4.2), è stata implementata un'operazione di join tra le tabelle MDM_NEGOZI e MDM_NAZIONI. Quest'ultima rappresenta l'output del framework aziendale applicato alla tabella NAZIONI fino al livello MDM. L'obiettivo

principale di questa operazione è il recupero della chiave surrogata associata alla tabella NAZIONI, necessaria per migliorare l'efficienza nelle successive fasi di analisi. Oltre a tale chiave, in base alle specifiche esigenze analitiche, è possibile estrapolare ulteriori informazioni da includere nella tabella OUT_NEGOZI, come ad esempio la nazione in cui è situato il negozio o il continente di appartenenza. Analogamente a quanto avviene nel mapping MDM, anche in questo caso viene applicato un filtro che consente di selezionare soltanto i record aggiornati o inseriti di recente. I dati così filtrati vengono infine inseriti nella tabella OUT_NEGOZI, che pur non contenendo lo storico, rappresenta un dataset pulito e consolidato, già predisposto per essere caricato nella tabella di pubblicazione finale.

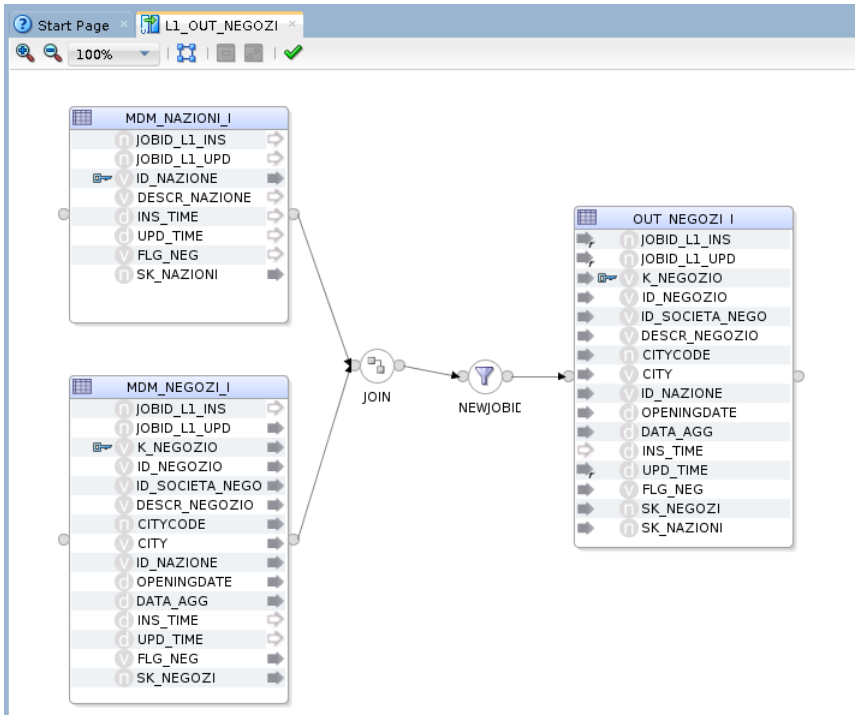

Figura 4.2. OUT della tabella NEGOZI

L'ultimo livello del framework aziendale è rappresentato dal livello L2, denominato Publication Area. Questo livello è composto da un singolo step operativo, identificato come **mapping PUB**, il cui scopo principale è quello di trasferire i dati, opportunamente preparati nello step precedente, verso le tabelle finali. Tali tabelle costituiscono il punto di accesso per la consultazione da parte degli utenti finali e rappresentano la base per la generazione dei report aziendali.

Durante questa fase conclusiva, il processo prevede innanzitutto un filtraggio dei dati basato sul valore del campo JOBID, in modo da selezionare esclusivamente i record che

non sono ancora stati elaborati. Successivamente, per ciascun insieme di chiavi identificative, viene individuato il record più aggiornato. Questo consente di garantire che le informazioni pubblicate siano sempre coerenti con l'ultima versione disponibile, assicurando così l'affidabilità e l'attualità dei dati consultabili da parte degli utenti.

Nel caso specifico della tabella oggetto di analisi, il mapping PUB ha il compito di trasferire i dati dalla tabella sorgente OUT_NEGOZI alla corrispondente tabella di destinazione PUB_NEGOZI (Figura 4.3). Questo trasferimento rappresenta una fase di consolidamento dei dati, che avviene solo dopo che sono stati superati con successo tutti i controlli di qualità e di integrità referenziale previsti nei passaggi precedenti.

Oltre al consueto filtro basato sul campo jobid_l1_upd, viene applicata una logica di selezione avanzata che sfrutta la funzione ROW_NUMBER(), calcolata in funzione delle chiavi primarie e dei campi di aggiornamento. Tale meccanismo consente di identificare in modo univoco, per ciascuna entità, il record più aggiornato, evitando così sia la duplicazione delle righe sia la presenza di record non più validi.

Infine, l'inserimento dei dati nella tabella PUB_NEGOZI avviene attraverso un'operazione di *merge*, basata sulla chiave surrogata definita nel mapping MDM. Questo approccio garantisce la coerenza, l'integrità e il corretto allineamento delle informazioni all'interno delle tabelle finali, che costituiscono la base per le attività di reportistica e consultazione da parte degli utenti.

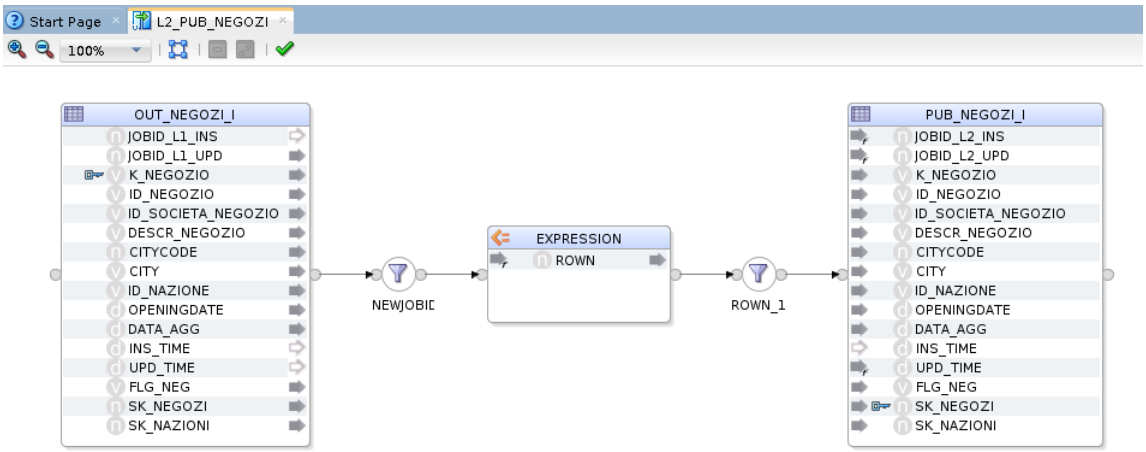

Figura 4.3. PUB della tabella NEGOZI

Prima dell'effettiva esecuzione dei processi di integrazione, tutte le variabili tecniche necessarie sono state opportunamente definite e inizializzate, al fine di garantire uno sviluppo continuo e privo di interruzioni. Ogni mapping sviluppato è stato poi sottoposto a una fase di test accurata all'interno di Oracle Data Integrator (ODI), con l'obiettivo di verificarne il corretto funzionamento, sia in termini di trasferimento dei dati sia per quanto riguarda le operazioni di trasformazione.

Nel corso dello sviluppo del sistema è stata introdotta una funzionalità avanzata che consente di gestire l'esecuzione dei mapping in modo sequenziale, sfruttando le variabili tecniche precedentemente definite. Tale approccio ha contribuito a ottimizzare l'intero processo di integrazione, con evidenti benefici in termini di riduzione del carico computazionale, diminuzione dei tempi di elaborazione e incremento complessivo dell'efficienza.

L'automazione dell'integrazione dei dati si realizza attraverso la definizione di una sequenza ordinata di operazioni, come mapping, procedure e altri passaggi, raccolte all'interno di un package. Questo componente rappresenta l'unità logica principale del workflow di integrazione, ed è progettato per generare uno scenario di produzione contenente il codice eseguibile necessario per ogni fase del processo [13].

Nel contesto di questa tesi, il termine package fa riferimento a un insieme di step organizzati secondo un diagramma di esecuzione, che definisce in modo esplicito l'ordine in cui devono essere svolte le attività previste. Oltre a gestire il flusso dei mapping tra tabelle sorgente e tabelle di destinazione, un package può includere azioni aggiuntive, come l'avvio di una procedura di reverse-engineering su un datastore, l'invio di notifiche via email, lo scaricamento e la decompressione di file, o l'implementazione di cicli di esecuzione con parametri variabili.

Generalmente, per ogni mapping viene creato un package specifico, che può successivamente essere incluso in una struttura gerarchica più articolata. In tale architettura, un package principale ha il compito di inizializzare le variabili tecniche e coordinare l'esecuzione ordinata dei package secondari. Questo approccio modulare non solo garantisce una maggiore flessibilità nella gestione del workflow di integrazione, ma rende anche più agevole il controllo, il monitoraggio e la manutenzione del processo, dall'acquisizione dei dati alla loro trasformazione e caricamento nel data warehouse.

La suddivisione dei package in sezioni distinte risulta particolarmente utile anche nella fase di test, in quanto permette di individuare con maggiore precisione eventuali anomalie o malfunzionamenti. A supporto di questa attività, lo strumento Operator Navigator, messo a disposizione da ODI, consente un monitoraggio dettagliato di ogni step di integrazione, offrendo una visione chiara dell'avanzamento e dell'esito di ciascun package eseguito.

Un ulteriore vantaggio derivante dall'impiego dei package è la possibilità di schedularne l'esecuzione automatica tramite la definizione di job. Questi ultimi possono essere configurati per avviarsi secondo una cadenza temporale prestabilita (giornaliera, settimanale, mensile) oppure in risposta a eventi specifici (trigger), rendendo il sistema di integrazione dei dati altamente automatizzato, adattabile e perfettamente allineato alle esigenze operative dell'ambiente aziendale.

Con l'aumentare del numero di mapping e procedure da gestire all'interno di un progetto di integrazione dati, la complessità del sistema tende a crescere in maniera significativa. Ogni procedura comporta infatti la definizione di uno o più mapping personalizzati, e la gestione simultanea di numerosi componenti può rapidamente rendere il processo più articolato e suscettibile a criticità. Questa crescente complessità non solo rallenta le attività di sviluppo, ma aumenta anche il rischio di errore sia nella fase di progettazione che durante l'implementazione operativa.

La gestione manuale di flussi di lavoro complessi, inoltre, può compromettere la coerenza dei dati, incidere negativamente sulle prestazioni del sistema e mettere a rischio l'affidabilità complessiva del progetto. Sebbene l'ambiente di test rappresenti uno strumento essenziale per l'identificazione di anomalie, non è raro che alcuni errori emergano solo in fasi avanzate, con possibili impatti rilevanti sulla qualità dell'integrazione e sulla consistenza delle informazioni.

Alla luce di tali considerazioni, risulta fondamentale disporre di un processo di sviluppo quanto più possibile standardizzato, efficiente e affidabile. In questo contesto, l'introduzione di soluzioni automatizzate per la generazione dei mapping si configura come un passaggio strategico. Automatizzare queste attività consente infatti di migliorare in modo significativo l'efficienza operativa, riducendo al minimo il rischio di errore umano e garantendo maggiore uniformità nelle procedure.

Oltre a garantire una migliore qualità e coerenza dei dati integrati, l'automazione contribuisce a ridurre sensibilmente i tempi di sviluppo, rendendo l'intero progetto più agile e reattivo rispetto alle esigenze in continua evoluzione dell'ambiente aziendale. In sintesi, la generazione automatica dei mapping si propone come un elemento chiave per il successo dei progetti di integrazione dati complessi, in grado di coniugare efficienza, affidabilità e riduzione dei rischi operativi.

4.2.2 Progettazione e Realizzazione dei Template

L'azienda ha da tempo manifestato un forte interesse verso l'automatizzazione dei processi di integrazione dei dati, adottando a tal fine la tecnologia Oracle Data Integrator (ODI) come strumento principale per la gestione dei flussi ETL. In particolare, erano già stati sviluppati e messi in produzione diversi script automatizzati relativi alle fasi iniziali del processo di integrazione, fino al livello ODS (Operational Data Store). In questa fase, i dati vengono estratti dalle fonti originarie e caricati senza subire modifiche strutturali, concentrandosi esclusivamente su operazioni di controllo della qualità e della correttezza dei dati stessi.

Tuttavia, le fasi successive del flusso di integrazione presentano un livello di complessità maggiore, soprattutto per quanto riguarda la realizzazione dei mapping. In questa parte del processo, infatti, diventa necessario trasformare la struttura delle tabelle sorgente per adattarle sia alle specifiche esigenze del cliente sia ai requisiti dello strumento di visualizzazione dei dati. Tale necessità rende difficile definire un modello standard applicabile in modo uniforme a tutti i casi, in quanto ogni mapping tende ad assumere caratteristiche personalizzate, configurandosi spesso come una soluzione costruita su misura.

A seguito dell'implementazione manuale dei mapping, è stato possibile individuare una serie di elementi ricorrenti all'interno dei progetti, che hanno evidenziato la possibilità di introdurre meccanismi di automazione anche nelle fasi più avanzate. A partire da queste osservazioni, è stata avviata un'attività di analisi sui template già sviluppati internamente, con l'obiettivo di verificare l'esistenza di uno standard aziendale condiviso e potenzialmente riutilizzabile. Tale iniziativa si inserisce in un più ampio percorso volto a uniformare le pratiche di sviluppo e garantire coerenza con le strategie di automatizzazione già adottate in precedenza dall'organizzazione.

Nel corso dell'attività di analisi, si è proceduto anche alla generazione automatica dei mapping relativi alle fasi precedenti il flusso MDM (Master Data Management), sfruttando i template di codice già esistenti. Questo passaggio ha rappresentato un punto di partenza fondamentale per l'adozione di soluzioni più efficienti e scalabili, contribuendo a strutturare un approccio sistematico al processo di integrazione.

Una volta implementati i template di codice e compilati correttamente i fogli Excel con le informazioni richieste, la generazione automatica dei mapping in Oracle Data Integrator (ODI) può essere eseguita seguendo una serie di passaggi ben definiti.

La sequenza operativa riportata di seguito si riferisce al caso in cui si crea un nuovo progetto, con tutte le connessioni da configurare ex novo. Nel caso invece in cui tutte le connessioni siano già presenti e configurate, molti dei primi step possono essere saltati, iniziando direttamente dal punto 7 in poi.

Di seguito la procedura dettagliata:

1. Avviare l'ambiente ODI Studio sulla macchina virtuale in cui è installato il software ODI Generator, e connettersi al repository del progetto.

2. Esportare da ODI Studio, all'interno della connessione relativa al progetto di tesi, il progetto vuoto e il modello precedentemente creato, salvandoli localmente.

3. Copiare il folder esportato sul desktop della macchina virtuale che ospita il software.

4. Importare nuovamente il progetto vuoto e il modello all'interno di ODI Studio, direttamente dalla macchina virtuale.

5. Configurare l'architettura logica (Logical Architecture) in ODI Studio, assegnando lo stesso nome utilizzato nel progetto originale.

6. Stabilire la connessione al repository ODI Studio all'interno dello strumento ODI Generator.

7. Inserire i file Excel compilati in una cartella dedicata sul desktop della macchina virtuale.

8. Aprire ODI Generator e connettersi sia al repository che all'ambiente ODI Studio (Figura 4.4).

9. Caricare il file Excel cliccando sull'icona corrispondente e selezionandolo dalla cartella desktop (Figura 4.5).

10. Selezionare il template di codice di riferimento e il foglio Excel corrispondente da utilizzare per la generazione del mapping (Figura 4.6).

11. Eseguire il codice generato selezionandolo completamente e cliccando sull'icona 'Run'.

    - Se il codice è corretto, il mapping verrà generato automaticamente all'interno di ODI Studio della macchina virtuale.

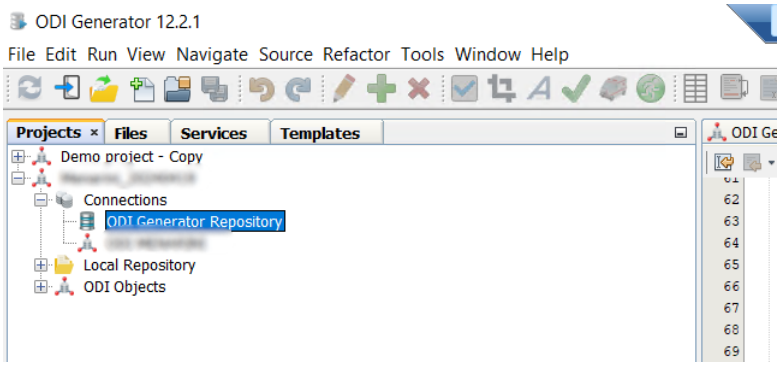

Figura 4.4. Interfaccia di ODI Generator

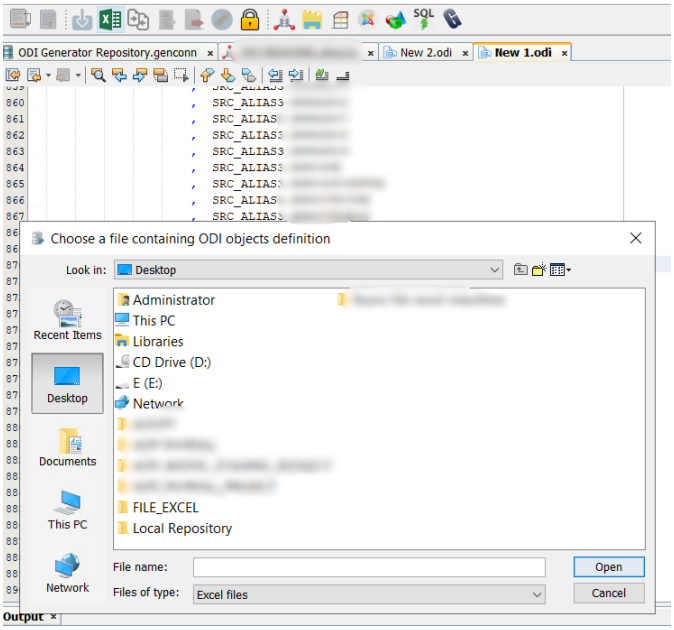

Figura 4.5. Creazione tramite fogli Excel su ODI Generator

- In caso di errore, verrà restituito un messaggio nella parte inferiore dello schermo; sarà quindi possibile correggere direttamente il codice oppure modificare il file Excel e ripetere il processo.

12. Una volta generati, i mapping possono essere esportati da ODI Studio in un nuovo folder all'interno della macchina virtuale.

13. Infine, è possibile importare i mapping nel progetto ODI effettivo, eseguendo l'operazione inversa di quanto fatto in precedenza. A questo punto, i mapping generati tramite gli script possono essere eseguiti su ODI Studio e integrati all'interno delle

procedure, permettendo così la loro schedulazione e l'automatizzazione dei processi di integrazione.

Questa procedura consente di automatizzare in modo efficiente la creazione dei mapping, riducendo sensibilmente il tempo necessario per le fasi più ripetitive e standardizzabili del processo, pur mantenendo il controllo e la possibilità di intervento manuale nei casi in cui si rendano necessarie modifiche o correzioni.

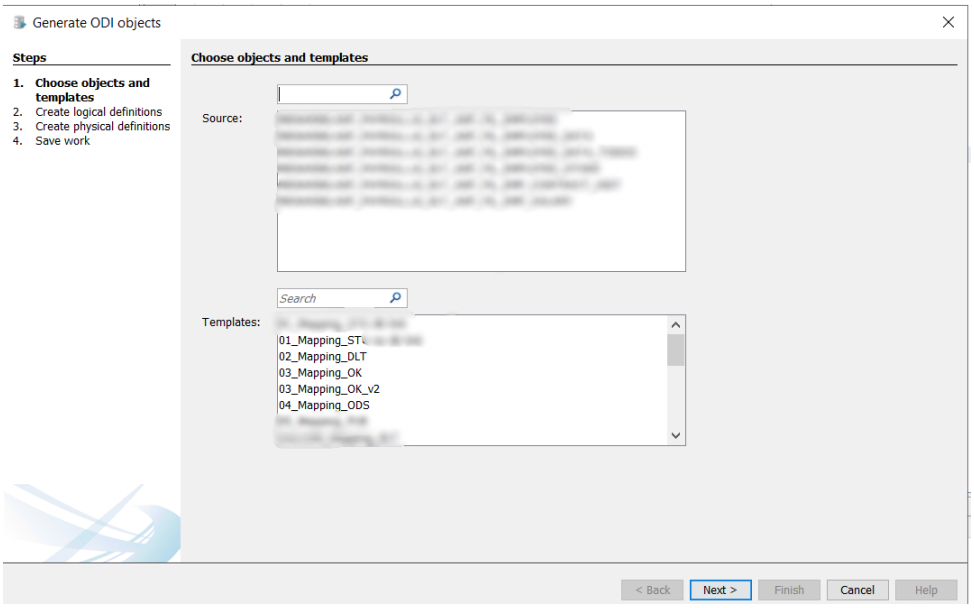

Figura 4.6. Genezione di oggetti su ODI Generator

L'analisi approfondita dei mapping implementati manualmente ha reso possibile l'individuazione di una serie di elementi ricorrenti e strutture comuni. A partire da tali osservazioni, sono stati sviluppati script mirati alla replicazione automatica di questi pattern, attraverso l'impiego di template specifici. Nel prosieguo del capitolo verranno illustrati nel dettaglio i principali elementi ripetitivi identificati e verrà descritto il processo attraverso cui tali elementi sono stati trasformati in modelli riutilizzabili, permettendo un'elevata automazione nella creazione dei mapping all'interno del progetto.

Come già anticipato nei capitoli precedenti, i template sviluppati sono stati implementati utilizzando il linguaggio Apache Velocity. L'adozione di questo linguaggio ha reso possibile il mantenimento della struttura delle query SQL, consentendo al contempo l'introduzione di elementi parametrici funzionali all'automazione del processo di generazione.

Apache Velocity è un linguaggio di template basato su Java, progettato per la generazione dinamica di contenuti testuali, tra cui codice sorgente, file XML, HTML e query SQL. La sua principale utilità risiede nella possibilità di automatizzare la produzione di script o configurazioni a partire da modelli predefiniti, che vengono compilati con dati

forniti in input. Questo lo rende particolarmente adatto a contesti in cui è richiesto un elevato grado di flessibilità, standardizzazione e riduzione delle attività manuali ripetitive.

Una delle funzionalità fondamentali offerte da Velocity è l'utilizzo dei **campi parametrici**, ovvero variabili definite all'interno del template che fungono da segnaposto per valori dinamici. Tali variabili, identificate dal simbolo $ (ad esempio $nomeTabella), vengono valorizzate al momento dell'esecuzione tramite un meccanismo di sostituzione automatica. I valori sono generalmente forniti da file esterni (come fogli Excel, JSON o XML) o da strutture dati elaborate da sistemi integrati. Questo meccanismo consente la generazione dinamica del codice, facilitando la riutilizzabilità dei template e riducendo la possibilità di errore dovuta all'inserimento manuale.

Nel caso specifico della presente tesi, Apache Velocity è stato utilizzato per la realizzazione di template dedicati alla generazione di script ETL. Tali template vengono compilati automaticamente a partire da fogli Excel strutturati, contenenti le informazioni necessarie alla definizione dei mapping, come i nomi delle tabelle sorgente e target, i campi di join e altre configurazioni tecniche. L'integrazione dei campi parametrici ha permesso di automatizzare in modo efficiente il processo, garantendo al contempo flessibilità e uniformità nella generazione dei flussi di integrazione dati.

Sulla base dell'analisi dei template sviluppati negli anni precedenti, è stata mantenuta la medesima struttura di base del codice. Questa si articola in due sezioni principali: la prima riguarda la definizione dei moduli specifici di ODI Studio, noti come Knowledge Modules (KM), componenti fondamentali che definiscono le modalità operative attraverso cui vengono eseguite le diverse fasi di un processo ETL. I Knowledge Modules sono suddivisi in categorie in base al tipo di operazione che gestiscono e offrono un'elevata flessibilità grazie alla possibilità di personalizzare il codice generato. In particolare, in questo contesto vengono impiegati:

- LKM (Loading Knowledge Module): responsabili della fase di caricamento iniziale dei dati dalla sorgente verso una staging area temporanea. Questa operazione può avvenire tramite diversi protocolli o tecnologie (ad esempio JDBC, FTP, o file system), a seconda della natura della sorgente. Gli LKM possono includere logiche di estrazione, trasformazione iniziale o controllo qualità dei dati, prima che questi vengano elaborati ulteriormente.
- IKM (Integration Knowledge Module): gestiscono la fase di integrazione e caricamento dei dati dalla staging area alla tabella di destinazione. Questi moduli determinano la modalità con cui i dati vengono inseriti, aggiornati o uniti alla destinazione, utilizzando strategie come insert-only, insert/update o merge.

La seconda parte del template è invece dedicata alla definizione della query di insert e delle logiche necessarie alla gestione degli elementi ripetitivi nei mapping, con l'obiettivo di automatizzare e semplificare tali configurazioni.

Il primo mapping analizzato è quello relativo alla tabella MDM_NEGOZI, ovvero il mapping di **MDM (Master Data Management)**, il cui obiettivo è consolidare e normalizzare dati provenienti da più sorgenti in una visione unificata e coerente. Questo

mapping presenta un elemento ricorrente comune a tutti i progetti esaminati: la creazione delle chiavi surrogate. Indipendentemente dal numero di sorgenti coinvolte, una volta che i dati vengono unificati in una tabella normalizzata, risulta infatti necessario associare a ciascun record un identificativo numerico univoco. Questa esigenza si traduce, dal punto di vista implementativo, nella corretta definizione di una sequenza numerica all'interno del database di progetto e nell'inserimento, tra le variabili di output della query, di un campo parametrico destinato a contenere la chiave surrogata. Tale campo viene definito nel foglio Excel di input e, grazie all'integrazione con il software aziendale, popolato automaticamente durante la generazione dello script.

Accanto a questo primo elemento standardizzabile, il mapping MDM prevede due ulteriori passaggi, più complessi da generalizzare in quanto strettamente legati alle specificità del progetto e, in particolare, alla struttura delle tabelle sorgente.

Il primo di questi riguarda l'unione dei dati provenienti da più sorgenti. A tal fine è stato implementato un ciclo che percorre tutte le tabelle specificate nel foglio Excel; al suo interno è inclusa una logica condizionale che consente di generare dinamicamente una query SELECT completa e corretta, includendo tutte le tabelle sorgenti e i relativi vincoli di join, così da garantire la coerenza del risultato finale.

Il secondo passaggio riguarda invece l'arricchimento del dato, ossia l'integrazione di informazioni aggiuntive provenienti da tabelle esterne. Anche in questo caso viene adottata una logica annidata: il ciclo esterno elabora tutte le tabelle di arricchimento indicate nel foglio Excel, mentre la logica condizionale interna valuta il numero effettivo di tabelle presenti e genera il codice necessario in modo logicamente corretto, assicurando l'inserimento coerente delle informazioni supplementari nel processo di integrazione.

Per quanto riguarda i Knowledge Modules, sono stati selezionati quelli tradizionalmente associati a questa tipologia di mapping, prestando particolare attenzione alla definizione del Loading Knowledge Module. In particolare, è stato utilizzato un modulo di tipo *merge* basato sulle chiavi primarie definite nel database sorgente. Anche in questo caso è stato fatto ricorso a variabili parametriche, il cui valore viene determinato a partire dalle informazioni inserite nel foglio Excel. Questa strategia consente una maggiore flessibilità, poiché permette di adattare il template a tabelle diverse semplicemente modificando le chiavi indicate nel file di input.

Infine, anche tutte le variabili tecniche, come ad esempio i JOBID associati alle operazioni di inserimento e aggiornamento, sono state gestite tramite l'utilizzo di campi parametrici. Nell'ambito del framework aziendale, la nomenclatura di tali variabili è stata resa uniforme, con l'obiettivo di standardizzare il flusso di integrazione e semplificarne la manutenzione nel tempo. Poiché l'unico elemento che varia da progetto a progetto è il nome del progetto stesso, da cui dipende la definizione delle variabili tecniche, anche questo valore è stato configurato come campo parametrico. Tale parametro viene valorizzato automaticamente attraverso il foglio Excel di input, garantendo così coerenza e riutilizzabilità del modello.

Seguendo il framework aziendale, il secondo mapping preso in esame è quello relativo al livello **OUT**. Analogamente al livello di Master Data Management, anche questo mapping prevede l'unione di più sorgenti dati. Tuttavia, in questo caso l'obiettivo è duplice: da un lato, raccogliere le informazioni richieste dallo strumento di data visualization; dall'altro,

derivare eventuali informazioni aggiuntive utili all'analisi. La logica implementata per lo sviluppo del relativo template risulta quindi simile a quella già descritta in precedenza.

Nello specifico, è stato sviluppato un ciclo che scorre le tabelle definite nel foglio Excel di input. All'interno del ciclo è stata inserita una logica condizionale (if...then...else) che, attraverso l'utilizzo di campi parametrici, consente di popolare correttamente l'elenco delle tabelle sorgenti e le relative condizioni di join. Questa logica viene attivata solo nel caso in cui siano presenti più tabelle di input, rendendo il processo dinamico e adattabile a diversi scenari progettuali.

All'interno di questo mapping, come anche negli altri livelli del flusso, è inoltre presente un filtro che consente di selezionare esclusivamente i record non ancora visualizzati. Tale meccanismo è implementato mediante l'uso combinato di campi parametrici e campi tecnici. I campi parametrici permettono di specificare il nome del progetto all'interno del quale sono stati definiti i campi tecnici; questi ultimi, grazie a una nomenclatura standardizzata, possono essere richiamati direttamente nel template senza ulteriori personalizzazioni. Questo approccio consente di riutilizzare lo stesso template su più progetti, modificando unicamente i valori nei fogli Excel di input.

Come nel caso del mapping MDM, anche in questa fase è stata posta particolare attenzione alla configurazione dei Knowledge Modules, in particolare dei LKM (Loading Knowledge Modules), fondamentali per il corretto caricamento dei dati nelle tabelle di destinazione. In questo mapping è stata adottata una logica di tipo *truncate-insert*, che prevede la creazione e la gestione di una tabella temporanea come supporto intermedio al caricamento finale.

L'ultimo livello oggetto di analisi è stato il mapping **PUB**. La sua struttura si articola principalmente in due componenti fondamentali: l'applicazione di un filtro basato sul campo JOBID e la definizione della funzione ROW_NUMBER(), utilizzata per individuare il record più aggiornato per ciascuna entità identificata dalla chiave primaria.

Per quanto riguarda il primo elemento, è stata adottata la medesima logica già descritta nel paragrafo precedente, basata sull'uso combinato di campi parametrici e tecnici per garantire la flessibilità e la riusabilità del template su progetti differenti.

La definizione della funzione ROW_NUMBER() è stata invece sviluppata facendo nuovamente ricorso ai campi parametrici, con l'obiettivo di rendere il template il più possibile configurabile e adattabile a contesti progettuali diversi. In questo modo, è possibile gestire con efficacia la selezione dei record più recenti senza la necessità di modificare manualmente il codice sorgente.

Infine, per quanto concerne la configurazione dei Knowledge Modules, è stata adottata una logica di tipo *merge*, analoga a quanto già implementato nel mapping MDM. In questo caso, però, il confronto tra i record avviene sulla base della chiave surrogata definita nel livello precedente, assicurando così l'aggiornamento corretto delle informazioni all'interno della tabella di destinazione.

Una volta completata l'implementazione dei template, è stato necessario verificarne il corretto funzionamento attraverso la compilazione dei file Excel, che costituiscono il principale strumento di input per il processo di generazione automatica dei mapping. Questi file contengono tutte le informazioni necessarie per consentire allo strumento ODI

Generator di generare i mapping in modo coerente con la struttura definita dai template, rendendo il processo parametrico, standardizzato e facilmente riutilizzabile semplicemente modificando il contenuto del file Excel per adattarlo ai diversi progetti.

Per ciascun mapping è stato predisposto un file Excel dedicato, all'interno del quale sono state inserite le specifiche necessarie alla generazione automatica dei livelli corrispondenti alle tabelle NEGOZI e NAZIONI. I file Excel seguono una struttura standardizzata, progettata per garantire la corretta interpretazione dei campi da parte dello strumento. In particolare, le colonne dei fogli di lavoro vengono popolate con informazioni testuali ricavate dal database, tra cui i nomi delle tabelle sorgente e di destinazione, i campi da includere, le eventuali regole di trasformazione e altri parametri fondamentali per la costruzione del mapping.

La struttura di ciascun file prevede tre fogli distinti: due presentano una struttura predefinita da compilare con i dettagli tecnici relativi ai mapping (come mostrato nelle Figure 4.7 e 4.8), mentre il terzo è un foglio descrittivo, in formato *readme*, che offre indicazioni pratiche e spiegazioni utili per la corretta compilazione.

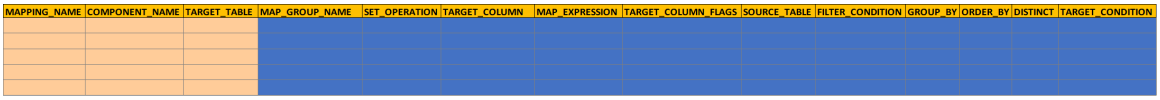

Figura 4.7. Foglio 'Mapping Data' del file Excel

È importante sottolineare che non tutte le colonne presenti nei fogli Excel devono necessariamente essere valorizzate. Tuttavia, alcuni campi risultano indispensabili per garantire il corretto funzionamento del processo di generazione. L'assenza o l'errata valorizzazione di tali elementi potrebbe infatti compromettere l'intera esecuzione automatica, causando errori o incoerenze nei mapping prodotti.

Per questo motivo, una parte centrale del progetto ha riguardato la corretta compilazione e validazione dei file di input, con particolare attenzione ai campi considerati obbligatori. La tabella riportata di seguito elenca i principali campi che devono essere valorizzati affinché la generazione automatica avvenga correttamente, accompagnati da una breve descrizione del loro significato e del ruolo che svolgono nel processo (Tabella 4.1).

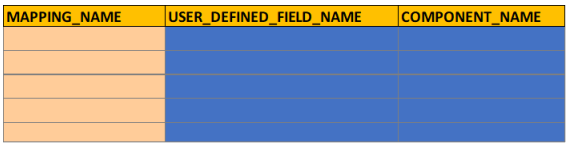

Figura 4.8. Foglio 'Properties' del file Excel

A seguito della corretta compilazione dei file Excel, è stato possibile caricare sia i template sia i file di input relativi a ciascun mapping sulla macchina virtuale che ospita

Nome della colonna	Descrizione
MAPPING_NAME	Identifica il mapping da generare, solitamente ottenuto concatenando il nome del progetto e quello della tabella da mappare.
TARGET_TABLE	Indica il nome della tabella target del mapping
TARGET_COLUMN	Elenca i nomi delle colonne presenti nella tabella di destinazione.
TARGET_COLUMN_FLAGS	Specifica eventuali vincoli associati a ciascun campo della tabella target (per esempio INSONLY per campi in solo inserimento, KEY per chiavi primarie).
MAP_EXPRESSION	Definisce l'espressione di mapping come combinazione del nome della tabella sorgente e del nome della colonna da mappare nella tabella target
SOURCE_TABLE	Indica i nomi delle tabelle sorgenti con relativi alias univoci. In caso di join, è riportata l'espressione completa del join
COMPONENT_NAME	Contiene i valori assegnati ai campi personalizzati (USER_DEFINED_FIELD_NAME), come il nome del progetto o campi tecnici specifici

Tabella 4.1. Principali campi richiesti nei file Excel di input

il software aziendale. Seguendo i passaggi descritti nei paragrafi precedenti, sono stati quindi generati con successo i mapping per entrambe le tabelle oggetto dell'integrazione. Nel capitolo successivo verranno presentati i risultati ottenuti a seguito del processo di generazione automatica delle mappature, insieme alle metriche utilizzate per valutarne l'efficacia e la correttezza.

Capitolo 5

Risultati

In questo capitolo vengono illustrati i risultati ottenuti dall'utilizzo del software aziendale per la generazione automatica dei mapping, realizzata tramite i template sviluppati nel corso di questo progetto e l'impiego di file di input parametrizzati. Dopo aver completato le fasi di progettazione, configurazione e caricamento dei dati sulla piattaforma, è stato possibile avviare con successo il processo di generazione dei mapping per le tabelle previste.

L'obiettivo principale di questo capitolo è valutare l'efficacia, l'efficienza e la manutenibilità della soluzione proposta, attraverso l'analisi di alcune metriche significative. Tra queste si includono:

- La percentuale di riutilizzo del codice, che misura quanto i template possano essere riutilizzati in progetti diversi o in mapping multipli senza modifiche sostanziali;
- Il tempo medio di sviluppo, confrontando il processo manuale di creazione dei mapping con quello automatizzato tramite template;
- La manutenibilità del sistema, intesa sia come facilità di intervento sulle correzioni, grazie alla possibilità di risolvere eventuali bug direttamente sui template, evitando modifiche ripetitive sui singoli mapping, sia come capacità di adattarsi nel tempo a nuove esigenze progettuali o cambiamenti nei requisiti tecnici, mantenendo una struttura solida e facilmente aggiornabile.

L'applicazione dei template implementati ha prodotto i risultati attesi: i mapping relativi alle due tabelle sono stati generati correttamente e, una volta esportati all'interno del progetto originale, è stato possibile eseguirli senza riscontrare errori. Questo ha permesso di incrementare il livello di automazione nel processo di creazione dei mapping. Come descritto in precedenza, i mapping iniziali erano già stati automatizzati negli anni passati; l'introduzione dei template anche per i mapping successivi ha quindi rappresentato un ulteriore progresso, contribuendo al completamento del processo di semi-automazione del framework aziendale sviluppato con lo strumento ODI. Tuttavia, poiché la generazione dei mapping richiede ancora la compilazione manuale dei file di input, il processo non può essere considerato completamente automatico.

Tempo medio di sviluppo

Uno dei principali vantaggi derivanti dalla realizzazione dei template sviluppati in questo progetto di tesi riguarda la significativa riduzione dei tempi necessari per l'implementazione dei mapping, con conseguenti benefici in termini di costi. L'utilizzo combinato del software aziendale e degli script appositamente creati consente di generare un elevato numero di mapping in tempi inferiori rispetto al processo manuale.

Nel caso di una creazione manuale dei mapping direttamente su ODI, infatti, l'intero processo potrebbe richiedere diversi giorni di lavoro, durante i quali gli operatori sono impegnati in attività ripetitive su una vasta quantità di tabelle da gestire. Oltre all'impegno temporale, questo approccio comporta un maggiore rischio di introduzione di errori durante la configurazione dei mapping, che a sua volta comporta ulteriore dispendio di tempo per la loro individuazione e correzione. Tale metodo manuale, oltre a risultare inefficiente, può compromettere la qualità complessiva del progetto, rendendo la generazione automatizzata attraverso template una soluzione decisamente più efficace e affidabile.

Al contrario, l'utilizzo di ODI Generator consente una significativa riduzione dei tempi di sviluppo, grazie a un processo semplificato che richiede la compilazione di un unico file per mapping, contenente tutte le specifiche necessarie relative alle diverse tabelle coinvolte nel progetto. I template, implementati nel corso di questo lavoro, automatizzano la generazione dei flussi, che vengono creati in modo rapido e coerente, garantendo una standardizzazione che facilita l'integrazione e l'esecuzione all'interno del progetto. Questo metodo non solo velocizza l'intero processo, ma contribuisce anche a ridurre il carico di lavoro manuale, liberando risorse e tempo preziosi. Inoltre, l'automatizzazione limita drasticamente il rischio di errori derivanti dalla natura ripetitiva e soggetta a distrazioni delle attività manuali, migliorando così la qualità e l'affidabilità dei mapping prodotti. In questo modo, l'approccio basato su template rappresenta un miglioramento significativo rispetto ai metodi tradizionali, in cui la configurazione manuale può risultare lenta, onerosa e incline a errori.

Per offrire un confronto quantitativo tra le due modalità di sviluppo, sono state stimate le tempistiche necessarie per la creazione dei mapping considerando due scenari distinti: la realizzazione manuale tramite ODI Studio e la generazione automatica tramite ODI Generator (Tabella 5.1). L'analisi è stata focalizzata esclusivamente sulla fase di creazione dei mapping a partire dal flusso di Master Data Management (MDM), che costituisce il cuore del progetto di tesi, assumendo che tutte le fasi preliminari fossero già state completate. Nel caso dell'implementazione manuale, è stato preso come riferimento un tempo medio di sei ore per la realizzazione dei mapping per una tabella. Per la generazione automatica, invece, sono stati considerati i tempi necessari per la compilazione del file di input e per l'esecuzione della generazione tramite ODI Generator, ossia il tempo richiesto una volta che i template sono stati predisposti e resi operativi.

Percentuale di riutilizzo del codice

Il codice implementato è stato sviluppato a partire dalle tabelle NAZIONI e NEGOZI, tenendo però sempre come obiettivo quello di generalizzare il più possibile la soluzione, in modo da renderla riutilizzabile in un'ampia varietà di progetti. Come discusso nei

	Implementazione manuale	**Implementazione automatica**
1 tabella	6 ore	2 ore
5 tabelle	30 ore	10 ore
50 tabelle	300 ore	100 ore

Tabella 5.1. Stima dei tempi di creazione dei mapping: approccio manuale vs automatico

capitoli precedenti, i mapping analizzati nel presente lavoro presentano una struttura altamente variabile a seconda del progetto specifico per cui vengono creati, rendendo di fatto complessa la standardizzazione e la generalizzazione del codice. Questa eterogeneità costituisce una sfida importante per garantire la flessibilità e la scalabilità della soluzione proposta. Per verificare l'efficacia e la versatilità dei template sviluppati, oltre ai test effettuati sulle tabelle NAZIONI e NEGOZI, sono stati condotti ulteriori esperimenti su un progetto di dimensioni ridotte, composto da due tabelle dei fatti e sei tabelle delle dimensioni. Questo secondo scenario ha consentito un controllo più accurato dei risultati ottenuti e ha permesso di confermare che la generazione dei mapping avviene correttamente, rispettando le specifiche definite all'interno dei file di input.

Facilità di manutenzione

Un ulteriore elemento utile per evidenziare i vantaggi derivanti dall'utilizzo dei template è la manutenibilità del sistema, intesa sia come facilità di intervento in fase di correzione, sia come capacità di adattarsi nel tempo a nuove esigenze progettuali o a modifiche nei requisiti tecnici.

Grazie all'adozione del software aziendale integrato con i template sviluppati, la gestione di bug ed errori risulta decisamente più rapida ed efficiente rispetto a un approccio manuale. In particolare, nel caso in cui venga rilevato un errore ricorrente in più mapping, l'approccio tradizionale richiederebbe l'intervento manuale su ciascun mapping interessato, con un notevole dispendio di tempo e risorse. Al contrario, con l'approccio automatizzato, una volta individuata la causa dell'errore è sufficiente intervenire direttamente sul template: in pochi passaggi è possibile rigenerare correttamente tutti i mapping aggiornati, senza doverli modificare singolarmente.

Allo stesso modo, nel caso in cui si verifichino variazioni nei requisiti o nei vincoli progettuali, come ad esempio modifiche ai nomi dei campi o alla definizione dei vincoli di colonna, l'utilizzo di ODI Generator consente di intervenire in maniera centralizzata, modificando esclusivamente il file di input o il template. In questo modo, le modifiche vengono propagate a tutti i mapping interessati, mantenendo la coerenza e riducendo drasticamente i tempi di intervento. Al contrario, con un approccio manuale, sarebbe necessario aggiornare ogni singolo mapping uno per uno, con un considerevole dispendio di tempo e risorse e un maggiore rischio di introdurre errori.

Capitolo 6

Conclusione

La presente tesi ha affrontato il tema dell'automazione nella creazione dei processi ETL, un ambito strategico per l'innovazione tecnologica e per il mantenimento della competitività nel mercato attuale. È stato dimostrato come l'uso di template per la generazione automatica dei mapping consenta un netto miglioramento in termini di efficienza operativa e riutilizzabilità del codice.

In passato, l'azienda Mediamente Consulting aveva già automatizzato gli step iniziali del flusso ETL, dedicati alla replica e certificazione dei dati sorgente, senza modificarne la struttura. Questo lavoro si è quindi concentrato sulle fasi successive del processo, a partire dal livello di Master Data Management (MDM), in cui prende avvio la trasformazione dei dati finalizzata alla costruzione della struttura del data warehouse in base alle specifiche esigenze analitiche dei clienti.

L'attività principale ha riguardato la progettazione e l'implementazione di script in Apache Velocity, i quali, tramite l'impiego di variabili parametriche, permettono di generare automaticamente i mapping ETL attraverso il software aziendale.

Il progetto si è sviluppato in diverse fasi. In primo luogo è stato analizzato il flusso ETL in uso e il sistema di integrazione fornito da Oracle. Successivamente, sono stati individuati i pattern ricorrenti nei mapping creati manualmente, che sono poi stati incorporati negli script per automatizzare le fasi successive del framework aziendale.

Nel dettaglio, sono stati sviluppati tre template specifici per i mapping MDM, OUT e PUB, relativi alla seconda parte del flusso ETL. Tali componenti completano il percorso di automazione già avviato in azienda, basato sulla tecnologia ODI (Oracle Data Integrator) per l'integrazione dei dati.

I risultati ottenuti confermano il raggiungimento degli obiettivi prefissati: è stato infatti possibile generare automaticamente i mapping per numerose tabelle, riducendo in modo significativo sia i tempi di sviluppo sia il numero di operazioni manuali. Il sistema si è dimostrato efficace, affidabile e facilmente riutilizzabile, richiedendo unicamente la corretta compilazione dei file di input per adattarsi a diversi progetti.

Sulla base delle soluzioni implementate, si possono tuttavia individuare diverse opportunità di miglioramento e ampliamento. Alcuni possibili sviluppi futuri, orientati all'evoluzione tecnica e all'estensione funzionale del sistema, saranno illustrati nella sezione successiva.

6.1 Sviluppi Futuri

Un primo possibile sviluppo del lavoro riguarda la naturale evoluzione del progetto verso forme sempre più avanzate di automazione. Il presente contributo si colloca, infatti, in una fase intermedia tra la generazione manuale dei mapping ETL e un futuro scenario in cui tali processi potranno essere completamente automatizzati mediante strumenti basati sull'intelligenza artificiale generativa. Attraverso gli script realizzati in questa tesi è stato possibile ridurre in modo significativo il tempo necessario per lo svolgimento di operazioni ripetitive e manuali. Tuttavia, la compilazione dei file di input in formato Excel, necessari per la generazione dei mapping, rappresenta ancora un'attività da svolgere manualmente.

Un'evoluzione naturale potrebbe dunque consistere nello sviluppo di un modello di machine learning addestrato a partire dai file XML generati da Oracle Data Integrator (ODI), in grado di produrre direttamente il codice necessario per i mapping sulla base di input strutturati, eliminando così la necessità della compilazione manuale dei fogli Excel. Questo approccio permetterebbe di aumentare ulteriormente l'automazione, rendendo il processo ancora più efficiente, scalabile e meno soggetto a errori umani.

Le potenzialità offerte dai modelli di intelligenza artificiale e machine learning nel contesto dell'integrazione dei dati non si esauriscono tuttavia nella sola generazione dei flussi ETL. Un'ulteriore direzione di sviluppo riguarda la creazione automatica della documentazione dei mapping. Nei progetti di grandi dimensioni, infatti, risulta spesso complesso e dispendioso ricostruire il data lineage, ovvero la sequenza di trasformazioni che porta dalla sorgente al dato visualizzato in uno strumento di business intelligence. In questo contesto, tecniche di machine learning potrebbero essere sfruttate per analizzare e tracciare automaticamente l'origine e l'evoluzione dei dati, rendendo possibile individuare con precisione il punto in cui intervenire in caso di errori o incongruenze nei dati finali.

In aggiunta, un possibile ampliamento futuro, particolarmente rilevante nel contesto aziendale, potrebbe riguardare l'estensione del sistema anche ad altri strumenti di integrazione dati oltre a Oracle Data Integrator. L'obiettivo sarebbe quello di rendere la generazione automatica dei mapping compatibile con una più ampia varietà di tecnologie presenti sul mercato, aumentando così la versatilità della soluzione proposta e garantendo una maggiore copertura rispetto alle diverse esigenze dei clienti.

Questa esigenza emerge anche da una considerazione pratica: attualmente, persino nel caso di Oracle Data Integrator, il software aziendale utilizzato non risulta compatibile con l'ultima versione dello strumento, rilasciata di recente. Ciò evidenzia la necessità di individuare e integrare nuovi strumenti o piattaforme capaci di supportare la generazione dei mapping tramite template, così da assicurare la continuità operativa e mantenere aggiornato il processo di automazione in linea con l'evoluzione tecnologica.

Tali sviluppi, oltre a rappresentare un significativo passo avanti in termini di automazione e gestione dei processi, offrirebbero un valore aggiunto concreto in termini di trasparenza, tracciabilità e affidabilità delle informazioni elaborate.

Bibliografia

[1] Vlad Diaconita Alexandra Maria Ioana Florea and Ramona Bologa. Data integration approaches using etl. *Database Systems Journal*, 6:19–27, 2015.

[2] Fernando Almeida. *Concepts and Fundaments of Data Warehousing and OLAP*. ISSUU Publishing, 2017.

[3] Samuel S. Conn. Oltp and olap data integration: a review of feasible implementation methods and architectures for real time data analysis. *Proceedings. IEEE SoutheastCon*, page 515–520, 2005.

[4] Filippo La Noce e Luigi D'Ercole. *Data Warehousing: Dal dato all'informazione*. FrancoAngeli, 2001.

[5] Mimi Safinaz Jamaluddin e Nurulhuda Firdaus Mohd Azmi. Extraction transformation load (etl) solution for data integration: A case study of rubber import and export information. *Jurnal Teknologi*, 78, 2015.

[6] G. Satyanarayana Reddy et al. Data warehousing, data mining, olap and oltp technologies are essential elements to support decision-making process in industries. *International Journal on Computer Science and Engineering 2.9*, 02:2865–2873, 2010.

[7] R. Bommasani et al. On the opportunities and risks of foundation models. URL https://arxiv.org/abs/2108.07258.

[8] Gartner. Gartner experts answer the top generative ai questions for your enterprise. URL https://www.gartner.com/en/topics/generative-ai.

[9] William H. Inmon. *Building the Data Warehouse*. John Wiley Sons, Inc., 2005.

[10] Bernard Marr. The essential skills that will define success in the ai era (and they're not what you think), . URL https://bernardmarr.com/the-essential-skills-that-will-define-success-in-the-ai-era-and-theyre-not-what-you-thin

[11] Bernard Marr. Human plus ai: Redefining work in the age of collaborative intelligence, . URL https://www.forbes.com/sites/bernardmarr/2025/03/07/human-plus-ai-redefining-work-in-the-age-of-collaborative-intelligence/.

[12] Natalia Miloslavskaya and Alexander Tolstoy. Big data, fast data and data lake concepts. *Procedia Computer Science*, 88:300–305, 2016.

[13] Oracle. Oracle data integrator. URL https://www.oracle.com/it/middleware/technologies/data-integrator.html.

[14] Margy Ross Ralph Kimball. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, Third Edition*. John Wiley Sons, Inc., 2013.

[15] Cem Dilmegani (AIMultiple Research). Top 100 genai applications with real-life examples in 2025. URL https://research.aimultiple.com/generative-ai-applications/.

[16] Jasna Rodic and Mirta Baranovic. Generating data quality rules and integration into etl process. *Proceedings of the ACM*, pages 65–72, 2009.

[17] Dhamotharan Seenivasan. Etl vs elt: Choosing the right approach for your data warehouse. *International Journal for Research Trends and Innovation*, 7:110–122, 2022.

[18] Abdeltawab M. Ahmed Hendawi Shaker H. Ali El-Sappagh and Ali Hamed El Bastawissy. A proposed model for data warehouse etl processes. *Journal of King Saud University-Computer and Information Sciences*, 23:91–104, 2011.

[19] Christian Janiesch Stefan Feuerriegel, Jochen Hartmann and Patrick Zschech. Catchword generative ai. *Business Information Systems Engineering*, 66.1:111–126, 2023.

[20] Umeshwar Dayal Surajit Chaudhuri and Vivek Narasayya. An overview of business intelligence technology. *Communications of the ACM*, 54:88–98, 2011.

[21] Robert Thanaraj Michele Launi Nina Showell Thornton Craig, Sharat Menon. Gartner magic quadrant for data integration tools. URL https://www.gartner.com/en/documents/5975271.

[22] Ishit Vachhrajani. Fuel your data with generative ai. URL https://aws.amazon.com/it/blogs/enterprise-strategy/fuel-your-data-with-generative-ai/.

[23] Boris Zaikin. Achieving data excellence: How generative ai revolutionizes data integration. URL https://dzone.com/articles/achieving-data-excellence-generative-ai.

[24] Adam Zewe. Mit researchers introduce generative ai for databases. URL https://news.mit.edu/2024/mit-researchers-introduce-generative-ai-databases-0708.