

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

---



**TESI DI LAUREA**

**Re-engineering del sistema di Business Intelligence a supporto delle attività di analisi di un'azienda leader nel settore degli pneumatici**

**Re-engineering of a Business Intelligence system supporting the analysis activities of a leading tire company**

Relatore

Prof. Domenico Ursino

Candidato

Pinciaroli Andrea

---

**ANNO ACCADEMICO 2021-2022**

Ad Alessandro,

*Siamo venuti al mondo come fratello e fratello;  
E ora andiamo di pari passo, non uno prima dell'altro*

William Shakespeare

## Sommario

Negli ultimi anni sempre più aziende hanno iniziato a prendere consapevolezza sull'importanza dell'acquisizione e organizzazione dei dati con il fine di eliminare inefficienze e attuare rapidi cambiamenti nei processi sulla base delle decisioni prese a partire da essi. Tale processo, sotto il nome di Business Intelligence combina business analytics, data mining, visualizzazione e infrastrutture per i dati. In questa tesi si discuteranno le procedure attuate per la reingegnerizzazione di una piattaforma di BI non più in grado di soddisfare le esigenze dell'utente. In particolare, sarà descritta la vecchia piattaforma, seguita dalla progettazione ed implementazione delle attività di reingegnerizzazione per concludere con l'analisi del nuovo sistema di BI.

**Keyword:** Business Intelligence; Business Analytics; Re-engineering; Data Engineering; Extract Trasform and Load; Data Platform; Data Warehouse; Oracle Obiee; Qlik Sense

<b>Introduzione</b>	<b>1</b>
<b>1 Il re-engineering</b>	<b>3</b>
1.1 Re-engineering in letteratura . . . . .	3
1.2 Definizione di re-engineering per una piattaforma di BI . . . . .	4
1.2.1 Framework di Processo . . . . .	4
1.2.2 Re-engineering e Reverse engineering . . . . .	5
1.3 Obiettivi del re-engineering . . . . .	5
1.3.1 Possibili svantaggi e criticità . . . . .	6
<b>2 Descrizione del sistema di BI di partenza: Oracle Obiee</b>	<b>7</b>
2.1 Introduzione ad Oracle Obiee . . . . .	7
2.2 Descrizione del DWH di partenza . . . . .	7
2.3 Definizione della home page . . . . .	8
2.4 Definizione dell'editor di analisi . . . . .	9
2.5 Creazione di analisi . . . . .	9
2.5.1 Istruzioni SQL e utilizzo delle richieste dirette al database . . . . .	10
2.5.2 Visualizzazione dei risultati . . . . .	11
2.6 Descrizione di Oracle BI Presentation Catalog . . . . .	12
2.7 Vantaggi e Svantaggi . . . . .	12
<b>3 Analisi dei requisiti</b>	<b>14</b>
3.1 Introduzione all'analisi dei requisiti . . . . .	14
3.2 Raccolta delle logiche di sviluppo del Back-end . . . . .	15
3.2.1 Inbound vs Outbound . . . . .	15
3.3 Creazione della documentazione di Reverse . . . . .	15
3.4 Raccolta logiche di sviluppo Front-end . . . . .	16
3.4.1 Creazione documenti Business Glossary . . . . .	16
<b>4 Progettazione delle attività di re-engineering</b>	<b>19</b>
4.1 Definizione della metodologia Agile . . . . .	19
4.2 Fase di avvio . . . . .	20
4.3 Pianificazione delle attività . . . . .	20
4.4 Fase di esecuzione . . . . .	21
4.4.1 Milestone di sviluppo front end . . . . .	22

---

4.5	Valutazione e chiusura . . . . .	22
<b>5</b>	<b>Implementazione delle attività di re-engineering</b>	<b>23</b>
5.1	Sviluppo nuove applicazioni . . . . .	23
5.1.1	Generazione dei QVD per tabelle dei fatti e dimensioni . . . . .	23
5.2	Fase di ETL e creazione del modello dati . . . . .	24
5.2.1	Creazione della reportistica . . . . .	25
5.2.2	Integration test e rilascio . . . . .	26
5.3	Sviluppo applicazioni dual run . . . . .	26
5.3.1	Traduzione delle query . . . . .	27
5.3.2	Aggiornamento della reportistica . . . . .	27
5.4	Fase di UAT . . . . .	27
5.5	Fase di training con gli utenti . . . . .	28
5.6	Chiusura della subject area . . . . .	28
<b>6</b>	<b>Esempi di funzionamento del nuovo sistema di Business Intelligence</b>	<b>29</b>
6.1	Applicazione dualrun, il caso "SupplyChain - NetKpi - Overstock" . . . . .	29
6.1.1	Estrazione dei dati dalle tabelle dei fatti . . . . .	29
6.1.2	Seconda fase di estrazione, join con le tabelle delle dimensioni . . . . .	30
6.1.3	Organizzazione dei campi e delle chiavi . . . . .	32
6.1.4	Creazione del modello dati . . . . .	34
6.1.5	Visualizzazione dei dati . . . . .	35
6.2	Nuova applicazione, "Planning - Logistic Export Monitor" . . . . .	37
6.2.1	Data load editor . . . . .	38
6.2.2	Visualizzazione dei dati . . . . .	39
<b>7</b>	<b>Discussione in merito al lavoro svolto</b>	<b>41</b>
7.1	Punti di Forza . . . . .	41
7.1.1	Progettazione ben definita . . . . .	41
7.1.2	Utilizzo di tecnologie innovative . . . . .	42
7.1.3	Team eterogeneo e altamente preparato . . . . .	42
7.2	Punti di debolezza . . . . .	42
7.2.1	Scarsa conoscenza dei dati . . . . .	43
7.2.2	Analisi e raccolta dei requisiti incomplete . . . . .	43
7.2.3	Modifica delle richieste e delle priorità del cliente . . . . .	43
7.3	Lezioni Apprese . . . . .	44
7.3.1	Importanza del processo di validazione dati . . . . .	44
7.3.2	Attenzione alle performance . . . . .	44
	<b>Introduzione</b>	<b>46</b>
	<b>Bibliografia</b>	<b>48</b>
	<b>Ringraziamenti</b>	<b>51</b>

---

## Elenco delle figure

---

1.1	Framework di processo . . . . .	5
2.1	Logo Oracle Obiee . . . . .	7
2.2	Home page Oracle Obiee . . . . .	8
2.3	Scheda Criteri Oracle Obiee . . . . .	10
2.4	Query processing . . . . .	10
2.5	Un esempio di dashboard Obiee . . . . .	11
2.6	Oracle BI Presentation Catalog . . . . .	12
3.1	Esempio di nuvola dei dati . . . . .	17
3.2	Esempio descrizione di un report Obiee . . . . .	18
4.1	Iterazione relativa alla metodologia Agile . . . . .	20
5.1	Processo di ETL . . . . .	24
5.2	Fase di ETL lato sviluppo front-end . . . . .	25
6.1	Estrazione e caricamento delle informazioni . . . . .	30
6.2	Concatenazione dei QVD mensili e pregresso . . . . .	31
6.3	Load QVD di ETL01 e join con tabelle delle dimensioni . . . . .	31
6.4	Store per creazione del QVD di ETL02 . . . . .	32
6.5	Load dei campi dal QVD di ETL03 . . . . .	34
6.6	Creazione della Link Table per il modello dei dati . . . . .	35
6.7	Panoramica di KPI Overstock . . . . .	35
6.8	Trend di KPI Overstock . . . . .	36
6.9	Global values di KPI Overstock . . . . .	36
6.10	Storico di KPI Overstock . . . . .	37
6.11	KPI Overstock in relazione alle factory . . . . .	37
6.12	Load dei dati direttamente da QVD tabella dei fatti . . . . .	38
6.13	Creazione della chiave per la Link Table . . . . .	39
6.14	Drop degli ID dalle tabelle dei fatti . . . . .	39
6.15	Report di Export Monitor . . . . .	40
6.16	Report di Export Monitor - EU . . . . .	40
7.1	Esempio di Preceding Load . . . . .	45

---

## Elenco delle tabelle

---

3.1	Esempio documento di Reverse . . . . .	16
3.2	Definizione del mapping dal DWH alla piattaforma dati . . . . .	17

Al giorno d'oggi i dati stanno assumendo un'importanza sempre maggiore nello sviluppo delle aziende. Una corretta organizzazione ed interpretazione dei dati può rappresentare un notevole vantaggio competitivo rispetto a chi ancora si limita ad utilizzare i dati senza averne una conoscenza vera e propria. L'analisi dei dati consiste nell'esplorazione, pulizia, trasformazione e modellazione con lo scopo ultimo di evidenziare informazioni che supportino le decisioni strategiche aziendali. Tali procedure vengono utilizzate come driver decisionali per la riduzione degli errori nella pianificazione delle strategie e nello sviluppo dell'azienda.

Grazie ai progressi compiuti nelle tecnologie dell'informazione le aziende possono ora raccogliere, elaborare, archiviare e diffondere grandi volumi di informazioni. È bene ricordare che i *dati* diventano *informazioni* solo quando vengono interpretati e da essi ne viene estratta conoscenza. I dati in passato venivano analizzati per prendere decisioni future; ad oggi, possono essere analizzati per prendere decisioni in tempo reale, evidenziare trend e svelare correlazioni ed anomalie nascoste utili per convalidare una teoria o un'ipotesi.

Molte aziende, in particolare le aziende leader di mercato di diversi settori, hanno già da tempo organizzato le proprie strutture IT in modo da riuscire a gestire l'enorme mole di dati ed estrarne conoscenza da essa. Come già affermato in precedenza, con gli sviluppi compiuti nelle tecnologie dell'informazione, alcune di queste strutture sono diventate già obsolete, o comunque non in grado di soddisfare le esigenze del business. Per questo sono state avviati progetti di re-engineering in modo da passare a nuove strutture di Business Intelligence (database, tool di visualizzazione) più performanti e adeguati al continuo cambiamento delle esigenze nelle decisioni strategiche.

Quando si esegue un processo di reingegnerizzazione di questo tipo, soprattutto in contesti aziendali molto grandi, è necessario eseguire una minuziosa progettazione ed analisi dei requisiti con chi utilizzerà la nuova piattaforma in modo da garantire il corretto proseguo delle analisi già in opera, migliorare le performance e aggiungere valore, ovvero facilitare ed ampliare le possibilità di interpretazione dei dati.

In questo elaborato vengono analizzate le fasi di progettazione ed implementazione del re-engineering della piattaforma di BI (Oracle Obiee) di un'azienda leader nel settore della produzione degli pneumatici. Nella prima fase sarà descritta la piattaforma di BI da reingegnerizzare; successivamente si passerà al processo di raccolta dei requisiti con il cliente e alla successiva progettazione e pianificazione del lavoro, per descrivere, poi, l'implementazione delle attività di re-engineering. Infine saranno mostrati due esempi di applicazioni Qlik Sense, comprensivi del processo di sviluppo, che vanno a sostituire i report Oracle Obiee.

L'obiettivo è quello di evidenziare le fasi ed i processi che permettono lo sviluppo delle



applicazioni Qlik Sense che sostituiranno i report della piattaforma di Business Intelligence attualmente in uso dal cliente. Si partirà dal processo di migrazione dei dati lato back-end per concentrarsi, poi, sulla creazione delle app di visualizzazione lato front-end. Lo scopo è quello di sottolineare la complessità e, quindi, l'organizzazione necessaria al successo di un progetto di tali dimensioni insieme alle funzionalità offerte dalla nuova piattaforma di Business Intelligence che aggiungono valore alle analisi aziendali.

La presente tesi è composta da otto capitoli strutturati come di seguito specificato:

- Nel Capitolo 1 sarà introdotto il concetto di re-engineering, dandone una definizione generale ed evidenziando vantaggi, svantaggi e possibili criticità.
- Nel Capitolo 2 verrà descritto il sistema di BI di partenza utilizzato dal cliente, ovvero Oracle Obiee.
- Nel Capitolo 3 verrà mostrata l'analisi dei requisiti necessaria per la pianificazione delle successive attività di re-engineering.
- Nel Capitolo 4 sarà descritta, a valle dell'analisi dei requisiti, la fase di progettazione delle attività di re-engineering.
- Nel Capitolo 5 verrà descritta l'implementazione delle attività di re-engineering analizzando ciò che avviene lato back-end e concentrandosi, poi, prevalentemente sulla reportistica lato front-end.
- Nel Capitolo 6 saranno mostrati due esempi di applicazioni Qlik Sense realizzate.
- Nel Capitolo 7 si discuterà in merito al lavoro svolto. In particolare saranno analizzati i punti di forza e debolezza del progetto per concludere, poi, con le nozioni acquisite.
- Infine, nel Capitolo 8 saranno tratte le conclusioni.

*In questo primo capitolo sarà introdotto il concetto di re-engineering. L'analisi inizierà da una breve introduzione storica per poi concentrarsi sull'applicazione di tale processo nel caso di studio in questione ovvero la piattaforma di Business Intelligence.*

*Sarà descritto il framework di sviluppo, ovvero l'insieme di operazioni da seguire necessarie per la corretta implementazione di tale metodologia e il raggiungimento dei risultati desiderati.*

*Nella parte finale del capitolo si evidenzierà la differenza tra i concetti di re-engineering e reverse engineering, spesso erroneamente confusi. Il capitolo si chiuderà analizzando vantaggi e svantaggi della reingegnerizzazione.*

### 1.1 Re-engineering in letteratura

Il concetto di reingegnerizzazione in un contesto aziendale prende il nome di Business Process re-engineering (BPR). Tale strategia di gestione aziendale viene in origine introdotta intorno agli inizi degli anni '90 con lo scopo di aiutare le organizzazioni (aziende, istituzioni, etc) a ripensare i processi lavorativi in atto con il fine di migliorare i servizi verso i clienti, di ridurre i costi operazionali e di aumentare la produttività e la qualità.

I primi sostenitori affermavano che un processo aziendale poteva essere suddiviso in un insieme di attività correlate logicamente con il fine di ottenere un risultato definito. In tale contesto reingegnerizzare significa ripensare l'intero ciclo di vita del processo in modo da renderlo più efficiente, piuttosto che ottimizzare in maniera iterativa i singoli sottoprocessi.

Michael Hammer, ex professore di informatica presso l'Istituto di tecnologia del Massachusetts, pubblicò, in quel periodo, un articolo dal nome *Reengineering Work: Don't Automate, Obliterate*. L'autore affermava che la maggior parte del lavoro che veniva svolto, in realtà, non aggiungeva effettivamente valore al cliente e che, quindi, fosse necessario che le aziende ripensassero l'utilizzo delle tecnologie informatiche che fino ad allora erano state utilizzate per automatizzare questo tipo di processi, piuttosto che renderli obsoleti a vantaggio di una riprogettazione totale del processo lavorativo.

Molte furono le critiche che minacciarono l'affermarsi delle tecniche di BPR, tra cui l'accusa di aumentare il controllo manageriale o giustificare il ridimensionamento, ovvero ridurre la forza lavoro. Nonostante ciò la reingegnerizzazione è stata ampiamente adottata anche grazie all'adozione del BPR da parte del settore della consulenza e grazie ai continui sviluppi tecnologici che rendono questa tecnica sempre più vantaggiosa.

## 1.2 Definizione di re-engineering per una piattaforma di BI

Come già anticipato nella sezione precedente, il re-engineering è un metodo utilizzato in campo manageriale con lo scopo di sopperire a risultati insoddisfacenti. Nella sua definizione più ampia, di BPR è stato sottolineato come la riprogettazione riguardi l'intero processo aziendale. Nel caso specifico di questa trattazione si analizzerà la realizzazione di tali metodologie applicate al singolo software, ovvero alla piattaforma di Business Intelligence.

Applicare queste tecniche ad un software significa che ciò che si sta eseguendo è un processo di Software Reengineering. La reingegnerizzazione del software, in questo caso la piattaforma di BI Oracle Obiee, nasce dalla necessità del cliente di rendere il processo di analisi di dati più efficiente e dinamico, adeguandosi alle nuove soluzioni offerte dal mercato.

Con *software reengineering* si intende il processo di modifica di un software legacy importante per un'azienda con il fine di migliorare le prestazioni, aumentare la qualità, l'utilità e il periodo di vita del software. Tale definizione si differenzia da un altro concetto che è quello del *forward reengineering*, ovvero la creazione di una nuova applicazione che sostituisca quella già esistente. La grande differenza tra i due processi sta nelle specifiche iniziali, poichè, per il software reengineering, è il sistema legacy che funge da specifica, mentre per il forward engineering il punto di partenza sono le nuove specifiche richieste che porteranno poi alla creazione del nuovo software. Nella reingegnerizzazione della piattaforma di BI in analisi, che come anticipato nell'introduzione, porterà ad una nuova soluzione attraverso l'utilizzo dell'applicazione Qlik Sense, ci si trova a metà tra i due processi appena descritti poichè, effettivamente, la nuova applicazione di reportistica dovrà riportare fedelmente le informazioni, utilizzando come specifica iniziale i report della vecchia piattaforma, tuttavia grazie alle caratteristiche del nuovo software, sarà possibile aggiungere valore all'analisi integrando nuove richieste da parte del cliente che si aggiungeranno alle maggiori potenzialità di analisi in real time e di visualizzazione offerte da Qlik Sense.

### 1.2.1 Framework di Processo

Nel corso degli anni con il continuo svilupparsi delle tecniche di reingegnerizzazione, si sono definite una serie di linee guida da seguire in modo da ottenere dei risultati di valore dall'applicazione di questa strategia aziendale. Non esiste un'unica successione sequenziale di operazioni per effettuare un processo di reingegnerizzazione; molto dipende da verso cosa si sta eseguendo questo tipo di operazione: software, processi fondamentali d'azienda, etc.

Nel caso in analisi in questa trattazione, ovvero la riprogettazione di una piattaforma di BI, la Figura 1.1 mostra in maniera efficace il framework, e quindi la serie di operazioni eseguite dai vari reparti aziendali (backend, funzionale, frontend) per ottenere un risultato capace di soddisfare il cliente.

Tale tipo di framework con le sue fasi viene adattato al caso specifico di analisi della riprogettazione di una piattaforma di BI. Nel seguito della trattazione questi step saranno analizzati in maniera più approfondita. In generale, è possibile dare una breve definizione di ciascuno come di seguito specificato:

- *Inventory Analysis*: possono essere semplicemente una serie di fogli di calcolo contenenti le informazioni relative alle applicazioni o, come nel caso in analisi, ai report.
- *Document Reconstructing*: documentazione che descrive il funzionamento della piattaforma originale
- *Reverse Engineering*: comprensione delle logiche di backend e frontend della piattaforma di BI.

- *Data Reconstructing*: processo di migrazione delle strutture dati dal vecchio database verso il nuovo.
- *Forward Engineering*: ricostruzione dei report riportando il contenuto informativo, rendendo l'analisi più efficiente e implementando anche nuove funzionalità per aggiungere valore al risultato finale

### 1.2.2 Re-engineering e Reverse engineering

La reingegnerizzazione e il reverse engineering sono due processi tecnici che sembrano simili e spesso vengono confusi. Entrambi sono utilizzati abitualmente nella gestione dell'obsolescenza di un qualche tipo di processo o software. Tuttavia, le due tecniche possono essere utilizzate nello stesso contesto come visto nella figura 1.1 dove il reverse engineering costituisce una fase operativa del processo totale di reingegnerizzazione, o come metodi separati che portano a risultati diversi.

Si parla di *re-engineering* quando si descrive il processo di ripensamento di dispositivi, flussi lavorativi o, nell'ambito informatico, applicazioni con il fine di velocizzare il raggiungimento di alcuni obiettivi. In generale, l'azienda, si trova di fronte a qualcosa che non soddisfa più le proprie esigenze, e per questo, si avvia questa riprogettazione dell'intero progetto o di alcuni singoli componenti.

Il *reverse engineering* consiste, invece, nel prendere un prodotto finito e scoprirne il suo funzionamento testandolo. In genere, questo può essere fatto all'interno di un'intera fase di riprogettazione, come nel caso in analisi, per comprendere il funzionamento della piattaforma di BI e del database esistente, così da poter poi poter ripetere le logiche nella nuova soluzione. Complessivamente questa tecnica viene effettuata dalle aziende per cercare di infiltrarsi nel mercato di un concorrente cercando di capire un prodotto attraverso l'identificazione dei componenti e delle relazioni esistenti tra di questi.

## 1.3 Obiettivi del re-engineering

Come già approfondito nelle sezioni precedenti, il processo tecnico di reingegnerizzazione è diventato imprescindibile per le aziende che vogliono affermarsi come competitors a livello

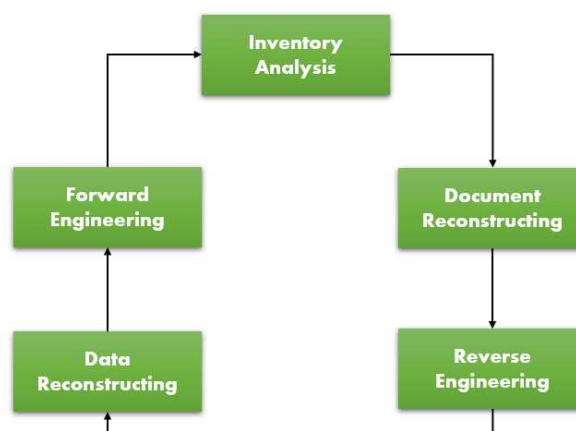


Figura 1.1: Framework di processo

mondiale. Nel caso in analisi riguardante una piattaforma di BI, tale processo permette di ottenere i seguenti vantaggi:

- *Aumento della produttività*: è possibile aumentare la produttività ottimizzando il database e i tempi di caricamento dei dati su Qlik in modo da velocizzare l'elaborazione
- *Continuità dei processi*: nel mentre che si sviluppa la nuova piattaforma gli utenti possono continuare ad utilizzare la vecchia per le analisi
- *Opportunità di miglioramento*: durante il processo di reingegnerizzazione è possibile implementare nuove funzionalità richieste dal cliente nei report e, quindi, ampliare l'orizzonte d'analisi
- *Riduzione dei rischi*: avendo a disposizione la piattaforma di partenza è sempre possibile confrontare la coerenza delle informazioni che vengono restituite dalla nuova soluzione.
- *Ottimizzazione*: il nuovo prodotto sarà più adeguato alle esigenze di business e l'analisi sarà più agevolata e più efficiente dal punto di vista prestazionale e temporale.

### 1.3.1 Possibili svantaggi e criticità

Il processo di reingegnerizzazione non è applicabile in ogni tipo di contesto aziendale poichè dipende molto da fattori quali dimensioni e disponibilità di risorse; per questo ne traggono vantaggio le grandi aziende. Questo processo tecnico non porta valore nell'immediato ma può essere considerato una sorta di investimento a lungo termine perchè, soprattutto nei contesti come quello in analisi, dove il progetto ha una durata complessiva di diversi anni, i risultati e lo sforzo complessivo sono difficili da valutare in real time. Richiede all'azienda, inoltre, un sostanziale investimento nella parte IT, una pianificazione adeguata e tanto lavoro di squadra, sia interno che con l'azienda di consulenza esterna che si occupa della reingegnerizzazione.

Una criticità di questa metodologia di riprogettazione deriva dal fatto che in passato molte aziende l'hanno utilizzata come pretesto per il ridimensionamento, ovvero per perseguire l'obiettivo di aumentare efficienza e produttività automatizzando, tramite le tecnologie, alcuni processi e riducendo il lavoro umano. La critica più frequente ai processi di reingegnerizzazione è stata il fatto che si concentra troppo sull'efficienza e sulla tecnologia a discapito del valore delle persone sottoposte a tale processo.

---

## Descrizione del sistema di BI di partenza: Oracle Obiee

---

*In questo secondo capitolo sarà fornita una descrizione generale della piattaforma di Business Intelligence attualmente utilizzata dal cliente, ovvero Oracle Obiee. Saranno mostrati, in primo luogo, il processo di integrazione dati e la sorgente che alimentano il sistema software. Si passerà poi ad una descrizione puntuale degli strumenti principali e delle feature per la creazione delle analisi. Verranno, quindi, descritte le opzioni che Obiee mette a disposizione per l'utente, soffermandosi su temi quali le potenzialità di visualizzazione offerte dalla piattaforma e il processo di estrazione delle informazioni dal database.*

### 2.1 Introduzione ad Oracle Obiee

Oracle Business Intelligence Enterprise Edition nasce in origine come un prodotto chiamato *nQuire* acquistato dall'azienda Siebel nel 2002. Tale società fu poi acquisita da Oracle nel 2005 che quasi lasciò inalterata la piattaforma di analisi già sviluppata dalla Siebel.

Oracle Obiee Figura 2.1, così comunemente abbreviato, è una soluzione completa ed integrata per tutte le esigenze di business aziendali. Tale software è costituito da strumenti quali un server di query scalabile ed efficiente, uno strumento di query ed analisi ad hoc, dashboard interattive e un motore per report aziendali.



**Figura 2.1:** Logo Oracle Obiee

Oracle Business Intelligence ha la possibilità di acquisire le informazioni da qualsiasi sorgente offrendo l'opportunità di effettuare elaborazioni analitiche online (OLAP) e di presentare tali elaborazioni attraverso diverse interfacce, quali interfacce web-based, dispositivi mobili, portali standard e la suite di Microsoft Office.

### 2.2 Descrizione del DWH di partenza

Nel caso in analisi, la piattaforma di Business Intelligence Oracle Obiee utilizza come sorgente dati un Data Warehouse (Oracle). Tale tecnologia è un tipo di sistema di data

management progettato per supportare le attività di Business Intelligence, in particolare gli analytics. I Data Warehouse servono ad eseguire query ed analisi e spesso contengono grandi quantità di dati storici provenienti da diverse fonti. Grazie a queste funzionalità un Data Warehouse può essere considerato una fonte unica, centralizzata ed affidabile per i dati aziendali.

L'integrazione dei dati da diverse sorgenti e la gestione stessa del Data Warehouse avvengono grazie allo strumento software Oracle Data Integrator (ODI). Tale applicativo offre una soluzione completamente unificata per la gestione di Data Warehouse complessi, combinando elementi di movimento, sincronizzazione e qualità dei dati.

In sintesi, con la piattaforma di Business Intelligence si riescono ad effettuare analisi efficaci poichè alla base si ha una struttura capace di gestire la moltitudine di dati eterogenei e che garantisce che le informazioni siano tempestive, di qualità e coerenti tra sistemi complessi.

## 2.3 Definizione della home page

Il punto di partenza per l'esecuzione dei task in Obiee è rappresentato dalla home page Figura 2.2. La home page è suddivisa in sezioni che consentono di avviare task o individuare oggetti. Essa include, inoltre, sezioni come Recenti o Preferiti, che permettono di accedere ad oggetti, analisi o visualizzazioni recentemente aggiornate o più utilizzate dagli utenti.

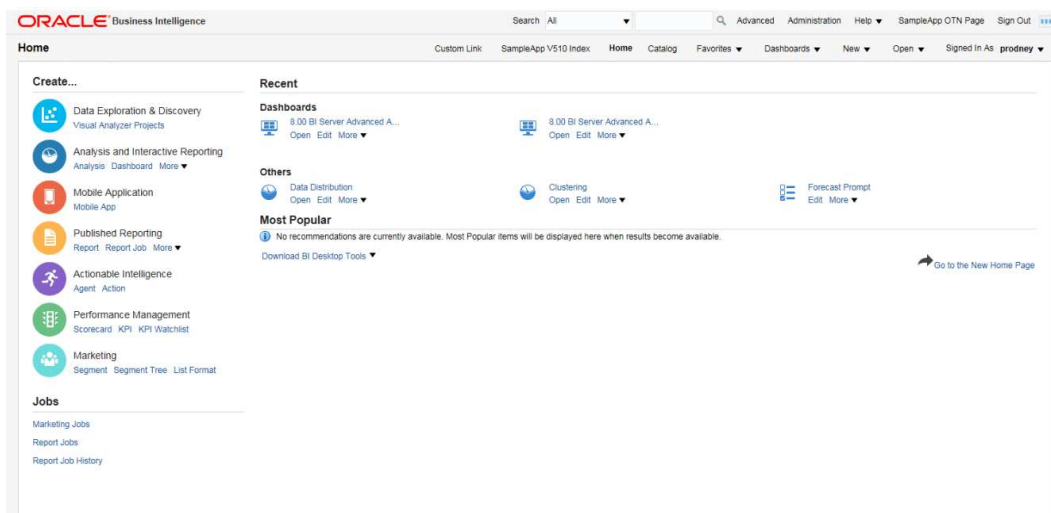


Figura 2.2: Home page Oracle Obiee

Nella colonna di sinistra viene mostrata la sezione *Crea*, contenente tutte le opzioni che l'utente ha a disposizione per le analisi. Nello specifico, tale sezione permette l'accesso a:

- *Data Exploration & Discovery*: permette di utilizzare la feature *Visual Analyzer*, che consistono in uno strumento rapido nell'esplorazione dei dati che permette all'utente di creare dashboard ricche di informazioni in tempi molto brevi.
- *Analysis and Interactive Reporting*: consente la creazione di analisi e report interattivi ovvero contenenti dei link che permettono la navigazione diretta tra un report e l'altro configurando delle action sulle colonne o su altri elementi del report.
- *Mobile Application*: viene utilizzata per la creazione di applicazioni di reportistica per dispositivi mobili.
- *Published Reporting*: permette di accedere al reporting pubblicato.

- *Actionable Intelligence*: consente di utilizzare elementi come le *action* per navigare verso contenuti correlati o, per richiamare operazioni.
- *Performance Management*: questa sezione contiene elementi quali scorecard, KPI e KPI Watchlist che permettono di creare una serie di indicatori per monitorare le metriche più importanti. Tali strumenti permettono di capire con quale efficacia l'azienda raggiunge gli obiettivi di business.
- *Marketing*: presenta strumenti di analisi specifici per il marketing

## 2.4 Definizione dell'editor di analisi

L'editor di analisi consente di interagire con le informazioni rappresentando i dati visivamente attraverso tabelle, grafici e tabelle pivot. Le viste che si creano possono essere, poi, raccolte e visualizzate nei dashboard. L'editor di analisi contiene una serie di schede tramite le quali possono essere eseguite determinate operazioni. Tali schede sono le seguenti:

- *Scheda criteri*: consente di specificare i criteri per un'analisi, inclusi filtri e colonne. Permette di specificare l'ordine di visualizzazione dei risultati, la formattazione di numeri, stili e colori. Consente di aggiungere formule alle colonne come funzioni di classificazione o percentili.
- *Scheda Risultati*: una volta ottenuti i risultati dall'analisi delle informazioni, è possibile creare le varie viste come grafici, tabelle o KPI.
- *Scheda Prompt*: consente di creare un prompt all'utente così da selezionare i valori per filtrare le analisi contenute all'interno di una dashboard. In questo modo l'utente può filtrare dinamicamente tutte le viste di una o più analisi dello stesso report.
- *Scheda Avanzate*: consente di modificare il codice XML e di esaminare l'istruzione SQL utilizzata per la generazione dell'analisi. È possibile riutilizzare l'istruzione SQL come base per la creazione di nuove analisi.

## 2.5 Creazione di analisi

Lo sviluppo di un'analisi consiste nell'estrazione dei dati e nella creazione delle visualizzazioni che saranno aggiunte ai dashboard. Tale processo ha inizio con la definizione di una query per la selezione dei dati, questa può essere di due tipi in base ai privilegi di cui dispone l'utente. È possibile, infatti, eseguire delle query dirette verso i database di back-end fisici o impostare l'analisi tramite un'istruzione SQL semplice i cui dati verranno, poi, manipolati nell'editor di analisi. Nel seguito tali istruzioni saranno trattate più specificatamente. Una volta recuperate le informazioni, è necessario utilizzare la scheda criteri (Figura 2.3) dell'editor di analisi per specificare i criteri di analisi tra cui colonne, filtri e formule. Nella colonna di sinistra, *Subject Area* della Figura 2.3 vengono mostrate le tabelle delle dimensioni e dei fatti, con i relativi campi ottenuti dall'esecuzione della query SQL. Nella parte centrale, invece, è presente un pannello nel quale vengono trascinati i campi in modo da selezionare le colonne d'interesse per lo sviluppo dell'analisi. Per le specifiche colonne è possibile eseguire una serie di operazioni quali selezione dell'ordine di visualizzazione e formattazione del contenuto. L'editor, inoltre, mette a disposizione una serie di funzioni (di aggregazione, per elaborazione stringhe, etc.) per applicare formule ai valori delle colonne. Per quanto riguarda la formattazione delle colonne si passa dalle funzioni di modifica dello stile fino alla creazione di istruzioni condizionali ed interattive. Nella scheda *Filters* saranno inseriti, invece, i valori che l'utente avrà a disposizione per effettuare il filtraggio delle colonne.



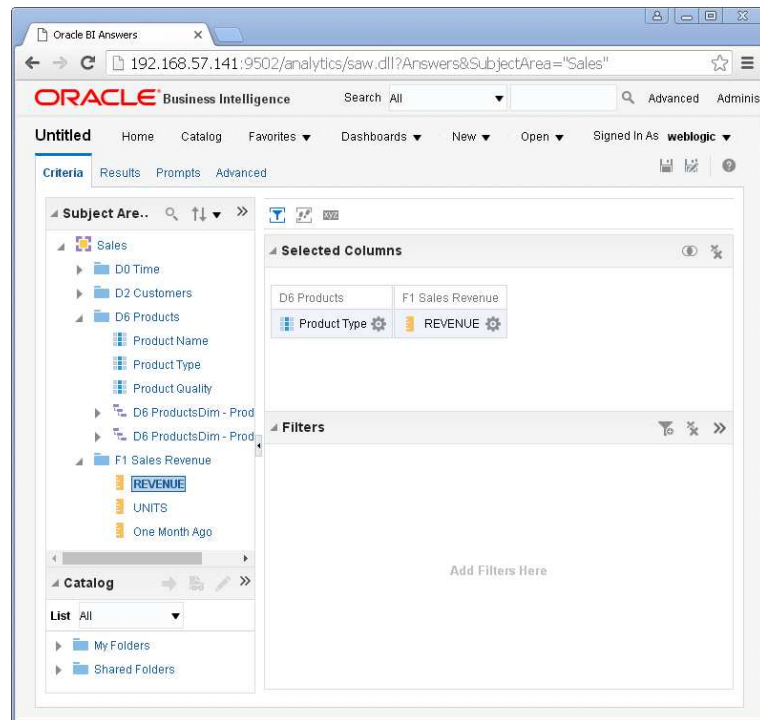


Figura 2.3: Scheda Criteri Oracle Obiee

### 2.5.1 Istruzioni SQL e utilizzo delle richieste dirette al database

Come già anticipato nella sezione precedente, la creazione di un'analisi in Obiee avviene mediante l'utilizzo di istruzioni SQL logiche. Tale tipo di SQL viene interpretato dal *BI Server*. Oracle Bi Server è un software che può integrare dati da molteplici risorse e agisce traducendo la query logica creata dall'utente in una o più query fisiche verso i diversi database fisici. Nella Figura 2.4 viene riportato tutto il processo affinché una query SQL possa essere interpretata dalle diverse sorgenti di dati.

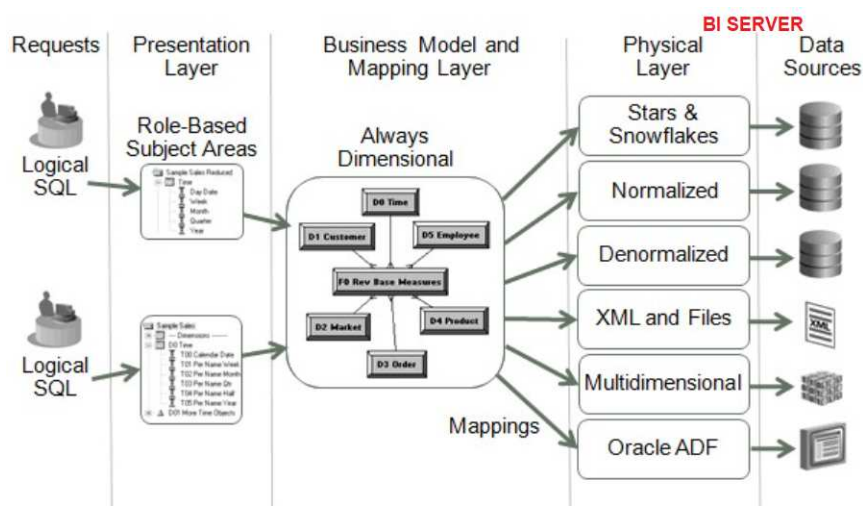


Figura 2.4: Query processing

Una query SQL in Obiee non differisce da una generica query se non per la possibilità di aggiungere funzioni specifiche per filtraggio, aggregazione e creazione delle formule per le

analisi nei report. Le istruzioni di cui si è discusso fin'ora non sono l'unico metodo con cui un utente può interfacciarsi con il database. Gli utenti con privilegi adeguati possono creare ed inviare richieste direttamente ai database di back-end fisici. Questi privilegi vengono definiti dagli amministratori lato presentation, nello specifico nella sezione Oracle BI Presentation Services Administration. Le richieste dirette al database hanno il vantaggio che esulano dalle impostazioni di sicurezza di BI Server e vengono utilizzate quando il requisito del report non può essere rappresentato in un modello dati o quando la progettazione di quest'ultimo è molto onerosa.

## 2.5.2 Visualizzazione dei risultati

I risultati delle analisi rappresentano l'output restituito da Oracle Bi Server corrispondente ai criteri d'analisi. Essi possono essere visualizzati tramite varie viste, ad esempio una tabella, un grafico o un indicatore. Gli utenti possono esaminare ed analizzare i risultati nel report Obiee oppure decidere di scaricare il dato come PDF o, più comunemente, all'interno di fogli di calcolo. L'insieme delle viste e degli oggetti relativi ad un'analisi vengono raccolti all'interno dei dashboard. Il contenuto viene visualizzato in una o più pagine del dashboard identificate dalle schede disposte nella parte superiore. Tra le schede si trova la barra degli strumenti che contiene i pulsanti che permettono di effettuare modifiche se si dispone delle autorizzazioni o, più comunemente, applicare filtri dinamici al dashboard. L'aggiunta di contenuto in un dashboard consente agli utenti di visualizzare vari tipi di dati organizzati in un unico punto. Gli oggetti che possono essere inclusi si dividono in:

- *Oggetti dashboard*: colonne, sezioni, testo, cartelle, nonché oggetti come grafici, KPI, così via.
- *Oggetti salvati dall'utente*: sono oggetti che vengono salvati dall'utente o da altri utenti all'interno di Oracle Obiee Presentatuon Catalog; tali oggetti possono essere analisi, prompt o specifici grafici precedentemente creati.

Nella Figura 2.5 viene riportato un esempio di dashboard Obiee contenente diversi grafici e filtri.

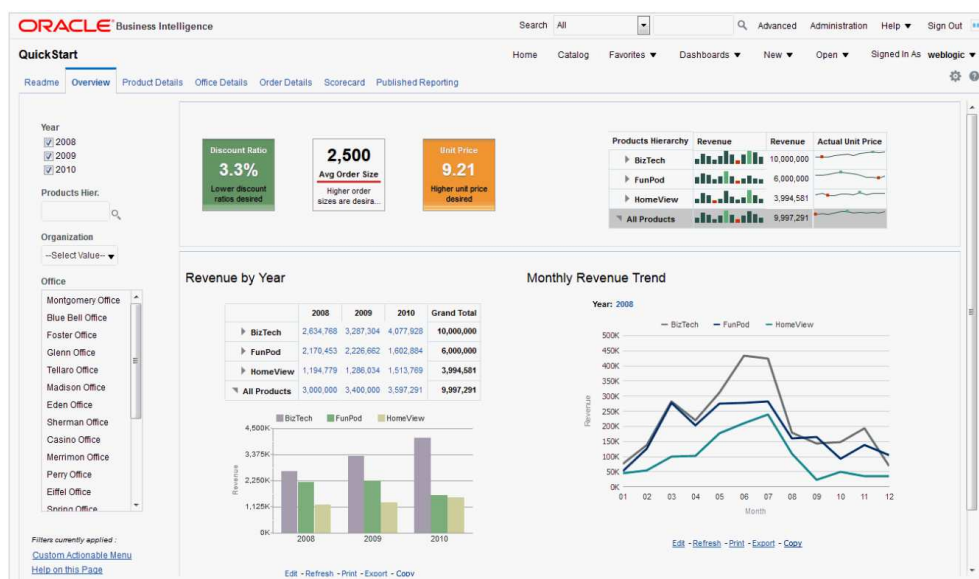


Figura 2.5: Un esempio di dashboard Obiee

## 2.6 Descrizione di Oracle BI Presentation Catalog

Le analisi, le dashboard e le visualizzazioni vengono poi salvate all'interno di Oracle BI Presentation Catalog (Figura 2.6). Il catalogo ha la funzione di memorizzare gli oggetti di Business intelligence e fornire un'interfaccia in cui gli utenti possono creare, accedere e gestire gli oggetti salvati. Tali oggetti sono organizzati in cartelle condivise o personali.

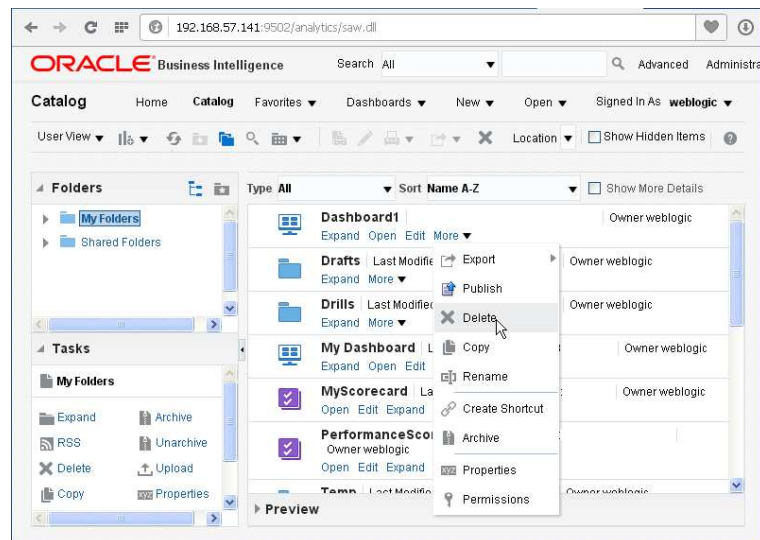


Figura 2.6: Oracle BI Presentation Catalog

Al catalogo possono accedere tre diverse categorie di utenti; pertanto, ciascun utente visualizza una versione diversa dell'interfaccia del catalogo. L'utente con meno privilegi è il consumatore di contenuti il quale può accedere solo a determinati oggetti in base alle analisi di suo interesse e può eseguire solo determinati task definiti dagli utenti di livello superiore, designer di contenuti e amministratore. I designer di contenuti hanno un accesso molto più ampio al catalogo poichè creano, modificano e risolvono i problemi degli oggetti. Questi privilegi vengono determinati, infine, dall'utente di livello maggiore che è l'amministratore, egli si occupa prevalentemente di task di gestione del catalogo. Nella parte in basso a sinistra della Figura 2.6 è definita la sezione *Tasks*, dove vengono specificate tutte le azioni che possono essere eseguite per un oggetto selezionato dal catalogo. Infine, per trarre massimo vantaggio dalle analisi, è possibile estrarre gli oggetti del catalogo quali dashboard e KPI in diversi formati come PDF o foglio di calcolo; tale operazione si esegue tramite un apposito task di esportazione.

## 2.7 Vantaggi e Svantaggi

Oracle Business Intelligence Enterprise Edition è un tool di analitica che offre molte funzionalità e che rende semplice l'analisi dei dati in molteplici scenari. I vantaggi offerti da questa piattaforma sono diversi. Obiee è una soluzione estremamente scalabile, si parte da una suite base su cui poter aggiungere moduli che amplificano le possibilità di analisi. È possibile connettersi a sorgenti di dati eterogenee sia di tipo strutturato che non. L'interfaccia e il processo di creazione delle analisi sono estremamente user friendly e strutturati; ciò permette la creazione di report interattivi e ad hoc per le esigenze di business del cliente. L'utente finale può accedere alla reportistica con pochi click e, grazie alle funzionalità di esportazione delle analisi, si semplifica il processo di condivisione dei risultati. Tuttavia Obiee presenta anche una serie di punti deboli tali per cui, ad oggi, non viene più considerato

come uno dei software leader a livello mondiale. Le operazioni che il cliente può eseguire sul report sono poco flessibili e limitate a quelle che lo sviluppatore ha definito, soluzioni di BI come Qlik Sense permettono la creazione di oggetti preconfezionati che rendono la modifica meno tecnica e più dinamica per l'utente finale. La scalabilità, analizzata come punto di forza, ha anche un aspetto negativo poiché il software ha un costo elevato, e più moduli si installano più il costo sale. Aggiungendo moduli aumentano anche le possibilità di analisi ma, allo stesso tempo, aumenta la complessità dell'architettura e quindi si ha un degrado delle performance. Infine, confrontando Obiee, con i moderni tool di BI si riscontrano similarità nelle funzionalità a discapito dell'aspetto datato e poco organizzato dei grafici.

*Nel presente capitolo verrà descritto il processo di raccolta ed analisi dei requisiti. L'analisi inizierà introducendo il processo e la figura aziendale preposta ad esso. In particolare, si vedrà quali sono le logiche di sviluppo back-end e front-end. Si procederà, inoltre, alla creazione della relativa documentazione sotto forma di fogli di Reverse e Business Glossary.*

### 3.1 Introduzione all'analisi dei requisiti

La raccolta e l'analisi dei requisiti è uno dei processi di pianificazione finalizzato all'acquisizione di tutte le informazioni necessarie a configurare in maniera efficiente i deliverable del progetto e la loro produzione. Questo processo consiste nell'identificazione delle diverse esigenze e delle aspettative degli stakeholder verso un determinato progetto. Per quanto detto finora, a valle di questa analisi, è necessario eseguire un'accurata fase di individuazione e gestione degli stakeholder, ovvero le figure economicamente coinvolte nel progetto. Questa raccolta può avvenire attraverso l'utilizzo di diverse tecniche quali:

- *Interviste one to one*: si tratta di incontri faccia a faccia con il cliente per la raccolta di requisiti ed aspetti funzionali. In questo tipo di incontri è importante che lo stakeholder esprima ogni tipo di valutazione concentrandosi sul perimetro di analisi che viene definito per ogni incontro.
- *Focus group*: questa modalità prevede un incontro con un gruppo di stakeholder per confrontare idee e selezionare le opzioni migliori.
- *Indagini*: la raccolta avviene mediante questionari che vengono inviati agli stakeholder i quali rispondono autonomamente. Una volta ottenute le risposte si valutano i risultati

Per quanto riguarda il caso di studio di questa trattazione, le figure coinvolte in questo processo sono gli analisti funzionali, per la raccolta dei requisiti, e il settore IT e Business lato cliente, come stakeholder. Gli analisti funzionali sono figure responsabili dell'analisi dei processi all'interno di un'azienda, con l'obiettivo di soddisfare le esigenze del cliente. Questa figura funge da collegamento tra gli utenti e il team di sviluppo. Oltre a forti competenze di comunicazione e gestione delle persone, l'analista funzionale deve possedere anche buone conoscenze tecniche, così da comprendere la fattibilità delle richieste del cliente. Dall'altra parte, invece, il settore IT è tutta quell'area che si occupa della corretta evoluzione del progetto di reingegnerizzazione della piattaforma di BI. Il business, infine, rappresenta tutti gli utenti finali che utilizzeranno le nuove applicazioni di analisi sviluppate su Qlik Sense.

## 3.2 Raccolta delle logiche di sviluppo del Back-end

In un progetto di reingegnerizzazione di una piattaforma di business intelligence, la prima fase di raccolta dei requisiti consiste nell'individuare le logiche di back-end che sono alla base della vecchia piattaforma. Come già anticipato nel precedente capitolo, Oracle Obiee è alimentato da un data warehouse in cui tutte le tabelle sono strutturate seguendo le logiche di tale sorgente dati. L'obiettivo è quello di portare tutti i dati e tutte le tabelle verso una nuova piattaforma di dati più efficiente e moderna. Questa operazione prevede la raccolta dettagliata delle logiche implementate lato DWH così da poter eseguire, poi, un mapping verso le logiche della piattaforma mantenendo il contenuto informativo originale. Questo tipo di raccolta e analisi avviene sia per tabelle già esistenti su DWH sia per nuovi sviluppi di tabelle che avverranno solo nella piattaforma dati. Le tabelle, ed in generale, le analisi sono suddivise per *subject area* e organizzate in diversi *stream*. Esempi di *subject area* sono lo stock dei componenti, le vendite per un certo tipo di mercato, i dati provenienti dalle recensioni dei clienti, etc. Gli *stream*, invece, sono le macro-aree di produzione ed analisi aziendali come, nel caso in analisi, la *Supply Chain* o *International Transport*. L'analista funzionale, sia per il caso di tabelle già esistenti sia per i nuovi sviluppi, ha il compito di comprendere con il cliente il perimetro dei dati in analisi per ogni *subject area*, quali tabelle devono essere mappate secondo le nuove logiche e come questa operazione deve essere eseguita. Tutta questa serie di informazioni è di fondamentale importanza per gli sviluppatori back-end poichè permette la creazione e la migrazione dei dati in accordo con le esigenze del cliente.

### 3.2.1 Inbound vs Outbound

Come specificato nella sezione precedente l'analista funzionale ha il compito di comprendere per ogni *stream* quale sia il perimetro di analisi dei dati e quali sono le tabelle che ne fanno parte. Nel progetto in analisi è stata effettuata una suddivisione in due macro-*stream* di tutte le tabelle. Tali strutture dati vengono separate in *Inbound* e *Outbound*. La parte di *Inbound* comprende tutte le tabelle per cui deve essere effettuato un processo di mapping e traduzione dal data warehouse verso la piattaforma dati. Questa categoria è quella più prioritaria nella quale si concentrano maggiormente le risorse poichè comprende tutte le tabelle, e quindi tutte le informazioni che sono necessarie al fine del completamento del progetto di reingegnerizzazione della piattaforma di BI. La parte di *Outbound*, invece, considera tutte le tabelle per cui viene effettuata una migrazione dalla piattaforma dei dati verso altri sistemi detti sistemi riceventi. Questo processo è stato richiesto dal cliente poichè certe informazioni sono necessarie ad altri settori aziendali al di fuori del business secondo logiche diverse da quelle che caratterizzeranno la piattaforma dei dati. La parte di *outbound*, ovviamente, ha priorità minore rispetto all'*inbound* poichè esula dall'obiettivo finale del progetto e richiede, anche, un onere maggiore, vista la necessità di comprendere, per ogni processo di mapping, le logiche del sistema ricevente.

## 3.3 Creazione della documentazione di Reverse

La raccolta delle logiche implementate nel data warehouse attraverso interviste one to one con il settore IT del cliente ha come fine la creazione della documentazione di reverse. Questo tipo di documento è di fondamentale importanza per gli sviluppatori di back-end poichè funge da linea guida per la corretta implementazione delle tabelle e delle informazioni nella nuova piattaforma dati. I documenti di reverse (Tabella 3.1) devono essere approvati da entrambi i lati, ovvero dal cliente e dall'azienda di consulenza, poichè solo così si riesce a

Nome Tabella DWH						
dw.table_name						
Filter	Eventuali filtri presenti					
Mapping Logics	Tabella formata dall'unione di due query					
Modalità di inserimento dati	Insert					
Column Name	Primary Key	Source Table	Source Column	Datatype	Destination Table	Destination Column
date_id		dw.tabella_sorgente	date_id union dw.tabella_sorgente2	Number	nome_tabella_DP	nome_colonna_DP

**Tabella 3.1:** Esempio documento di Reverse

portare avanti il progetto di reingegnerizzazione evitando possibili nuove richieste del cliente quando si è già in corso di sviluppo.

Questo concetto ribadisce l'importanza che hanno nel processo l'analista funzionale e le conoscenze tecniche che gli permettono, a priori, di stabilire cosa è fattibile e cosa no. I documenti di reverse, in genere, sono strutturati sotto forma tabellare utilizzando i fogli di calcolo. Per ogni subject area vengono definite tutte le tabelle lato DWH che rappresentano il perimetro dati da analizzare. Viene fornita una descrizione dei campi di ciascuna tabella, a partire dal tipo fino alla descrizione generale. Per ciascuna di queste tabelle viene riportato, infine, come sarà mappata nella piattaforma dati. Questo tipo di operazione non è, banalmente, un mapping uno a uno tra DWH e data platform poiché è nell'interesse del cliente aggiungere valore nel progetto di reingegnerizzazione. La differenza nelle logiche implementative della piattaforma dati e le nuove richieste del cliente portano a modifiche come la separazione di una tabella del DWH in due tabelle della piattaforma dati, la concatenazione di due campi del DWH in un unico campo della piattaforma dati o, infine, l'aggiunta di nuove tabelle e nuovi campi che non esistevano nel DWH. Tale processo porterà poi lo sviluppatore back-end all'implementazione dell'area.

## 3.4 Raccolta logiche di sviluppo Front-end

La fase di raccolta dei requisiti lato front-end avviene al termine del processo di sviluppo ed implementazione delle tabelle lato back-end. Questo processo viene eseguito in parte dall'analista funzionale e in parte dallo sviluppatore front-end. L'analista funzionale, in prima battuta, si occupa di reperire le informazioni relative ai report implementati tramite Obiee. Tali informazioni saranno, poi, descritte all'interno di una documentazione apposita che fungerà da punto di partenza per lo sviluppatore. Anche in questo caso, l'analista non solo ha accesso diretto alla reportistica di Obiee ma si occupa di organizzare incontri one to one con il cliente al fine di ottenere un quadro completo del report allo stato attuale ed eventuali nuove aggiunte. Lo sviluppatore, in questa fase, ha il compito di analizzare in maniera tecnica le dimensioni, e soprattutto le misure (KPI) chiave, del report per valutarne la fattibilità e la chiarezza con il cliente. Ottenute tutte le informazioni necessarie per lo sviluppo dei report, la fase di raccolta dei requisiti non termina qui. Al rilascio dell'applicazione seguono infatti, una serie di test insieme agli utenti di business da cui possono emergere nuove richieste per migliorare l'esperienza di utilizzo.

### 3.4.1 Creazione documenti Business Glossary

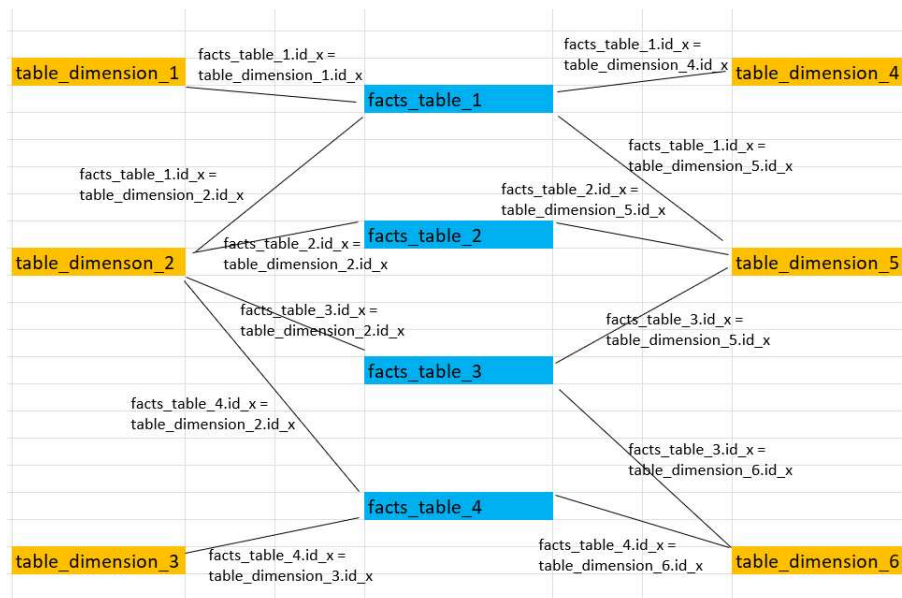
Nella sezione precedente si è discusso del ruolo dell'analista funzionale nella raccolta dei requisiti per lo sviluppo del front-end. Tutte le informazioni necessarie allo sviluppo della reportistica vengono racchiuse all'interno di un documento, in genere un foglio di calcolo, chiamato Business Glossary. Quest'ultimo è un documento in cui lo sviluppatore trova tutti dati necessari per lo sviluppo di nuove applicazioni Qlik Sense. La prima sezione del documento contiene il vero e proprio business glossary (Tabella 3.2) ovvero tutte le

informazioni relative al perimetro di analisi dei dati per la subject area dell'app da sviluppare. Tale perimetro è descritto attraverso l'utilizzo di una tabella dove vengono riportati tutti gli attributi delle tabelle interessate lato DWH e il relativo mapping lato piattaforma dati.

Subject Area	Nome Attributo Obiee	Campo DWH	Description	Tabella DWH	Campo DP	Tabella DP	Join Link
SupplyChain	Region	REGION_DESCRIPTION	Europe, Apac, etc.	dw.nome_tabella_1	des_region	edp_l1.nome_tabella_1	tabella1.id_product = tabella4.id_product
SupplyChain	Area	DES_AREA	Description of geographic area	dw.nome_tabella_2	area	edp_l1.nome_tabella_2	
SupplyChain	Product	PRODUCT_CODE	Description of cod_product	dw.nome_tabella_3	cod_product	edp_l1.nome_tabella_3	

**Tabella 3.2:** Definizione del mapping dal DWH alla piattaforma dati

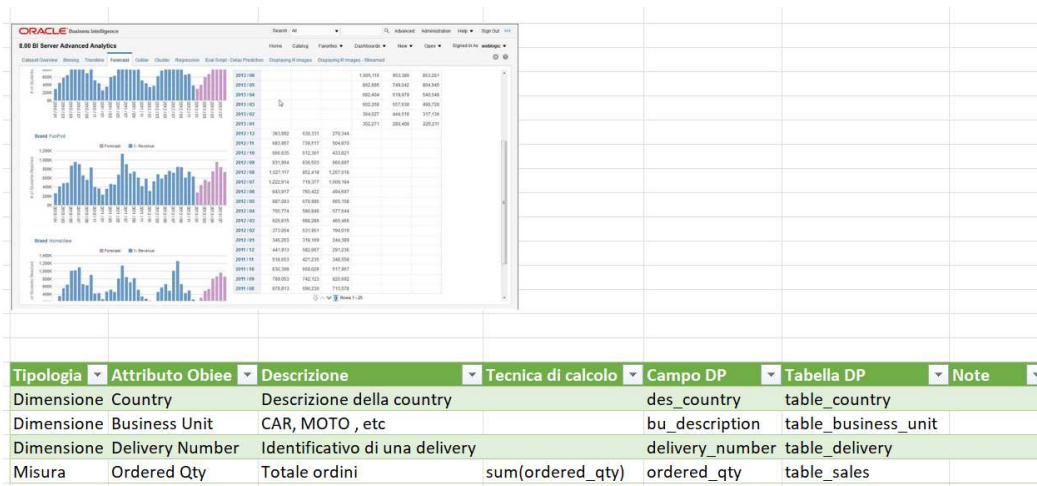
Sviluppare una nuova applicazione Qlik Sense sulla base di report già esistenti in Obiee significa avere piena conoscenza del modello dei dati che sta alla base dell'app in modo da riportare correttamente i dati. Per questo motivo, all'interno del Business Glossary, viene inserita anche la nuvola dati. Quest'ultima (Figura 3.1) è un diagramma contenente tutte le tabelle dei fatti e tutte le tabelle delle dimensioni necessarie alla creazione del modello dei dati. I fatti vengono posti centralmente e, per ciascuno di essi, vengono inserite le join con le dimensioni specificandone la chiave.



**Figura 3.1:** Esempio di nuvola dei dati

All'interno del documento vengono, poi, creati una serie di sheet contenenti gli screen delle diverse dashboard dei report Obiee. Negli stessi fogli, l'analista funzionale riporta con esattezza i campi da utilizzare come dimensioni e le tecniche di calcolo che permettono la creazione delle misure e dei KPI. Questo tipo di fogli (Figura 3.2), insieme alla nuvola dati sono le informazioni più critiche e necessarie ai fini dello sviluppo e, per questo, è fondamentale che tali specifiche siano chiare e sottoposte ad un rigoroso processo di validazione. La principale causa nei ritardi delle consegne sta proprio nell'incomprensione di questi requisiti iniziali.





**Figura 3.2:** Esempio descrizione di un report Obiee

Per completezza, infine, vengono riportate anche le query SQL utilizzate direttamente su Obiee per la creazione dei report così che lo sviluppatore abbia un ulteriore punto di riferimento nel caso ci siano dubbi nell'interpretazione delle dimensioni o delle misure.

---

## Progettazione delle attività di re-engineering

---

*In questo capitolo verranno descritte le fasi che scandiscono il ciclo di vita del progetto: fase di avvio, pianificazione, esecuzione ed infine valutazione e chiusura.*

*Verrà approfondita la metodologia di sviluppo del software Agile applicata all'interno dell'azienda di consulenza.*

*Nella descrizione della progettazione delle attività sarà posta particolare attenzione alla parte relativa al front-end, caso di studio della trattazione.*

### 4.1 Definizione della metodologia Agile

Nell'ambito della realizzazione di un progetto, o nello specifico di un software, diversi sono i metodi di sviluppo che possono essere applicati. Il più tradizionale e storicamente utilizzato è quello a *cascata*. Tale metodologia prevede che il processo di realizzazione di un software sia strutturato su più fasi sequenziali che, in generale, sono:

- *Analisi dei requisiti;*
- *Progettazione;*
- *Sviluppo;*
- *Collaudo;*
- *Manutenzione;*

Questo modello prevede che, per passare alla fase successiva, quella precedente sia completamente terminata. L'idea alla base è che i progetti siano lineari e ben definiti, caratteristiche che permettono di predisporre un piano ben dettagliato. Con il tempo, però, i progetti sono diventati sempre più grandi e difficili da gestire, con un elevato numero di task da eseguire, che comportano correzioni complesse. L'approccio rigido tradizionale, caratterizzato da un estremo formalismo e basato su task progressivi, non era più adatto a gestire la dinamicità dei nuovi progetti. In merito a ciò, è emersa la necessità di sviluppare nuovi approcci che permettessero di superare i limiti della metodologia a cascata. L'idea alla base della metodologia *Agile* è la risposta dinamica al cambiamento, per questo motivo è da considerarsi un approccio iterativo. Ogni iterazione (Figura 4.1), il più corta possibile, comprende tutte le fasi di progetto e concorre al raggiungimento dello scopo finale. Il metodo



**Figura 4.1:** Iterazione relativa alla metodologia Agile

iterativo aiuta nel velocizzare lo sviluppo del processo e permette di consegnare degli output di valore (*deliverable*) alla fine di ogni iterazione.

I valori su cui si basa tale metodologia sono:

- *Individui e interazioni più che processi e strumenti;*
- *Software funzionante più che la documentazione esaustiva;*
- *Collaborazione con il cliente più che negoziazione dei contratti;*
- *Rispondere al cambiamento più che seguire un piano.*

L'obiettivo finale è la piena soddisfazione del cliente; l'utilizzo di tale metodologia può consentire di abbattere costi e tempi di sviluppo aumentandone la qualità.

## 4.2 Fase di avvio

Il processo di inizio di un progetto Scrum è identico a quello di inizio di un progetto tradizionale in termini di pianificazione e assegnazione delle risorse. Tuttavia, ci sono alcune attività che devono essere portate a termine prima di iniziare il primo Sprint (breve periodo di tempo in cui lo scrum team lavora per completare un'attività). All'interno del team i manager ricoprono il ruolo di *product owner*, la figura che ha come obiettivo la massimizzazione del valore generato dal prodotto. Il product owner e lo scrum team (membri del team di sviluppo) collaborano insieme al cliente per determinare l'ordine con cui le diverse applicazioni relative alle subject area devono essere rilasciate. Una volta che ciò è stato definito viene creato il *Backlog* di prodotto. Quest'ultimo, in questo caso, è un elenco di tutte le app che devono essere sviluppate e consegnate. Ogni elemento di questo documento è un task.

## 4.3 Pianificazione delle attività

Nel contesto Agile la pianificazione è di fondamentale importanza. La differenza con il project management tradizionale sta nel fatto che non si perde tempo iniziale per elaborare

una pianificazione dettagliata su lungo periodo con la relativa documentazione. Nella metodologia Agile la pianificazione è continua, rapida e si evolve adattandosi ai cambiamenti. Come descritto nella sezione precedente, il risultato finale della fase di avvio è il Backlog di prodotto contenente tutti i task che devono essere svolti per i rilasci.

Nel progetto di re-engineering della piattaforma di Business Intelligence vengono definiti due task-list di questo tipo, una per il back end e una per il front end. La task-list, in generale, è un documento che ricopre un ruolo fondamentale nella pianificazione delle attività di sviluppo. All'interno di questo documento vengono riportate tutte le *subject area* da sviluppare, l'owner (membro del team di sviluppo a cui viene assegnata una certa area, gli open point e le scadenze. Questo file viene aggiornato periodicamente dagli analisti funzionali o dagli stessi membri del team di sviluppo. Molto importante è la sezione relativa agli *open point* dove vengono riportate le criticità che impediscono il corretto sviluppo dell'applicazione. I manager utilizzano la task-list per avere un quadro generale delle attività in corso e per intervenire dove si presentano criticità bloccanti per i rilasci. L'aggiornamento di tale documentazione avviene, prevalentemente, durante i *daily stand up meeting*.

Il daily stand up è una riunione di breve durata dove si discute sullo stato di avanzamento del progetto tra manager e membri del team di sviluppo. Tale attività viene pianificata due volte a settimana ed è molto importante poichè, dai riscontri e dagli open points emersi, si capisce se le scadenze verranno rispettate o c'è la necessità di inserire ulteriore supporto. Con la stessa periodicità, il manager e i consultant, partecipano ai SAL insieme ai clienti. Il SAL non è altro che una riunione stabilita per garantire e verificare l'avanzamento del progetto. A valle di un SAL, spesso, la pianificazione delle attività subisce variazioni proprio in relazione agli aggiornamenti e alle modifiche richieste dal cliente.

## 4.4 Fase di esecuzione

La fase di realizzazione del progetto può essere considerata come la fase più critica poichè è quella che coinvolge più risorse e, di conseguenza, l'impatto degli errori è più elevato. È proprio per questo che è opportuno che tutto il lavoro venga svolto a valle dell'accurata fase di pianificazione descritta nella sezione precedente. L'analisi della fase di realizzazione del progetto si concentrerà, per tutta la trattazione, sulla creazione delle applicazioni Qlik Sense, e quindi sullo sviluppo del front end.

Il software di Business Intelligence Qlik Sense è un applicativo che può operare come istanza in locale o in cloud. L'azienda cliente, in questo caso, mette a disposizione degli sviluppatori un'istanza Qlik che gira sopra a dei server di proprietà aziendale. Tutte le applicazioni, in questo modo, sono a disposizione del settore IT, degli utenti del business e degli sviluppatori del progetto. Ad ogni risorsa vengono fornite delle credenziali che permettono di accedere ai diversi ambienti necessari per lo sviluppo delle applicazioni. Le applicazioni, nella maggior parte dei casi, mostreranno i dati aggiornati giorno per giorno.

Per il cliente è molto importante, non solo che i dati riportati siano corretti, ma anche che i ricarichi avvengano negli orari opportuni in base alle esigenze lavorative. Per tutte queste motivazioni, nella fase di realizzazione, si fa particolare attenzione alle performance delle singole applicazioni. È compito dei manager fornire indicazioni tecniche e fare in modo che gli sviluppatori prestino attenzione a questi dettagli durante lo sviluppo.

Durante la fase di esecuzione del front end è, inoltre, necessaria una forte cooperazione con le risorse del back end perchè spesso solo in questa fase ci si accorge di eventuali errori che sono stati fatti durante la creazione delle strutture. Nella prossima sezione saranno descritti in maniera dettagliata tutti gli step che uno sviluppatore front end segue per il corretto rilascio di un'applicazione Qlik Sense.

#### 4.4.1 Milestone di sviluppo front end

Il processo di sviluppo di un'applicazione Qlik Sense si compone in una serie di step che devono essere seguiti in maniera rigorosa dallo sviluppatore. Questi step possono essere visti come delle *milestones*, ovvero, eventi che indicano il raggiungimento di obiettivi intermedi durante la fase di realizzazione.

Quando si parla di applicazioni di reportistica che riguardano il mondo dei dati, generalmente, le aziende sviluppano e testano questi prodotti lavorando su più ambienti. Lo sviluppatore, per prima cosa, crea e testa l'applicazione lavorando nell'ambiente di *quality*. Un ambiente può essere considerato come un insieme di impostazioni in cui i software vengono testati, sviluppati o effettivamente utilizzati dagli utenti finali.

L'ambiente di *quality* è una impostazione dove le tabelle dei fatti e delle dimensioni non sono soggette ad aggiornamenti periodici e, quindi, lo sviluppatore crea e testa l'applicazione considerando un intervallo temporale fisso di analisi. In questo ambiente, inoltre, chi sviluppa è libero di eseguire test e modifiche senza intaccare le analisi degli utenti del business.

Una volta che si raggiunge un buon livello di correttezza delle informazioni, l'applicazione viene rilasciata all'interno di appositi streams a cui hanno accesso gli utenti del settore IT del cliente. Quest'ultimi hanno il compito di verificare e validare la correttezza dei dati riportati all'interno dell'applicazione e segnalare eventuali anomalie da correggere.

Terminato il confronto con il settore IT, l'app può considerarsi pronta per la successiva fase di rilascio verso gli utenti finali ovvero coloro che la utilizzeranno quotidianamente per lo sviluppo delle analisi di business. Prima che ciò avvenga, l'applicazione viene portata in ambiente di produzione dove funzionerà a regime e non potrà più essere modificata. Per il rilascio agli utenti vengono programmati una serie di meeting in cui lo sviluppatore (insieme a dei responsabili del reparto IT del cliente) svolge sessioni di training fornendo un'introduzione generale su Qlik Sense e mostrando tutte le funzionalità dell'applicazione.

### 4.5 Valutazione e chiusura

La valutazione di un'applicazione e la successiva dismissione di una subject area sono processi piuttosto complessi che dipendono da molti fattori. Prima del rilascio, in genere, si svolgono una serie di test denominati *integration test* dove si cerca di riprodurre le misure del front end direttamente su database. Questo tipo di test viene eseguito per verificare se, durante il processo di ETL, alcune informazioni si sono perse a causa di possibili errori nella creazione del modello di dati o nei collegamenti tra tabelle. Una volta che l'*integration test* si chiude con esito positivo si è abbastanza confidenti sulla bontà dell'applicazione.

La seconda fase di valutazione spetta al cliente che confronta i dati dell'applicazione Qlik Sense con i report della vecchia piattaforma di BI Oracle Obiee. In un confronto di questo tipo si possono trovare molte differenze poichè durante il processo di migrazione tra il DWH e la piattaforma dati il contenuto informativo ha subito diverse modifiche. In generale, però, i dati non devono avere un delta molto grande con quanto precedentemente riportato.

Terminate le analisi da parte del cliente possono emergere delle segnalazioni che, a stretto giro, vengono corrette dagli sviluppatori. A questo punto l'applicazione soddisfa i requisiti del cliente e possono essere avviate le procedure di dismissione dell'area.

Dopo un certo intervallo temporale da quando l'applicazione è stata rilasciata e ha funzionato a regime, si prepara una documentazione generale sul suo funzionamento e sulla struttura. Questa documentazione, e la stessa applicazione, passeranno in carico all'AMS, acronimo di *Application Managemnet Services*, una specifica area aziendale che si occupa della manutenzione sia delle strutture di back end sia delle applicazioni di front end rilasciate in ambiente di produzione.

---

## Implementazione delle attività di re-engineering

---

*Nel presente capitolo verranno descritte tutte le fasi del processo di sviluppo di un'applicazione Qlik Sense. Verrà sottolineata la differenza tra creazione e rilascio di nuove applicazioni e traduzione di applicazioni definite "dualrun". Infine, verranno descritti la fase di validazione dell'applicazione con il cliente e i successivi training con gli utenti del business.*

### 5.1 Sviluppo nuove applicazioni

Come visto finora, il processo di re-engineering della piattaforma di Business Intelligence consiste nella migrazione dei dati dal data warehouse alla nuova piattaforma dati e nello sviluppo delle applicazioni di reportistica Qlik Sense che sostituiranno le app Obiee. Il cliente, prima dell'avvio del progetto, aveva già sviluppato alcune app in Qlik Sense con i dati che, però, puntavano al vecchio DWH. Quando si parla di nuovi sviluppi si intende la creazione di applicazioni relative ad una determinata subject area che esistono, attualmente, solo nella vecchia piattaforma di Business Intelligence, ovvero Obiee. Questo tipo di app devono essere implementate a partire dalla creazione del modello dati, cioè dall'individuazione delle tabelle dei fatti e delle dimensioni e delle loro connessioni. Tali informazioni vengono ricavate durante la fase di reverse e creazione della documentazione (ad esempio Business Glossary) con il cliente. L'unico metro di confronto che si ha a disposizione è l'applicazione Obiee con la quale si paragonano i dati e si cerca di riprodurre in maniera più fedele possibile la parte di visualizzazione, dei grafici e report.

#### 5.1.1 Generazione dei QVD per tabelle dei fatti e dimensioni

A monte degli sviluppi di un'applicazione, come descritto nei precedenti capitoli, viene creata la nuvola dati contenente le tabelle dei fatti e le relative join con le tabelle delle dimensioni. La creazione del modello dei dati all'interno di un'applicazione Qlik Sense avviene nel *Data Load Editor*. Tale sezione permette il caricamento dei dati utilizzando linguaggio SQL, ovvero query dirette verso il database. Si caricano i campi necessari all'analisi, si definiscono le join, e Qlik, in base ai collegamenti tra i campi e le join stesse, crea il modello dei dati. Nel caso di applicazioni che recuperano dati da tabelle di dimensioni medie, l'utilizzo di query dirette al database risulta essere la soluzione più efficiente e veloce. Nonostante ciò, la maggior parte delle applicazioni si basa su tabelle dei fatti e dimensioni contenenti decine e centinaia di milioni di record. In queste situazioni, l'utilizzo di query dirette sarebbe la causa di lunghi tempi di caricamento e di notevole degrado delle performance. Qlik Sense, oltre all'utilizzo

delle query, consente la creazione dei QVD. Un file QVD (QlikView Data) è un file contenente una tabella di dati esportata da Qlik Sense. È un formato di Qlik nativo che può essere letto o scritto solo da Qlik Sense o Qlik View. Tale file è ottimizzato per la velocità di lettura dei dati restando comunque di dimensioni molto compatte. In genere, la lettura di un file QVD è circa 10-100 volte più veloce rispetto alla lettura da altre sorgenti. I file QVD vengono utilizzati in generale per le seguenti motivazioni:

- *Incremento della velocità di caricamento*: eseguendo in memoria il buffer di blocchi di dati di grandi dimensioni, l'esecuzione dello script diventa notevolmente più veloce
- *Diminuzione del carico sui server del database*: ciò consente di ridurre il carico di lavoro sui database esterni e di ridurre il traffico di rete.
- *Consolidamento dei dati di più app*: con l'utilizzo dei file QVD uno script può combinare dati provenienti da un numero qualsiasi di app.
- *Incrementale*: spesso i QVD vengono utilizzati per il ricarico incrementale caricando esclusivamente nuovi record da tabelle o database che si aggiornano periodicamente.

La creazione di un QVD avviene attraverso l'utilizzo del comando *store* posto alla fine di uno script relativo al caricamento di una tabella. Nello specifico caso in analisi, esistono molte tabelle sia dei fatti che delle dimensioni che vengono utilizzate da molte applicazioni differenti. Proprio per questo i QVD vengono utilizzati in maniera preponderante poiché si ottengono tutti i benefici dal punto di vista delle performance e si risparmia tempo nella creazione del modello dei dati. Quando si sviluppa una nuova applicazione che recupera dati da nuove tabelle, si procede con la creazione dei QVD per queste ultime e poi si crea il modello dati basandosi proprio su questi file. Affinché i QVD siano periodicamente aggiornati sono state create delle app preposte che ricaricano un QVD ogni volta che la tabella ad esso collegata viene aggiornata.

## 5.2 Fase di ETL e creazione del modello dati

Le applicazioni di reportistica Qlik Sense mostrano, in genere, misure e dimensioni relative ad una specifica area di sviluppo aziendale, necessarie agli utenti del business per fare analisi ed estrarre conoscenza. Questi dati sono il frutto di un'accurata elaborazione che avviene nella fase di ETL. ETL (Extract/Transform/Load Figura 5.1), è il processo di raccolta dei dati da diverse sorgenti e dalla loro successiva organizzazione e centralizzazione in un unico repository.



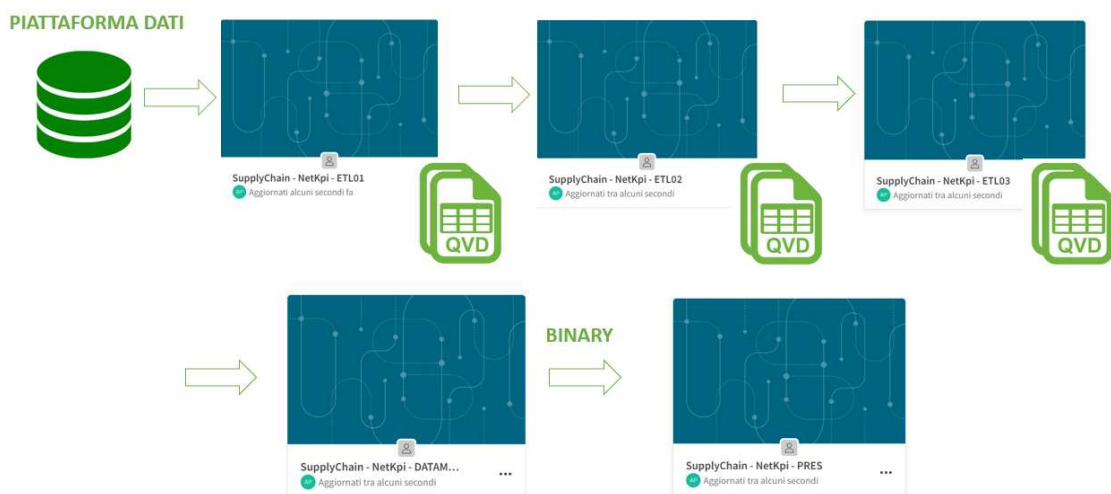
Figura 5.1: Processo di ETL

Il processo ETL è generalmente nelle seguenti tre fasi:

- *Estrazione*: L'obiettivo finale è quello di ottenere dati puliti e accessibili. I dati "grezzi" vengono estratti dalle molteplici sorgenti che vanno a comporre la piattaforma dei dati.

- *Trasformazione*: Questa fase è la più critica poichè ai dati vengono applicati i filtri e le regole aziendali necessarie a soddisfare i requisiti.
- *Caricamento*: L'ultima fase del processo di ETL prevede il caricamento dei dati estratti verso una nuova destinazione.

Nel caso di studio in analisi, tutto il processo viene eseguito sia per la parte di back-end e sia per la parte di front-end. Un applicazione Qlik Sense che viene rilasciata all'utente è proprio il frutto dell'esecuzione di questo tipo di processo. La fase di ETL lato front-end si ottiene attraverso la creazione di più applicazioni. Per il rilascio di ogni applicazione viene seguito un determinato schema dove si creano tante applicazioni per quante sono le fasi di ETL necessarie all'elaborazione del dato per quella determinata area di analisi. Si parte con la creazione di un app di *ETL01*, dove avviene la fase di estrazione del dato attraverso query dirette al database. Tale estrazione viene salvata nei file QVD che possano essere ripresi dalle successive applicazioni. Si procede con l'app di *ETL02* dove viene caricato il QVD precedentemente salvato. Vengono eseguite delle elaborazioni dei dati estratti e i risultati vengono salvati in un file QVD apposito. A questo punto i dati, se necessario, vengono processati da una terza fase di ETL, altrimenti, vengono caricati all'interno di un'applicazione apposita (*DATAMODEL*) in cui avviene la creazione del modello dei dati. Tale modello viene, infine, caricato all'interno dell'app di *presentation* attraverso l'utilizzo dell'istruzione *binary* di Qlik Sense. Nell'app di presentation vengono mostrati i grafici e i KPI come da richiesta da parte del business. La Figura 5.2 mostra il processo di sviluppo delle applicazioni fino all'app di presentation.



**Figura 5.2:** Fase di ETL lato sviluppo front-end

### 5.2.1 Creazione della reportistica

Terminata la fase di ETL e creazione del modello dati, l'applicazione dispone di tutte le informazioni pulite e organizzate in accordo con le necessità del business. La creazione dei grafici e dei KPI avviene nell'app di presentation. All'interno di tale applicazione vengono creati tanti sheet quante sono le dimensioni d'interesse per l'area aziendale a cui si fa riferimento. Nel caso in analisi, generalmente, il grafico più utilizzato è quello della tabella. Spesso, infatti, gli sheet sono composti da tabelle che mostrano numerose dimensioni (ad esempio codici prodotto, delivery, date, etc) correlate a una serie di misure (KPI aziendali).



Nella parte alta dello sheet vengono posizionati i filtri, richiesti dagli utenti del business, che loro permettono di estrarre analisi più specifiche. Per questo tipo di report il compito dello sviluppatore è fare molta attenzione alle performance. Queste tabelle possono arrivare a caricare milioni di dati, e ciò rallenta molto l'apertura dell'app e la stessa fruizione del contenuto. Lo sviluppatore, può quindi, agire lato ETL cercando di migliorare le performance di caricamento del dato o, in accordo con gli utenti, limitare la numerosità dei dati in visualizzazione ad un range specifico per le necessità d'analisi. Oltre questo tipo di report, vengono utilizzati, anche, i grafici scelti tra quelli più utilizzati e di impatto (ad esempio a barre, lineari, misuratori e combinati).

### 5.2.2 Integration test e rilascio

Una volta che l'applicazione è completa si procede con la fase di testing. Questo lavoro è di estrema importanza e buona parte del tempo necessario per lo sviluppo viene dedicata proprio a questa fase. La prima cosa che si verifica è che i valori delle misure riportate nell'app di presentation siano coerenti con quello che si ha nella piattaforma dati. Si analizzano le logiche di calcolo delle misure nel front-end per poi riprodurle tramite query SQL (integration test). Questa procedura permette di avere la certezza che nessuna informazione si è persa durante i vari passaggi della fase di ETL.

Il secondo step da svolgere per la fase di testing consiste nel fare il confronto dei valori ottenuti con ciò che mostrava la vecchia applicazione basata su Oracle Obiee. Non è sempre possibile avere un confronto uno a uno in questa fase, poichè il processo di migrazione del dato nel back-end potrebbe aver subito modifiche in accordo con il cliente. Riguardo a questi due tipi di test, si cerca di porre maggiormente l'attenzione sull'integration test, poichè avere la certezza del corretto passaggio delle informazioni dal database all'applicazione di presentation permette, poi, in caso di errori a back o logiche di mapping errate, di rimediare effettuando un semplice ricaricamento dell'app. Il fallimento dell'integration test blocca il rilascio dell'app e, quindi, il mancato rispetto della data di scadenza. L'azienda di consulenza sensibilizza fortemente gli sviluppatori su questa fase di testing poichè il rilascio di un'applicazione non correttamente testata provoca molte segnalazioni da parte degli utenti di business e, quindi, una coda di lavoro che mette in difficoltà il rilascio dei successivi deliverable. Terminata la fase di testing si procede con il rilascio dell'applicazione in ambiente di produzione. Tale rilascio viene segnalato all'IT del cliente che esegue una prima validazione. A valle di questo passaggio si procede con il rilascio verso gli utenti del business che conoscono in maniera dettagliata il dato e che quindi possono segnalare eventuali incongruenze.

## 5.3 Sviluppo applicazioni dual run

Tutto il processo descritto finora riguarda lo sviluppo di applicazioni Qlik Sense volte a sostituire applicazioni presenti solo nella vecchia piattaforma di Business Intelligence, Oracle Obiee. Oltre a ciò, esistono anche delle applicazioni Qlik Sense che sono state sviluppate dal cliente, ma che attingono i dati dal vecchio DWH. Questo tipo di app vengono definite *dualrun* poichè è stato effettuato il passaggio da Obiee a Qlik Sense, ma la sorgente dei dati è ancora quella obsoleta del DWH. La procedura di sviluppo di questo tipo di app è leggermente diversa da quanto descritto finora. L'obiettivo finale è sempre quello di ottenere un app Qlik che, però, utilizzi come sorgente la nuova piattaforma dati. Solitamente lo sviluppo di queste app è più veloce rispetto alla creazione totalmente da zero poichè, in questo caso, si risparmia molto tempo nella fase iniziale di analisi dei campi e creazione del modello dei dati, passaggio che è già stato svolto dal cliente. Il primo step che bisogna

seguire, che è anche il più importante, è quello di effettuare un'attenta analisi delle strutture utilizzate (tabelle) in modo che prima di iniziare lo sviluppo si abbia un quadro generale su ciò che manca ed eventualmente si possa richiedere la creazione di queste strutture mancanti prima della data di rilascio.

### 5.3.1 Traduzione delle query

L'operazione più importante per lo sviluppo delle applicazioni dualrun è la traduzione delle query. Queste applicazioni, come le altre descritte finora, possono essere formate da una successione di diverse app in base a quante sono le fasi di ETL necessarie all'elaborazione dei dati. Ogni app presenta una serie di query per il recupero dei dati su DWH. Tali query sono costituite da tabelle e campi che attingono e fanno riferimento alle vecchie logiche del DWH. L'operazione di traduzione consiste nel mapping di tali strutture verso quelle presenti nella nuova piattaforma dati. Questa traduzione può essere semplice quando le logiche di mapping sono rimaste quasi uno-ad-uno, e quindi si interviene modificando esclusivamente il nome delle tabelle e dei campi. In questo caso le join vengono lasciate inalterate e si fa sì che la query vada ora a puntare alla nuova piattaforma dati. Spesso, però, le logiche di mapping delle tabelle subiscono modifiche; alcuni campi possono essere aggregati in altri, o, addirittura eliminati. Tutto ciò rende più complesso il processo di traduzione e di recupero delle informazioni. Proprio per questo, anche in questa fase, è bene spendere del tempo per eseguire degli integration test approfonditi così da verificare sin dalle prime fasi di ETL se, con la nuova traduzione, risultano perdite di dati. Nonostante alcuni campi possono avere un nome diverso nella piattaforma dati rispetto al DWH, questi vengono rinominati attraverso l'utilizzo dell'alias così che si possano riutilizzare tutti i successivi passaggi di rielaborazione già formulati dal cliente.

### 5.3.2 Aggiornamento della reportistica

Gli sheet delle applicazioni dualrun sono formati da grafici creati dall'area IT del cliente. Una volta completato il processo di traduzione delle query, l'applicazione dovrà presentare i dati recuperati dalla nuova piattaforma dati. Lo sviluppatore ha il compito di verificare che tutti i dati siano riportati in maniera corretta sui grafici che erano stati creati in precedenza. Il processo di migrazione può far sì che alcuni campi, o alcune regole di calcolo, subiscano modifiche e questo poi si ripercuote su come i grafici dei report sono stati impostati.

Si procede analizzando dimensioni e misure di ogni grafico in modo da verificare la presenza di possibili errori. Oltre a ciò, l'area IT può richiedere l'aggiunta di nuove visualizzazioni; tale esigenza nasce direttamente dagli utenti del business durante il processo di analisi. Lo sviluppatore front-end si occupa di aggiungere queste nuove visualizzazioni facendo attenzione che ciò non comprometta il corretto funzionamento dell'applicazione, come era allo stato dell'arte.

## 5.4 Fase di UAT

L'acronimo *UAT* (User Acceptance Testing) descrive la parte finale dello sviluppo di un software. Lo scopo principale di questo test è convalidare il software rispetto ai requisiti aziendali. Nel caso in analisi tale fase inizia dopo il rilascio dell'applicazione. Si procede effettuando il rilascio verso l'IT che, dopo una prima analisi, comunica agli utenti finali del business che l'applicazione è disponibile e può essere testata. La convalida dell'app passa per gli utenti del business proprio per la conoscenza specifica dei requisiti aziendali. La fase di UAT ha una durata di alcune settimane durante le quali vengono concordati dei meeting

circa due volte a settimana. A queste riunioni partecipano gli sviluppatori dell'applicazione, l'IT e gli utenti del business lato cliente. Lo sviluppatore presenta, di volta in volta, i vari report che compongono l'app e si mette in ascolto delle domande presentate dagli utenti del business. Lo scopo è quello di fare in modo che gli utenti possano sfruttare a pieno le potenzialità dell'app per le analisi che dovranno svolgere. Con il susseguirsi dei meeting, gli utenti approfondiscono le analisi sull'app e, possibilmente, espongono le problematiche incontrate durante tale processo. Le segnalazioni più frequenti sono relative a mancanza di informazioni, misure non corrette o richieste di migliorie dal punto di vista dell'utilizzo dell'app. Lo scopo finale della fase di UAT è fare in modo che l'applicazione soddisfi gli utenti del business sia in usabilità sia nel contenuto informativo. Raggiunto tale risultato, si può passare alla successiva fase di training.

## 5.5 Fase di training con gli utenti

La fase di *training* con gli utenti è l'ultimo processo prima della dismissione dell'area. Tale fase può avere inizio solo dopo la convalida del rilascio dell'applicazione da parte dell'IT e degli utenti incaricati alla validazione. Alla fase di training partecipano tutti gli utenti del business, a livello globale dell'azienda cliente, che hanno interesse nell'utilizzo dell'app per le analisi. Anche in questo caso si vanno a schedulare una serie di riunioni in cui lo sviluppatore, insieme all'IT e ad alcune figure manageriali, si occupa di presentare l'applicazione.

A queste riunioni, generalmente, partecipano utenti da tutto il mondo e, per questo, vengono svolte totalmente in lingua inglese. La presentazione, in questa fase, parte da una descrizione generale di quello che è Qlik Sense per poi scendere più nei dettagli tecnici implementativi dell'applicativo. Si forniscono informazioni generali sul tool Qlik Sense poichè alcuni utenti possono non aver mai utilizzato tale piattaforma. Questi meeting hanno una durata di diverse ore poichè si cerca di lasciare molto spazio alle domande; questo fa sì che tutti i dubbi possano essere chiariti in un'unica sede e che gli utenti dispongano di una certa conoscenza di base quando utilizzeranno l'applicazione. È importante ricordare come l'utente sia abituato a svolgere analisi utilizzando le app di Oracle Obiee, le cui funzionalità sono diverse rispetto a quelle che poi si ritroverà nell'app Qlik Sense.

## 5.6 Chiusura della subject area

Terminata la fase di training si procede con la chiusura della subject area. Tale attività indica che l'applicazione è stata correttamente sviluppata, è funzionante e che sono stati soddisfatti i requisiti aziendali concordati con l'IT del cliente. Il passaggio successivo consiste nella redazione di una documentazione tecnica riguardante l'applicativo rilasciato. All'interno di questa documentazione viene descritto in maniera generale tutto il processo di sviluppo back-end e front-end. Vengono raccolti gli script relativi alle section del data load editor e le regole di calcolo dei principali KPI presenti nei report. Questo documento viene consegnato all'area AMS (Application Management Services). Il team AMS offre un servizio di assistenza applicativa per tutte le entità che sono state rilasciate e validate dal cliente. Esso prende in carico le richieste del cliente attraverso l'apertura di ticket per tutte le problematiche che possono riguardare strutture back-end e applicazioni front-end già rilasciate. Grazie a questo servizio gli sviluppi procedono senza che ci siano interruzioni dovute al malfunzionamento di entità già rilasciate.

---

## Esempi di funzionamento del nuovo sistema di Business Intelligence

---

*Nel presente capitolo verrà mostrato l'intero processo di sviluppo di un'applicazione. Saranno analizzati entrambi i casi descritti nei precedenti capitoli; ovvero lo sviluppo di applicazione dualrun e nuove app. La descrizione del processo inizierà mostrando la fase di estrazione dei dati dell'ETL01 per proseguire con le successive elaborazioni fino alla creazione dell'app relativa alla visualizzazione.*

### 6.1 Applicazione dualrun, il caso "SupplyChain - NetKpi - Overstock"

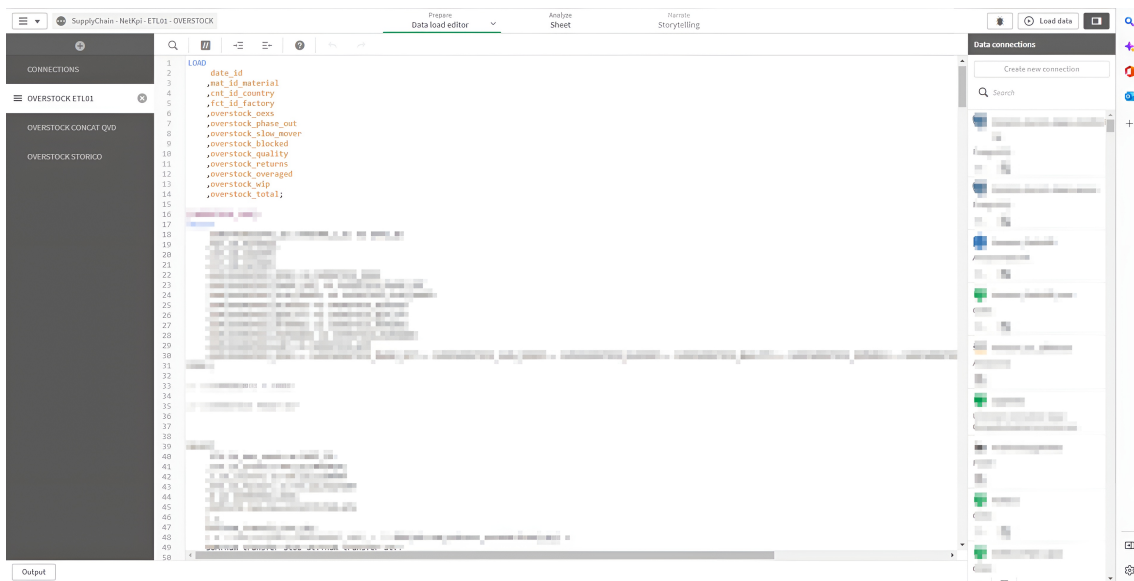
Il primo esempio che analizziamo riguarda lo sviluppo di un'applicazione dualrun, in particolare l'app di "SupplyChain - NetKpi - Overstock". Il nome di ogni applicazione non è scritto in maniera casuale, bensì segue uno specifico pattern in modo che subito si capisca a quale area o stream si fa riferimento. In questo caso si tratta della subject area relativa alla Supply Chain, ovvero la catena di approvvigionamento. Di questa specifica area, l'applicazione mostrerà i dati dei principali KPI relativi all'overstock, ovvero le scorte ed i volumi dei vari componenti contenuti nei magazzini a livello mondiale. Questa applicazione è di tipo dualrun poichè era già stata implementata in Qlik Sense, ma le query puntavano al vecchio DWH. Il lavoro svolto è stato quello di tradurre tutte le query in modo da recuperare le informazioni dalla nuova piattaforma dati, riadattare le logiche per la creazione del nuovo modello dati ed infine adeguare la grafica dell'app di presentation sulla base delle linee guida aziendali utilizzate in tutte le altre applicazioni Qlik Sense.

#### 6.1.1 Estrazione dei dati dalle tabelle dei fatti

La prima fase di sviluppo per un'applicazione dualrun consiste nella traduzione delle query. Tale operazione viene generalmente eseguita utilizzando un tool di amministrazione di database. Lo sviluppatore, con l'aiuto dell'analista funzionale, "mappa" i campi e le tabelle del DWH con i corrispettivi riferimenti della nuova piattaforma dati. Completato questo passaggio si procede con la validazione dei dati; si prendono in considerazione le misure più importanti al fine della creazione dei report e si verifica che le regole di calcolo restituiscano un risultato il più vicino possibile a quello che si aveva con la query nel DWH. In base alle tabelle o ai campi che si considerano possono esserci leggere squadrature che, però, non incidono sulle analisi.

Terminata la validazione della nuova query si procede con l'implementazione in Qlik. La prima applicazione della catena è quella di ETL01, dove avviene l'estrazione dei dati dal database e il salvataggio di questi all'interno di un apposito QVD. La query di ETL01 estrae i dati e le misure dalle tabelle dei fatti, ovvero le strutture dove i record vengono identificati come *fatti* non modificabili e vengono aggiunti progressivamente alla tabella sotto forme di flusso o blocchi di grandi dimensioni. In questa fase si ha anche un pre-filtraggio del dato attraverso opportune join con le tabelle delle dimensioni; tali filtri di minor rilievo rispetto alle fasi successive.

Nella Figura 6.1 viene mostrato uno screen rappresentante una *section* del data load editor dell'applicazione di ETL01 relativa all'Overstock. Nella parte in alto viene specificata l'istruzione *Load* che permette il caricamento dei campi elencati in giallo. Questi campi vengono estratti dalla query che punta alla nuova piattaforma dati.



**Figura 6.1:** Estrazione e caricamento delle informazioni

Questa applicazione, in particolare, si aggiorna una volta al giorno ed è impostata per recuperare i dati in maniera incrementale a partire dal primo giorno del mese fino al giorno precedente al reload. Proprio per questa modalità di ricarica si è preferito impostare la creazione dei QVD su base mensile. Alla fine dello script viene applicata una *store* in modo da avere come risultato finale un QVD per ogni mese.

Oltre ai dati dell'anno corrente, l'applicazione necessita di recuperare le informazioni anche degli anni passati. Tale recupero avviene modificando gli intervalli temporali della query. Anche in questo caso si procede con la creazione di un QVD apposito.

Il passaggio successivo (Figura 6.2) consiste nella concatenazione dei QVD creati su base mensile con quello relativo al pregresso. Tale operazione avviene in una *section* apposita dove viene eseguita la *Load* dei campi da entrambi i QVD. L'istruzione *concat* di Qlik permette la creazione di un'unica tabella che, infine, viene salvata in un terzo QVD comprensivo di tutti i dati dell'overstock. Tale tabella viene eliminata attraverso l'istruzione *DROP TABLE*; questo per velocizzare il reload dell'applicazione evitando il salvataggio di una struttura di dimensioni elevate.

### 6.1.2 Seconda fase di estrazione, join con le tabelle delle dimensioni

Nel caso specifico dell'applicazione di Overstock è necessario implementare una seconda fase di estrazione e filtraggio delle informazioni in modo da ottenere i dati desiderati.

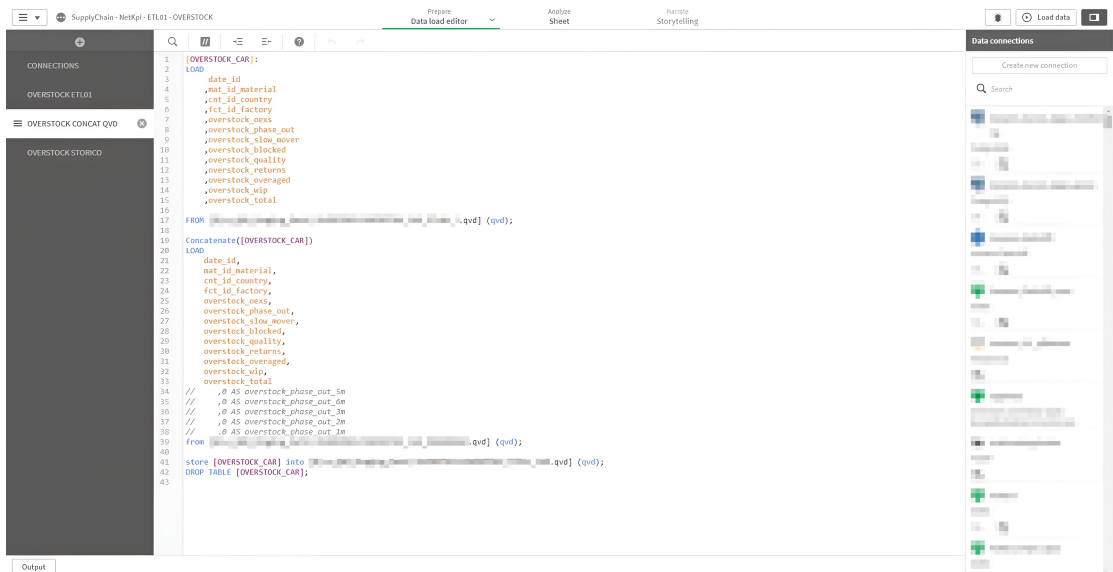


Figura 6.2: Concatenazione dei QVD mensili e progresso

Tale operazione avviene all'interno di una seconda applicazione che viene appositamente denominata ETL02.

Come anticipato nel precedente capitolo, grazie all'utilizzo dei QVD, l'applicazione riesce a caricare tutti i dati estratti dalla prima fase di ETL01. Questa operazione avviene grazie all'utilizzo dell'istruzione `Load *` seguita dal path di storage del QVD. Una volta che questi dati sono stati recuperati (Figura 6.3), nell'applicazione di ETL02 vengono eseguite diverse *join* con tabelle delle dimensioni contenenti decine o milioni di record. Tali tabelle contengono gli attributi lungo i quali la tabella dei fatti calcola la metrica.

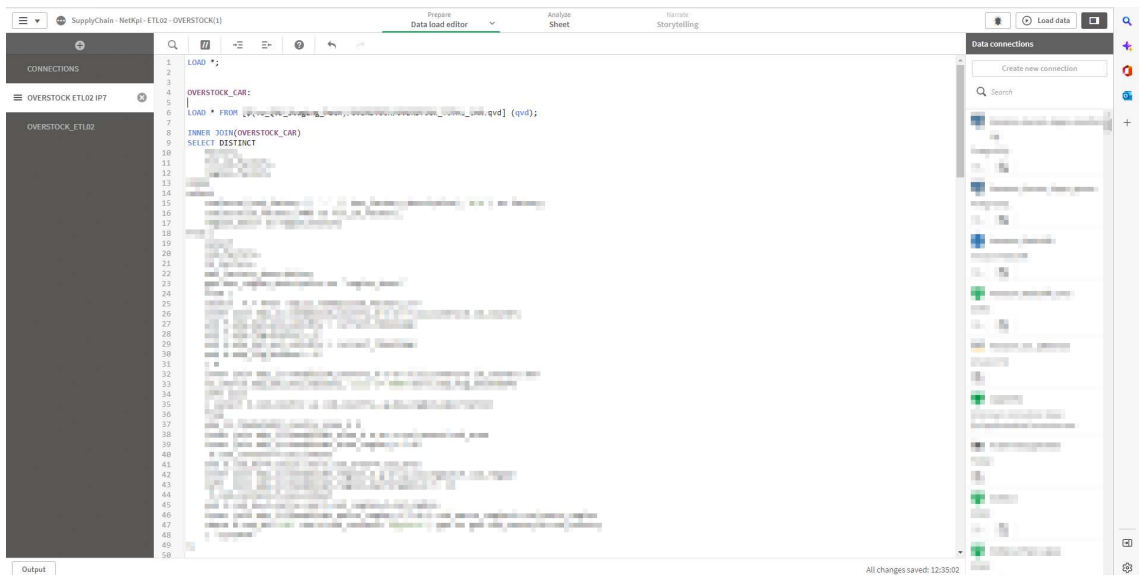


Figura 6.3: Load QVD di ETL01 e join con tabelle delle dimensioni

Questa seconda fase di ETL viene eseguita in un applicazione diversa rispetto alla prima in modo da migliorare al massimo le performance. Si cerca, inoltre, di isolare possibili anomalie nel dato raggruppando le join con le tabelle delle dimensioni più grandi all'interno di un applicazione apposita.

L'applicazione, infine, salva il risultato del filtraggio ottenuto (Figura 6.4) attraverso le join all'interno di un QVD di ETL02 che sarà poi recuperato per le successive elaborazione dall'app di ETL03.

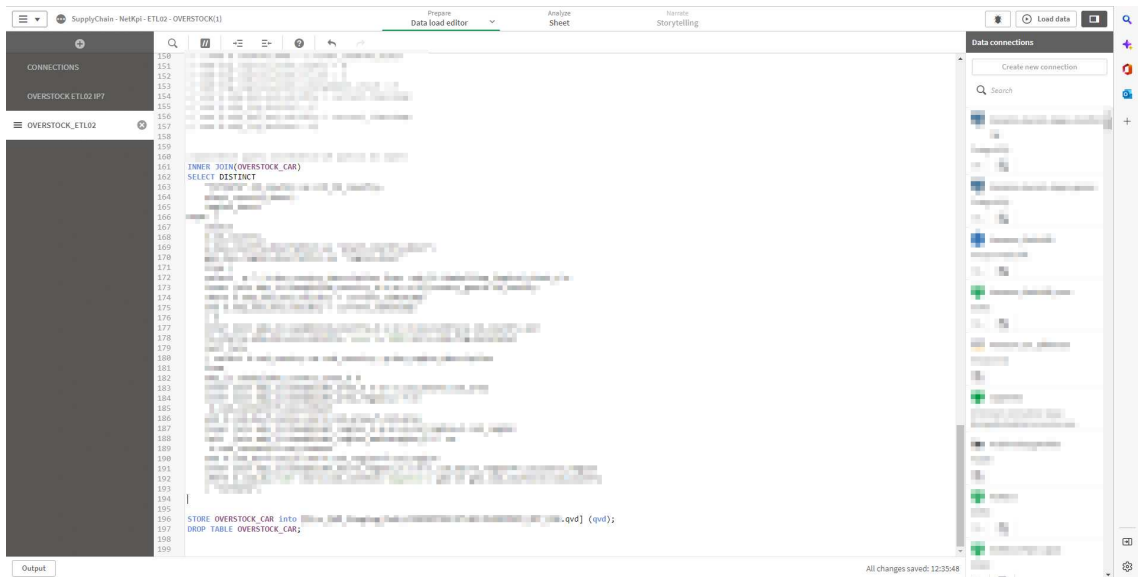


Figura 6.4: Store per creazione del QVD di ETL02

### 6.1.3 Organizzazione dei campi e delle chiavi

Con l'applicazione di ETL02 termina la fase di estrazione delle informazioni dalle tabelle dei fatti e di join con le dimensioni. Lo step successivo consiste nell'organizzazione dei campi e delle chiavi. Questo tipo di operazione avviene all'interno di un'applicazione apposita, l'ETL03 (Listato 6.1). Come per le altre app della catena, anche in questo caso i campi vengono caricati attraverso la Load del QVD di ETL02. Il listato di seguito riportato mostra come i campi vengano processati all'interno dell'app. Nella prima parte si può osservare come alcuni campi vengano caricati tramite una concatenazione per la creazione della *PK\_LINK\_TABLE\_DESC*. All'interno di questa concatenazione è possibile che dei campi subiscano delle elaborazioni (ad esempio. *plant\_region\_desc*) sulla base delle esigenze di visualizzazione nell'app finale. La *PK\_LINK\_TABLE\_DESC* viene creata raggruppando solo un determinato set di campi. Questi, infatti, a differenza degli altri riportati in seguito, sono dei campi che sono in comune con altre subject area. Nella creazione del modello di dati dell'applicazione di Overstock, questi campi saranno recuperati anche da altre tabelle dei fatti poiché alcune misure possono essere calcolate prendendo informazioni da due subject area diverse. La creazione della *Link Table* serve a creare una terza tabella dei fatti quando si hanno due o più tabelle dei fatti con diversi campi in comune. Grazie a questa terza tabella è possibile andare a collegare più tabelle dei fatti che presentano gli stessi campi con le dimensioni. In alternativa, Qlik Sense in presenza di campi comuni tra più tabelle, cerca di effettuare i collegamenti creando chiavi sintetiche o referenze cicliche. Una chiave sintetica non è altro che un campo anonimo che Qlik crea quando due o più tabelle dei dati hanno due o più campi in comune. Tale chiave è formata da tutte le combinazioni ricorrenti dei valori dei campi comuni; la presenza di tali chiavi indica un modello dei dati errato.

Listing 6.1: Script di ETL03

```

[OVERSTOCK]:
LOAD

    seasonality &'%'& premium &'%'& costruction_rc &'%'&
    cl_description &'%'& marked &'%'& rim_band &'%'& rim &'%'&
    NULL() &'%'& bu &'%'& market_segment &'%'& comm_group &'%'&
    brand &'%'& prd_status &'%'& mat_gestional_status &'%'& date_id &'%'&
    tread_pattern &'%'& NULL() &'%'& mf_coddesc_matfam &'%'& NULL() &'%'&
    oem_brand

    &'%'& NULL() &'%'& NULL()
    &'%'& NULL() &'%'& NULL() &'%'& factory &'%'& prestige

    &'%'& if(plant_region_desc = 'APAC', 'APAC',
    IF(plant_region_desc = 'EUROPE', 'EUROPE',
    IF(plant_region_desc = 'RUSSIA / NORDICS / MEAI', 'RUSSIA /
    NORDICS / MEAI'
    IF(plant_region_desc = 'NORTH AMERICA', 'NORTH AMERICA',
    IF(plant_region_desc= 'RUSSIA / NORDICS / MEAI',
    'RUSSIA / NORDICS / MEAI',
    IF( plant_region_desc = 'SOUTH AMERICA', 'SOUTH AMERICA',
    'OTHER'))))))) //AS REGION_OVERVIEW

AS PK_LINK_TABLE_DESC,

    plant_region_desc      as      OVERSTOCK_PLANT_REGION_DESC,
    plant_country_descr    as      OVERSTOCK_PLANT_COUNTRY_DESCR,
    region_factory         as      REGION_FACTORY,

    overstock_oexs AS OVERSTOCK_OEXS,
    overstock_phase_out AS OVERSTOCK_PHASE_OUT,
    overstock_slow_mover AS OVERSTOCK_SLOW_MOVER,
    overstock_blocked AS OVERSTOCK_BLOCKED,
    overstock_quality AS OVERSTOCK_QUALITY,
    overstock_returns AS OVERSTOCK_RETURNS,
    overstock_overaged AS OVERSTOCK_OVERAGED,
    overstock_wip AS OVERSTOCK_WIP,
    overstock_total AS OVERSTOCK_TOTAL

STORE OVERSTOCK into [#####] (qvd);
DROP TABLE OVERSTOCK;

```

La ridenominazione di alcuni campi e la creazione della Link Table vengono salvati nel QVD di ETL03 che funge da input per l'app relativa alla creazione del modello dati, ovvero il *Datamodel*.



### 6.1.4 Creazione del modello dati

La creazione del modello dati inizia con il caricamento dei campi processati dall' ETL03 che vengono salvati all'interno di una tabella temporanea. La Figura 6.5 mostra l'istruzione Load con la quale si recuperano le informazioni dal QVD. È bene fare attenzione alla funzione: `AUTONUMBER(PK_LINK_TABLE_DESC) AS PK_LINK_TABLE`

Questa funzione Qlik restituisce un valore intero univoco per ciascun valore distinto dell' *expression* calcolato che viene passato come parametro. Come visto nella precedente sezione, `PK_LINK_TABLE_DESC` è una chiave molto complessa formata da diversi campi. L' *autonumber* viene utilizzato per creare una rappresentazione compatta in memoria della chiave che velocizza di molto il processo di caricamento dell'intera applicazione.

I campi caricati vengono, poi, concatenati con i dati provenienti da un secondo QVD contenente lo storico dei due anni precedenti sulla base della data di reload.

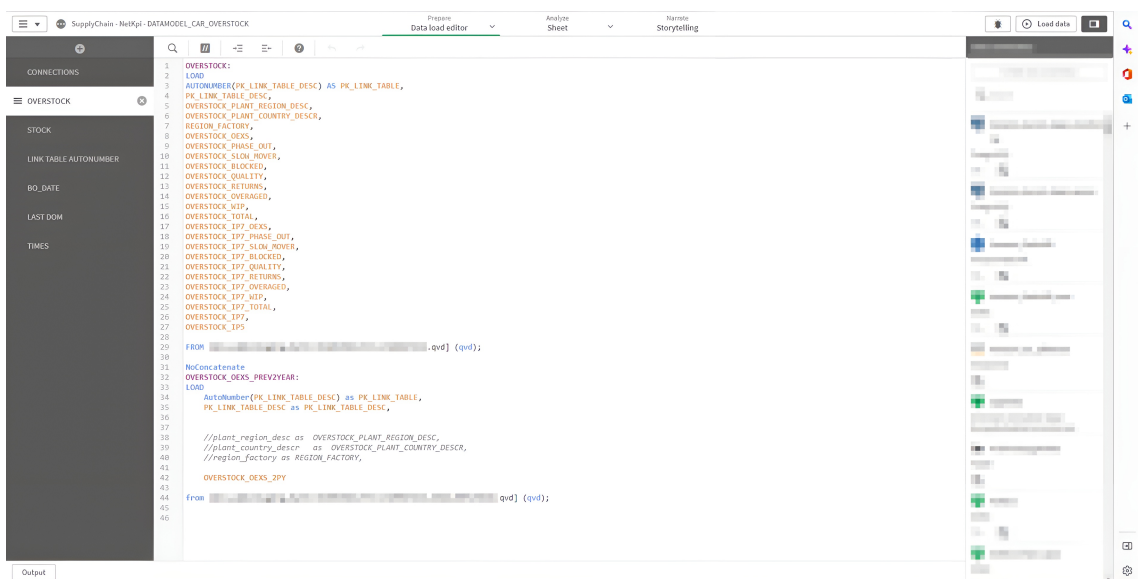


Figura 6.5: Load dei campi dal QVD di ETL03

Lo step successivo consiste nella creazione della Link Table che consente il collegamento tra le tabelle dei fatti che vengono caricate all'interno dell'app. Come si può notare nella colonna di sinistra della Figura 6.5, è presente una sezione dove vengono caricati dei dati relativi allo *Stock*. Questo indica che per questa specifica app esistono dei campi in comune che derivano da due fact table diverse. La soluzione a questo problema si può vedere nella Figura 6.6. Nella section `LINK_TABLE_AUTONUMBER` viene creata la tabella `LINK_TABLE`. Questa utilizzerà come chiave primaria il campo `LINK_TABLE` creato in precedenza attraverso la funzione `autonumber` ed estrarrà tutti i campi che sono in comune con la tabella dei fatti dello stock dalla `PK_LINK_TABLE_DESC`. Tale chiave era stata creata nella fase di ETL03 contenente tutti i campi comuni. L'istruzione `SUBFIELD` permette l'estrazione singola dei campi concatenati. Lo stesso tipo di operazione verrà eseguito per lo Stock. Grazie a questa modalità di caricamento, verrà utilizzata un'unica chiave primaria come collegamento tra le tabelle (la `PK_LINK_TABLE`) evitando, inoltre, la creazione di chiavi sintetiche dai campi estratti dalla `PK_LINK_TABLE_DESC`

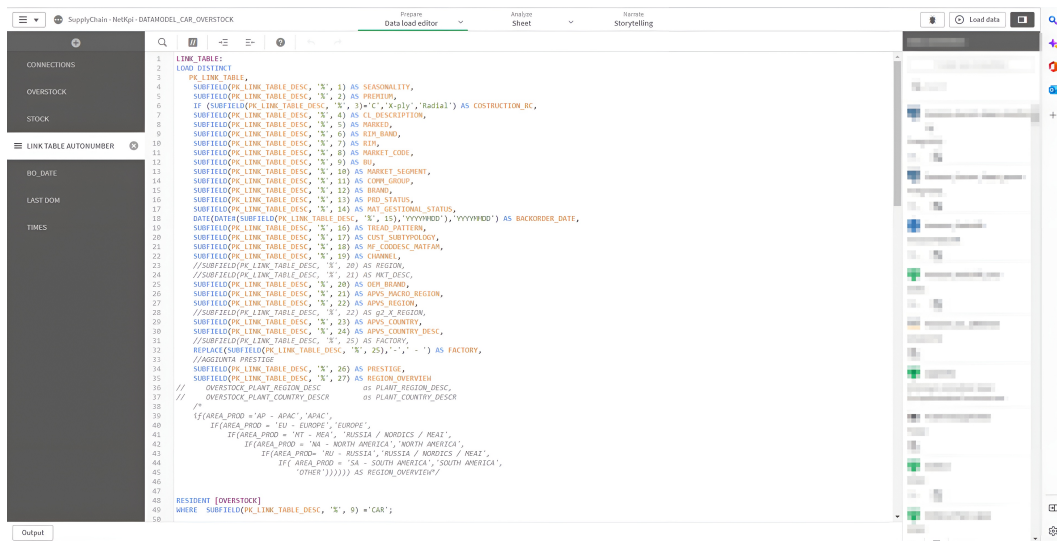


Figura 6.6: Creazione della Link Table per il modello dei dati

### 6.1.5 Visualizzazione dei dati

Terminato il processo di datamodel, i dati sono pronti per essere visualizzati nei report. L'applicazione finale della catena è quella, appunto, di *presentation*. Diversamente rispetto alle altre app, l'applicazione di datamodel non crea un QVD finale contenente il modello dati. In questa parte finale, infatti, i dati vengono recuperati attraverso l'utilizzo della *BINARY*:

*BINARY*: [PATH]

Tale istruzione permette il caricamento dei dati da un'altra app Qlik Sense, nello specifico l'app di datamodel. Il primo foglio (Figura 6.7) presenta una serie di misure raccolte sotto forma tabellare. Sono presenti dei *button* (Hide Region, Hide Countries) che permettono di ampliare le dimensioni di esplorazione del dato. Nella parte in alto di tutti i fogli, inoltre, sono presenti una serie di filtri che permettono di modificare dinamicamente uno dei parametri in modo da poter eseguire diverse tipologie di analisi.

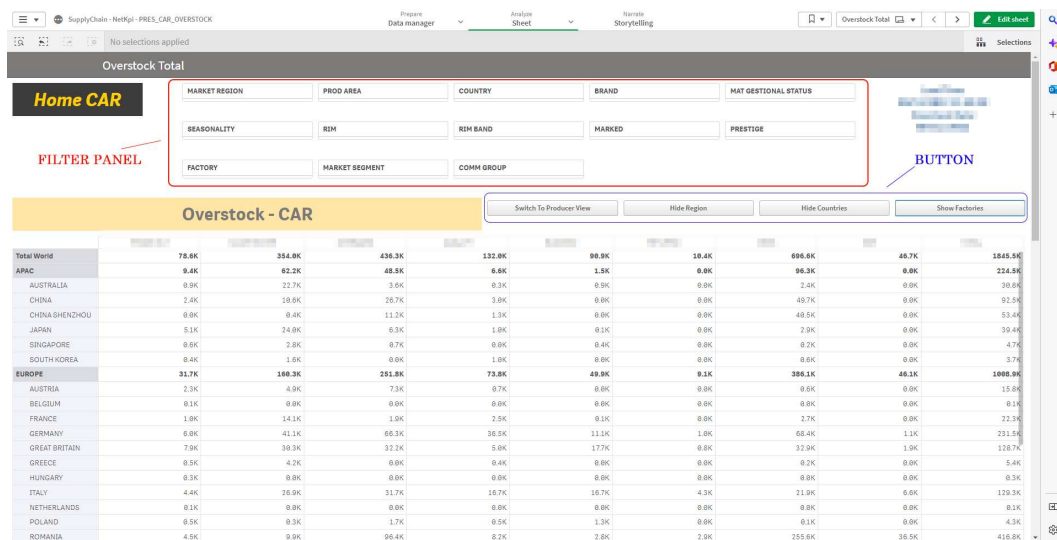


Figura 6.7: Panoramica di KPI Overstock

Il secondo foglio (Figura 6.8) presenta, invece, il dettaglio delle misure esplorate analizzando i trend mensili e annuali. In questo caso sono stati utilizzati due *line graph* di Qlik Sense. Tramite un apposito pulsante, si seleziona il mese d'interesse che viene mostrato nel grafico in alto. Nella parte in basso, invece, viene mostrato lo storico annuale.

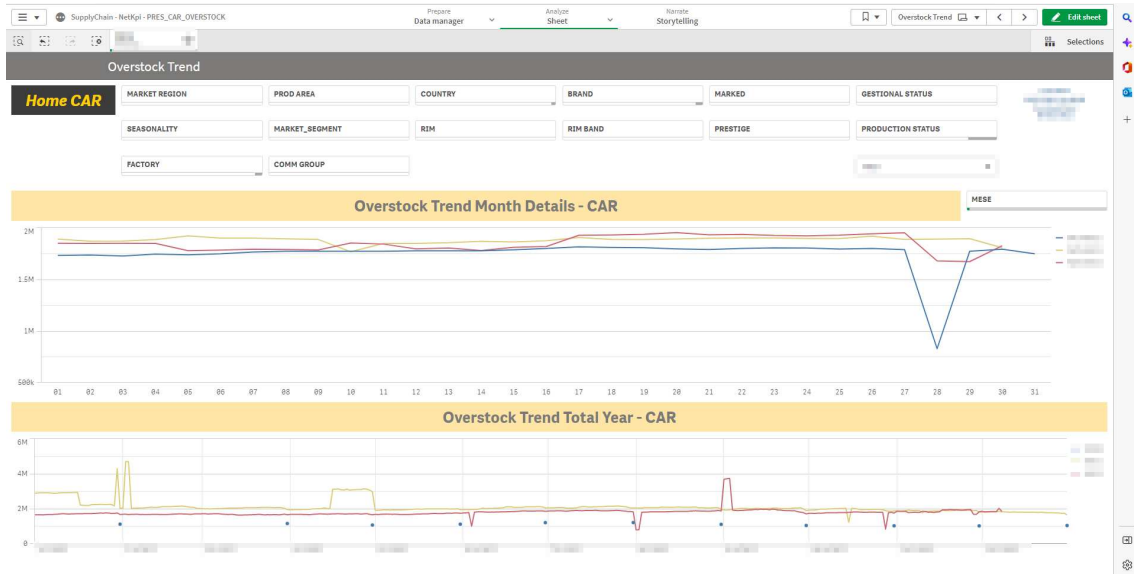


Figura 6.8: Trend di KPI Overstock

Un altro tipo di report che spesso viene mostrato nelle applicazioni riguarda il dettaglio di alcune misure in relazione alla collocazione geografica. Per rendere tale analisi più espressiva viene utilizzato il grafico *mappa* (Figura 6.9) di Qlik Sense. Tale grafico viene popolato con campi contenenti le informazioni sulle country; in base al valore della misura, questa viene visualizzata in maniera specifica sulla mappa.

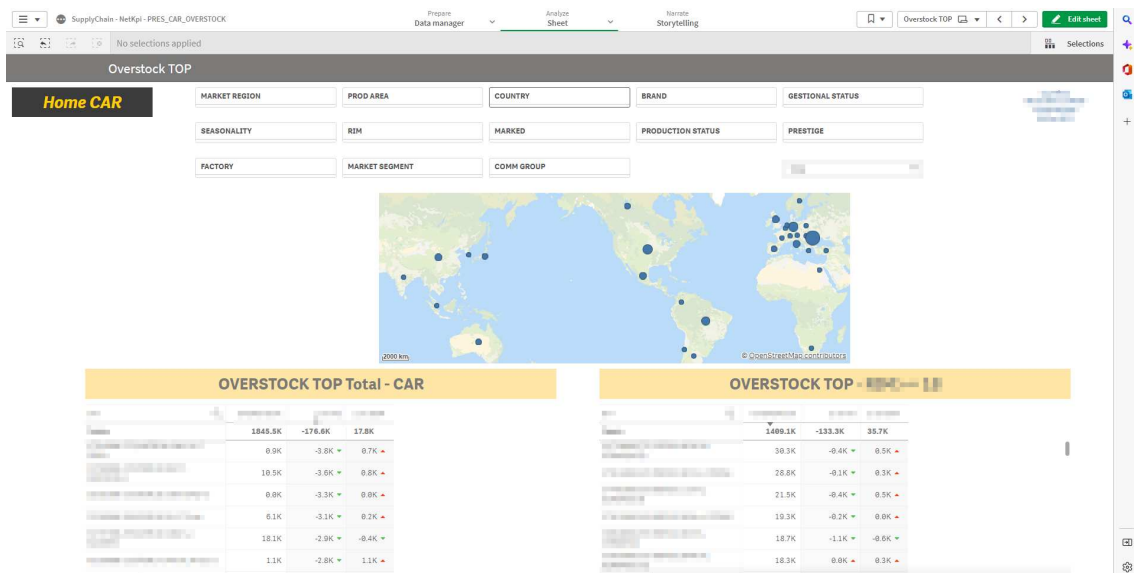


Figura 6.9: Global values di KPI Overstock

Gli ultimi due fogli dell'app mostrano uno lo storico del totale delle misure in analisi suddiviso per mese e anno, l'altro un focus sull'andamento dei trend raggruppati per le varie factory del cliente. Per lo sheet relativo allo storico, in Figura 6.10 è stato utilizzato un *Bar*

chart per le somme dei KPI suddivise su base mese/anno. Per lo sheet relativo alle factory (Figura 6.11) , invece, sono stati utilizzati due *line graph* per mostrare i valori percentuali dei KPI; anche in questo caso, viene preventivato l'andamento mese/anno suddiviso per le varie factory.



Figura 6.10: Storico di KPI Overstock

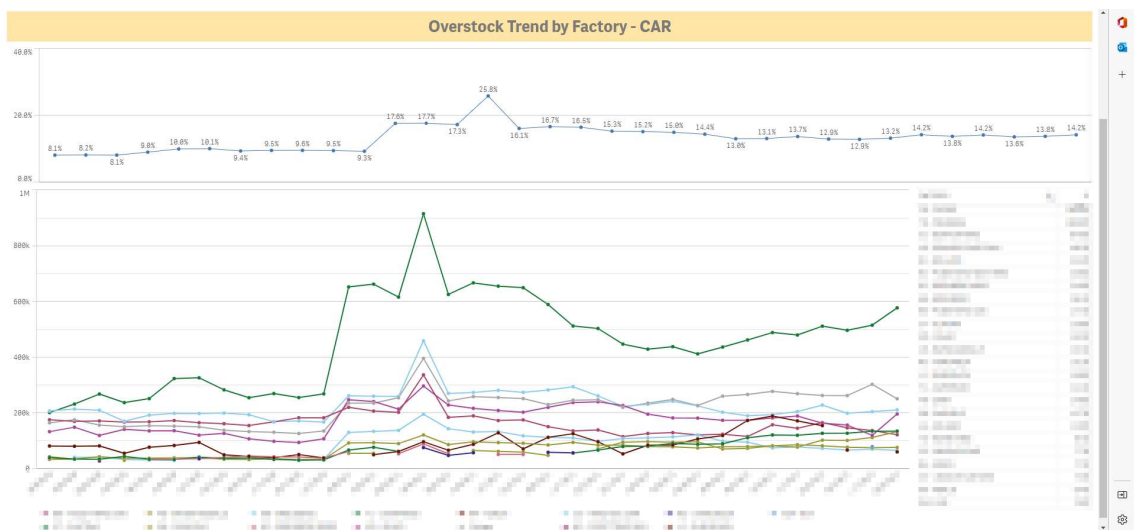


Figura 6.11: KPI Overstock in relazione alle factory

## 6.2 Nuova applicazione, "Planning - Logistic Export Monitor"

In questa sezione sarà analizzato lo sviluppo di una nuova applicazione Qlik a partire dalle informazioni recuperate dal report Obiee. La subject area di analisi per l'app di Logistic Export Monitor è la *pianificazione* che comprende tutti i dati e le informazioni necessarie all'azienda per la definizione degli obiettivi e delle azioni da effettuare per raggiungerli. L'app di Logistic Export Monitor, in particolare, mostra le quantità dei principali KPI relative all'export degli pneumatici dal paese di produzione verso un determinato paese cliente.

Nelle sezioni che seguono si analizzerà la differenza nello sviluppo rispetto all'applicazione dualrun per poi mostrare come sono stati creati i report associati.

### 6.2.1 Data load editor

La metodologia di sviluppo di una nuova applicazione è diversa rispetto allo sviluppo di un'app dualrun. L'obiettivo è quello di replicare le tabelle del DWH e le visualizzazioni che vengono utilizzate nei report Obiee. Una volta che le strutture dati sono state create, lo sviluppatore front end può iniziare il processo di creazione dell'app. Si inizia dalla semplice creazione di una nuova app vuota, in cui si creerà direttamente il modello dati. Non essendoci query dualrun da tradurre, si lavora recuperando i dati direttamente dai QVD. Le tabelle dei fatti create appositamente per la nuova applicazione vengono processate da applicazioni molto complesse con l'unico scopo di produrre un QVD che riporti il contenuto informativo intero della tabella. Quando si hanno tutti i QVD a disposizione allora si può iniziare con il caricamento dei dati nello stesso modo analizzato per le applicazioni dualrun. Nella Figura 6.12 viene mostrato un esempio di caricamento dati direttamente da QVD. Questi dati andranno a popolare una tabella Qlik Sense che, tramite le istruzioni di join, sarà collegata con le tabelle delle dimensioni. Quest'ultime vengono create e popolate allo stesso modo di come è stato fatto per le tabelle dei fatti. Le tre sezioni principali del data load editor sono:

- *Facts*: Dove sono caricati i QVD delle tabelle dei fatti.
- *Dimensions*: Presenta i QVD delle tabelle delle dimensioni.
- *Link Table*: Consente la creazione della Link Table.

Nella figura sottostante è anche possibile osservare come, alla chiusura dell'istruzione *Load*, sia possibile inserire una condizione di *where* utilizzabile, ad esempio, per limitare l'intervallo temporale di caricamento.

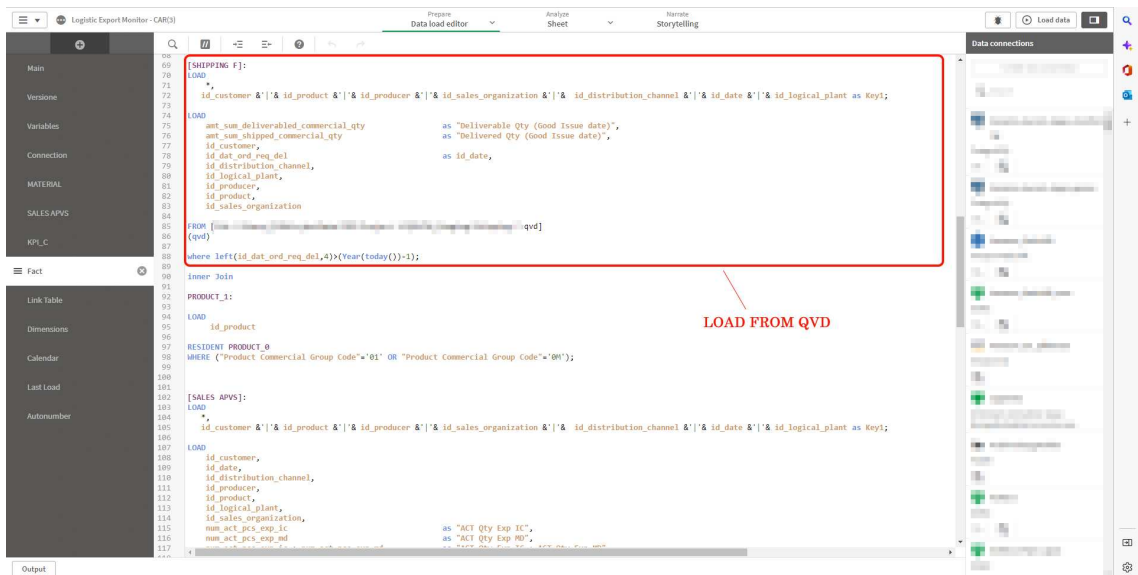
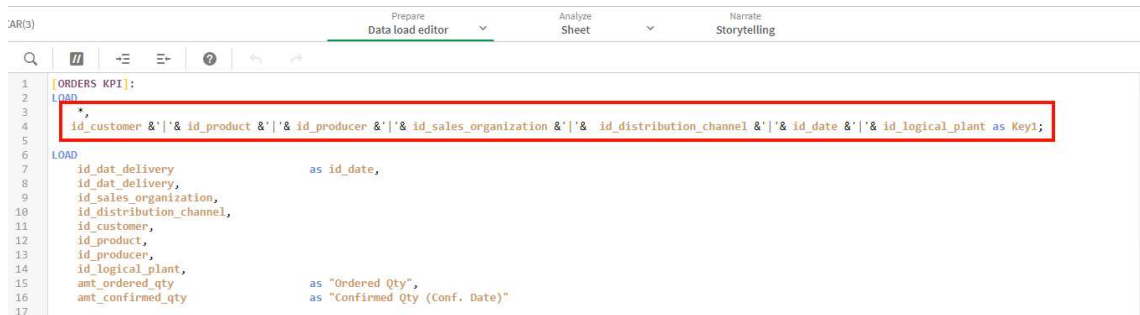


Figura 6.12: Load dei dati direttamente da QVD tabella dei fatti

Questa applicazione è di particolare interesse soprattutto per la complessità del modello dati sottostante. Tale modello è formato da sei tabelle dei fatti e ben sette tabelle delle dimensioni. La complessità sta nel fatto che ogni tabella dei fatti si collega in join con tutte e sette le tabelle delle dimensioni. Le chiavi utilizzate per effettuare questi collegamenti sono

i campi *ID* che sono denominati allo stesso modo tra tabelle delle dimensioni e dei fatti. Questa problematica ricalca l'analisi effettuata nelle precedenti sezioni relativa alla creazione di chiavi sintetiche in presenza di campi comuni tra diverse tabelle da parte di Qlik Sense. La soluzione, anche in questo caso, si ottiene con la creazione di una link table. Per ogni tabella dei fatti si procede con la creazione di un'unica chiave denominata *Key1* (Figura 6.13). Tale chiave consente la concatenazione di tutti i campi *ID* con i quali le tabelle dei fatti vanno in join con le tabelle delle dimensioni.



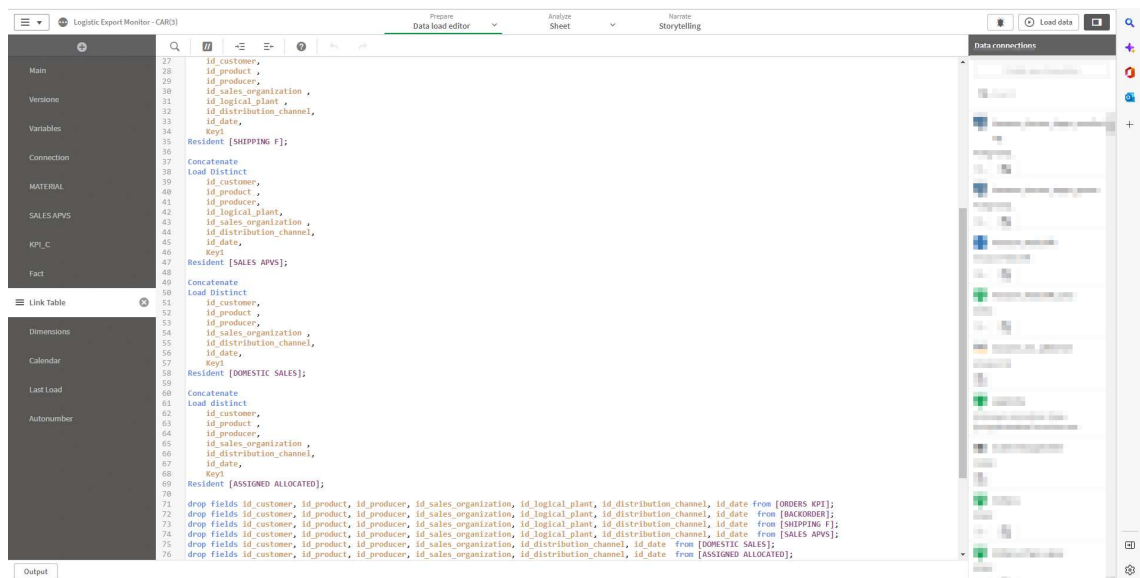
```

1 [ORDERS KPI]:
2 LOAD
3 *,
4 id_customer &'| '& id_product &'| '& id_producer &'| '& id_sales_organization &'| '& id_distribution_channel &'| '& id_date &'| '& id_logical_plant as Key1;
5
6
7 LOAD
8   id_dat_delivery
9   id_dat_delivery,           as id_date,
10  id_sales_organization,
11  id_distribution_channel,
12  id_customer,
13  id_product,
14  id_producer,
15  id_logical_plant,
16  amt_ordered_qty           as "Ordered Qty",
17  amt_confirmed_qty        as "Confirmed Qty (Conf. Date)"

```

Figura 6.13: Creazione della chiave per la Link Table

Questo nuovo campo *Key1* diventa la nuova chiave primaria che collega le tabelle dei fatti con le tabelle delle dimensioni nella tabella d'appoggio Link Table. Il passaggio finale per la corretta implementazione di questa struttura consiste nella cancellazione dei campi *ID* dalle tabelle dei fatti in modo che non si vadano a creare le chiavi sintetiche. Nella Figura 6.14 viene mostrata l'apposita section in cui tutti questi campi vengono caricati all'interno di tabelle d'appoggio per poi essere cancellati tramite l'utilizzo dell'istruzione *DROP FIELDS*.



```

27 id_customer,
28 id_product,
29 id_producer,
30 id_sales_organization,
31 id_logical_plant,
32 id_distribution_channel,
33 id_date,
34 Key1
35 Resident [SHIPPING F];
36
37 Concatenate
38 Load Distinct
39 id_customer,
40 id_product,
41 id_producer,
42 id_logical_plant,
43 id_sales_organization,
44 id_distribution_channel,
45 id_date,
46 Key1
47 Resident [SALES APVS];
48
49 Concatenate
50 Load Distinct
51 id_customer,
52 id_product,
53 id_producer,
54 id_sales_organization,
55 id_distribution_channel,
56 id_date,
57 Key1
58 Resident [DOMESTIC SALES];
59
60 Concatenate
61 Load Distinct
62 id_customer,
63 id_product,
64 id_producer,
65 id_sales_organization,
66 id_distribution_channel,
67 id_date,
68 Key1
69 Resident [ASSIGNED ALLOCATED];
70
71 drop fields id_customer, id_product, id_producer, id_sales_organization, id_logical_plant, id_distribution_channel, id_date from [ORDERS KPI];
72 drop fields id_customer, id_product, id_producer, id_sales_organization, id_logical_plant, id_distribution_channel, id_date from [BACKORDER];
73 drop fields id_customer, id_product, id_producer, id_sales_organization, id_logical_plant, id_distribution_channel, id_date from [SHIPPING F];
74 drop fields id_customer, id_product, id_producer, id_sales_organization, id_logical_plant, id_distribution_channel, id_date from [SALES APVS];
75 drop fields id_customer, id_product, id_producer, id_sales_organization, id_distribution_channel, id_date from [DOMESTIC SALES];
76 drop fields id_customer, id_product, id_producer, id_sales_organization, id_distribution_channel, id_date from [ASSIGNED ALLOCATED];

```

Figura 6.14: Drop degli ID dalle tabelle dei fatti

## 6.2.2 Visualizzazione dei dati

Conclusa la creazione del modello dei dati è possibile procedere con la creazione dei report. Le visualizzazioni dovranno seguire il più fedelmente possibile quanto veniva riportato nelle dashboard Obiee. Per l'applicazione in analisi sono stati sviluppati due fogli. In entrambi è stato utilizzato l'oggetto *container* di Qlik Sense. Il contenitore permette di aggiungere



visualizzazioni in uno spazio limitato. Queste ultime possono essere mostrate o nascoste in base alle condizioni definite. Si è deciso di utilizzare questo strumento poichè, come si vede dalla Figura 6.15, il report presenta tante tabelle per quante sono le country verso cui l'azienda esporta pneumatici. Creare una tabella per ogni country avrebbe reso l'applicazione molto più pesante e meno intuitiva da analizzare; in questo modo, invece, l'utente può passare da una country all'altra restando nello stesso report. L'area rossa in figura evidenzia le varie country analizzate nel report a cui, a ciascuna delle quali, corrisponde una tabella diversa. I dati vengono mostrati in base all'anno e al filtro selezionato.

Export Type	Region	Area	Market Code	Market Descr	DC Code	Q1	Q2	Q3	Q4	YTD	YTD%		
Totals						139,391	49,970	27,691	14,276	109,183	42,712	23,638	18,646
CH-GREATER CHINA	AP-APAC	CH-GREATER CHINA	608	CHINA	1	5,318	0	417	48	1,158	1,693	490	480
CH-GREATER CHINA	AP-APAC	CH-GREATER CHINA	608	CHINA	2	2,450	0	0	0	574	1,844	608	608
CH-GREATER CHINA	AP-APAC	CH-GREATER CHINA	627	HONG KONG	1	381	166	191	107	108	0	0	0
JP-JAPAN	AP-APAC	JP-JAPAN	611	JAPAN	1	879	0	271	232	875	438	438	438
KR-SOUTH KOREA	AP-APAC	KR-SOUTH KOREA	619	SOUTH KOREA	1	648	332	2	0	703	0	0	0
KR-SOUTH KOREA	AP-APAC	KR-SOUTH KOREA	619	SOUTH KOREA	2	5,781	0	7,782	50	8,844	62	718	718
PC-PACIFIC	AP-APAC	PC-PACIFIC	524	OTHER PACIFIC	1	1	0	0	0	0	0	0	0
PC-PACIFIC	AP-APAC	PC-PACIFIC	610	NEW ZEALAND	1	67	27	0	0	0	0	0	0
PC-PACIFIC	AP-APAC	PC-PACIFIC	614	AUSTRALIA	1	2,193	0	184	182	977	-40	-40	0
SS-SOUTH EAST ASIA	AP-APAC	SS-SOUTH EAST ASIA	628	TAIWAN	1	482	152	639	353	377	0	0	0
SS-SOUTH EAST ASIA	AP-APAC	SS-SOUTH EAST ASIA	613	OTHER ASIA	1	23	0	0	0	0	0	0	0
SS-SOUTH EAST ASIA	AP-APAC	SS-SOUTH EAST ASIA	620	PHILIPPINE	1	2	2	0	0	0	0	0	0
SS-SOUTH EAST ASIA	AP-APAC	SS-SOUTH EAST ASIA	625	SINGAPORE	1	389	0	1	1	411	0	0	0
SS-SOUTH EAST ASIA	AP-APAC	SS-SOUTH EAST ASIA	626	THAILANDIA	1	144	18	527	203	288	0	0	0
SS-SOUTH EAST ASIA	AP-APAC	SS-SOUTH EAST ASIA	647	MALAYSIA	1	53	9	0	0	0	0	0	0
SS-SOUTH EAST ASIA	AP-APAC	SS-SOUTH EAST ASIA	648	INDONESIA	1	21	9	0	0	90	0	0	0
IC_Eu	EU-EUROPE	EC-EU-CENTRAL ACCOUNTS	581	EU CENTRAL ACCOUNTS	1	613	0	-58	-58	43,703	5,840	4,475	904
IC_Eu	EU-EUROPE	EE-EAST EUROPE	601	POLAND	1	1,954	881	0	0	3,472	200	828	200
IC_Eu	EU-EUROPE	EE-EAST EUROPE	602	HUNGARY	1	1,260	0	0	0	0	0	0	0
IC_Eu	EU-EUROPE	EE-EAST EUROPE	603	CZECH REPUBLIC	1	1,190	0	0	0	44	0	0	0
IC_Eu	EU-EUROPE	EE-EAST EUROPE	605	SLOVAKIA	1	39	1,251	0	0	0	0	0	0
IC_Eu	EU-EUROPE	EE-EAST EUROPE	668	MONTENEGRO	1	12	0	0	0	8	0	0	0
IC_Eu	EU-EUROPE	EE-EAST EUROPE	679	CYPRUS	1	11	0	0	0	0	0	0	0
IC_Eu	EU-EUROPE	EE-EAST EUROPE	680	ALBANIA	1	27	0	0	0	0	0	0	0
IC_Eu	EU-EUROPE	EE-EAST EUROPE	681	BOSNIA	1	24	0	0	0	75	9	0	0
IC_Eu	EU-EUROPE	EE-EAST EUROPE	682	RUSSIA	1	23	0	0	0	92	50	0	0
IC_Eu	EU-EUROPE	EE-EAST EUROPE	683	CROAZIA	1	38	0	0	0	0	0	0	0
IC_Eu	EU-EUROPE	EE-EAST EUROPE	683	CROAZIA	2	0	0	0	0	84	0	0	0

Figura 6.15: Report di Export Monitor

Il secondo foglio in Figura 6.16, invece, mostra gli stessi dati di quello appena descritto ma condensa il contenuto informativo solo sulle country appartenenti all'Europa.

Export Type	Region	Area	Market Code	Market Descr	DC Code	Q1	Q2	Q3	Q4	YTD	YTD%			
Totals						139,391	49,970	27,691	14,276	29%	41,712	23,638	18,646	37%
CA-CANADA						1,474	1,824	54	51	3%	6	0	0	0%
CH-GREATER CHINA						8,149	165	608	156	95%	3,541	1,098	1,098	665%
IC_Eu						74,907	44,734	2,339	2,030	9%	27,729	17,137	11,393	23%
IC_extra EU						748	2,081	3,351	967	46%	388	388	388	19%
JP-JAPAN						879	0	271	232		438	438	438	
KR-SOUTH KOREA						6,439	332	7,754	50	15%	62	718	718	216%
MX-MEXICO						748	290	801	412	142%	568	-10	0	-3%
NO-NORDICS						24,781	282	10,816	9,391	3350%	8,863	3,397	2,707	960%
PC-PACIFIC						2,265	27	184	182	674%	-40	-40	0	0%
RU-RUSSIA & C.I.S.						374	45	0	0	0%	0	0	0	0%
SA-SOUTH AFRICA						932	0	248	247		0	0	0	
SS-SOUTH EAST ASIA						1,114	190	1,167	557	259%	0	0	0	0%
US-U.S.A.						16,149	0	6	0		912	912	1,914	

Figura 6.16: Report di Export Monitor - EU

---

## Discussione in merito al lavoro svolto

---

*Nel presente capitolo verrà presentata una discussione in merito al lavoro svolto. In particolare, si analizzeranno i punti di forza e i punti di debolezza che caratterizzano il progetto. Il capitolo si concluderà poi con la descrizione delle lezioni apprese nell'ambito del lavoro svolto.*

### 7.1 Punti di Forza

Si conclude, con il precedente capitolo, la descrizione del progetto in analisi e del lavoro svolto nella reingegnerizzazione di una piattaforma di Business Intelligence. Si procederà, ora, con una discussione critica di quanto svolto all'interno del progetto. Tale discussione è volta a mettere in luce i punti di forza e i punti di debolezza osservati durante il lavoro. Per punti di forza, generalmente, si intendono fattori interni all'azienda che possono essere controllati e che determinano il successo di un progetto.

#### 7.1.1 Progettazione ben definita

Come già introdotto all'interno della trattazione, uno dei punti di forza più importanti per il corretto proseguo del progetto sta nella fase di progettazione. Per la complessità e le dimensioni del progetto, il contratto stipulato prevede una durata delle attività lavorative di alcuni anni. Quando si ha a che fare con progetti di tali dimensioni vengono, necessariamente, utilizzate molte figure diverse. Nel caso in analisi il team operativo si compone di circa quaranta figure professionali. Oltre alla numerosità, è bene ricordare che non tutto il team sia composto da personale della stessa azienda di consulenza poichè, per necessità progettuali, sono state coinvolte anche terze parti (aziende esterne) per la delega di alcune attività e per ridurre il carico di lavoro complessivo. Le figure manageriali, responsabili del progetto, hanno definito una roadmap molto stringente ma, allo stesso tempo, ben strutturata.

Uno dei punti di forza maggiore sta, sicuramente, nella definizione delle milestones e dei pattern di sviluppo per il back-end e di front-end. Tali pattern permettono allo sviluppatore di avere chiaro quali sono le procedure e quale è il risultato che il cliente si aspetta per la chiusura degli sviluppi. La dettagliata organizzazione delle attività è data anche dalla creazione di documenti centralizzati, ai quali hanno accesso le risorse del tema di sviluppo, dove nel dettaglio vengono riportate le attività da svolgere.

Su questi documenti, ogni risorsa riporta lo status di update dei lavori sulle attività in carico ed eventuali criticità che possono provocare ritardi nei rilasci. Molta importanza, infine, viene data al fattore comunicativo. Tutto il team deve essere allineato sulle attività in corso



d'opera e sulle possibili problematiche. Questo allineamento permette che le risorse possano supportarsi reciprocamente in maniera efficiente ed evita la sospensione degli sviluppi qualora una risorsa risulti indisponibile per un certo lasso di tempo.

In conclusione, la progettazione così dettagliata delle attività di sviluppo e, se necessario di recupero, è il punto di forza maggiore che permette la consegna dei deliverable e il corretto svolgimento delle attività. Tali tecniche di definizione delle attività e gestione del team derivano, in maniera preponderante, dalla grande esperienza pregressa delle figure manageriali coinvolte nel progetto altamente qualificate per questi contesti di tali dimensioni.

### 7.1.2 Utilizzo di tecnologie innovative

Il progetto di reingegnerizzazione della piattaforma di Business Intelligence permette allo sviluppatore di interfacciarsi con diversi software, lato back-end e front-end, innovativi e utilizzati su larga scala nel mondo del lavoro. Per quanto riguarda il back-end le tecnologie più interessanti utilizzate sono *Talend* e *Amazon AWS*. Tutto il processo di migrazione dei dati dal DWH alla nuova piattaforma dati avviene grazie al software di data integration *Talend*. Tale software permette di definire il processo di unione di dati provenienti da più sorgenti in una vista unificata effettuando, inoltre, operazioni di trasformazione ed elaborazione sui dati stessi. Tale software, ad oggi, è uno dei tool più potenti ed utilizzati nel mondo della *data analytics* per quanto riguarda la parte di *data integration*.

Lavorare ad un progetto di tali dimensioni e complessità permette la conoscenza profonda di questi strumenti e quindi un vantaggio dal punto di vista delle competenze personali. Lato front-end, invece, la tecnologia più utilizzata è sicuramente *Qlik Sense*. Si fa riferimento ad uno dei principali tool a livello mondiale per lo sviluppo di analytics visive. La complessità del progetto ha permesso ad alcuni sviluppatori di lavorare a stretto contatto direttamente con l'azienda produttrice di *Qlik Sense* per mettere in campo specifiche soluzioni per alcune casistiche progettuali. Tutto ciò, unito all'utilizzo continuo del software, permette di ottenere una conoscenza di primo livello facilmente spendibile nel mondo del lavoro.

### 7.1.3 Team eterogeneo e altamente preparato

Un team di lavoro eterogeneo è composto da membri con diversi background, esperienze e competenze. Questa diversità può essere una risorsa preziosa per il team, poiché ogni membro può apportare il proprio contributo unico e avere una prospettiva diversa sui problemi e su come risolverli. Un team eterogeneo può essere particolarmente efficace quando è anche preparato. Ciò significa che ogni membro del team ha le competenze e le conoscenze necessarie per svolgere al meglio il proprio lavoro e contribuire alla missione del team. Inoltre, lavorare in un team eterogeneo e preparato può essere estremamente gratificante perché ogni membro può imparare dalle esperienze e dalle conoscenze degli altri. In questo modo, il team può crescere e migliorare insieme, diventando sempre più forte e capace di affrontare sfide nuove e impegnative. In conclusione, un team di lavoro eterogeneo e preparato rappresenta una combinazione vincente che può portare a risultati eccezionali. Con la diversità di vedute e di competenze, il team può trovare soluzioni creative ai problemi, mentre la preparazione dei membri garantisce che queste soluzioni vengano attuate in modo efficiente e accurato.

## 7.2 Punti di debolezza

I punti di debolezza di un progetto possono rappresentare una sfida importante per il team di lavoro. Tuttavia, riconoscere e affrontare questi punti di debolezza può aiutare a garantire il successo del progetto.

### 7.2.1 Scarsa conoscenza dei dati

In un progetto di reingegnerizzazione di una piattaforma di Business Intelligence è di fondamentale importanza che il contenuto informativo non subisca alterazioni durante il processo di migrazione dal DWH alla nuova piattaforma dati. Spesso, però, alcune logiche vengono modificate e per questo è possibile che alcuni dati subiscano delle modifiche. Una buona conoscenza intrinseca del dato o della misura/KPI da parte dello sviluppatore consentirebbe la rilevazione anticipata delle squadrature o di valori inverosimili.

Nel caso analizzato nella trattazione, la conoscenza del dato da parte dello sviluppatore è quasi nulla. I dati che vengono analizzati fanno riferimento a specifiche tecniche relative al settore della produzione degli pneumatici. Da parte dello sviluppatore esterno all'azienda cliente è difficile fornire una corretta interpretazione di ciò che si va a mostrare nelle visualizzazioni. Come unico metro di paragone si hanno i report Obiee e la possibilità di interfacciarsi direttamente con il cliente e gli utenti del business. Avere una scarsa conoscenza dei dati può portare a decisioni errate che possono compromettere la qualità del lavoro e il successo del progetto. Un possibile spunto di miglioramento sarebbe quello di ricevere dal cliente, per ogni applicazione da sviluppare, una documentazione dettagliata sulle dimensioni e sulle misure. In questa documentazione potrebbero essere presenti delle brevi descrizioni che permetterebbero allo sviluppatore di avere più conoscenza dell'informazione che sta trattando e, quindi, capire in anticipo se un certo valore o una certa visualizzazione sono sensate o meno.

### 7.2.2 Analisi e raccolta dei requisiti incomplete

L'analisi e la raccolta dei requisiti incomplete in un progetto possono rappresentare una sfida importante per il team di lavoro. I requisiti sono le specifiche che descrivono ciò che il progetto deve fare e come deve essere realizzato. Le cause di un'analisi e una raccolta dei requisiti incomplete possono essere diverse. Ad esempio, il team di lavoro potrebbe non avere una comprensione adeguata delle esigenze del cliente o dell'utente finale. Inoltre, potrebbe esserci una mancanza di comunicazione tra il team di lavoro e il cliente o l'utente finale, il che può portare a una mancata comprensione delle loro esigenze. Nel caso in analisi spesso si è riscontrato questo tipo di problematica che si ripercuote poi sui corretti sviluppi delle applicazioni dal lato tecnico. La corretta raccolta dei requisiti e delle specifiche tecniche permette, infatti, allo sviluppatore di concentrarsi esclusivamente sul lato tecnico dello sviluppo. Quando, invece, queste informazioni non sono corrette, o sono comunque parziali, è necessario ripetere ulteriormente il lavoro di raccolta dei requisiti e, quindi, spendere tempo che può compromettere la qualità del lavoro e il successo del progetto.

### 7.2.3 Modifica delle richieste e delle priorità del cliente

I cambiamenti nelle richieste del cliente possono portare a ritardi e problemi di budget, mentre i cambiamenti nelle priorità possono modificare l'intero piano di lavoro.

Le modifiche delle richieste e delle priorità dal cliente possono essere dovute a diverse cause. Ad esempio, il cliente potrebbe non avere una comprensione adeguata delle conseguenze di questi cambiamenti o potrebbe avere esigenze che cambiano nel corso del tempo. Inoltre, potrebbero esserci cambiamenti nell'ambiente di lavoro o nella situazione del cliente che richiedono un adeguamento delle richieste e delle priorità.

Le modifiche delle richieste e delle priorità dal cliente possono essere frustranti per il team di lavoro e possono portare a ritardi e problemi di budget. Tuttavia, è importante che il team di lavoro sia flessibile e collaborativo e che lavori con il cliente per trovare soluzioni che soddisfino le sue esigenze senza compromettere il successo del progetto.

In conclusione, le modifiche delle richieste e delle priorità dal cliente possono rappresentare una sfida per il team di lavoro, ma lavorare in modo collaborativo con il cliente può aiutare a superare questa sfida.

## 7.3 Lezioni Apprese

Le lezioni apprese in un progetto rappresentano una parte importante del processo di miglioramento continuo. Rifletterci a fine progetto può aiutare il team di lavoro a imparare da eventuali errori e a trovare modi per fare meglio la prossima volta.

Le lezioni apprese possono riguardare una varietà di argomenti, come la pianificazione, la gestione del team, la gestione del budget e la comunicazione. Ad esempio, il team potrebbe aver scoperto che la pianificazione accurata è stata cruciale per il successo del progetto o che la comunicazione efficace è stata fondamentale per evitare problemi. Di seguito si analizzeranno due punti, tra i più importanti, sui quali è stata data molta attenzione e che sono tra le principali cause che portano al completamento del progetto in maniera efficiente.

### 7.3.1 Importanza del processo di validazione dati

La validazione dei dati è l'ultima fase del processo di sviluppo di un'applicazione prima del rilascio. Spesso si tende a sottostimare il tempo da dedicare a questo passaggio a causa delle stringenti scadenze quando, in realtà, necessiterebbe di molta più attenzione. Il processo di validazione dei dati si può compiere in due modi; nel caso in cui si avesse a disposizione il report Obiee da sostituire o l'applicazione Qlik Sense dualrun, allora si procede con un confronto tra i valori delle nuove dashboard con quelli di riferimento. Questo tipo di validazione è però approssimativa e non garantisce la bontà del risultato finale poichè, come già descritto nella trattazione, spesso le logiche di traduzione dal DWH alla piattaforma dati subiscono modifiche. Nella maggior parte dei casi, quindi, il confronto one-to-one con i dati già esistenti non è sufficiente. Tramite questo tipo di confronto è comunque possibile avere una prima idea sull'ordine dei grandezza o su cosa bisogna aspettarsi.

Il test più importante che garantisce la validità di ciò che si sta mostrando è l'*integration test*. Si riproducono le misure calcolate a front-end attraverso le query SQL che sono state utilizzate nelle fasi di ETL01 e ETL02. Queste query permettono di riprodurre la misura ed estrarre il valore direttamente dal database. Questo tipo di test permette di confermare che tutto ciò che si sta mostrando nella reportistica è coerente con quanto presente su database. Una volta che l'app sarà rilasciata, nel caso si presentassero delle segnalazioni, lo sviluppatore sarà più facilitato nella risoluzione della problematica e potrà garantire all'utente l'allineamento tra database e dashboard. Questo tipo di test può richiedere più di una giornata lavorativa ma, se di esito positivo, permette poi di evitare la lunga coda di lavoro che si crea post rilascio quando l'applicazione non è stata correttamente testata e, quindi, i dati mostrati non vengono validati dal cliente.

### 7.3.2 Attenzione alle performance

La maggior parte delle applicazioni che vengono create sono formate da più processi di ETL che estraggono ed elaborano dati da tabelle delle dimensioni e dei fatti popolate da milioni di record. In generale, un'applicazione Qlik Sense, per completare il processo di caricamento dei dati può impiegare più di un'ora di tempo in totale. Il cliente, in base alle analisi da effettuare o per il significato stesso di una specifica informazione, può segnalare l'esigenza di avere a disposizione i dati aggiornati di una certa applicazione ad uno specifico orario.

Come già analizzato nei precedenti capitoli, inoltre, non solo il tempo di caricamento sono influenzati dalle dimensioni dei dati di una certa applicazione, ma anche l'usabilità stessa dell'app. Per questa serie di motivi e per altri, durante il processo di sviluppo di un'applicazione è necessario fare attenzione a massimizzare le performance e rendere più efficiente possibile il processo di caricamento. Le figure manageriali esperte a livello tecnico si occupano periodicamente di creare delle brevi lezioni in cui vengono forniti suggerimenti tecnici e si cerca di fare in modo che gli sviluppatori seguano sempre più questo tipo di modello per lo sviluppo di un'applicazione.

Alcune delle tecniche più utilizzate per il miglioramento delle performance che sono state analizzate sono:

- *Utilizzo dei QVD*: la lettura di dati da un file QVD è in genere 10-100 volte più veloce rispetto alla lettura da altre sorgenti di dati.
- *Autonumber*: questa funzione dello script restituisce un valore intero univoco per ciascun valore calcolato distinto del parametro rilevato durante l'esecuzione dello script. Essa può essere utilizzata per creare una rappresentazione compatta di memoria che rappresenta una chiave complessa. In particolare, viene utilizzata per le chiavi primarie delle link table spesso formate da molti campi concatenati. Questa funzione può ridurre i caricamenti di un'app anche di 10 minuti.
- *Where conditions*: possibile (sulla base delle esigenze di visualizzazione) si cerca di inserire una condizione di *where* che possa limitare il caricamento dei dati da una tabella.
- *Preceding Load*: si tratta di un'istruzione *LOAD* che viene caricata da un'istruzione *LOAD* sottostante (Figura 7.1), senza la specifica di un qualificatore di espressione quale *FROM* o *RESIDENT* come si farebbe normalmente. In questo modo è possibile impilare qualsiasi numero di istruzioni *LOAD*. L'istruzione in fondo viene valutata per prima, quindi viene valutata l'espressione sopra e, così via, fino ad arrivare alla valutazione dell'istruzione nella parte superiore. Questa procedura aumenta le performance poichè il caricamento dei dati viene lasciato alla *LOAD* sottostante mentre le trasformazioni dei dati viene demandato all'espressione precedente.

```
32
33 SimplePreceding:
34 LOAD
35 *,
36     [To Date] - [From Date] as Duration
37 ;
38 LOAD
39     Date(Date#(FromDate, 'YYYYMMDD'), 'DD MMM YYYY') as [From Date],
40     Date(Date#(ToDate, 'YYYYMMDD'), 'DD MMM YYYY') as [To Date]
41 FROM ..\SourceData\Durations.csv
42 (txt, codepage is 1252, embedded labels, delimiter is ',', msq);
43
```

Figura 7.1: Esempio di Preceding Load

In questa trattazione si è discusso il progetto di reingegnerizzazione di una piattaforma di Business Intelligence per un'azienda leader nel settore della produzione di pneumatici. Prima di tutto è stata descritta una panoramica generale sulla definizione di re-engineering, fornendo alcuni cenni storici per poi concentrarsi nello specifico nel caso in analisi della piattaforma di Business Intelligence. Si è passati, poi, alla descrizione della piattaforma su cui viene effettuato il processo di reingegnerizzazione, ovvero Oracle Obiee. È stata proposta una descrizione del Data Warehouse in via di dismissione e delle principali funzionalità che Obiee mette a disposizione per la creazione delle analisi e delle visualizzazioni. Successivamente, è stata trattata l'analisi dei requisiti, in cui sono stati descritti i processi che portano alla raccolta delle informazioni per lo sviluppo delle applicazioni relative alla nuova piattaforma di Business Intelligence. Riguardo a questa fase è stata mostrata in dettaglio come avviene la creazione di due dei documenti più importanti per il processo di re-engineering, i documenti di Reverse ed i Business Glossary.

Il Capitolo 4 ha trattato la tematica relativa alla progettazione delle attività. Si è parlato, quindi, della metodologia di sviluppo Agile e delle diverse fasi progettuali quali avvio, pianificazione ed esecuzione. In particolare, è stata fatta un'analisi approfondita su tutto ciò che riguarda lo sviluppo front-end, focus della trattazione e del lavoro svolto. Nel Capitolo 5 sono state presentate le linee guida per il corretto sviluppo di un'applicazione Qlik Sense, a partire dal recupero dei dati attraverso opportune fasi di ETL, per concludere, poi, con il processo di validazione e il rilascio verso gli utenti finali.

Successivamente, nel Capitolo 6, sono state mostrate due applicazioni Qlik Sense illustrandone nel dettaglio il processo di caricamento e le dashboard create appositamente per le analisi. Nel Capitolo 7, infine, è stata fatta una discussione sul lavoro svolto analizzando punti di forza e debolezza del progetto per concludere evidenziando le nozioni acquisite.

La partecipazione a questa attività lavorativa permette di comprendere come approcciarsi ad un progetto di tali dimensioni e quali sono le tecnologie innovative nell'ambito della Business Intelligence a cui le aziende leader fanno riferimento.

Questo progetto, sicuramente, può essere migliorato sotto il punto di vista della conoscenza della natura del dato. Lo sviluppatore si trova ad elaborare informazioni ed applicare regole di calcolo a dati di cui non conosce realmente il significato e questo provoca errori che solo l'utente finale può segnalare. Molte delle nuove applicazioni Qlik Sense mostrano dati sotto forma tabellare che poi vengono estratti ed analizzati dall'utente attraverso l'utilizzo di Excel.

Come possibile sviluppo futuro sarebbe opportuno cercare di modellare il più possibile le dashboard sulla base delle esigenze dell'utente finale, così da incentivare maggiormente il

processo di analisi direttamente su Qlik Sense migliorando, inoltre, le competenze tecniche su tale nuova piattaforma.

- ANGELI, F. (2010), «L'analisi dei requisiti : processo tecnico o gestionale», *Project Manager. Fascicolo 3*, 2010).
- ARNOLD, R. S. (1994), «Software reengineering: a quick history», *Communications of the ACM*.
- ATZENI, P., CERI, S., FRATERNALI, P., PARABOSCHI, S. e TORLONE, R. (2018), *Basi di dati*, McGraw-Hill.
- BEAULIEU, A. (2020), *Learning SQL: Generate, Manipulate, and Retrieve Data*, O'Reilly.
- BERGAMASCHI, F., BIANCONI, D. e MATTAVELLI, A. (2022), *Business intelligence per le PMI. Manuale per professionisti e imprenditori*.
- BERGMANN, T. e KARWOWSKI, W. (2019), «Agile Project Management and Project Success: A Literature Review», *Part of the Advances in Intelligent Systems and Computing book series (AISC, volume 783)*.
- BLOKDYK, G. (2020), *Qlik A Complete Guide - 2020 Edition*, The art of Service.
- BOWEN, J. (2012), *Getting Started with Talend Open Studio for Data Integration*, Packt Publishing.
- BULUSU, L. (2016), «Using OBIEE for Operational BI», *Oracle Business Intelligence and Essbase Solutions Guide*.
- CALDWELL, G. (2020), *Agile Project Management: The Complete Guide for Beginners to Scrum, Agile Project Management, and Software Development*.
- EARLY, S. (2019), «The Role of a Customer Data Platform», *IT Professional ( Volume: 20, Issue: 1, January/February 2018)*.
- FRACHERI, S. (2020), *Business Process Reengineering. Una guida pratica per mappare e reingegnerizzare i processi aziendali*, Franco Angeli.
- GOLFARELLI, M. e RIZZI, S. (2006), *Data Warehouse. Teoria e pratica della progettazione*, McGraw-Hill.
- JENNINGS, R. (2019), *Agile Project Management: The Ultimate Step By Step Guide to Learn Agile Project Management to Complete Your Goals with Maximum of Results*.
- JOHNSTON, G. (2017), *Business Process Re-engineering: A Simple Process Improvement Approach to Improve Business Performance*.

- KIHN, M. e O'HARA, C. (2021), *Customer Data Platforms: Use People Data to Transform the Future of Marketing Engagement*.
- LABBE, P., ANJOS, C., SOLANKI, K. e DIMASO, J. (2022), *Hands-On Business Intelligence with Qlik Sense: Implement self-service data analytics with insights and guidance from Qlik Sense experts*, Packt Publishing.
- MAHLER, M. (2022), *Mastering Qlik Sense: Expert techniques on self-service data analytics to create enterprise ready Business Intelligence solutions*, Packt Publishing.
- MORA, J. M. L. (2020), «Qlik sense implementation: dashboard creation and implementation of the test performance methodology», *Diss.*
- NAHT, I. (2020), *A Software Re-engineering Approach for Efficient Requirements Analysis: SE3 three-degree freedom to software requirements management and release management*, Lambert Academic Publisher.
- O'DONOVAN, M. (2019), *Qlik Sense for Beginners*, [www.TECHSTUFFY.com](http://www.TECHSTUFFY.com).
- RADA, R. (1999), *Re-Engineering Software: How to Re-Use Programming to Build New, State-of-the-Art Software*, Routledge.
- REIS, J. e HOUSLEY, M. (2022), *Fundamentals of Data Engineering: Plan and Build Robust Data Systems*, O'Reilly.
- SHUKLA, A. e DHIR, S. (2016), «Tools for Data Visualization in Business Intelligence: Case Study Using the Tool Qlikview», *Advances in Intelligent Systems and Computing*.
- SOLANKI, K. (2022), *Implementing Qlik Sense: Design, Develop, and Validate BI solutions for consultants*, Packt Publishing.
- TROYANSKY, O., GIBSON, T., LEICHTWEIS, C. e BJORK, L. (2015), *QlikView Your Business: An Expert Guide to Business Discovery With Qlikview and Qlik Sense*, Wiley.
- VAKOLA, M. e REZGUI, Y. (2000), «Critique of existing business process re-engineering methodologies: The development and implementation of a new methodology», *Business Process Management Journal*.
- VASSILIADIS, P., SIMITSIS, A. e SKIADOPOULOS, S. (2002), «Conceptual modeling for ETL processes», *DOLAP '02: Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP*.
- WARD, A., SCREEN, C. e KHAN, H. (2017), *Oracle Business Intelligence Enterprise Edition 12c: Build your organization's Business Intelligence system*.
- WILCOX, T., JIN, N., FLACH, P. e THUMIM, J. (2019), «A Big Data platform for smart meter data analytics», *Computers in Industry*.
- WONG, R. (2018), *Mastering Reverse Engineering: Re-engineer your ethical hacking skills*, Packt Publisher.



### Siti web consultati

- Qlik – [www.qlik.com](http://www.qlik.com)
- Oracle Business Intelligence Enterprise Edition – [www.oracle.com](http://www.oracle.com)
- Google Scholar – [www.scholar.google.com/](http://www.scholar.google.com/)
- IEEE – [www.ieee.org](http://www.ieee.org)
- ScienceDirect – [www.sciencedirect.com](http://www.sciencedirect.com)
- SpringerLink – [www.springerlink.com](http://www.springerlink.com)
- Wikipedia – [www.wikipedia.org](http://www.wikipedia.org)

---

## Ringraziamenti

---

Al termine di questo elaborato, vorrei ringraziare tutte le persone che mi hanno sostenuto durante il mio percorso universitario.

Il primo ringraziamento va ai miei genitori che mi hanno permesso di portare a termine questo percorso in piena serenità. Il vostro incoraggiamento è stato ciò che mi ha maggiormente sostenuto nel conseguimento di questo obiettivo, dandomi forza e coraggio per attraversare i periodi di difficoltà.

Un ringraziamento speciale va ad Alessia, la mia ragazza. Grazie per tutto il tempo che mi hai dedicato. Grazie perché ci sei sempre stata.

Ringrazio tutti i colleghi universitari con cui ho superato i vari ostacoli e condiviso momenti di gioia nel raggiungimento dei traguardi.

Ringrazio di cuore i miei coinquilini: Luca, Paolo e Federico. È stato fantastico condividere questa esperienza insieme a voi.

Grazie alla mia tutor Valeria Marzana e a tutti i miei colleghi dell'azienda Deloitte, per avermi accolto e messo a mio agio e, soprattutto, per avermi insegnato tanto.

Vorrei ringraziare il Prof. Ursino, che con estrema professionalità e disponibilità mi ha aiutato a svolgere quest'esperienza di tirocinio, accompagnandomi, poi, nel lavoro di tesi.

Infine, un ringraziamento particolare va a mio fratello Alessandro. Il mio più fidato consigliere. Nonostante i chilometri che ci separano so che potrò sempre contare su di te.