

POLITECNICO DI TORINO



Corso di Laurea Magistrale in
INGEGNERIA MATEMATICA

Tesi di laurea Magistrale

Business Intelligence tramite piattaforma Qlik Sense: un caso di studio

Relatore:

Tania CERQUITELLI

Tutor Aziendale:

Sandro GATTI

Candidato:

Valentina MANZONE

19 Marzo 2021

Sommario

Introduzione.....	5
1 LA BUSINESS INTELLIGENCE	7
1.1 LA STORIA DELLA BUSINESS INTELLIGENCE	7
1.2 CARATTERISTICHE PRINCIPALI DELLA BUSINESS INTELLIGENCE	9
1.3 STRUTTURA DELLA BUSINESS INTELLIGENCE.....	10
1.3.1 BASE DATI.....	11
1.3.2 ETL	12
1.3.3 DATA WAREHOUSE	13
1.3.4 SISTEMI OLAP	17
1.3.5 REPORTISTICA.....	18
1.4 BUSINESS ANALYTICS	20
1.5 IL COMPITO DEL CONSULENTE.....	21
2 LA PIATTAFORMA QLIK SENSE	23
2.1 CONFRONTO CON IL MODELLO RELAZIONALE BASATO SU QUERY.....	27
2.2 COMPONENTI DI QLIK SENSE	28
2.3 CARICAMENTO DATI	31
2.4 CREAZIONE DEI REPORT.....	35
2.5 SELF-SERVICE DI QLIK SENSE	41
3 IL PROGETTO DI BUSINESS INTELLIGENCE SU QLIK SENSE.....	44
3.1 BASE DATI E PROCESSI DI ETL.....	45
3.2 ETL VENDITE	48
3.3 ANALISI VENDITE.....	49
3.5 CALCOLO PROVVIGIONI	52
3.5.0 PROCEDURA APPLICATIVA DEL CALCOLO DELLE PROVVIGIONI.....	52
3.5.1 LETTURA VENDITE	54
3.5.2 RECUPERO PROVVIGIONI A FRONTE DI RITIRO PERMUTA	57
3.5.3 CONTROLLO DELLA MAPPATURA.....	59
3.5.4 CALCOLO SUI SINGOLI STOCK_ID VENDUTI	62
3.5.5 PROVVIGIONI SUL SINGOLO VENDITORE.....	66
3.5.6 GESTIONE GAD DISCREZIONALE.....	68
3.5.6 STORICIZZAZIONE PROVVIGIONI.....	71
3.6 PERFORMANCE VENDITORE.....	71
3.6.1 CARICAMENTO DATI.....	72

3.6.2	PRESENTATION.....	73
3.7	TREND PERFORMANCE VENDITORE.....	75
3.7.1	CARICAMENTO DEI DATI	75
3.7.2	PRESENTATION.....	76
3.8	SCHEDULAZIONE TASK DI AGGIORNAMENTO E CARTELLE DEI PROGETTI	79
	Conclusioni.....	81
	Ringraziamenti	83
	Bibliografia	84

Introduzione

Il concetto di Business Intelligence (BI) comprende le attività, i processi e le competenze tecnologiche che rientrano nell'ambito del sistema informativo aziendale [1].

Quest'ultimo negli anni sta ricoprendo un ruolo sempre più importante dovuto alle esigenze dell'era dell'informazione e dell'evoluzione applicata all'economia e al management. Di conseguenza si è giunti ad un aumento dei dati disponibili per l'analisi e alla richiesta da parte del business di tempi di reazione inferiori con complessità crescenti.

Inoltre il sistema informatico ha il compito di risolvere problemi riguardo al patrimonio dei dati. I dati spesso provengono da fonti diverse ed eterogenee e quindi c'è il bisogno di unificare le strutture e i formati; i dati possono anche essere ridondanti, imprecisi e incompleti e c'è bisogno di operazioni di "pulizia" dei dati; oppure non sono facilmente fruibili e integrabili e quindi occorre trasformarli in informazioni di facile lettura per gli attori aziendali.

Il sistema di Business Intelligence, pertanto, ha lo scopo di valorizzare il patrimonio dei dati aziendali.

L'obiettivo di questa tesi è quindi quello di analizzare i passaggi fatti nella progettazione e sviluppo di un sistema di Business Intelligence.

Per comprendere meglio viene utilizzato, come caso di studio, un progetto su cui ho lavorato durante il mio percorso di tirocinio curriculare presso l'azienda Bios Management che si occupa di consulenza nel campo della direzione e organizzazione aziendale e di sistemi di Performance e Business Intelligence.

Nel primo capitolo viene data una panoramica generale sulla storia e le caratteristiche principali della BI. In particolare vengono descritti la procedura di caricamento dati (*ETL*), la basi dati che li contiene (*DATA WAREHOUSE*), i sistemi *OLAP* con la loro struttura a forma di cubo, ed infine la parte di *front-end* cioè la parte del sistema che gestisce l'interazione con l'utente.

Nel secondo capitolo invece viene presentato lo strumento di BI che viene utilizzato per il progetto in esame, *Qlik Sense*. Vengono quindi illustrate le principali caratteristiche e il confronto con il modello relazionale, i componenti di Qlik, lo schema di funzionamento del software e un accenno all'interfaccia grafica.

Nel terzo capitolo si entra nel cuore del progetto, presentando la struttura e quali sono i passaggi per creare l'output finale. Il progetto in esame riguarda la concessionaria "Gino S.p.A." e precisamente l'analisi del Calcolo delle Provvigioni e delle Performance dei venditori.

Capitolo 1

LA BUSINESS INTELLIGENCE

La prima definizione che si può dare per Business Intelligence è la seguente [1]:

la BI è un insieme di processi aziendali per raccogliere dati e trasformarli in informazioni utili per prendere decisioni.

In realtà la BI è molto di più, è un sistema di modelli, metodi, processi, persone e strumenti che per mezzo di elaborazioni, analisi e aggregazioni trasformano le informazioni in un formato più leggibile e comprensibile, utilizzabili come supporto alle decisioni strategiche, tattiche e operative per migliorare l'efficienza e la performance dell'azienda.

Inoltre attraverso Internet e altri strumenti di comunicazione è possibile, per tutti gli utenti, condividere e utilizzare le informazioni.

Tuttavia per arrivare al concetto di BI usato ai giorni nostri è stato necessario più di mezzo secolo.[2]

1.1 LA STORIA DELLA BUSINESS INTELLIGENCE

La prima volta che si trova il termine *Business Intelligence* è nel 1865 nell'opera "Cyclopedia of Commercial and Business Anecdotes" di Richard Millar Devens. Egli utilizzò tale termine per descrivere la capacità del banchiere Sir Henry Furnese, che raccogliendo informazioni politiche e di mercato, precedeva regolarmente la concorrenza. Furnese diventò ben presto conosciuto come il "finanziatore corrotto" perché utilizzava le sue conoscenze per fini non del tutto leciti, ma la sua idea di raccogliere informazioni sulle condizioni del Business ebbe notevole sviluppo in seguito.

Nel 1958 lo scienziato Hans Peter Luhn scriveva, nell'articolo "A Business Intelligence System", di un sistema dedicato alla trasmissione di informazioni nelle varie aree di organizzazione (scientifiche, industriali e governative). Infatti nella scia del boom dopo la seconda guerra mondiale tali settori richiedevano un modo per organizzare e semplificare i dati scientifici e tecnologici in rapida crescita. Per Luhn la BI era una strada per comprendere facilmente e velocemente una quantità enorme di informazioni per prendere le migliori decisioni possibili. Proprio per questo motivo Luhn è considerato oggi il "padre fondatore della Business Intelligence".

Nel periodo tra gli anni '50 - '70 il mondo del business iniziò a utilizzare il computer per la memorizzazione e archiviazione dei dati, in particolare nel 1956 fu inventato l'hard disk e furono creati anche altri strumenti tecnologici (floppy disk, laser disk, etc.). Questo portò alla nascita del primo database management system, un sistema software con lo scopo di creare e interrogare i database. Dal 1970 nascono anche i primi vendor di software BI che proponevano i loro tool capaci di accedere e organizzare i dati. Tuttavia si trattava di una tecnologia nuova e imprecisa, ancora difficile da usare.

La svolta è stata negli anni '80 - '90 quando iniziarono a nascere tool di BI specifici che utilizzavano i primi veri e propri data warehouse, cioè un grande insieme di dati organizzati in modo da essere condivisi e riutilizzabili. I tempi di accesso ai dati si ridussero notevolmente e i dati che inizialmente erano memorizzati in diversi database ora venivano storicizzati in un unico punto. Inoltre vennero adottati alcuni elementi che sono componenti fondamentali per la BI oggi: il tool ETL (Extract, Transform, Load) e il software OLAP (Online Analytical Processing).

Gli ultimi anni del '90 e inizio 2000 vennero battezzati come la fase "Business Intelligence 1.0". La BI iniziava a diventare conosciuta e sempre più vendor diventavano competitivi. Le principali caratteristiche erano produrre report e organizzare e visualizzare i dati in maniera adeguata. Tuttavia il business necessitava ancora del supporto IT (Information Technology) perché gli utenti non erano capaci di eseguire task da soli e inoltre il tempo per realizzare report era ancora elevato.

Nel nuovo millennio gli sviluppi tecnologici rendevano disponibili capacità di calcolo che pochi anni prima erano impensabili migliorando anche di velocità. Sono nate soluzioni tecnologicamente nuove come ad esempio processi real-time per prendere decisioni considerando i più recenti dati disponibili, oppure accessi self-service dove anche senza particolari competenze tecniche è possibile arrivare ad analizzare base dati non particolarmente complesse.

Oggi, sempre più piattaforme introducono le opzioni self-service in modo che gli utenti finali possano esplorare e utilizzare i dati in modo autonomo e la diffusione della tecnologia su supporti mobili modifica anche le richieste del Business che si orienta verso applicazioni grafiche con aggiornamenti spesso real-time. L'esigenza di strumenti

sempre più veloci ed economici viene soddisfatta attraverso il “Cloud BI” che ospita il software su Internet riducendo i costi di storicizzazione e rendendo l’accesso ai dati più conveniente e veloce.

La BI non è più solo un corredo aggiuntivo ma diventa fondamentale per le aziende se vogliono rimanere competitive.

Ai giorni nostri esistono numerosi strumenti di Business Intelligence che consentono alle imprese di individuare ed analizzare i dati interni ed esterni. Il progetto che viene presentato in questa tesi utilizza la piattaforma Qlik Sense.

1.2 CARATTERISTICHE PRINCIPALI DELLA BUSINESS INTELLIGENCE

Utilizzando la BI, le persone dell’azienda non devono preoccuparsi di reperire i dati e conservarli ma possono interagire con i dati stessi per ricavare informazioni. E’ di fondamentale importanza capire che la BI da sola non risolve i problemi ma servono le persone capaci di interpretare correttamente le informazioni.

Le principali caratteristiche di un sistema di BI sono [3]:

- Facilità d’uso e velocità: il sistema deve essere in grado di trattare grandi volumi di dati in un formato facile da leggere e veloce nei tempi di risposta.
- Integrazione e storicizzazione: i dati che provengono da fonti diverse, interne ed esterne all’azienda, devono essere integrati mantenendo la loro storia.
- Identificazione dei trend e delle anomalie: gli strumenti interattivi devono permettere di identificare i trend e le anomalie nei dati.
- Adattabilità nel tempo: la BI deve essere capace di affrontare e sfruttare le inevitabili evoluzioni tecnologiche trasformandole in opportunità di miglioramento aziendale.
- Indipendenza dall’IT: gli utenti devono essere in grado, da soli, di crearsi i propri report senza essere legati all’area IT (Self Service BI).
- Sicurezza: gli strumenti di BI devono garantire il controllo dell’accesso ai dati che sono spesso sensibili.

1.3 STRUTTURA DELLA BUSINESS INTELLIGENCE

Gli elementi che costituiscono la BI possono essere suddivisi in 3 macro aree come si può vedere dalla *Figura 1.1.* [1]

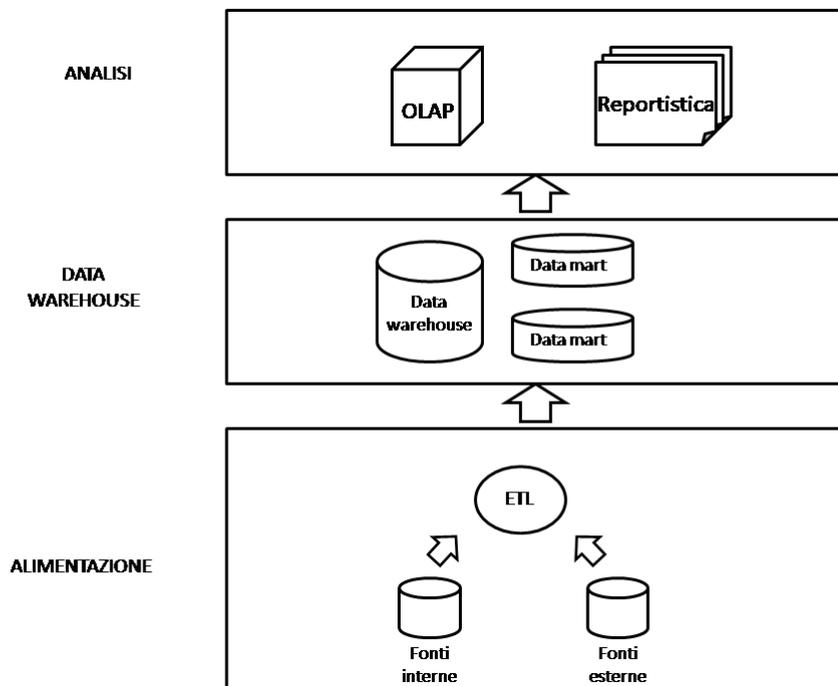


Figura 1. 1 Struttura della BI

La prima macro area è quella *dell'Alimentazione dei dati*. In molti casi le fonti dati provengono da diverse applicazioni che utilizzano software basati su tecnologie diverse, cioè database relazionali (Microsoft SQL Server, Oracle, MySQL, Microsoft Excel) e database non relazionali (txt, csv, xml).

L'utilizzo di più software porta a dover uniformare e rendere coerenti i dati e quindi si ricorre al processo di *ETL* per poi fornire i dati puliti al *data warehouse*. Quest'ultimo fa parte della seconda macro area da cui si parte per le attività di analisi.

E infine la terza macro area è quella dell'*analisi* in cui troviamo gli strumenti *OLAP* e i tool di *reportistica* che costituiscono il front-end cioè la fase di presentazione dei dati all'utente.

Di seguito sono presenti delle brevi descrizioni sugli elementi fondamentali delle macro aree che sono state accennate.

1.3.1 BASE DATI

Una base dati (database in inglese) è un insieme di dati correlati tra di loro utilizzando tecniche di modellazione dei dati e gestiti sulle memorie di massa tramite dei software [1].

Quest'ultimi sono i DBMS (Data Base Management System)

Un database deve avere tre caratteristiche per essere efficiente:

- Consistenza: i dati devono essere utili per le applicazioni dell'utente
- Sicurezza: i dati non devono essere danneggiati
- Integrità: gli utenti autorizzati a fare delle modifiche sui dati devono avere la garanzia di non provocare danni sul database.

Come è stato accennato sopra vi sono due tipi di database che possono contenere i dati usati poi nel Data warehouse, database relazionali e database non relazionali [4].

DATABASE RELAZIONALI

La struttura dei database relazionali è basata sul modello relazionale, costituito da una tabella in cui le colonne rappresentano degli attributi e le righe sono popolate dai dati che prendono il nome di record o tupla e a cui viene assegnato un ID univoco chiamato chiave. Ogni dato prende così valore se associato al suo attributo.

L'utilizzo delle tabelle garantisce un modo efficiente per conservare le informazioni e per accedervi. Inoltre il modello relazionale utilizza il linguaggio SQL (Structured Query Language) per poter accedere al database ed eseguire semplici operazioni come aggiungere, aggiornare o modificare i dati attraverso l'uso di query. L'utilizzo di operazioni logiche come il *join* che utilizza le chiavi esterne per collegare le tabelle è dovuto al fatto che le informazioni complesse vengono distribuite all'interno del database e ricomposte a richiesta dell'utente.

Uno dei concetti principali del modello relazionale è la normalizzazione. I dati sono smistati in tabelle proprio per evitare la ridondanza di informazioni che potrebbero causare problemi all'integrità dei dati e rallentamenti sulle operazioni di aggiornamento.

Tuttavia la frammentazione delle informazioni porta ad un rigido controllo sulle relazioni e sulla validità dei dati e attribuisce poca flessibilità al modello.

DATABASE NON RELAZIONALI

I database non relazionali, noti anche come NOSQL (Not Only SQL) si differenziano da quelli relazionali principalmente per la struttura; infatti i dati non sono più distribuiti in tabelle ma in documenti. Ogni documento raggruppa tutti i dati associati ad un'entità

come se fosse un oggetto e quindi le informazioni sono tutte disponibili nello stesso documento.

La caratteristica principale è appunto l'assenza di uno schema fisso, il documento contiene tutti i campi necessari senza essere definiti e quindi è possibile arricchire il documento liberamente.

Questo tipo di dati è molto usato nel caso di big data, cioè quando si ha una grande quantità di dati, per le caratteristiche di flessibilità e velocità dei database non relazionali.

1.3.2 ETL

I dati sorgenti, come è già stato accennato, vengono estratti e trasformati grazie al processo di ETL (Extract, Transform, Load) per poi essere caricati all'interno del data warehouse. [1]

Questo processo di alimentazione presenta 3 fasi, come lo stesso termine ETL preannuncia.

- **ESTRAZIONE:** questa fase consiste nel prelevare i dati dalle fonti che possono essere numerose ed eterogenee. A causa della mole di dati è necessario estrarre solo i dati utili per l'analisi in modo da ridurre i tempi di trasformazione dei dati. Ci sono due modalità di estrazione: una statica che prende una fotografia dei dati, oppure una selettiva che ad ogni caricamento estrae solo quei dati variati rispetto al precedente.
- **TRASFORMAZIONE:** è la fase più complessa perché si occupa di migliorare la qualità dei dati cioè realizzare un'unica versione dei dati ripulita, integrata e uniformata. Ad esempio fonti diverse potrebbero avere dati in formati diversi nel descrivere l'indirizzo di un cliente oppure avere valori diversi per il medesimo attributo, è quindi indispensabile procedere con un'operazione di pulitura. Partendo dai dati grezzi di difficile utilizzo forniti dall'estrazione si giunge ad un insieme di dati pronti per la fase finale del processo di ETL.
- **CARICAMENTO:** questa è l'ultima fase del processo ETL in cui i dati estratti e trasformati vengono caricati nel data warehouse. Il caricamento o il ricaricamento per aggiornare i dati avviene di solito nei periodi in cui gli utenti non sono operativi.

1.3.3 DATA WAREHOUSE

Quando si considerano grandi imprese che gestiscono una quantità molto grande di dati e pretendono una certa velocità nel ricevere i dati, l'utilizzo di database diventa insufficiente.

I limiti principali che presentano i sistemi operazionali sono i seguenti:

- Evoluzione dei dati: i dati cambiano in continuazione a causa delle operazioni effettuate e quindi non si possono considerare attendibili in un processo decisionale.
- Tempi di attesa delle interrogazioni: poiché i dati sono distribuiti in diverse tabelle, nel caso di interrogazioni complesse bisogna ricercare i dati su diverse tabelle e quindi ne risente la velocità della ricerca.
- Mancanza di dati storici: l'utilizzo dei database è valido nei processi riguardanti il momento attuale ma senza far riferimento al passato perché i dati non vengono conservati.
- Formati diversi: le diverse applicazioni possono usare tecnologie diverse.

Nasce quindi la necessità di utilizzare un sistema aziendale che utilizzi il data warehouse.

Il data warehouse è una base dati che ha lo scopo di memorizzare i dati provenienti dalle fonti operazionali lavorati dal processo di ETL e di essere il punto di partenza per le attività analitiche e le applicazioni di front-end [1]. Senza l'utilizzo del data warehouse i dati, non essendoci un unico punto dove salvarli, sarebbero sparsi in più basi dati e quindi l'analisi sarebbe difficile da implementare e i procedimenti di uniformazione e pulizia dovrebbero essere ripetuti per ogni analisi. Il data warehouse può essere utilizzato per una interrogazione diretta oppure per estrarre solo una parte dei dati creando dei *data mart*.

Il *data mart* è un raccogliitore di dati che riguarda un solo processo di business all'interno dell'area aziendale. Esso contiene l'immagine di una porzione dei dati che saranno consultati dagli utenti. È importante notare che c'è un solo data warehouse mentre è possibile che si siano più data mart e possono essere derivati da esso oppure essere i blocchi di base per la sua costruzione. L'uso del solo data warehouse massimizza la coerenza e l'integrazione dei dati ma potrebbe essere poco gestibile e lento nella risposta alle query data la grande mole di dati. D'altra parte i *data mart* semplificano l'accesso ai dati, garantiscono maggiore sicurezza dato che si fa riferimento ad un insieme più piccolo di dati, ma si potrebbe rischiare di perdere il controllo sull'architettura creando confusione e problemi di coerenza.

Le caratteristiche principali del data warehouse sono le seguenti [1]:

- I dati sono integri e consistenti. L'integrità è garantita dalle regole implementate nel processo di ETL, le quali rendono i dati ad ogni caricamento consistenti. Al contrario le fonti operazionali non contengono i dati completi ma c'è bisogno dell'intervento di fonti esterne e questo crea confusione e inaffidabilità dei dati.
- Nel data warehouse viene conservata la storia dei dati. In un sistema operativo le informazioni sono relative ad un periodo piuttosto ridotto senza presentare un quadro storico di quanto analizzato; invece il data warehouse mostra un fenomeno in un intervallo di tempo molto esteso evidenziando l'evoluzione nel tempo.
- Mentre le fonti operazionali vengono utilizzate esclusivamente dall'area operativa, il data warehouse è rivolto soprattutto all'area direzionale perché la BI è utile per trovare tattiche e strategie.
- Il data warehouse è non volatile [4]. Gli utenti possono solo leggere i dati ed essi possono essere aggiornati o aggiunti solo con procedure eseguite in determinati periodi di non attività degli utenti. Questa caratteristica comporta una progettazione più semplice rispetto a quella in cui gli utenti possono anche modificare i dati come avviene nelle applicazioni operazionali.
- Il data warehouse è orientato al soggetto [4]. I dati sono organizzati con l'obiettivo di produrre più informazioni possibili mettendo in secondo piano il problema della ridondanza, per questo il data warehouse è più interessato ai temi aziendali specifici rispetto alle applicazioni o alle funzioni.

Il data warehouse è costituito da due tipi di tabelle, la *tabella dei fatti* e la *tabella delle dimensioni*. [1]

Il *fatto* rappresenta un evento dell'azienda che si vuole analizzare (vendite, acquisti, etc.). All'interno della *tabella dei fatti* troviamo le misure che descrivono quantitativamente gli eventi (ad esempio l'incasso delle vendite) ma anche le chiavi delle dimensioni che danno un contesto ai dati (prodotto, negozio, data). E' molto importante inserire le *dimensioni* perché determinano la granularità cioè il livello di dettaglio più fine a cui i dati saranno rappresentati, più il livello è elevato, più l'analisi è flessibile ma potrebbe perdere in prestazioni. Utilizzando query SQL o il motore OLAP (descritto nel *paragrafo 1.3.4*) si possono fare operazioni di aggregazione sui livelli di gerarchia delle dimensioni. Tuttavia non sempre hanno senso, per esempio per le vendite l'importo del venduto può essere sommato sul tempo e sul prodotto ma se si parla di giacenza in un magazzino non ha significato sommare le quantità sul tempo. La *tabella dei fatti* generalmente occupa molto più spazio rispetto alle *tabelle delle dimensioni* perché vengono memorizzati all'interno i dettagli degli eventi in un arco di tempo (per esempio nel caso di una catena di supermercati nella tabella delle vendite si hanno migliaia di record che coprono un determinato periodo).

Nelle *tabelle delle dimensioni* invece abbiamo gli elementi che costituiscono le attività aziendali (ad esempio il cliente, il prodotto, il tempo). Ogni colonna della tabella è rappresentata dagli *attributi*, cioè le caratteristiche analitiche che descrivono i dati presenti nelle *tabelle dei fatti*. Gli *attributi* possono essere distribuiti in gerarchie che danno la possibilità di organizzare i dati da un livello di massima aggregazione ad uno di massimo dettaglio. Ad esempio nella dimensione *Negozi* può esserci la gerarchia *nome del negozio, città, regione, stato*. La massima aggregazione si ha prendendo tutti i negozi ma si possono anche aggregare per stato fino ad arrivare ad una aggregazione per singolo negozio (vedi *figura 1.2*).

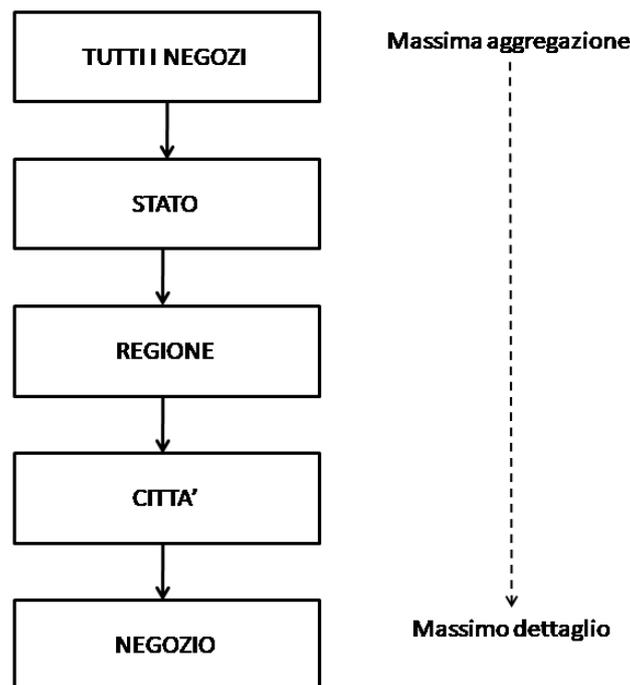


Figura 1. 2 Gerarchia della dimensione Negozi

Molto importante è l'uso della *chiave primaria*. Essa è un codice univoco per ogni entità che serve a collegare le tabelle delle dimensioni a quelle dei fatti.

Nel data warehouse i dati vengono memorizzati attraverso i così detti *schemi a stella* e a *fiocco di neve*, che rappresentano graficamente le diverse entità e le relazioni tra esse.

In uno *schema a stella* si ha una sola tabella dei fatti che è collegata, tramite le chiavi esterne, alle tabelle delle dimensioni ma tra queste ultime non vi sono collegamenti. Ad esempio la tabella dei fatti può essere una tabella di vendite che ha come attributi *quantità* e *guadagno* mentre le tabelle delle dimensioni possono essere quelle relative al tempo, al prodotto e al negozio (vedi *Figura 1.3*).

In uno *schema a fiocco di neve* i dati di una dimensione vengono frazionati in più tabelle dando l'aspetto di un fiocco di neve. Considerando l'esempio precedente la tabella della dimensione relativa ai negozi può essere ancora suddivisa nella tabella relativa alle città, e quindi nella prima ci sarà solo più la chiave della città associata al negozio mentre nella seconda ci saranno i dettagli (vedi *Figura 1.4*). Questo tipo di schema riduce lo spazio di archiviazione grazie alla rimozione di colonne ridondanti dalle tabelle per evitare di ripetere più volte la stessa voce. Inoltre se fosse necessario una modifica questa viene fatta una volta sola, ma d'altra parte aumenta il costo di ricostruzione dell'informazione.

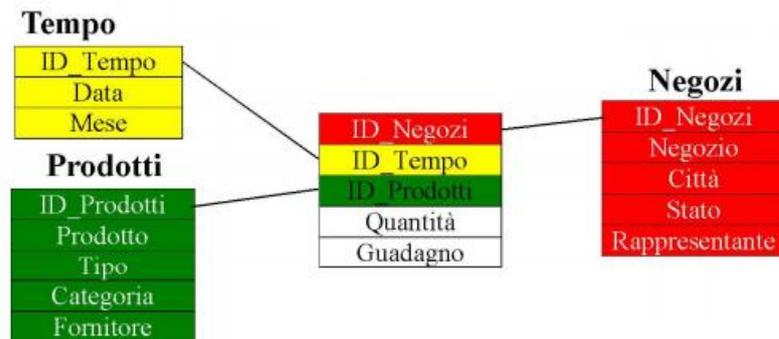


Figura 1. 3 Schema a stella¹

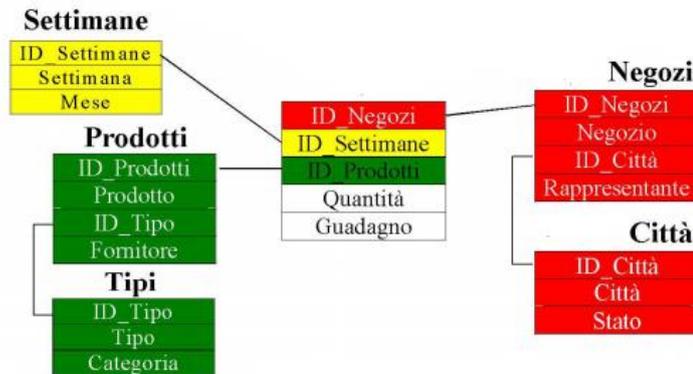


Figura 1. 4 Schema a fiocco di neve¹

¹«Progettazione di Data Warehouse» Available: https://dbdmg.polito.it/wordpress/wp-content/uploads/2020/05/2-DWprog_x6pdf.pdf (Slide 36-37)

1.3.4 SISTEMI OLAP

Il termine OLAP (On-line Analytical Processing) descrive un insieme di tecniche software che permettono all'utente finale di superare i problemi di manipolazione e di velocizzare l'analisi di grandi quantità di dati. [1]

La base dati OLAP ha una struttura multidimensionale che prende il nome di *Cubo*. Essa calcola le aggregazioni per ogni combinazione degli attributi. Secondo la metafora del cubo gli spigoli sono gli attributi e le intersezioni tra essi individuano una cella che contiene i valori aggregati delle misure. Il calcolo delle possibili combinazioni forma una struttura OLAP che contiene le risposte per ogni singola combinazione.

La caratteristica principale dei sistemi OLAP è la navigabilità dei dati che si concretizza nelle 5 categorie di operazioni seguenti:

- Drill-down: con questa operazione si passa da un'aggregazione maggiore ad un livello minore (ad esempio da *Anno* a *Mese*).
- Roll-up: è l'operazione contraria del drill-down.
- Slicing: è un'operazione di riduzione delle dimensioni di analisi, riduce la dimensione del cubo prendendone una "fetta". Viene utilizzata per ottenere totali in base a dimensioni diverse o per fare aggregazioni.
- Dicing: estrae un sottoinsieme di informazioni mediante un filtraggio.
- Pivoting: è un'operazione di rotazione delle dimensioni cambiando la modalità di visualizzazione dei dati.

I sistemi OLAP possono essere di vario tipo, alcuni più flessibili e altri ad alte prestazioni:

- *MOLAP* (Multidimensional OLAP): approccio noto anche come OLAP, esegue un'aggregazione preliminare dei dati. In questo modo l'utente può ricevere i risultati praticamente immediati a patto che le dimensioni rientrino in quelle predefinite. La caratteristica della velocità però penalizza la flessibilità delle analisi.
- *ROLAP* (Relational OLAP): estrae e aggrega i dati in tempo reale in base alle necessità dell'utente. Grazie alla sua scalabilità delle architetture di database a supporto del processo decisionale è l'approccio più diffuso ed è flessibile perché non ha dimensioni predefinite. Tuttavia può risultare lento perché richiede un'elaborazione più complessa nella creazione delle tabelle.
- *HOLAP* (Hibrid OLAP): compensa i punti deboli degli altri due approcci.

La scelta del tipo di database OLAP non è un'operazione banale e dipende molto dall'ipotesi delle possibili interrogazioni dell'utente.

Se da una parte la struttura OLAP fornisce una modalità intuitiva e performante nelle analisi vi sono anche dei problemi.

DATA SPARSITY

I sistemi OLAP generano una base dati molto estesa perché per ogni combinazione di valori si ha un dato, tuttavia molte di queste combinazioni sono nulle e quindi porta ad avere un grande database ma con un'elevata sparsità che influisce sull'efficienza dello spazio occupato.

Questa caratteristica è molto evidente nei sistemi MOLAP poiché utilizzano strutture sia per dati aggregati sia per quelli di dettaglio, mentre non è presente nei sistemi ROLAP poiché sono presenti solo aggregazioni dove è molto meno probabile trovare sparsità.

DATA EXPLOSION

Questo fenomeno consiste nel fatto che il calcolo dei valori delle aggregazioni può occupare molto più spazio dei dati di dettaglio. La data explosion può essere dovuta alla sparsità dei dati poiché la presenza di pochi valori favorisce la prevalenza dei dati calcolati, ma anche l'utilizzo di tante dimensioni e livelli di gerarchia possono influire sulla data explosion.

Una possibile soluzione è creare più cubi con poche dimensioni anziché uno solo in modo da ridurre la sparsità e le aggregazioni.

Inoltre i sistemi OLAP sebbene lavorino molto bene sulle aggregazioni dei dati, hanno difficoltà invece a usare il dato a livello atomico e richiedono una struttura denormalizzata per migliorare le prestazioni di accesso ma creando in questo modo chiavi ridondanti e ripetizioni.

1.3.5 REPORTISTICA

Dopo aver raccolto, trasformato, integrato e salvato i dati all'interno del data warehouse bisogna usarli per il processo decisionale.

Gli strumenti di front-end servono proprio a diffondere le informazioni ottenute dalla BI. Uno di questi strumenti è la Reportistica. [1]

Il report è un documento, elettronico o cartaceo, che contiene i dati rappresentati in tabelle e grafici, analizzati poi dagli utenti finali. Esso è uno strumento molto comodo per diffondere la BI e in particolare analizzare e aggregare i dati o effettuare calcoli.

I report inizialmente erano in formato cartaceo, ma con le nuove tecnologie si sono spostati sui dispositivi mobili, che oltre ad essere più fruibili dagli utenti sono diventati anche più interattivi.

La grande varietà del Business e gli infiniti progetti possibili richiesti dagli utenti fanno sì che non è fattibile raggruppare e definire dei report predefiniti, utilizzabili per qualsiasi azienda, ma al contrario ognuno di esse ha esigenze differenti che portano a delle analisi differenti.

Tuttavia si possono individuare tre tipologie di reportistica: strategica, tattica e operativa.

REPORTING STRATEGICO

I dati sono aggregati per mostrare un andamento su delle grandezze chiave definite dall'utente e per questo motivo è molto utile l'uso di grafici. I report di questo tipo analizzano i dati generalmente settimanalmente o mensilmente, ad esempio l'andamento delle vendite può essere un report realizzato mensilmente.

REPORTING TATTICO

Nel reporting tattico si ha un'aggregazione minore dei dati perché viene usato per analisi di controllo confrontando gli andamenti delle grandezze prese in esame e gli obiettivi decisi. La frequenza può variare da giornaliera a mensile a seconda del problema. Un report di esempio può essere riguardo ai risultati delle vendite in una settimana piuttosto che in un mese.

REPORTING OPERATIVO

Questa tipologia è molto differente dalle prime due; l'aggregazione in questo caso è minima, i dati sono di dettaglio e quindi l'uso di grafici non è efficace. Un esempio di reporting operativo può riguardare l'analisi sulle vendite giornaliere di un'azienda.

Nella costruzione del report bisogna tener conto di alcuni aspetti fondamentali. Prima di tutto bisogna utilizzare un layout semplice, definendo font, colori e regole di impaginazione (ad esempio i filtri possono essere inseriti in alto). Nelle pagine del report è buona norma l'utilizzo di diversi elementi, tabelle e grafici, che rispecchino al meglio l'analisi da effettuare; tuttavia troppi oggetti all'interno del report possono provocare confusione agli utenti.

I report sono riempiti per la maggior parte dai grafici che possono essere grafici lineari, a torta, a barre, a dispersione, etc. Essi devono essere strutturati in modo che siano di

facile lettura e quindi bisogna inserire una descrizione sintetica dell'oggetto, anche con l'uso della legenda se necessario. Inoltre è importante anche l'uso dei colori per differenziare i vari raggruppamenti che si vengono a formare, magari utilizzando colori più decisi per evidenziare qualche determinato valore. Una descrizione più dettagliata delle varie tipologie di grafici è presente nel *Paragrafo 2.4*.

Le caratteristiche fondamentali che un report deve avere sono le seguenti:

- Rilevanza e pertinenza: i dati visibili devono essere solo quelli utili alle analisi.
- Attendibilità: i dati devono essere attendibili per non essere dannosi al progetto.
- Tempestività: i report devono essere aggiornati e forniti periodicamente agli utenti.
- Navigabilità: riceve un valore aggiuntivo l'utilizzo di operazioni drill-down sui dati e la possibilità di avere dei record collegati tra loro.

Il report è un buono strumento per realizzare rendiconti per gli utenti che devono prendere decisioni o analisi che utilizzano dati di dettaglio. Invece un altro strumento di reportistica è la Dashboard, utilizzata specialmente nei sistemi di Corporate Performance Measurement (CPM) per misurare e interpretare le performance delle aziende monitorando quindi il loro andamento. Le dashboard sono delle rappresentazioni che contengono molte informazioni sotto forma di grafici che traducono degli indicatori di performance, in modo da essere più intuitivi e comprensibili. Questo tipo di reportistica infatti è generalmente indicato per utenti di livello strategico che possano usarle come punto di inizio per analisi ulteriori.

In conclusione, per l'implementazione della reportistica gli utenti hanno bisogno del supporto dell'IT e quando le richieste sono elevate possono portare ad un rallentamento del sistema. Una possibile soluzione è la self-service BI, essa permette agli utenti finali di agire senza il supporto del personale informatico in modo da permettere una navigabilità più ampia in un tempo minore. Questo però non significa che l'IT non sia più fondamentale perché ha ancora il compito di governare i dati, gestire la sicurezza e monitorare le operazioni svolte dagli utenti.

1.4 BUSINESS ANALYTICS

La Business Intelligence è riuscita a rivelare dei segnali che possono essere utili all'imprenditore quando si tratta di prendere decisioni sullo sviluppo della sua attività. Infatti il successo di un'azienda è anche quello di riuscire a capire tali segnali attraverso

l'intuito e il proprio pensiero soggettivo; sarebbe utile invece trovare uno strumento che possa farlo in modo oggettivo e automatizzato. Questo tipo di analisi la BI non è in grado di farlo in modo completo perché ha due limiti principali [5].

Il primo è quello di non essere in grado di prevedere il futuro di un'azione, infatti la BI offre un'analisi molto precisa su dati del passato o del presente rispondendo alle domande su cosa sta accadendo e il motivo, ma non riesce ad anticipare cosa potrebbe succedere in futuro.

Il secondo limite sta nel fatto che la BI non è in grado di rispondere a delle domande che non siano state pensate nella fase di progettazione dei dati. Ad esempio in un'analisi in cui ci sono le informazioni relative al cliente e al fornitore, se non risulta esserci una relazione tra queste due entità ed eventualmente manca il rapporto con il tempo o area geografica, non è possibile fare un'analisi completa.

Con la Business Analytics (BA) è possibile colmare queste lacune, infatti permette di trovare soluzioni per prevedere come si svilupperanno gli eventi. La BA può includere funzioni di ottimizzazione, statistica, analisi di testi e il data mining.

Quindi l'unione di BI e BA porta valore aggiunto all'azienda, poiché la Business Intelligence fa un'analisi descrittiva sui dati passati e presenti mentre la Business Analytics si concentra in un'analisi predittiva, cioè basata su cosa può accadere in futuro se si sceglie una determinata decisione, e prescrittiva, cioè basata su decisioni da prendere in seguito alle analisi svolte. Ad esempio, per un'azienda di Marketing è interessante vedere, oltre ad un'analisi sulle vendite avvenute, anche come i prezzi e i sistemi di vendita influenzino la scelta di una marca piuttosto che un'altra.

1.5 IL COMPITO DEL CONSULENTE

La maggior parte delle aziende non ha le abilità tecniche e di progettualità interne per affrontare un progetto di BI in autonomia. È frequente che si rivolga a una società di consulenza che la supporti in questo processo di trasformazione che impatta in modo sensibile sulla capacità di reazione al mercato e quindi sulla sua competitività.

Il compito del consulente è quello di analizzare le informazioni gestite nell'azienda e creare report utilizzati per attività decisionali o di misurazione delle performance. Egli offre ai clienti delle soluzioni logiche o tecniche di organizzazione e rielaborazione dei dati dell'azienda in modo da garantire un incremento delle capacità di organizzazione aziendale.

In particolare il consulente deve principalmente effettuare queste attività:

- Individuare le basi dati da utilizzare, e assicurarsi che i dati siano completi e aggiornati
- Organizzare in diverse applicazioni l'implementazione assicurando di soddisfare le esigenze espresse dal cliente
- Impostare la fase di pulizia e trasformazione del dato (ETL)
- Definire la struttura dati, quindi scegliere le dimensioni e le metriche che dovranno essere utilizzate e il modello più opportuno da usare
- Gestire per ogni base dati il corretto tempo di aggiornamento e verificare la coerenza delle schedulazioni
- Gestire la visibilità delle diverse aree dati assegnando a ciascun utente o tipo di utente regole che condizionano il completo accesso ad una certa pagina o oggetto di analisi o che ne parzializzano la vista

Quindi per effettuare un progetto di Business Intelligence si parte dal problema principale, si verificano i dati di cui si ha bisogno e si propone una possibile soluzione. Successivamente si realizza una o più applicazioni a seconda del progetto, e si prova a testare il risultato usando degli esempi; questa fase può essere eseguita anche con il cliente che può fornire degli esempi e proporre eventuali modifiche. Infine, una volta convalidato il prototipo, vengono assegnati i diritti di accesso a tutti gli utenti dell'azienda e viene rilasciato a loro il report. Durante la fase di test vengono messe a punto le letture delle base dati, mentre dopo il rilascio segue spesso una fase di manutenzione per recepire modifiche ed ottimizzazioni che il cliente chiede man mano che impara ad utilizzare il tool.

Capitolo 2

LA PIATTAFORMA QLIK SENSE

Nell'evoluzione della Business Intelligence, la creazione del Data Warehouse e l'introduzione degli strumenti di ETL ed OLAP hanno permesso di estendere e migliorare le query e la navigazione. Tuttavia il bisogno di avere sempre più funzionalità rende i sistemi più complessi e l'analisi diventa più lenta.

Ormai alla BI non è richiesto solo di fornire uno strumento che visualizza i dati in modo semplice e utile ma di abilitare l'esplorazione del dato per avere un'analisi più completa ed efficace. Qlik Sense ha tutte queste caratteristiche [7].

Qlik Sense è un software di ultima generazione della società Qliktech International, un'azienda leader mondiale nell'ambito BI che si impegna ad aiutare le aziende ad ottenere informazioni utili dai dati. Nata dalla richiesta a due professori universitari svedesi da parte di un'azienda divenuta poi una delle multinazionali più grandi e potenti al mondo (anche grazie al supporto ricevuto in fase di analisi del mercato), Qlik propone soluzioni analitiche con visualizzazioni flessibili, interattive e di facile lettura in modo da permettere a tutti gli utenti, indipendentemente dalle competenze, un'analisi approfondita per migliorare le prestazioni della propria azienda.

L'*app* è l'elemento centrale di Qlik Sense; è possibile creare le proprie app per poterle poi riutilizzare, modificare e condividere con altri utenti. Un' app consiste in uno o più fogli contenenti visualizzazioni. Le visualizzazioni contengono grafici, tabelle e rappresentazioni simili di dati, combinate con altre informazioni.

Il vantaggio principale che possiede Qlik è il *motore associativo* e la sua capacità di lavorare in "RAM" [8]. Questa tecnica all'avanguardia permette di caricare tutti i dati nella memoria legando in modo naturale rispetto al nostro modo di pensare tabelle con granularità e cardinalità diverse dando quindi la possibilità di avere sotto controllo l'intera storia dei dati. L'architettura associativa permette di esplorare in ogni direzione senza limiti, infatti il sistema non ha query predefinite e risponde prontamente ad ogni

selezione fatta senza richiedere tempi di attesa [9]. Una selezione, può essere fatta in qualsiasi grafico o oggetto della *presentation*, facendo clic e selezionando un valore alla volta oppure eseguendo un tracciamento e quindi selezionando contemporaneamente più valori; questo secondo tipo di selezione prende il nome di *lazo*. Ad esempio si può fare clic su un grafico a torta selezionando un particolare valore della dimensione interessata (vedi *Figura 2.1*); oppure in un grafico a dispersione si può tracciare un cerchio a mano per selezionare dei determinati punti (vedi *Figura 2.2*).

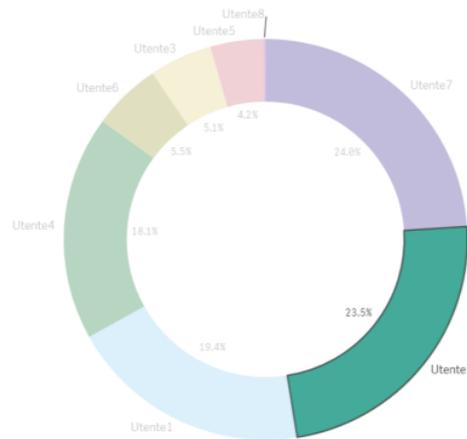


Figura 2. 1 selezione in un *grafico a torta*²

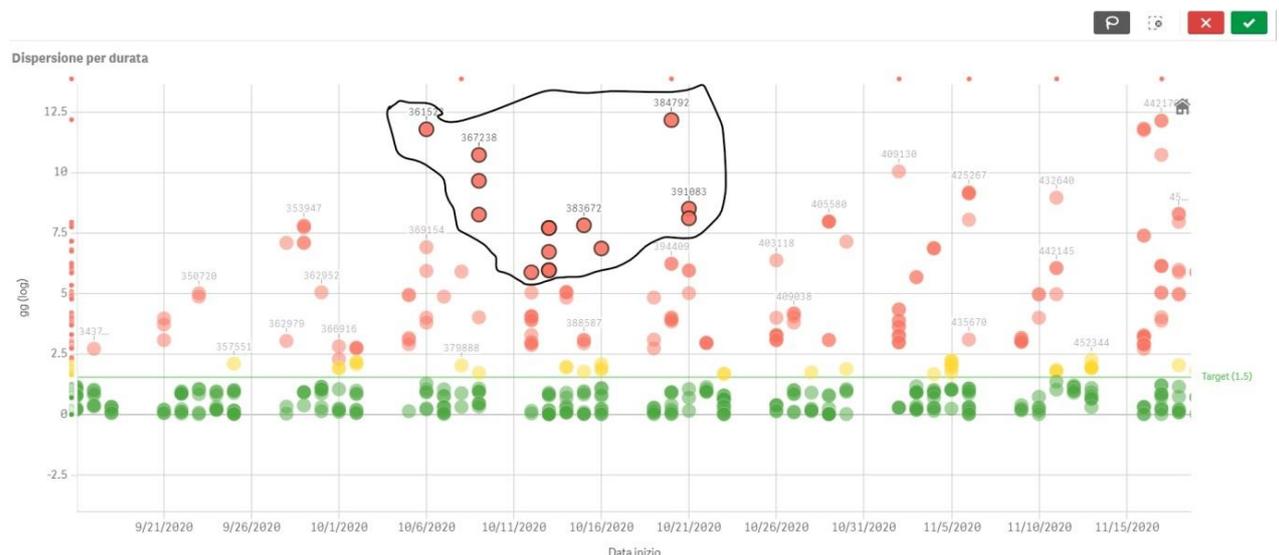


Figura 2. 2 selezione *lazo* in un *grafico a dispersione*²

Ci sono anche altri modi per fare selezioni che differenziano Qlik dagli altri strumenti di BI. Il primo è quello di cercare un campo sotto il pulsante “selezioni”; esso permette di

² Figura presa da una applicazione demo di Qlik Sense realizzata durante il tirocinio in Bios Management.

individuare i possibili campi (attributi) che hanno la parola cercata e i rispettivi valori distinti, infatti, come abbiamo accennato, i dati sono tutti a portata di mano e le ricerche sono facili e veloci. Ad esempio se si vogliono vedere i diversi tipi di prodotto in un’analisi di vendite di alimentari basterà cercare nelle selezioni “product”, Qlik mostrerà tutti i campi che hanno la parola “product” e quindi l’utente potrà selezionare un valore direttamente da lì (vedi *Figura 2.3*).

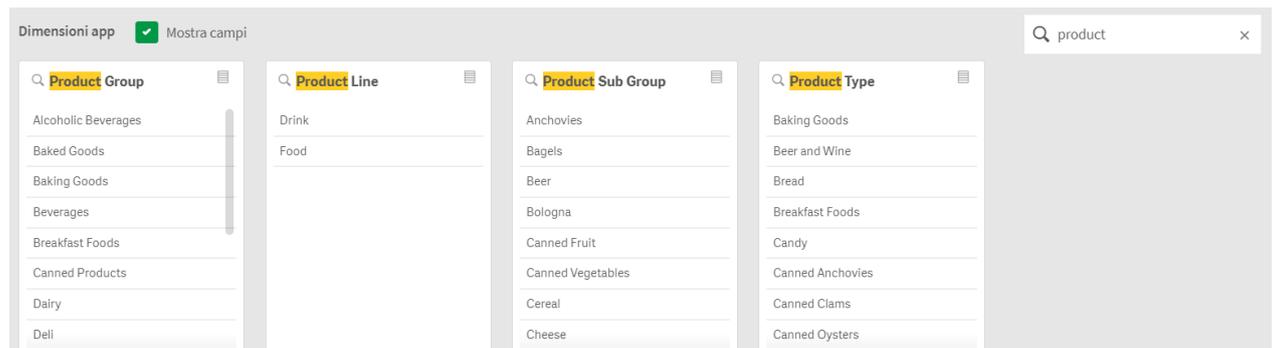


Figura 2. 3 Ricerca tra i campi della base dati .³

Se invece si vuole cercare un particolare valore, Qlik è in grado di eseguire ricerche nell’intera base dati dell’app e riferire anche il campo in cui è presente tramite la *ricerca intelligente*. Facendo sempre riferimento all’esempio precedente scrivendo la parola “drink” in corrispondenza della lente il sistema mi anticipa i possibili valori che contengono quella parola e i loro relativi campi (vedi *Figura 2.4*).

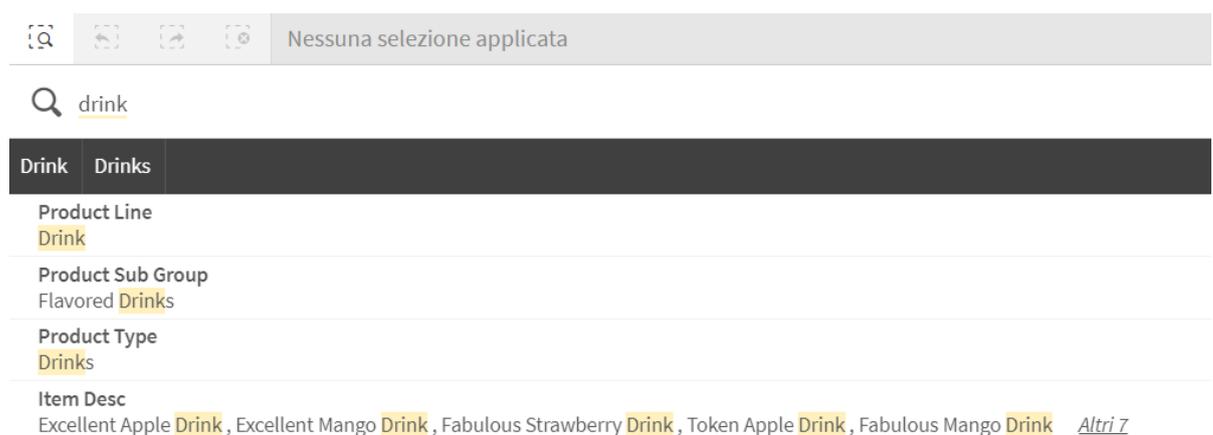


Figura 2. 4 Uso della ricerca intelligente. ³

³ Figura presa da un’applicazione di Tutorial fornita dal sito di Qlik Sense

Una caratteristica interessante è che le selezioni inserite sono *a tutto tondo*, cioè tutte le visualizzazioni lungo l'applicazione risentono della selezione mantenendo un contesto unificato e le analisi vengono aggiornate immediatamente al nuovo contesto. In questo modo l'utente può fare una serie di osservazioni lungo tutta l'analisi a differenti livelli. Inoltre se alcune selezioni sono ricorrenti, l'utente può salvarle tramite un "*segnalibro*" che potrà richiamare quando vuole, oppure sviluppare grafici che a prescindere dalle selezioni correnti mostrano quella specifica selezione preferita abilitando logiche di analisi di varianza che con altri tool avrebbero effort decisamente superiori. Infine se una particolare selezione è sempre presente in tutte le analisi si può bloccare, in questo modo se vengono cancellate le selezioni questa invece rimane attiva a meno che non si sblocchi di nuovo.

Unico nel suo genere è il così detto "*potere del grigio*", un elemento estremamente potente che permette di vedere i valori non correlati nell'analisi [10]. Per fare questo Qlik utilizza quattro colori che evidenziano le associazioni dei dati (vedi *Figura 2.5*):

- *Bianco*, per gli elementi che è possibile selezionare
- *Verde*, per gli elementi selezionati
- *Grigio chiaro*, per i componenti alternativi che potrebbero ancora essere selezionabili
- *Grigio scuro*, per gli elementi esclusi non selezionabili.

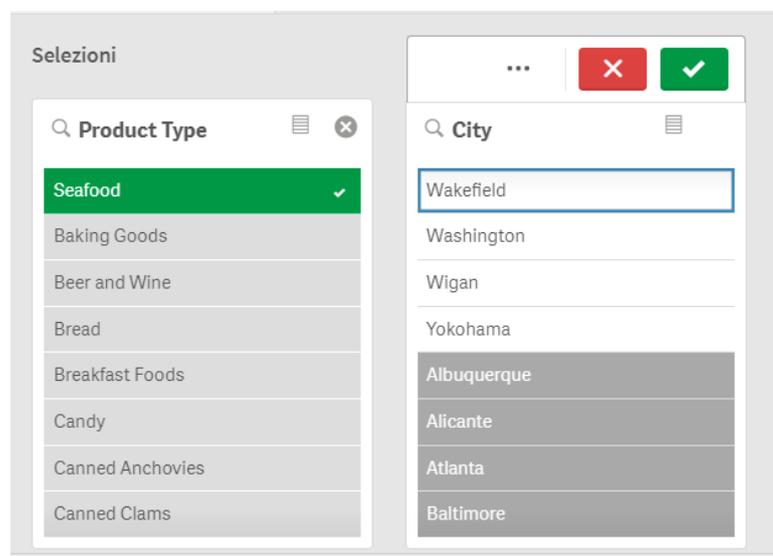


Figura 2. 5 Il potere del grigio.⁴

Considerando l'esempio delle vendite, se si seleziona "Seafood" nel campo *Product Type* gli altri valori diventano grigio chiaro cioè alternativi, mentre nel campo *City* si può

⁴ Figura presa da un'applicazione di Tutorial fornita dal sito di Qlik Sense

vedere ad esempio che quel tipo di cibo è venduto a Washington (bianco) ma non in Atlanta (grigio scuro).

Di conseguenza il motore di Qlik è considerato flessibile non solo per la libertà di analisi che permette ma anche perché evidenzia anche ciò che è escluso dalla selezione.

In conclusione, come per alcuni strumenti di BI, con Qlik è possibile effettuare analisi *self-service*, che potrebbe far pensare ad un'analisi senza regole ma che in realtà lascia la libertà agli utenti di muoversi sui dati con regole ben precise cioè utilizzando le dimensioni e misure fornite dalla BI.

Qlik non dà solo agli utenti la possibilità di crearsi i propri report ma dà anche suggerimenti su come strutturare l'analisi. Infatti un'altra caratteristica fondamentale di Qlik Sense è quella dell'Intelligenza Aumentata che non ha l'obiettivo di sostituire l'uomo ma soltanto di supportarlo nella scelta e di dare informazioni utili sulla base dei dati nel sistema. In particolare il software può suggerire all'utente delle tipologie di analisi (grafici e altri oggetti di interfaccia) in base al set di dati che ha a disposizione (vedi Paragrafo 2.5). Questo rende la piattaforma utile per un'analisi completa e dettagliata ma anche per dare spazio ad un'esplorazione libera che potrebbe portare a scoperte sorprendenti.

Il vantaggio dell'analisi *self-service*, oltre al fatto di evitare colli di bottiglia per le numerose richieste ai responsabili della BI, è quello di creare fiducia negli utenti. Infatti gli utenti finali molto spesso non si fidano del dato perché non sanno come è stato generato e devono solo prenderlo per certo, invece con la possibilità di utilizzarlo loro stessi iniziano a fidarsi.

2.1 CONFRONTO CON IL MODELLO RELAZIONALE BASATO SU QUERY

Generalmente gli strumenti di BI "moderni" hanno ancora un metodo di analisi basato su query [8]. Questo tipo di approccio separa il livello dei dati da quello dell'applicazione rendendolo lineare e predefinito ma comporta lunghe implementazioni e costi elevati. L'analisi con i sistemi tradizionali viene fatta solo su un sottoinsieme di dati parziali riuniti usando SQL join, risponde alle domande da cui nasce il progetto e i possibili tipi di domande che potrebbero essere fatte dagli utenti in seguito devono essere pensate in anticipo. Tutti gli altri dati sono lasciati da parte e quindi si ha una storia parziale con un approccio gerarchico dei dati; se devono essere fatte delle altre analisi o si vuole cambiare il punto di vista allora dovranno essere create altre query complesse. Questa

procedura prende il nome di “ask, wait, answer” perché ad ogni domanda c’è un periodo di attesa.

Per dare una visione più completa, nella *Tabella 2.1* vengono messe a confronto le principali differenze con Qlik Sense.

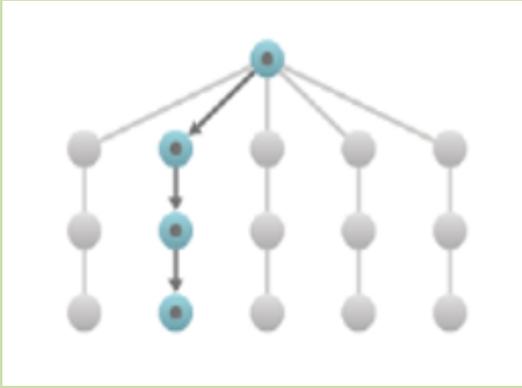
SQL	QLIK SENSE
Esplorazione lineare all’interno di sottoinsiemi di dati	Riunisce tutti i dati per esplorare senza limitazioni
Query lente e preparazione continua dei dati	Potenti operazioni istantanee di calcolo che aggiornano l’analisi dopo un clic
Gerarchie predefinite e percorsi fissi 	Esplorazione in ogni direzione 
Risposta a domande follow-up con ricostruzione di query	Gestione delle domande <i>follow-up</i> alla velocità del pensiero
Storia incompleta, valori solo filtrati	Storia completa, evidenziata associazione dei dati (<i>potere del grigio</i>)

Tabella 2.1: confronto con il modello relazionale⁵

2.2 COMPONENTI DI QLIK SENSE

Qlik Sense esiste sia nella versione Desktop che Enterprise. La prima lavora solo sul computer locale, non si possono pubblicare e condividere le apps e la funzione di

⁵ Le due immagini della tabella sono state prese dal sito <https://www.qlik.com/it-it/products/associative-difference>

sicurezza non è supportata. Invece Qlik Sense Enterprise è la versione completa di Qlik Sense, essa include la visualizzazione dei dati self-service per consentire agli utenti di esplorare i dati e analisi guidate per allineare gli utenti a un processo aziendale.

I componenti principali di Qlik Sense Enterprise sono l'*Hub* e la *QMC (Qlik Management Console)*.

L'*Hub* è il punto d'ingresso sulla piattaforma (vedi *Figura 2.6*). Gli utenti possono visualizzare le *apps*, i progetti sviluppati e da cui lo sviluppatore può derivare nuove applicazioni o modificarne di esistenti. All'interno dell'*Hub* le applicazioni si trovano negli *streams*, cartelle in cui vengono raggruppate le *apps* e in cui possono esserci diversi diritti di accesso. Ogni utente può avere accesso all' *hub* se possiede una licenza e può accedere solo agli *stream* a cui è stato dato il permesso e quindi visualizzare solo le *apps* di propria competenza. In alto è presente anche una cartella "Lavoro" dove sono presenti le *apps* create e non pubblicate. Infine tra gli *streams* creati ve n'è uno fornito da Qlik che si chiama "Monitoring apps" che contiene due applicazioni di Qlik Sense. La prima applicazione si chiama "License Monitoring" e traccia un'analisi sull'uso delle licenze, a quali *streams* e a quali *apps* gli utenti hanno accesso e quanto ognuno utilizza le applicazioni. L'altra applicazione è "Operations Monitor" che fornisce informazioni sull'ambiente Qlik Sense server, includendo l'uso della CPU e la memoria del server, le attività degli utenti e i task di ricaricamento delle *apps*.

La *QMC* ha una funzione di amministrazione, essa gestisce le licenze degli utenti a Qlik e definisce i permessi e le regole di modifica, lettura e accesso alle *apps*. In particolare le parti fondamentali della *QMC* che si possono vedere nella *Figura 2.7* sono le seguenti:

- Apps: in cui si possono vedere tutte le *apps* presenti che possono essere pubblicate negli *stream*, duplicate o eliminate. Inoltre esse possono essere esportate e possono esserne importate dalle altre.
- Data connections: in questa sezione è possibile modificare o creare nuove connessioni alle sorgenti dati da utilizzare nell'editor caricamento dati.
- Streams: in cui si possono vedere tutti gli *streams* disponibili che possono essere modificati ed eliminati; da questa sezione si possono creare inoltre le regole d'accesso degli utenti e si possono creare nuovi *streams* o inserire nuove *apps*. Di default Qlik Sense include due *streams*: *Everyone*, a cui hanno accesso tutti gli utenti e *Monitoring apps* di cui accennato sopra.
- Task: da questa sezione si possono vedere gli stati dei task di tutte le *apps*. Essi sono usati per ricaricare i dati; possono essere automatici quando il ricaricamento avviene secondo una regola (che può essere un orario specifico oppure dopo il caricamento di un'altra applicazione) oppure il ricaricamento può essere fatto in maniera manuale.
- License Management: contiene tutte le informazioni sulle licenze assegnate e quelle disponibili; inoltre per ogni licenza viene dato il dettaglio su che tipo di

licenza è. Agli utenti può essere assegnato un accesso di tipo Professional o Analyzer [14]. L'accesso Professional è dato agli utenti che devono accedere a tutte le funzionalità di Qlik Sense, infatti con questa licenza si possono creare, modificare e pubblicare fogli o creare direttamente le apps e si ha accesso a tutte le funzionalità disponibili compresa l'amministrazione. Invece l'accesso Analyzer è dato agli utenti che utilizzano le apps create, essi possono solo creare segnalibri, stampare oggetti ed esportare i dati di un oggetto nei fogli. Tuttavia per ogni licenza si possono dare diverse funzioni sulle diverse apps; ad esempio si può scegliere di dare il permesso di vedere un determinato stream ma all'interno di esso di visualizzare solo alcune app piuttosto che soltanto dei fogli di essa.

- **Extensions:** le estensioni di visualizzazione permettono di estendere le capacità di visualizzazione di Qlik Sense; per creare un'estensione è necessario un editor di codice. Da questa pagina è possibile aggiungerne una, modificarla o eliminarla. Un esempio di estensione, che è stata usata anche nei progetti presentati in questa tesi, è il pulsante di reload che permette di caricare i dati direttamente dal foglio di presentation e quindi permette di aggiornare l'app anche ad un'utente analyzer.

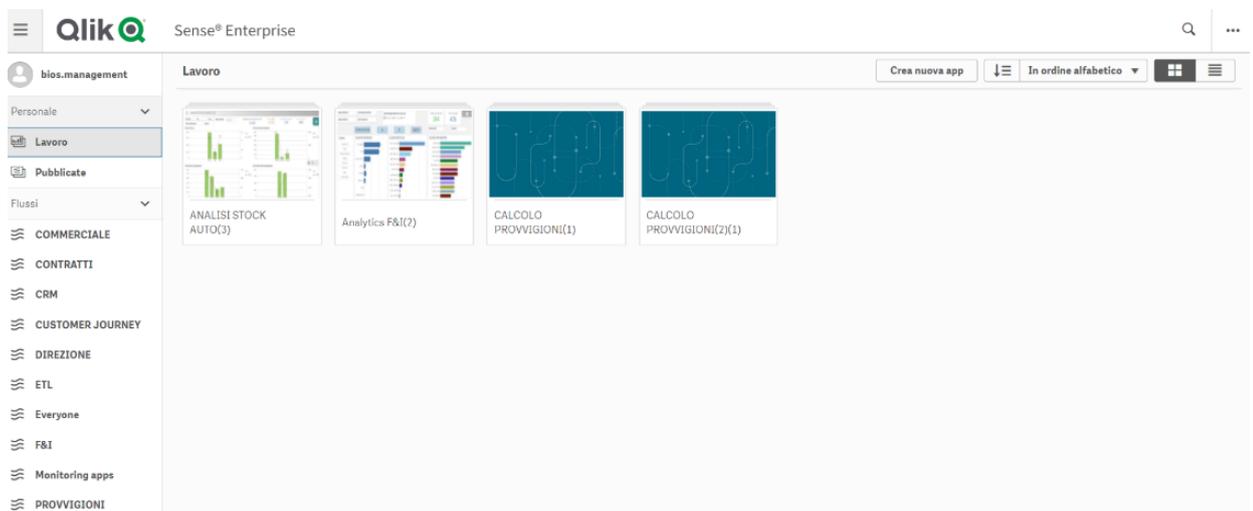


Figura 2. 6 Pagina dell'Hub ⁶

⁶ Figura presa dall'Hub di Qlik Sense relativo ai progetti della concessionaria Gino S.p.A.

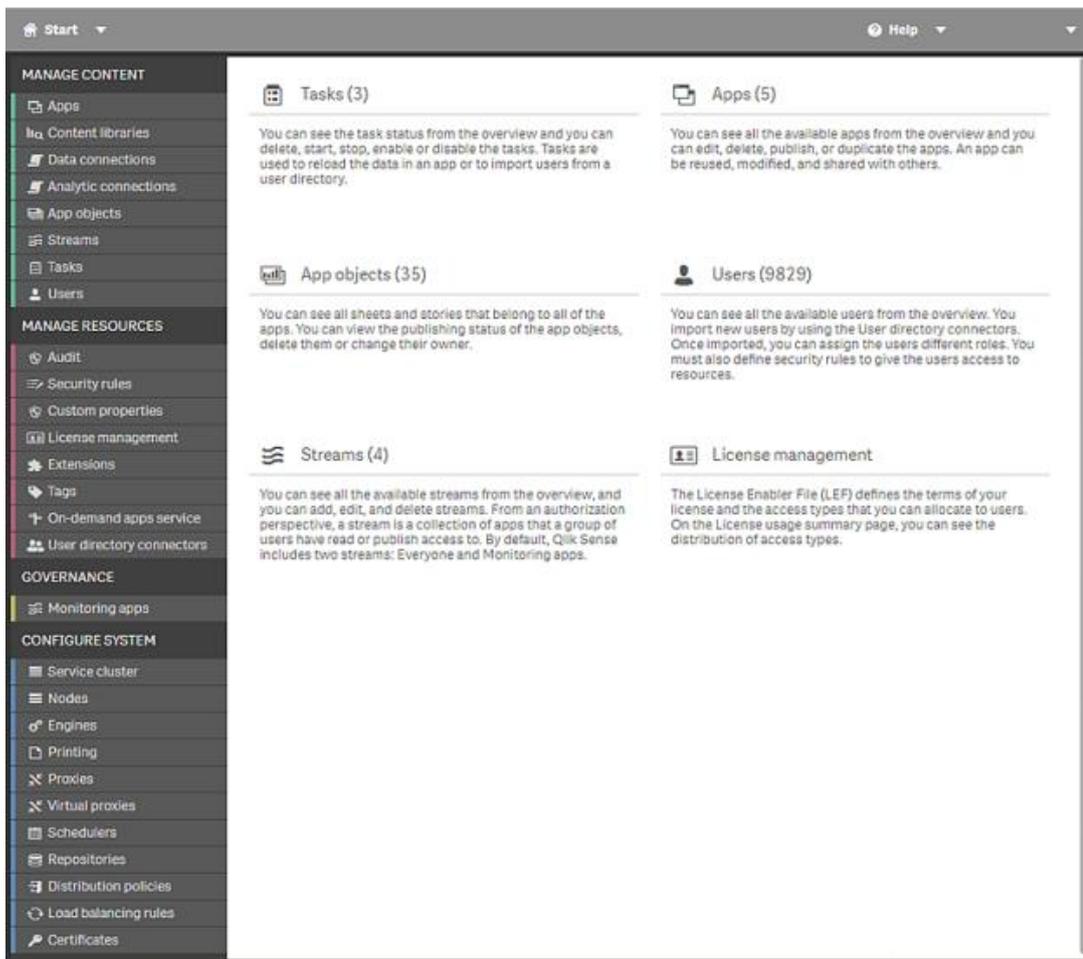


Figura 2. 7 Pagina principale della QMC ⁷

2.3 CARICAMENTO DATI

Lo script di Qlik Sense per il caricamento dei dati è assegnato all'*editor caricamento dati* che carica i dati da una sorgente collegandola all'app. Una sorgente può essere un file di dati (ad esempio Excel o csv) oppure un database.

La particolarità di Qlik è la creazione di una connessione dati che consente di salvare il collegamento alle sorgenti dati comunemente utilizzate. Infatti nel caricamento dati è presente l'elenco delle connessioni salvate raggruppate per spazi a cui appartengono. In questo modo scegliendo la connessione, Qlik crea in automatico la parte di script che carica i dati.

⁷ Figura presa dal sito https://help.qlik.com/it-IT/sense/February2019/Subsystems/ManagementConsole/Content/Sense_QMC/QMC-start-page.htm

Caricando i dati si crea una tabella con i relativi campi, ma quest'ultimi possono anche essere creati direttamente nello script utilizzando le infinite istruzioni a disposizione di Qlik, una sorta di "dialetto" del linguaggio SQL. Oltre ai campi possono essere create anche delle variabili. E consuetudine nominarle precedendole con "v_" da differenziarle dai campi. Esse possono essere create con il comando *set* se si vuole assegnare un valore preciso, mentre con il comando *let* se è frutto di un calcolo. Le variabili create da script possono essere viste lato presentation ed essere modificate dall'utente o meno in base alle esigenze del progetto (vedi *Paragrafo 2.4*)

Quando l'applicazione è complessa ci possono essere più tabelle dei fatti e queste possono essere gestite con diversi approcci.

Una delle possibilità è quella di usare l'operazione *join*. Essa consente di eseguire un'unione di due tabelle creandone una sola; inoltre può essere preceduta dai prefissi *left*, *right* e *inner* per limitare i dati (nei progetti descritti nel Capitolo 3 sono presenti numerosi esempi di questa operazione).

Un'altra opzione è la *Concatenate* che forza la concatenazione di due tabelle, essa può essere utilizzata indipendentemente dal fatto che le tabelle abbiano nomi di campo differenti. In sostanza i records saranno la somma dei records delle tabelle originarie e se ci sono dei duplicati, essi non vengono rimossi i duplicati. Tuttavia può essere pericoloso quando si caricano due tabelle con gli stessi campi perché Qlik Sense concatenerà in automatico il contenuto delle due tabelle. Se si vuole evitare la concatenazione bisogna introdurre l'istruzione *NoConcatenate* tra il caricamento delle due tabelle.

Un'istruzione particolare di Qlik è lo *store* che permette di fare una copia dei dati caricati che vengono salvati in un formato nativo di Qlik, *qvd* (QlikView Data) oppure in un comune formato testo meno performante [11]. File in formato *qvd* possono essere ricaricati in altre applicazioni Qlik per fare ulteriori modifiche ed altre analisi. L'uso dei *qvd* può avere diversi scopi. Il primo è l'incremento della velocità di caricamento, un file *qvd* può essere 10-100 volte più veloce rispetto alla lettura di altre sorgenti esterne. Inoltre diminuisce il carico sui server di database, infatti quando diversi script utilizzano gli stessi dati basterà caricarli una volta sola in un file *qvd* e le app utilizzeranno quest'ultimo riducendo il carico di lavoro sui database esterni. Infine i file *qvd* sono utili molto spesso per caricare solo i nuovi record da un database in crescita.

Dopo il caricamento, Qlik Sense associa i dati cercando i campi comuni in tutte le tabelle. Se alcune associazioni tra tabelle vogliono essere evitate basterà rinominare i campi nello script.

Nel sistema di *visualizzazione modello dati* viene presentata la struttura dei dati caricati (vedi *Figura 2.8*). In particolare le associazioni tra le tabelle sono definite da linee e per ogni tabella caricata viene mostrato il nome della tabella e l'elenco dei campi. Inoltre

quando viene selezionata una tabella Qlik mostra i dettagli di essa, tra cui i vari record, le correlazioni con le altre tabelle e se sono presenti duplicati o valori nulli (vedi *Figura 2.9*).

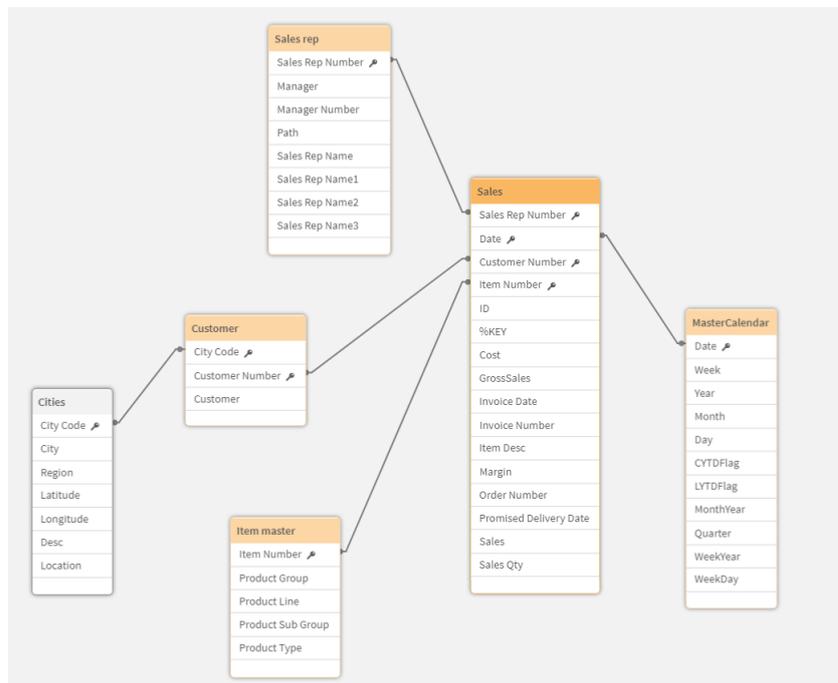


Figura 2. 8 Visualizzazione modello dati ⁸

▼ Anteprima

Aggiungi come dimensione

Aggiungi come misura

Customer Number		Sales								
		Customer Number	Item Number	ID	%KEY	Cost	GrossSales	Invoice Date	Invoice Number	Item Desc
Densità	100%	10012226	10696	1	3428	-513.15	-573.3835	1/12/2012 12:00:00 AM	318960	Cutting Edge Sliced Ha
Rapporto sottogruppo	93.1%	10012226	10009	2	3429	-105.93	-204.6638	1/12/2012 12:00:00 AM	318960	Washington Cranberry
Presenta duplicati	true	10012226	10385	3	3430	-88.07	-165.8016	1/12/2012 12:00:00 AM	318960	Moms Sliced Ham
Valori distinti totali	683	10012226	10215	4	3431	-43.12	-118.3703	1/12/2012 12:00:00 AM	318960	Tip Top Lox
Valori totali attuali	636	10012226	10965	5	3432	-37.98	-102.3319	1/12/2012 12:00:00 AM	318960	Just Right Beef Soup
Valori non null	96466	10012226	10901	6	3433	-49.37	-85.5766	1/12/2012 12:00:00 AM	318960	Fantastic Pumpnickle
Tag	Skey Snumeric Sinteger	10012226	10681	7	3434	-45.81	-68.4399	1/12/2012 12:00:00 AM	318960	Gorilla 1% Milk
		10012226	10808	8	3435	-13.56	-67.3833	1/12/2012 12:00:00 AM	318960	Real Cheddar

Figura 2. 9 Anteprima tabella selezionata nella visualizzazione modello dati ⁸

Tra le regole da seguire nel caricamento ci sono due temi molto importanti: le *chiavi sintetiche* e i *riferimenti circolari*.

⁸ Figura presa da un'applicazione di Tutorial fornita dal sito di Qlik Sense

Le *chiavi sintetiche* sono presenti quando più tabelle hanno due o più campi in comune creando delle chiavi composite [12]. Il sistema in questi casi risponde creando una tabella con i campi in comune in tutte le possibili combinazioni. La presenza di chiavi sintetiche a volte può nascondere un modello dati scorretto, creando una situazione che mette a rischio la capacità di navigazione associativa del tool. In quel caso bisogna verificare se i campi in comune abbiano lo stesso ruolo e in caso negativo bisogna rinominarli per evitare legami. Ad esempio nella *Figura 2.10* si può osservare che è stata creata una chiave sintetica perché le due tabelle [sheet1] e [Region] hanno in comune i due campi *Region* e *Region code*, tuttavia la prima tabella può evitare di avere il campo *Region* perché le basta il codice per identificare la regione.

Invece il *riferimento circolare* si ha quando tra le tabelle vi sono più percorsi di associazione possibili (vedi *Figura 2.11*) [13]. Essi sono sempre da evitare perché può portare ad ambiguità di comprensione. Un modo semplice per superare il problema è rinominando o eliminando i campi che creano il loop. Ad esempio nella *Figura 2.11* la tabella Table1 ha il campo "Date" in comune con la Table2, questa ha in comune con la tabella Region il campo "Week" che a sua volta ha in comune "SalesKey" con la Table1.

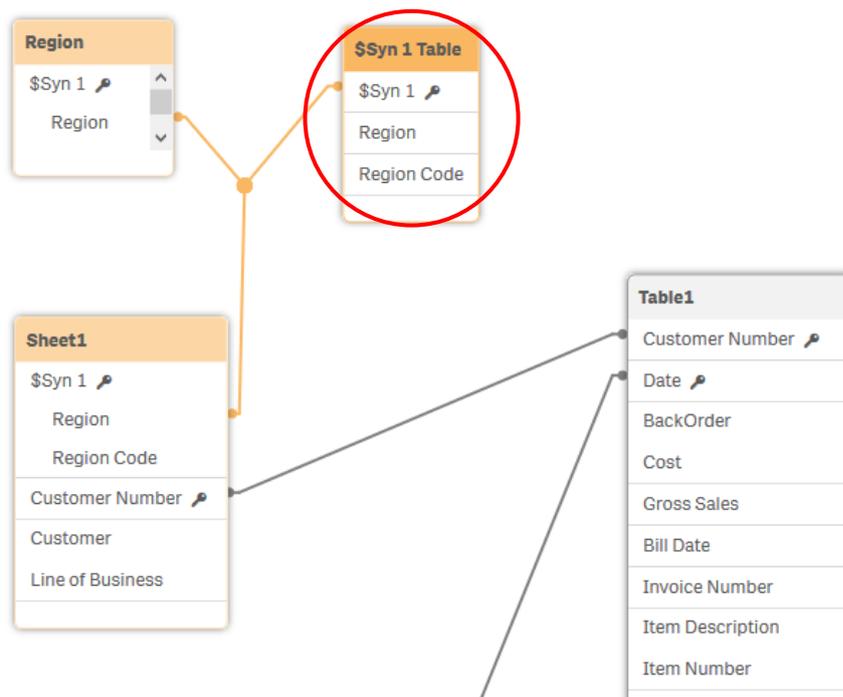


Figura 2. 10 Chiave sintetica cerchiata in rosso: le due tabelle Sheet1 e Region hanno due campi in comune, "Region code" e "Region" ⁹

⁹ Figura presa dal sito https://help.qlik.com/it-IT/cloud-services/Subsystems/Hub/Content/Sense_Hub/LoadData/synthetic-keys.htm



Figura 2. 11 Riferimento circolare.¹⁰

2.4 CREAZIONE DEI REPORT

Dopo aver caricato i dati dall'Editor di caricamento dati, l'analisi dell'app viene effettuata su fogli interattivi di *presentation* (vedi Figura 2.12). Ogni foglio ha un titolo e può contenere diversi oggetti che servono a rendere leggibile i dati caricati. Gli oggetti, per essere visualizzati, hanno bisogno di *dimensioni* e *misure*.

Le *dimensioni* sono create dai campi delle tabelle dei dati e determinano il loro raggruppamento nella visualizzazione [15]. Di solito esse rappresentano le fette di un grafico a torta oppure l'asse x nei grafici.

Le *misure* invece sono i calcoli che vengono fatti nelle visualizzazioni, esse sono generalmente funzioni di aggregazioni (somma, media, etc.) di più dimensioni [16]. Negli oggetti possono essere una colonna nelle tabelle oppure definire il diametro delle palle nei grafici a dispersione.

Nell'analisi può essere utile dover definire un gruppo di valori di dati da considerare, ad esempio confrontare delle vendite in diversi anni. Per fare questo in Qlik si fa ricorso alla *Set Analysis* che consente di definire un gruppo senza tener conto dalle selezioni correnti [17]. Essa deve essere utilizzata in una funzione di aggregazione ed è racchiusa tra parentesi graffe, inoltre può essere integrata soltanto nelle espressioni per i grafici e non nello script di caricamento dati. Un esempio è riportato nel *Paragrafo 3.7.2*.

¹⁰ Figura presa dal sito https://help.qlik.com/it-IT/cloud-services/Subsystems/Hub/Content/Sense_Hub/LoadData/circular-references.htm

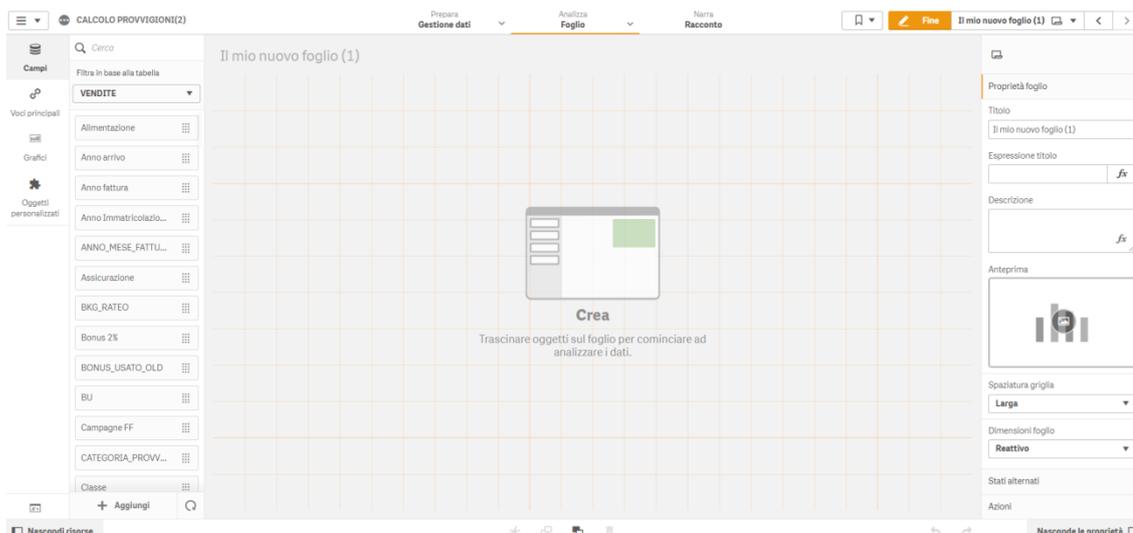


Figura 2. 12 Presentation di Qlik Sense (lato professional) ¹¹

Qlik dà la possibilità di prendere direttamente i campi caricati nel pannello delle risorse posto nella parte sinistra della presentazione cliccando la voce “campi” e trascinandoli direttamente sopra l’oggetto presente nel foglio e decidere se usarlo come dimensione o misura.

La scelta degli oggetti è data sotto la voce “Grafici”, quelli usati più frequentemente sono i seguenti:

- *KPI*, è un oggetto che mostra uno o due valori di misura che servono per visualizzare delle prestazioni. Ad esempio in un’analisi di Vendite può essere usato per mostrare il numero di vendite in un determinato periodo e la percentuale rispetto a tutto l’anno, inoltre è possibile modificare il colore e il simbolo a seconda di un’espressione, nella *Figura 2.13* il colore è verde perché le vendite sono in crescita rispetto all’anno precedente.



Figura 2. 13 Esempio di KPI in Qlik Sense ¹²

- *Tabella*, è utile quando si vogliono mostrare i dettagli di un’analisi, può contenere diverse dimensioni e misure. Nella *Figura 2.14* sono appunto presenti

¹¹ Figura presa da un’applicazione di Qlik Sense relativa ad un progetto per la concessionaria Gino S.p.A.

¹² Figura presa da un’applicazione di Tutorial fornita dal sito di Qlik Sense

i dettagli delle vendite di ogni cliente, in particolare le vendite fatte e le relative quantità e il margine prodotto con una riga di totali in grassetto.

Customer	Sales	Quantity	Margin (%)	# of Invoices	Average Sales per Invoice
Totals	\$104,852,674.81	1,816,372	4127.8%	38,314	\$2,736.67
A-2-Z Solutions	\$196,298.49	1,418	3841.7%	58	\$3,384.46
A-ARVIN Laser Resources	\$4,053.05	25	3792.6%	13	\$311.77
A Superior System	\$103,728.12	868	4074.5%	167	\$621.13
A&B	\$92,120.60	891	4202.9%	18	\$5,117.81
A&G	\$12,502.61	133	4708.0%	12	\$1,041.88
A&R Partners	\$30,392.45	156	3409.9%	6	\$5,065.41
A1 Datacom Supply	\$259,599.52	5,830	4025.7%	111	\$2,338.73
a2I	\$451.64	14	5983.7%	9	\$50.18
A2Z Solutions	\$69,977.36	454	4121.1%	94	\$744.44
AA-Wizard	\$94,209.44	917	4660.6%	41	\$2,297.79

Figura 2. 14 Esempio di tabella in Qlik Sense in cui sono presenti i dettagli sulle vendite dei vari clienti ¹²

- *Tabella pivot*, è usata per raccogliere i dati per evidenziare le relazioni tra gruppi diversi, ha sulle righe le dimensioni e sulle colonne le misure come è mostrato nella Figura 2.15 in cui le marche sono suddivise nelle due società e come misure vengono conteggiati i contratti di auto suddivisi anche in base al tipo di auto (nuova, usata); inoltre le misure vengono calcolate per ogni mese.

SOCIETA Q		MARCA Q		Mese Q	Valori	
				gen		
				CONTRATTI #	# NUOVO	# USATO
Totali				609	103	406
GINO_AUTO	Totali			389	156	233
	ALTRE MARCHE			85	8	77
	BMI			3	3	0
	BMW			27	0	27
	MINI			54	56	2
GINO_SPA	Totali			220	87	133
	A.M. Veicoli Commerciali			3	0	3
	A.M. Vetture			104	8	96
	Alfa Romeo			0	0	0
	Aston Martin			0	0	0
	Lotus			1	0	1
	Maserati			3	0	3
	Mercedes-Benz Veicoli Commerciali			0	4	0
	Mercedes-Benz Vetture			01	1	0
	Smart			06	27	0
	Volvo			02	27	0

Figura 2. 15 Esempio di tabella pivot ¹¹

- *Misuratore*, mostra una singola misura in un range prestabilito, offre una rappresentazione immediata delle prestazioni. Può essere indicato ad esempio nel mostrare il margine prodotto in un'azienda attraverso un cruscotto colorato in base all'intervallo oppure definendo con un'espressione la posizione in cui

cambiare colore; in questo modo diventa immediato capire se il valore è positivo o meno (vedi *Figura 2.16*).

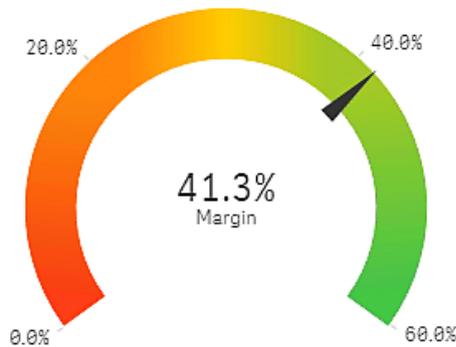


Figura 2. 16 Esempio di Misuratore¹³

- *Grafici*, ci sono numerosi tipi di grafici, ad esempio il *grafico lineare* per mostrare le tendenze nel corso del tempo come si può vedere in *figura 2.17*, in particolare sono presenti due linee che rappresentano due diverse dimensioni; il *grafico a barre* per confrontare più valori come in *Figura 2.18* dove per ogni mese sono presenti le quantità prodotte in tre diversi anni; oppure infine il *grafico a dispersione* per visualizzare la correlazione tra più misure, ad esempio vedere il rapporto che c'è tra il numero di vendite e il margine prodotto (*Figura 2.19*).

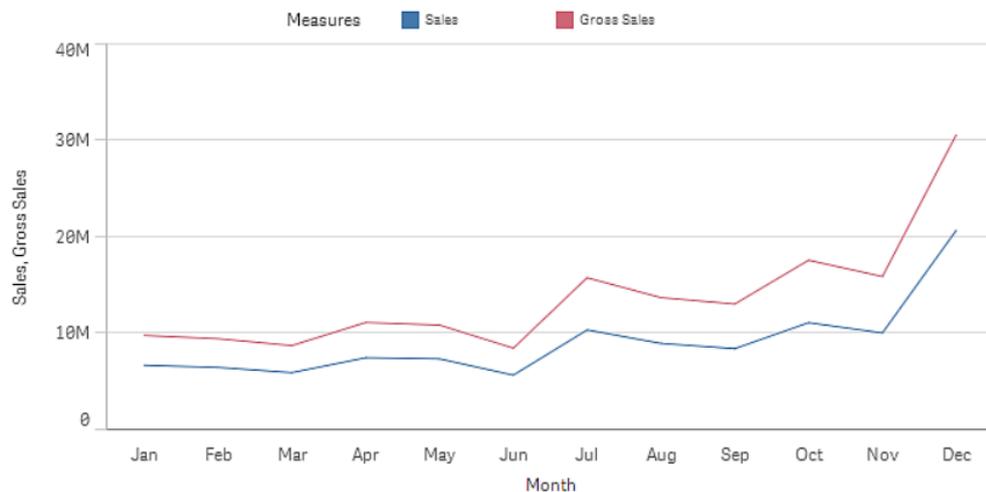


Figura 2. 17 Esempio di grafico lineare¹³

¹³ Figura presa da un'applicazione di Tutorial fornita dal sito di Qlik Sense

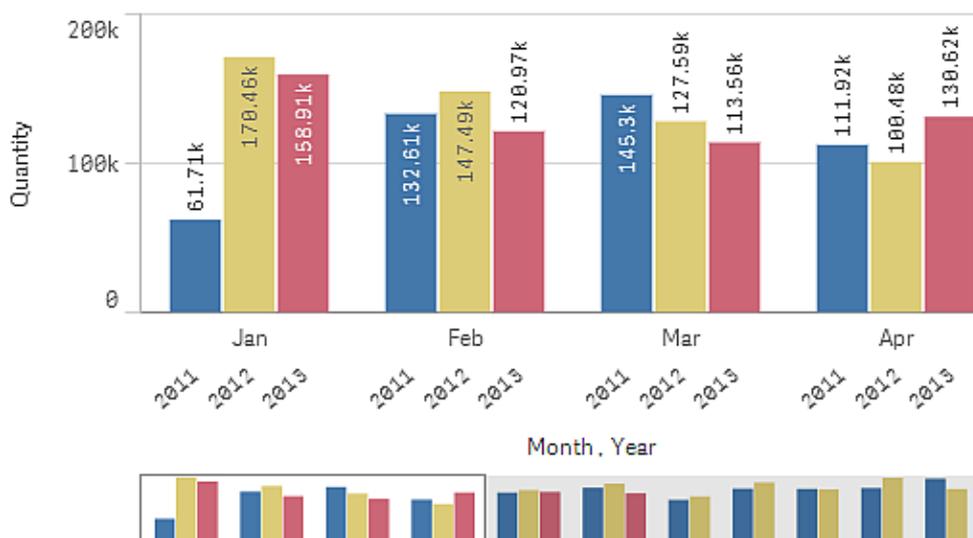


Figura 2. 18 Esempio di grafico a barre ¹³

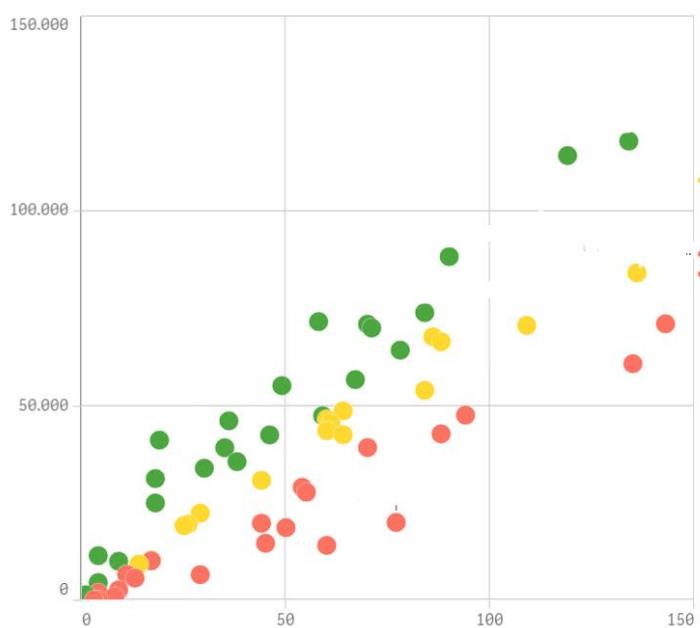


Figura 2. 19 Esempio di grafico a dispersione ¹⁴

- *Mappa*, è un oggetto complesso che offre una visualizzazione geografica dei dati cioè permette di visualizzare la distribuzione geografica di luoghi di interesse. Per esempio potrebbe essere utile osservare dove si concentrano maggiormente le vendite in Italia utilizzando un gradiente di colori, come mostrato in *Figura 2.20*. Questo è stato possibile perché nei dati è presente un campo che mi assegna le province ad ogni vendita.

¹⁴ Figura presa da un'applicazione di Qlik Sense relativa ad un progetto per la concessionaria Gino S.p.A.



Figura 2. 20 Esempio di mappa ¹⁴

- *Pulsante*, è un oggetto molto usato per selezionare dei valori di un campo e quindi creare dei filtri, per muoversi all'interno dei fogli ma anche per aprire un sito web. Ad esempio nella *Figura 2.21* sono presenti due pulsanti a cui vengono date le azioni di selezionare un particolare valore di due dimensioni diverse contemporaneamente (in questo caso le dimensioni sono la città e la categoria), inoltre il colore dello sfondo dei pulsanti è dinamico perché se il pulsante è selezionato lo sfondo diventa bianco.

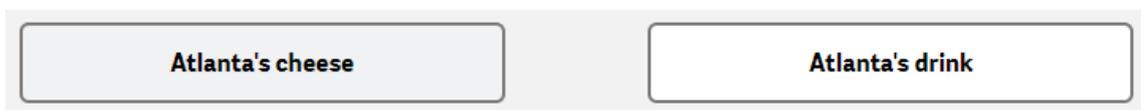


Figura 2. 21 Esempio di pulsante ¹⁵

Oltre i grafici classici sono a disposizione degli oggetti personalizzati in cui sono presenti le estensioni caricate esternamente e quelle date in dotazione da Qlik.

Dalla Presentation è possibile creare anche le variabili che possono essere modificate poi dagli utenti. In basso a sinistra infatti è presente un pulsante con scritto "*variabili*" in cui dare un nome alla variabile e assegnarle un valore. Esse vengono poi visualizzate di solito attraverso un'estensione di Qlik presente negli oggetti personalizzati che si chiama *Variable Input* da cui l'utente può direttamente cambiare il valore della variabile senza dover avere accesso al caricamento dati. Questo particolare oggetto può avere diversi formati, come cella dove scrivere un valore liberamente, una cella con i valori

¹⁵ Figura presa da un'applicazione di Tutorial fornita dal sito di Qlik Sense

possibili o ancora come cursore, quest'ultimo tipo può essere usato ad esempio per decidere rispetto a quanto storico di dati prendere (vedi *Figura 2.22*).

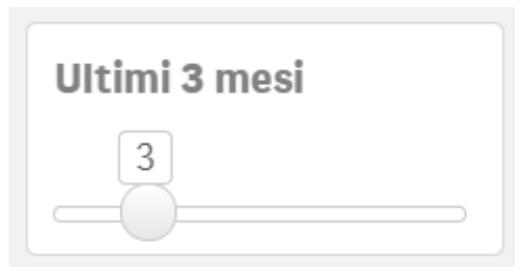


Figura 2. 22 Esempio di Variabile input ¹⁶

Nella creazione dei fogli, per ogni oggetto è presente una finestra a destra con le relative proprietà. In particolare è presente una sezione riguardante i dati, in cui sono presenti le dimensioni e misure usate per quell'oggetto e per ognuna di esse può essere scelta la formattazione dei dati. Un'altra sezione è dedicata all'ordinamento in cui si può scegliere in che ordine vedere i dati, può essere usato un ordinamento per misura, per una determinata dimensione oppure seguendo una formula. Dal punto di vista dell'aspetto vi è una sezione apposita in cui decidere se dare un titolo, sottotitolo e nota a piè di pagina dell'oggetto, piuttosto che definire lo stile o i colori.

Quando le dimensioni o misure vengono utilizzate frequentemente in più fogli è consigliabile salvarle come voci principali nel pannello risorse. Nella creazione di una dimensione principale è possibile selezionare dei colori per i singoli valori e scegliere tra dimensione singola o dimensione di drill-down. Il vantaggio di utilizzare la voce principale è dovuto al fatto che, quando vengono effettuate modifiche ad essa, queste vengono applicate a tutti gli elementi in cui la voce principale è inserita. Ad esempio se si deve cambiare una misura che viene utilizzata più volte, con l'utilizzo di una misura principale basta modificare questa per aggiornare tutte le istanze.

Un altro motivo per la creazione delle voci principali è quello di permettere agli utenti con una licenza *professional* di esplorare i dati creando i propri fogli e grafici che utilizzano le dimensioni e misure principali preimpostate. Diventa così possibile effettuare un'analisi self-service.

2.5 SELF-SERVICE DI QLIK SENSE

Come è stato accennato prima Qlik Sense dà la possibilità agli utenti di poter creare i propri fogli di analisi. In particolare l'utente dopo aver applicato un filtro e andando nelle

¹⁶ Figura presa da una applicazione demo di Qlik Sense realizzata durante il tirocinio in Bios Management.

selezioni ha la possibilità di osservare le informazioni strategiche che il motore cognitivo di Qlik ha scoperto. Ad esempio in un'analisi di vendite se si seleziona uno stato (Country=CAN) e si clicca sulle informazioni strategiche Qlik mostra che il 5% dei dipendenti (i dipendenti vengono presi dal campo *EmployeeName*) è escluso dalla selezione e che quindi la selezione fatta porta ad una possibile perdita del 37,8% delle vendite annuali (date dal campo *YearlySales*) (vedi Figura 2.23).

Inoltre l'utente può trovare nuove informazioni strategiche digitando su cosa vuole essere fatta l'analisi e il motore cognitivo prepara in automatico dei grafici adatti in base all'argomento che potranno poi essere aggiunti ad un ulteriore foglio di analisi creato dall'utente.

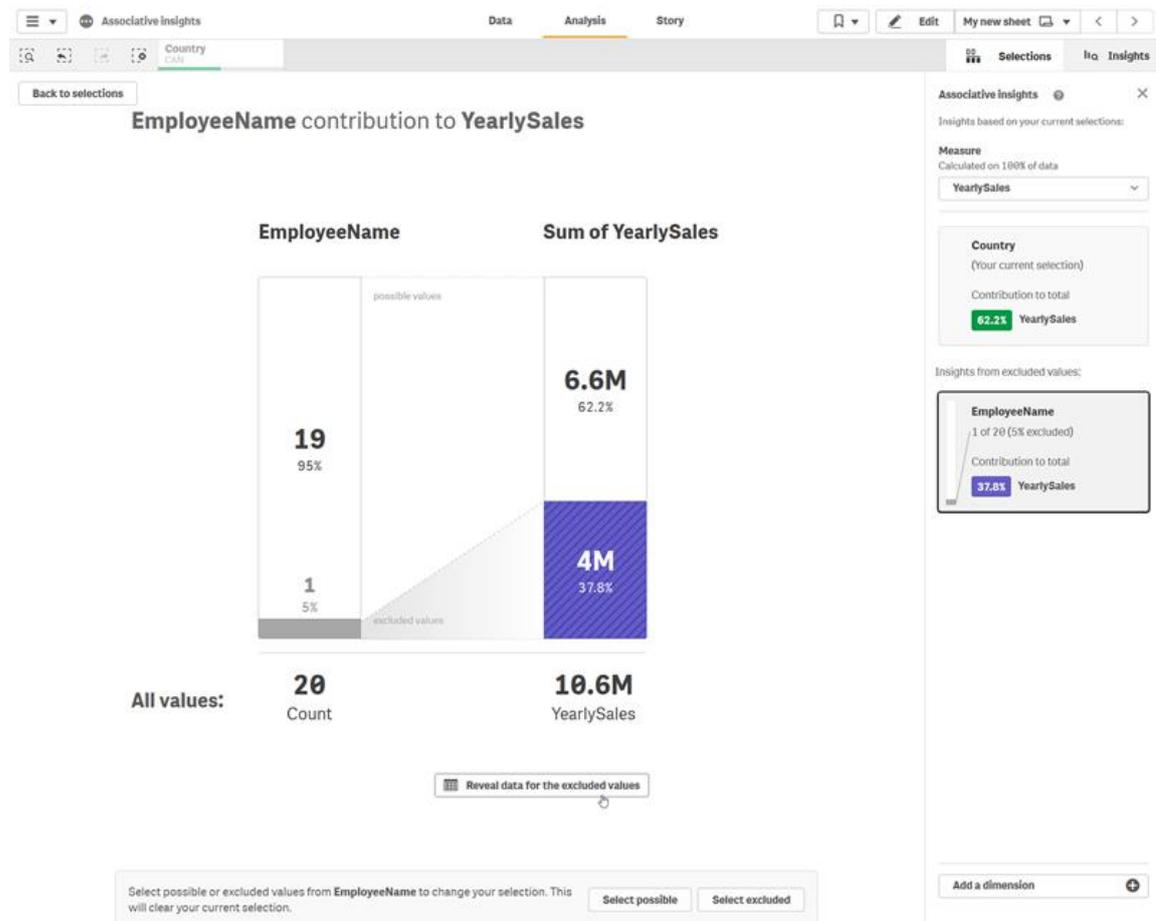


Figura 2. 23 Informazioni strategiche di Qlik Sense ¹⁷

L'utente che vuole aggiungere dei propri fogli non può modificare quelli predefiniti creati ma può solo duplicarli e applicare le modifiche su questi. Egli può decidere quali

¹⁷ Figura presa dal sito https://help.qlik.com/it-IT/sense/November2020/Subsystems/Hub/Content/Sense_Hub/Selections/associative-insights.htm

oggetti usare e quindi aggiungervi le dimensioni e misure ma può anche decidere di essere consigliati da Qlik Sense e trascinare soltanto i campi nel foglio e osservare cosa consiglia di utilizzare Qlik in base ai campi fornitogli. Questo è grazie all'intelligenza aumentata che ha questa piattaforma. Dopo aver creato i fogli, se l'utente ha il permesso, può anche pubblicarli e renderli visibili dagli altri utenti.

Infine attraverso la piattaforma Qlik Sense è possibile creare delle presentazioni in slide che prende il nome di *Racconto* (vedi *Figura 2.24*) [18]. Infatti per ogni oggetto è possibile fare una fotografia che potrà essere aggiunta ai fogli del racconto. Questa soluzione è molto utile quando c'è bisogno di una presentazione dell'analisi effettuata perché nel caso in cui si voglia approfondire qualche tema, attraverso le slide del racconto si può passare direttamente alle visualizzazioni da cui provengono le istantanee.

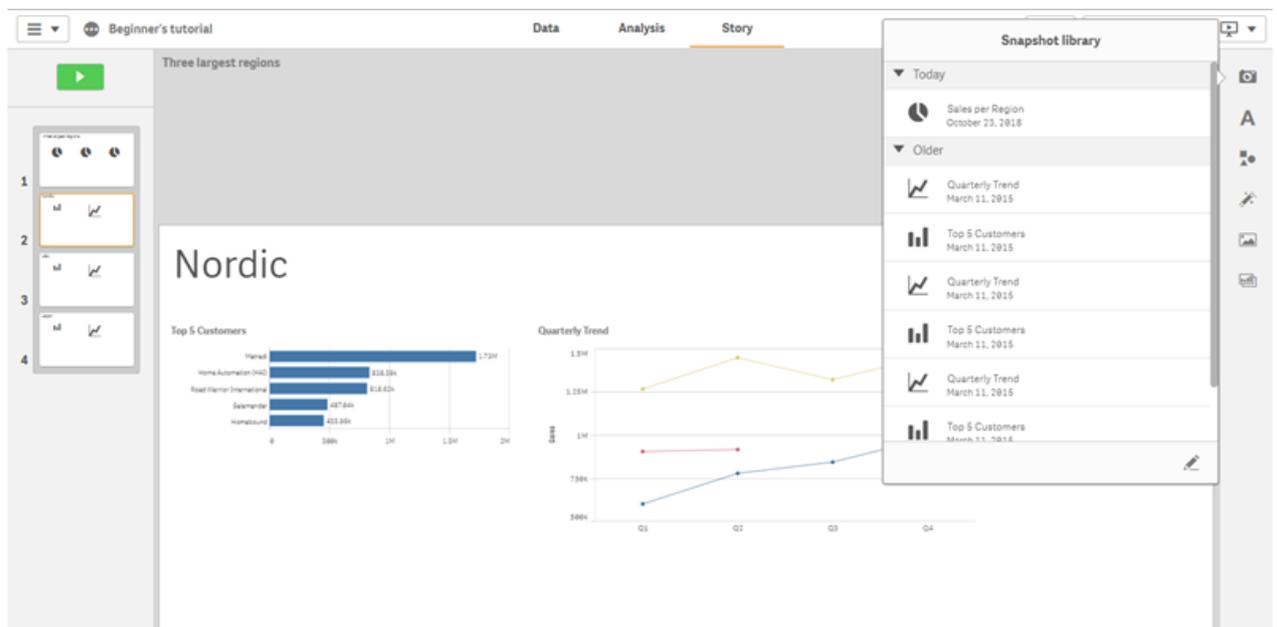


Figura 2. 24 Esempio di Racconto ¹⁸

¹⁸ Figura presa dal sito https://help.qlik.com/en-US/sense/November2020/Subsystems/Hub/Content/Sense_Hub/StoryTelling/Snapshots/snapshot-library.htm

Capitolo 3

IL PROGETTO DI BUSINESS INTELLIGENCE SU QLIK SENSE

Nel 1959 nasce la prima concessionaria Mercedes-Benz nella provincia di Cuneo, la *Gino Rag. Felice & figlio snc* e nel 1972 diventa *Gino Società per Azioni (S.p.A.)* con sede a Cuneo [19]. Negli anni *Gruppo Gino* si è ingrandito diventando uno dei dealer di riferimento nel nord d'Italia, ha aperto numerose sedi sul territorio nazionale, in Piemonte, Liguria fino a raggiungere il centro Italia con la Toscana, ed è arrivato a rappresentare 10 case costruttrici con particolare attenzione alle premium (Mercedes, Bmw, Volvo etc.).

Nel 2020 l'azienda ha iniziato a collaborare con l'azienda *Bios Management* con numerosi progetti utilizzando Qlik Sense come piattaforma di BI al fine di consolidare e migliorare la propria performance. Tra i progetti svolti, in questa tesi saranno presentati quelli del *Calcolo delle Provvigioni* e delle *Performance dei venditori*. Entrambi i temi erano sviluppati utilizzando il gestionale aziendale e completando le funzioni non disponibili con Excel e la richiesta dell'azienda è stata di automatizzare, grazie alla BI, l'intero processo recuperando sicurezza del calcolo e riducendo in modo drastico il volume delle attività manuali.

La flessibilità di Qlik Sense e l'esperienza di Bios Management ha così reso possibile la modifica delle regole interne di calcolo delle provvigioni, abilitando una logica più business oriented che premia maggiormente il merito: durante la fase di realizzazione è stato fornito un tool di simulazione con cui, applicando il nuovo calcolo parametrico sulle vendite passate, è stato possibile tarare le impostazioni dello stesso arrivando ad una messa a punto in modo veloce ed efficace. Il calcolo provvigionale ha una competenza mensile e si basa sulle vendite avvenute in un certo mese sulla base della data della fattura di vendita.

Anche il *Calcolo delle Performance* si basa sulla lettura delle stesse vendite del Calcolo Provvisoriale e l'analisi ha una competenza mensile. Le performance vengono calcolate per ogni venditore e si basano sulle vendite

Inoltre il cliente ha richiesto un'ulteriore applicazione *Trend Performance* per poter vedere il trend delle Performance nei vari periodi precedenti.

3.1 BASE DATI E PROCESSI DI ETL

La piattaforma Qlik può utilizzare, da un punto di vista di architettura, un DWH esterno dove i vari flussi di ETL immagazzinano i dati provenienti dalle base dati operative; questa possibilità, che con altre piattaforme è di fatto necessaria, con Qlik può essere realizzata direttamente dalla BI, evitando l'utilizzo di più tecnologie e permettendo dei sostanziali risparmi a livello di costo.

Gino Spa utilizza al momento uno dei quattro principali gestionali per dealer auto (DMS): il progetto è stato sviluppato in modo da poter supportare qualsiasi sistema legacy di base, prescindendo dalle sue strutture informative e basandosi su output normalizzati

Essendo un solo sistema e limitate fonti dati, la scelta è stata quella di implementare direttamente su Qlik la parte di ETL per fornire le basi dati normalizzate alle applicazioni di BI realizzate con Qlik Sense. In questo paragrafo viene spiegato come avviene il flusso dei dati a partire dalla base dati sorgente delle applicazioni fino alle dashboard utente come si può osservare dallo schema in *Figura 3.1*.

SOURCE:

È l'elenco delle fonti dati, cioè i file aggiornati quotidianamente e forniti dal gestionale <erp_vendor> che si trovano in una directory chiamata *FTP*, e dei file di mapping, che contengono regole di classificazione o raggruppamento, gestiti a livello di progetto per singola applicazione. Inoltre è stata aggiunta un'ulteriore cartella *INPUT* dove sono stati inseriti dei file excel che contengono dati aggiuntivi da inserire nelle fonti dati perché servono nelle analisi ma non erano forniti dal gestionale.

ETL I LIVELLO:

È uno strato di applicazioni Qlik dedicate a preparare le base dati intermedie che possono essere rese disponibili a più applicazioni con evidenti risparmi di tempo e risorse. Dalla schedulazione di *reload* delle applicazioni di ETL deriva la possibile

schedulazione di aggiornamento delle applicazioni visibili al Business. Nel nostro caso abbiamo l'applicazione *ETL VENDITE* che legge i file presi nella directory FTP e salvo per motivi di timing di aggiornamento a richiesta utente non è visibile al Business.

QVD DWH:

È il repository che contiene le tabelle dati pronte per essere utilizzate nelle diverse applicazioni presenti nella USER APPS. La struttura può essere considerata una sorta di Data warehouse dove sono raggruppati tutti i dati che saranno poi usate nello strato superiore.

Nella cartella VENDITE sono presenti tabelle di dati estratti e rielaborati dallo strato applicativo ETL di primo livello, che vengono ricaricati con una frequenza indicata che dipende dalla disponibilità del dato e dalle necessità del Business.

Nel caso si necessiti di analisi con stratificazione temporale (automatica o manuale) in queste cartelle saranno presenti serie di file di tipo QVD con suffisso temporale, generati dallo strato di ETL.

In caso di problemi da parte dei sistemi sottostanti operativi le applicazioni di tipo ETL talvolta ricostruiscono o riclassificano dati mancanti o errati: il DWH associativo è il patrimonio informativo dell'azienda e rappresenta spesso l'unico punto di continuità che permette analisi di trend anche in situazione di discontinuità dei tool gestionali operativi.

USER APPS:

È lo strato visibile al Business suddiviso in stream che gestiscono l'accesso abilitando specifici gruppi di utenti. Al di sopra di ogni applicazione è indicata la schedulazione di *reload*. Nel caso di *ANALISI VENDITE*, l'applicazione di analisi risulta al tempo stesso estrattore dei dati sorgenti per le altre applicazioni presenti nello stream *PROVVIGIONI* e visualizzatore degli stessi.

Nei prossimi paragrafi verranno analizzate le applicazioni Qlik Sense presenti nello schema precedente.

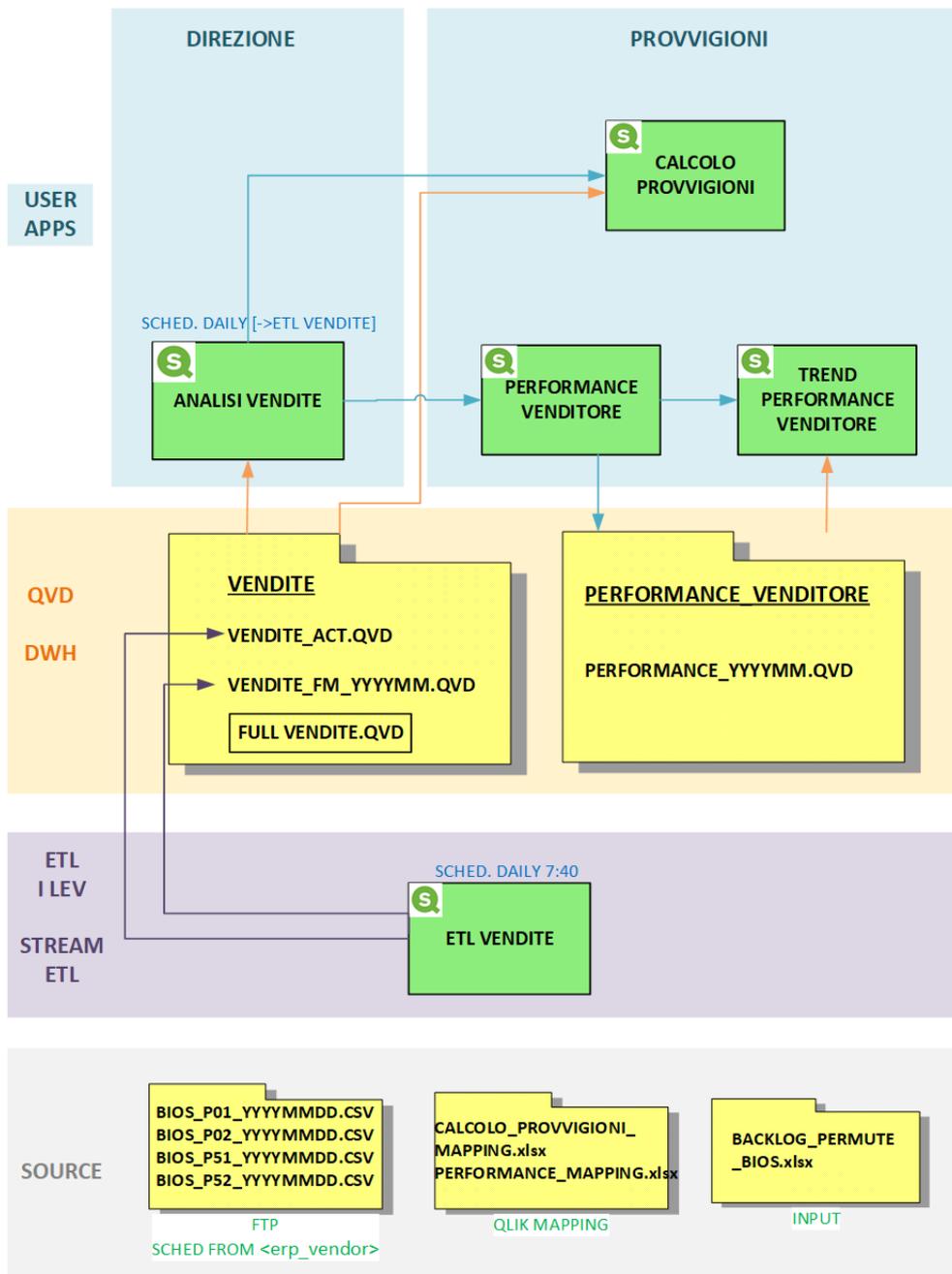


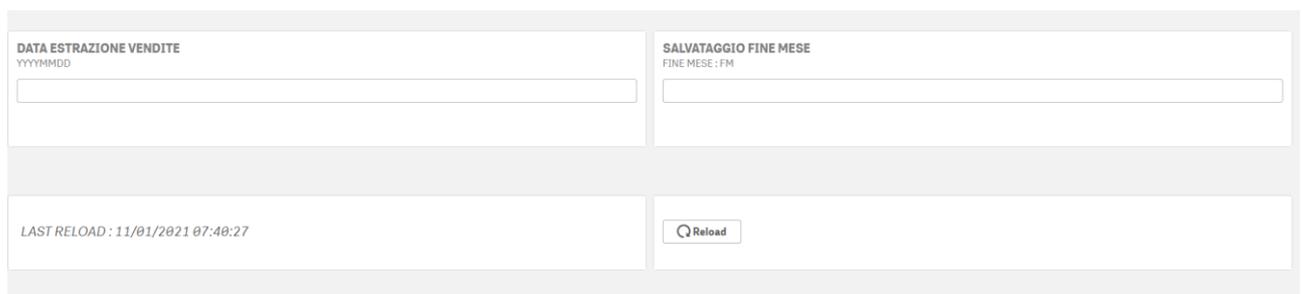
Figura 3. 1 Schema flusso dati

3.2 ETL VENDITE

L'applicazione *ETL VENDITE* legge quattro file aggiornati quotidianamente nella directory *FTP* da <erp_vendor>. I file sono in formato "csv" corrispondenti ai dati delle vendite delle società *Gino S.p.A.* e *Gino Auto* suddivisi a loro volta in vendite di auto nuove e usate.

In particolare l'applicazione va a leggere le vendite effettuate da *Gruppo Gino* aggiornate all'ultimo caricamento avvenuto nella serata del giorno precedente e genera un unico file in formato "qvd" (*VENDITE_ACT.QVD*) che viene salvato nella cartella *QVD_DWH/VENDITE* e che contiene tutte le informazioni relative a tutte le vendite dei quattro file aggiornati ogni giorno.

Inoltre l'app *Qlik Sense ETL VENDITE* consente al Controllo di Gestione di storicizzare il dettaglio del venduto di un determinato "Fine Mese" in qualsiasi momento. Questa funzione permette ad un utente designato di congelare e storicizzare i dati di un certo mese subito dopo la riunione di condivisione ed approvazione dei dati mensili (M3). Infatti nella sua unica pagina di presentation (vedi *Figura 3.2*) l'utente può inserire la data di estrazione delle vendite, che andrà a popolare la variabile *v_analysis_date*, scrivere 'FM' nel box salvataggio fine mese, che sarà inserito nella variabile *v_store_mode*, ed infine cliccando il pulsante di *reload* i dati verranno caricati. Essendo la variabile *v_store_mode*= 'FM' allora verrà creato un ulteriore QVD oltre quello *actual* (relativo alle vendite caricate il giorno precedente) chiamato *VENDITE_FM_YYYYMM.QVD* che conterrà le vendite prese dai file di <erp_vendor> che nel nome hanno la data *v_analysis_date* che ha segnato l'utente.



The screenshot shows a dashboard interface for ETL VENDITE. It features two main input sections at the top. The left section is titled "DATA ESTRAZIONE VENDITE" and includes a sub-label "YYYYMMDD" above a text input field. The right section is titled "SALVATAGGIO FINE MESE" and includes a sub-label "FINE MESE: FM" above another text input field. Below these sections, on the left, is a status indicator "LAST RELOAD: 11/01/2021 07:40:27". On the right, there is a button labeled "Reload" with a circular refresh icon.

Figura 3. 2 Dashboard ETL VENDITE

L'*ETL VENDITE* viene schedulata giornalmente alle ore 7.40 salvando sempre un QVD che contiene le vendite *actual* ed è predecessore dell'applicazione *ANALISI VENDITE*.

3.3 ANALISI VENDITE

L'applicazione *ANALISI VENDITE* era stata già richiesta da Gruppo Gino in un periodo precedente per analizzare le vendite e poter effettuare anche confronti tra situazioni temporali differenti; tutte le viste e le logiche di analisi del cliente sono state recepite ed automatizzate nella BI di Qlik Sense migliorando il processo di analisi ed abilitando nuove logiche che in precedenza avrebbero implicato costi notevolmente superiori, tali da mettere in dubbio la loro economicità.

I file letti dall'applicazione sono *VENDITE_ACT.QVD* e tutti quelli del tipo *VENDITE_FM_YYYYMM.QVD* salvati rispettivamente in modalità *actual* e *FM* (fine mese).

In questa fase i dati vengono elaborati per permettere un'analisi più veloce e immediata. Ad esempio a tutti i campi numerici viene sostituito il punto con la virgola (vedi *Figura 3.3*) per non fare confusione con i numeri decimali e il divisore delle centinaia per cui molto spesso porta a conteggi errati nelle misure. Questa operazione viene fatta con il comando **REPLACE** che ad ogni valore del campo datogli in prima posizione va a sostituire la stringa in seconda posizione con quella in terza posizione.

```
35 REPLACE("Prezzo di Acquisto", '.', ',') AS "Prezzo di Acquisto",
36 REPLACE("Prezzo di Vendita", '.', ',') AS "Prezzo di Vendita",
37 REPLACE("Prezzo di Vendita con IVA", '.', ',') AS "Prezzo di Vendita con IVA",
38 "Margine 1",
39 "Margine 1 %",
40 REPLACE("Ripr Esterni", '.', ',') AS "Ripr Esterni",
41 REPLACE("Giro Margine", '.', ',') AS "Giro Margine",
42 REPLACE("Ripr Interni", '.', ',') AS "Ripr Interni",
43 "Quota Parte RipInt> 30gg da Fattura mezzo",
44 "Tot Ripr",
45 "Inc % Tot Ripr Prezzo",
46 REPLACE("Provvigioni Passive", '.', ',') AS "Provvigioni Passive",
47 "Margine 2",
48 "Margine 2 %",
49 REPLACE("Campagne FF", '.', ',') AS "Campagne FF",
50 REPLACE("Margine 3", '.', ',') AS "Margine 3",
51 REPLACE("Margine 3 %", '.', ',') AS "Margine 3 %",
```

Figura 3. 3 Parte di script di ANALISI VENDITE

In assenza di Data Warehouse l'eventuale indisponibilità o non correttezza di alcuni campi utilizzati per le analisi rende necessario il calcolo delle dimensioni direttamente nell'applicazione. Infatti ogni applicazione può avere delle regole di classificazione, raggruppamento, clusterizzazione o selezione dei record che non sarebbero implementabili senza una tabella controllata ed aggiornata dal Business. La possibilità di

associare di volta in volta alcuni elementi in gruppi per eseguire un controllo dinamico può essere eseguita in un file di mapping.

Una dimensione calcolata, che è rilevante anche nel calcolo delle Provvigioni, è *DRIVER CANALE DI VENDITA*. Infatti nelle Provvigioni di Gruppo Gino il calcolo considera diversi parametri a seconda del tipo di vendita effettuata (ad esempio fatta ad un privato piuttosto che ad un commerciante).

Nella casistica del campo *Tipo di Vendita* = 'Commerciante' il cliente ha definito un ulteriore suddivisione che distingue *Noleggi*, *Flotte*, *Auto-Immatricolazione* e *Commerciante*. Questa dimensione non è disponibile in modo pulito nella base dati, quindi una specifica dimensione è stata calcolata direttamente, attraverso degli *if* annidati, secondo la seguente regola:

```
se [Ragione Sociale Cliente] appartiene al Mapping DRIVER_COMMERCIANTE
allora inserisco le suddivisioni date dal Mapping; altrimenti
    se [Provvigioni Passive]>0 allora 'COMMERCIANTE'; altrimenti
se il VENDITORE appartiene al Mapping VENDITORI_VENDITE_A_COMMERCIANTI
allora 'COMMERCIANTE';
altrimenti 'PRIVATO'.
```

Il codice relativo nello script di Qlik Sense è

```
IF(INDEX(UPPER("Tipi di vendita"), 'COMMERCIANTE')>0
OR INDEX(UPPER("Tipi di vendita"), 'TRADING')>0 ,
'COMMERCIANTE',
APPLYMAP('MAP_COMMERCIANTE_NOLEGGI_FLOTTE', RAGIONE_SOCIALE_CLIENTE,
if(WildMatch(upper(RAGIONE_SOCIALE_CLIENTE), '*ALPHABET*', '*ARVAL*', '*CHARTERWAY*')>0,
'NOLEGGI',
IF("Provvigioni Passive">0, 'COMMERCIANTE',
if(Applymap('MAP_VENDITORI_VENDITE_COMMERCIANTI_FROM', "Nome Venditore")<DATA_ANALISI_D
AND Applymap('MAP_VENDITORI_VENDITE_COMMERCIANTI_TO', "Nome Venditore")>DATA_ANALISI_D,
'COMMERCIANTE',
'PRIVATO'))))) AS DRIVER_CANALE_VENDITA,
```

Le funzioni presenti nel codice sono così definite:

- **INDEX**: ricerca la stringa 'COMMERCIANTE' nel campo "Tipi di vendita" e individua la posizione iniziale se trovata.
- **UPPER**: applica il carattere maiuscolo a tutti i caratteri della stringa in input "Tipi di vendita" in modo da uniformare del tutto la scrittura.
- **WILDMATCH**: confronta il primo parametro RAGIONE_SOCIALE_CLIENTE con tutti i parametri seguenti e restituisce la posizione dell'espressione corrispondente. In realtà in questo caso va a vedere se nei valori del campo è presente una di quelle parole perché l'asterisco permette la presenza di altre lettere prima e dopo.

3.5 CALCOLO PROVVIGIONI

Il calcolo delle provvigioni, cioè della parte di retribuzione del Venditore che si basa sul lavoro svolto durante il mese, tiene conto di tre punti.

Il primo punto è la Provvigione del Ferro, in particolare per ogni vettura fatturata viene maturata una provvigione in base a dei criteri e ogni venditore ha un obiettivo di pezzi fatturati in base alla filiale in cui lavora e il brand che rappresenta.

Il secondo punto è la Provvigione F&I che è una percentuale delle Provvigioni dei servizi Finanziari. Tale percentuale può assumere diversi valori in base a fasce stabilite.

L'ultimo punto è invece il Bonus usato, un ulteriore premio per le vendite di auto usate presenti in magazzino oltre a un periodo definito come parametro.

L'analisi ha due livelli di calcolo, nel primo vengono calcolate le provvigioni sul singolo stock auto venduto, mentre nel secondo livello viene calcolata la Provvigione totale sul singolo venditore.

Per il fine di questa tesi viene descritto nel dettaglio solo il primo punto per vedere la procedura che è alla base del calcolo, i punti rimanenti seguono un'analogia procedura.

3.5.0 PROCEDURA APPLICATIVA DEL CALCOLO DELLE PROVVIGIONI

Nella *Figura 3.5* viene rappresentata la procedura di calcolo che viene eseguita mediante il *Reload* dell'applicazione lanciato dall'utente in funzione della presenza di nuovi dati di vendita disponibili a seguito della riunione M3 mensile o per necessità di verifica dell'andamento delle provvigioni nel mese in corso o nel mese precedente prima dell'ufficializzazione M3.

Di seguito la serie di step previsti:

- 1) Import dati di Vendita da chiusura M3 o Actual.
- 2) Recupero Provvigioni a fronte di ritiro Permuta.
- 3) Verifica del mapping con segnalazione incompletezza tabelle utente relative ai venditori o alle categorie provvigionali necessarie per gestire la totalità delle vetture vendute nel mese.
- 4) Calcolo di I livello: calcolo provvigioni per ogni record presente nelle vendite e considerato valido per il calcolo delle provvigioni.


```

1 SET ThousandSep='.';
2 SET DecimalSep=',';
3 SET MoneyThousandSep='.';
4 SET MoneyDecimalSep=',';
5 SET MoneyFormat='#.##0,00 €;-#.##0,00 €';
6 SET TimeFormat='hh:mm:ss';
7 SET DateFormat='DD/MM/YYYY';
8 SET TimestampFormat='DD/MM/YYYY hh:mm:ss[.fff]';
9 SET FirstWeekDay=0;
10 SET BrokenWeeks=0;
11 SET ReferenceDay=4;
12 SET FirstMonthOfYear=1;
13 SET CollationLocale='it-IT';
14 SET CreateSearchIndexOnReload=1;
15 SET MonthNames='gen;feb;mar;apr;mag;giu;lug;ago;set;ott;nov;
16 SET LongMonthNames='gennaio;febbraio;marzo;aprile;maggio;giu
17 SET DayNames='lun;mar;mer;gio;ven;sab;dom';
18 SET LongDayNames='lunedì;martedì;mercoledì;giovedì;venerdì;s
19 SET NumericalAbbreviation='3:k;6:M;9:G;12:T;15:P;18:E;21:Z;
20
21 SET v_mode='1';
22
23 LET v_vendite_period=if('${v_run_type}'='WIP','ACT','FM_&');
24 LET v_rappel_file=if('${v_run_type}'='WIP','','${v_period}')
25 let v_vendite_period_d=monthstart(makedate(left('${v_period}',

```

Figura 3. 6 Divisione in sezioni dello script dell'applicazione Calcolo Provvigioni

3.5.1 LETTURA VENDITE

Il calcolo delle Provvigioni ha una competenza mensile e quindi si basa sulle vendite avvenute in un certo mese sulla base della fattura di vendita.

Come abbiamo accennato sopra è l'utente che sceglie il periodo di analisi tramite la prima pagina di presentation (vedi Figura 3.7).



Figura 3. 7 Pagina di Intro nella presentation dell'applicazione Calcolo Provvigioni

Infatti egli deve inserire il periodo in formato anno-mese e scegliere tra DEF e WIP.

Utilizzando il pulsante DEF si selezionano le vendite storicizzate dopo l'M3 del periodo precedente; mentre il calcolo può avvenire durante il mese per verificare l'andamento delle provvigioni in maturazione utilizzando il pulsante WIP. L'utilizzo del pulsante WIP evita l'esecuzione degli step della procedura a partire dallo step 4).

A livello di script la lettura delle vendite ha una sezione dedicata da cui prende il nome in cui vengono caricati i dati presi nella cartella QVD_DWH/VENDITE salvati dall'applicazione ETL VENDITE. Per fare questo i pulsanti WIP e DEF scrivono su una variabile chiamata *v_run_type* e il periodo sulla variabile *v_period*, quindi viene creata un'altra variabile, *v_vendite_period*, in questo modo:

```
LET v_vendite_period=if('$ (v_run_type)'='WIP','ACT','FM_'&'$(v_period)');
```

cioè assume il valore 'ACT' se l'utente ha schiacciato WIP altrimenti 'FM_(v_period)' perché l'utente vuole il calcolo relativo al fine mese. Inoltre viene utilizzato il comando LET perché la variabile assume un valore calcolato.

A questo punto tramite il comando LOAD viene creata una tabella chiamata [VENDITE] che carica i dati dal file relativo alle richieste dell'utente (FROM [lib://QVD_DWH/VENDITE/VENDITE_\$(v_vendite_period).QVD]). È però importante notare che di tutti i dati vengono caricati solo quelli con Anno e Mese Fattura relativo al periodo scelto, in modo da avere solo le vendite di quel mese.

Tuttavia bisogna caricare alcuni campi indispensabili per il calcolo delle Provvigioni dal file FULL_VENDITE.QVD perché sono stati calcolati solo nella dashboard ANALISI VENDITE e quindi non presenti nel QVD di ETL VENDITE. Per fare questo si utilizza il comando *left join*:

```
[VENDITE]:  
LOAD  
    [SUB_BU],  
    [SOCIETA],  
    [TIPO_FILE],  
    [Stock],  
    [Commissione],  
    [Targa],  
    [Anno arrivo],  
    [Mese arrivo],  
    [Anno fattura],  
    [Mese fattura],  
    [Ubicazione],  
    [BU],  
    [Sub BU],  
    [Tipo Mezzo],  
    [Marca],  
    [Classe],  
    [Macro Raggrupp/Macro-Modello],  
    [Modello],  
    [...]  
    REPLACE (CAPITALIZE ([Nome Venditore]), '_old', '') AS [VENDITORE_LK]  
FROM [lib://QVD_DWH/VENDITE/VENDITE_$(v_vendite_period).QVD] (qvd)  
where [Anno fattura]=left ($(v_period),4) and num(month([Data fattura vendita]))=right ($(v_period),2) ;
```

```

left join

LOAD distinct

    Stock,
    maxstring( COMMERCIANTE_PRIVATO)as COMMERCIANTE_PRIVATO ,
    maxstring(DRIVER_CANALE_VENDITA) as DRIVER_CANALE_VENDITA,
    maxstring(ANNO_MESE_FATTURA) as ANNO_MESE_FATTURA,
    maxstring(TIPO_VENDITA_USATO) as TIPO_VENDITA_USATO
FROM [lib://QVD_DWH/          /VENDITE/FULL_VENDITE.QVD]
(qvd)
where [Anno fattura]=left( $(v_period),4) and num(month([Data fattura vendita]))=right( $(v_period),2)
group by Stock;

```

esso consente di eseguire un'unione della tabella [VENDITE] con quella che carica i dati dal file *FULL_VENDITE.QVD* prendendo come collegamento lo stock della vendita. il risultato è una sola tabella che ha tutti i campi delle due tabelle ma che prende solo i valori della prima.

I campi caricati che hanno maggiore rilevanza per il calcolo della Provvigione sulla singola vendita sono i seguenti:

- Nome Venditore: rinominato VENDITORE_LK per poterlo collegare successivamente con altre tabelle, è di fondamentale importanza perché le Provvigioni vengono calcolate per singolo venditore e quindi rappresenterà la chiave univoca.
- Provv Fin e Provv Pol: sono le provvigioni dei finanziamenti e le provvigioni delle polizze vendute insieme all'auto che servono nel calcolo intermedio delle provvigioni F&I.
- Marca: si intende la marca del veicolo venduto, serve per definire alcuni parametri nel calcolo delle provvigioni
- GG stock: indicano quanti giorni il veicolo era in magazzino, servono per attribuire un aumento della provvigione nel caso in cui i giorni superano una certa soglia.
- CLASSE_ANZIANITA: è un campo creato prendendo due soglie definite in presentation, a cui viene attribuito c1, se i GG stock sono minori della prima soglia, c2 se è compresa tra le due soglie, c3 altrimenti.
- Stock: è il codice del veicolo venduto, rappresenterà la chiave per il primo livello di calcolo basato sulla vendita.
- SUB_BU: indica se il veicolo è 'NUOVO' oppure 'USATO', nei due casi le provvigioni vengono calcolate diversamente.
- Driver Canale di Vendita: campo calcolato nella dashboard Analisi Vendite, per i diversi valori vengono usati parametri di calcolo differenti.
- Prezzo di Vendita con IVA: serve come base del calcolo della Provvigione del Ferro.

Inoltre vengono aggiunti tre campi che assumeranno lo stesso valore per tutti i record per poter differenziarli da quelli relativi alle permutate presentati nel *paragrafo 3.5.2*:

- BKG_RATEO: che assume il valore 0, cioè non è prevista provvigione per la permuta.
- TIPO_RECORD: ogni record preso dalle vendite sarà di tipo Standard (STD).
- RECORD_MESE: è un campo valorizzato a uno che mi permetterà di sommare le vendite effettive del Venditore che servono nel calcolo per il raggiungimento dei propri obiettivi.

Infine l'utente nella parte di presentation ha a disposizione una pagina chiamata "Dettaglio Vendite" con tutte le informazioni relative alle vendite di quel mese. Questa pagina è molto utile quando viene selezionato un particolare Venditore perché mostrerà solo le vendite relative a quel venditore e i relativi dettagli per ogni sua vendita eseguita (vedi Figura 3.8).

Società	Stock	Targa	Anno arrivo	Mese arrivo	Anno fattura	Mese fattura	Data Fattura a Cliente	Data fattura vendita	BU	SUB_BU	Marca	Classe	Mac
Totale													
GINO AUTO			2020	09	2020	11	23/11/2020	23/11/2020	Vetture	NUOVO	MINI	Mini	F68
GINO AUTO			2020	11	2020	11	27/11/2020	27/11/2020	Vetture	NUOVO	BMW	BMW Serie 1	F48
GINO AUTO			2020	09	2020	11	16/11/2020	16/11/2020	Vetture	NUOVO	MINI	Mini	F68
GINO AUTO			2020	09	2020	11	12/11/2020	12/11/2020	Vetture	NUOVO	BMW	BMW serie X	G01
GINO AUTO			2018	05	2020	11	30/11/2020	30/11/2020	Vetture	USATO	BMW	BMW Serie 3	Seri
GINO AUTO			2020	10	2020	11	18/11/2020	18/11/2020	Vetture	NUOVO	MINI	Mini	F68
GINO AUTO			2020	11	2020	11	30/11/2020	30/11/2020	Vetture	NUOVO	MINI	Mini	F55

Figura 3. 8 Pagina di dettaglio delle Vendite nell'applicazione Calcolo Provvigioni con selezione di un Venditore.

3.5.2 RECUPERO PROVVIGIONI A FRONTE DI RITIRO PERMUTA

Nel sistema provvigionale precedente ai venditori che vendevano un'auto e ne ritiravano un'altra, veniva pagata solo una parte della provvigione totale mentre la restante veniva attribuita nel momento in cui la macchina ritirata veniva venduta. Nel nuovo sistema questa regola non è stata inserita ma bisogna comunque poter pagare la parte rimanente a quei venditori quando l'auto che avevano ritirato verrà venduta. Poiché l'informazione della permuta non è presente nelle vendite, è stato richiesto al cliente di fornire un file per poter recuperare tale informazione. In particolare è stato fornito un file excel BACKLOG_PERMUTE_BIOS che contiene le seguenti informazioni:

- STOCK_ID: è il codice dell'auto che era stata venduta quando c'è stata la permuta, ovvero sarà l'auto su cui deve essere pagata la provvigione.
- NOME_VENDITORE: è il nome del venditore a cui deve essere pagata la provvigione.
- Provvigione X%: è la parte di provvigione da versare al Venditore.

- STOCK_PERMUTA: è il codice dell'auto che era stata ritirata che servirà come chiave per cercare quando è stata venduta e quindi quando pagare la restante Provvigione.

L'obiettivo è quello di realizzare dei record fittizi con il Venditore che aveva fatto la permuta e lo stock relativo all'auto che aveva venduto in passato da aggiungere alla tabella [VENDITE] quando l'auto della permuta viene venduta.

Per fare questo viene creata una tabella leggendo tutte le vendite dal file VENDITE_ACT.qvd e ponendo tutti i campi numerici a zero ma senza prendere il campo Venditore. Attraverso una *right join* con il file BACKLOG_PERMUTE_BIOS.xlsx si andrà a prendere solo quei record nelle vendite che hanno lo stock uguale allo STOCK_PERMUTA a cui vengono aggiunti i campi del file. Quindi viene salvata la tabella in un qvd che contiene tutte le vendite delle auto di permuta con il Venditore che aveva fatto la pratica, l'importo da versargli e lo stock della macchina che aveva venduto. Per poter considerare solo le vendite relative al periodo richiesto viene creata la tabella [BKG] che legge il qvd appena salvato e crea i campi TIPO_RECORD che assume il valore Backlog (BKG) e RECORD_MESE che assume il valore 0 perché non verranno conteggiate nelle vendite del mese; ma attraverso la seguente *inner join* si andranno a considerare solo i record che hanno Data Fattura di quel periodo:

INNER JOIN

```
LOAD distinct Stock,
      [SUB_BU]

FROM [lib://QVD_DWH/          /VENDITE/VENDITE_$(v_vendite_period).QVD]
(qvd)
where num(year([Data fattura vendita]))=left($(v_period),4)
and num(month([Data fattura vendita]))=right($(v_period),2) ;
```

A questo punto è possibile rinominare il campo STOCK_ID come Stock in modo da avere un collegamento con la tabella [VENDITE] ed eliminare l'altro perché la Provvigione era relativa all'auto che era stata venduta in precedenza con il comando *drop field*:

```
DROP FIELD Stock;

NOCONCATENATE
BKG_F:
LOAD *,
      STOCK_ID AS Stock
RESIDENT BKG;
drop table BKG;
RENAME TABLE BKG_F TO BKG;
```

Infine quando viene creata la tabella [VENDITE] (vedi *Paragrafo 3.5.1*) vengono aggiunti i record della tabella [BKG]:

CONCATENATE

```
LOAD *
RESIDENT BKG;
DROP TABLE BKG;
```

L’istruzione **CONCATENATE** impone la concatenazione con la tabella subito sopra che in questo caso è [VENDITE], ma questo può essere fatto solo se le due tabelle hanno gli stessi campi, ecco perché in [BKG] vengono inseriti gli stessi campi ma forzati a zero.

L’istruzione **RESIDENT** è equivalente alla **from** ma si usa quando si fa riferimento ad una tabella già creata precedentemente. Inoltre, poiché la tabella [BKG] ha gli stessi campi di [VENDITE], essa deve essere eliminata per evitare una chiave sintetica con tutti i campi. Questo viene fatto con il comando *drop table*.

3.5.3 CONTROLLO DELLA MAPPATURA

Oltre ai dati di vendita che provengono dalle elaborazioni dell’applicazione *Analisi Vendita* per eseguire il calcolo delle Provvigioni devono essere completamente popolate rispetto ai dati contenuti nelle vendite alcune tabelle di mappatura necessarie all’applicazione. È frequente l’utilizzo di file di mapping in Excel.

In questo progetto tra le tante mappature, ci sono quelle che sono state create per poter definire dei Venditori esclusi dal calcolo, oppure dividendoli in gruppi definiti da team o ancora assegnargli il tipo (Dipendente, Agente) che peserà sul calcolo (vedi *Figura 3.9*).

	A	B	C	D
1	VENDITORE	TEAM_VENDITA	TIPO_VENDITOR	SUB_BU_VENDITORE
2	Venditore 1	Capovendita Roma	Venditore	NUOVO
3	Venditore 2	Capovendita Milano	Venditore	NUOVO
4	Venditore 3	Capovendita Venezia	Venditore	NUOVO
5	Venditore 4	Capovendita Padova	Venditore	NUOVO
6	Venditore 5	Capovendita Verona	Venditore	NUOVO
7	Venditore 6	Capovendita Napoli	Venditore	NUOVO
8	Venditore 7	Team dei Venditori	Venditore	USATO
9	Venditore 8	Team di Milano	Capovendita	NUOVO
10	Venditore 9	Team di Roma	Venditore	NUOVO
11	Venditore 10	Team di Napoli	Venditore	USATO
12	Venditore 11	Team di Padova	Venditore	NUOVO
13	Venditore 12	Capovendita Verona	Venditore	NUOVO

Figura 3. 9 Esempio di mappatura in Excel

Dal punto di vista dello script, per fare un Mapping bisogna prima creare una tabella (con il comando *mapping load*) che abbia due campi, uno che sarà di collegamento con

la tabella che vogliamo ampliare e un altro che andrà ad assegnare il valore in caso di collegamento, quindi si usa il comando *ApplyMap*.

In particolare nella sezione Mapping sono state create tutte le tabelle Mapping; ad esempio per vedere se il Venditore (preso dal file delle *Vendite*) è considerato nel calcolo Provvigionale sono state create le seguenti tabelle Mapping:

```
MAP_ESCLUSO:
MAPPING LOAD DISTINCT
    VENDITORE,
    'ESCLUSO' AS ESCLUSO
FROM [lib://MAPPING/CALCOLO_PROVVIGIONI/CALCOLO_PROVVIGIONI_MAPPING.xlsx]
(ooxml, embedded labels, table is Esclusi);

[VENDITORE_TEAM]:
LOAD
    [VENDITORE] AS [VENDITORE_LK],
    [VENDITORE] AS VENDITORE_TEAM,
    [TEAM_VENDITA],
    [TIPO_VENDITORE]
FROM [lib://MAPPING/CALCOLO_PROVVIGIONI/CALCOLO_PROVVIGIONI_MAPPING.xlsx]
(ooxml, embedded labels, table is Team);

MAP_TEAM:
MAPPING LOAD DISTINCT
    VENDITORE_TEAM,
    'PERIMETRO'

RESIDENT VENDITORE_TEAM;
```

E, quando si caricano i dati nella sezione di lettura delle vendite, sono presenti questi calcoli:

```
APPLYMAP('MAP_ESCLUSO',VENDITORE_LK,
APPLYMAP('MAP_TEAM', VENDITORE_LK,
'TO BE MAPPED')) AS PERIMETRO,
```

In questo modo ad ogni record corrispondente ad una vendita viene aggiunto un campo 'PERIMETRO' che assume il valore *ESCLUSO* se il venditore associato a quella vendita è presente nel foglio di Mapping 'Esclusi', altrimenti *PERIMETRO* se è presente nel foglio di Mapping 'Team', altrimenti *TO BE MAPPED* perché non era presente in nessun Mapping e quindi non rientra in nessuna tipologia di calcolo ma quel Venditore ha fatto delle Vendite.

Inoltre il secondo step verifica che non vi siano mappature incomplete tra quelle strettamente necessarie per il calcolo ed alimenta il foglio 'Check' (vedi *Figura 3.10*) che, oltre a mostrare un indicatore semaforico per ciascuno mapping critico, può visualizzarne il dettaglio evidenziando i record mancanti. Nel caso di mappature

mancanti il calcolo non verrà eseguito fino a quando i file non verranno aggiornati e l'applicazione ricaricata.

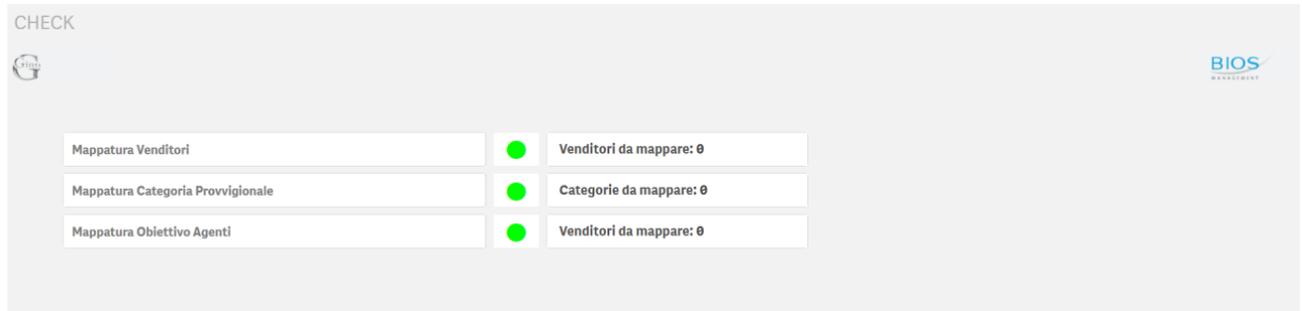


Figura 3. 10 Pagina di check sulla Mappatura dell'applicazione del Calcolo Provvigioni

Nell'esempio precedente è necessario informare l'utente se qualche venditore non è stato mappato perché potrebbe essere frutto di una dimenticanza.

Nel caricamento dati c'è una sezione apposta, "Check Completamento Calcolo" che permette di fare tale controllo. Tenendo presente sempre lo stesso esempio, si procede come segue:

```

CK_V:
LOAD
    distinct VENDITORE_LK
RESIDENT VENDITE
WHERE LEFT(PERIMETRO,12)='TO BE MAPPED'

CK_V_TOT:
load count( distinct CK_V.VENDITORE_LK) as CK_V
RESIDENT CK_V;

LET v_ck_v=IF(NoOfRows('CK_V')>0,peek('CK_V'),0);
    
```

Cioè vengono create due tabelle, in *CK_V* sono presenti i Venditori presi dalla tabella Vendite che hanno come parametro 'TO BE MAPPED', mentre nella tabella *CK_V_TOT* vengono contati quanti Venditori sono presenti nella tabella *CK_V*; viene quindi creata una variabile che mi dice il numero di Venditori non mappati. In particolare con la funzione *peek()* individua il valore del campo, in questo caso il numero di venditori da mappare; tuttavia se non ce ne sono la tabella [*CK_V*] non viene creata e quindi non permetterebbe di contare i venditori, per risolvere questo problema bisogna utilizzare un if in cui si testa prima se esistono righe nella tabella [*CK_V*] tramite la funzione *NoOfRows()* e in caso negativo si mette zero.

La variabile creata viene utilizzata anche in presentation per decidere di che colore mostrare il semaforo (se la variabile è maggiore di 0 il semaforo sarà rosso). Infine dopo aver fatto gli stessi controlli sulle altre mappature necessarie viene creata

un'ulteriore variabile, `v_check_err`, che assume il valore 1 se almeno una delle mappature ha bisogno di aggiornamenti altrimenti 0; questa variabile serve per bloccare il calcolo delle Provvigioni nel caso in cui una mappatura non sia completa. Il codice per fare questo è il seguente:

```
LET v_check_err= if($(v_ck_cp)>0 or $(v_ck_v)>0 or $(v_ck_a)>0,1,0);

if '$(v_check_err)'='1' then

trace mapping error!! ;
exit script;

end if
```

Nel caso in cui la variabile assume il valore 1 allora il caricamento dati viene interrotto senza eseguire le sezioni successive tramite il comando `exit script`.

3.5.4 CALCOLO SUI SINGOLI STOCK_ID VENDUTI

Dopo aver verificato che i Mapping coprano completamente le casistiche presenti nei dati di vendita del periodo selezionato viene eseguito un primo livello di calcolo che agisce su ogni singolo record di vendita.

Sui singoli record vengono aggiunti i seguenti altri campi calcolati o derivati dal mapping che verranno poi utilizzati durante le routine di calcolo delle provvigioni:

- `K_anzianita`: è un coefficiente che viene assegnato ad ogni vendita in base ai giorni in stock e alla marca. Nel dettaglio viene creata una tabella [K_ANZIANITA] dal foglio Coefficiente anzianità del Mapping in questo modo:

`K_ANZIANITA`:

```
Crosstable (KA_CLASSE_ANZIANITA, KA_ANZIANITA, 1)

LOAD Marca AS KA_MARCA,
      C1,
      C2,
      C3

FROM [lib://MAPPING/CALCOLO_PROVVIGIONI/CALCOLO_PROVVIGIONI_MAPPING.xlsx]
(ooxml, embedded labels, table is Coefficiente_anzianità);
```

Il comando `crosstable` consente di trasformare una tabella incrociata in una tabella lineare (vedi *Figura 3.11*). Per collegare questo campo alla vendita si usa una left join tra la tabella [VENDITE] e questa, in cui i campi di collegamento sono la CLASSE_ANZIANITA e la Marca:

LEFT JOIN

```
LOAD KA_MARCA as Marca,  
     KA_CLASSE_ANZIANITA AS CLASSE_ANZIANITA,  
     KA_ANZIANITA AS K_ANZIANITA  
RESIDENT K_ANZIANITA;
```

Si noti che nella tabella [K_ANZIANITA] la classe e la marca hanno nomi diversi per non creare chiavi sintetiche ma per poter andare in join bisogna avere lo stesso nome, quindi vengono rinominato usando il comando AS.

In questo modo ad ogni vendita è associato un valore di anzianità.

Marca	C1	C2	C3
Smart	1.1	1.2	1.3
Mercedes-Benz Vetture	1.1	1.5	1.6
Mercedes-Benz Veicoli Commerciali			
BMW	1	1	1
Aston Martin	1	1	1
Lotus	1	1	1
MINI	1	1	1
BMW i	1	1	1
Maserati	1	1	1
Volvo	1	1	1

K_ANZIANITA

KA_MARCA	KA_CLASSE_ANZIANITA	KA_ANZIANITA
Smart	C1	
Smart	C2	
Smart	C3	
Mercedes-Benz Vetture	C1	
Mercedes-Benz Vetture	C2	
Mercedes-Benz Vetture	C3	
Mercedes-Benz Veicoli Commerciali	C1	
Mercedes-Benz Veicoli Commerciali	C2	

Figura 3. 11 Tabella incrociata (sopra) presente nell' excel di Mapping. Tabella lineare (sotto) ottenuta con il comando *cross table* che collega la marca e la classe con un indice anzianità

- CATEGORIA_PROVVIGIONALE: in base a Marca e MacroModello ad ogni record viene associata nello stesso modo descritto sopra mediante mapping una categoria provvigionale che potrà assumere i valori 'a', 'b', 'l'.
- DIPENDENTE_AGENTE: è un campo creato per distinguere il calcolo Provvigionale per Dipendenti e Agenti. L'utente ha creato quindi uno *foglio* del file xls di Mapping chiamato "Dipendente" in cui sono presenti tutte le informazioni necessarie per i dipendenti. Per prima cosa vengono aggiunte le seguenti informazioni con una *left join* tra [VENDITE] e

```

LOAD DISTINCT
  VENDITORE AS VENDITORE_LK,
  1 AS DIPENDENTE_F,
  RAL,
  FASCIA AS FASCIA_RAL

FROM [lib://MAPPING/CALCOLO_PROVVIGIONI/CALCOLO_PROVVIGIONI_MAPPING.xlsx]
(ooxml, embedded labels, table is Dipendente);

```

Il campo DIPENDENTE_F, che assume il valore 1 solo per i venditori che sono presenti in quel foglio di Mapping, serve per definire il campo DIPENDENTE_AGENTE come segue:

```

IF(DIPENDENTE_F=1, 'DIPENDENTE', 'AGENTE') AS DIPENDENTE_AGENTE,

```

Inoltre nello stesso foglio di Mapping sono presenti anche le informazioni relative ai gettoni (valore e percentuale riconosciuti al dipendente nel calcolo delle provvigioni) e quindi tramite la *left join* vengono aggiunti i seguenti campi:

```

LOAD
  FASCIA_1 as FASCIA_RAL,
  CATEGORIA_PROVVIGIONALE,
  GETTONE_DIPENDENTE,
  GETTONE_PERCENTUALE_DIPENDENTE
FROM [lib://MAPPING/CALCOLO_PROVVIGIONI/CALCOLO_PROVVIGIONI_MAPPING.xlsx]
(ooxml, embedded labels, table is Dipendente);

```

che si collegano al venditore grazie alla fascia *ral*, che raggruppa i dipendenti in base al reddito, aggiunta nel passo precedente.

Infine nello stesso modo vengono aggiunti il campo OBIETTIVO_PEZZI che servirà nel calcolo successivo per singolo venditore.

Nel caso invece dell'agente, i gettoni vengono definiti in base alla marca e categoria provvigionale della vettura tramite il foglio di Mapping "Categoria_provv" come visto prima.

Oltre ai campi riportati in precedenza e derivanti completamente da Mapping xls gestiti dall'utente, l'applicazione utilizza una serie di parametri che l'utente può modificare direttamente sul front end Sense dell'applicazione (vedi *Figura 3.12*). I parametri inseriti nelle caselle riempiono delle variabili che vengono poi usate nei vari calcoli.

livello 1	Provvigione ferro		Provvigione F&I		
GETTONE COMMERCIANTE AGENTE (€)	GETTONE COMMERCIANTE (€)	LIMITE LOW F&I PER PEZZO	LIMITE HIGH F&I PER PEZZO		
SOGLIA LUXURY (€)	% PROV FERRO LUXURY	% LOW F&I PER PEZZO AGENTE	% MID F&I PER PEZZO AGENTE	% HIGH F&I PER PEZZO AGENTE	
USATO % VALORE FATTURA DIPENDENTE	USATO/KM0 % VALORE FATTURA AGENTE	% LOW F&I PER PEZZO DIP	% MID F&I PER PEZZO DIP	% HIGH F&I PER PEZZO DIP	
Bonus usato			Value4u		
GG STOCK BONUS USATO		% PROV F&I VALUE4U DIPENDENTE	% PROV F&I VALUE4U AGENTE		
BONUS USATO > 180 GG DIPENDENTE	BONUS USATO > 180 GG AGENTE	Last reload: 15/02/2021 12:35:06			Reload

Figura 3. 12 Pagina dei parametri dell'applicazione Calcolo Provvigioni

Dopo aver descritto tutti i campi e parametri necessari si può procedere nella definizione del calcolo Provvigioni del Ferro. Nella tabella [VENDITE] viene quindi prima aggiunto un campo che mi calcola la percentuale o il valore di provvigione sulla vendita:

```
LOAD *,
  IF(DIPENDENTE_AGENTE='AGENTE',
  IF(K_ANZIANITA=1,
  if(ORDER_STOCK='a stock',GETTONE_STOCK, GETTONE_VALORE),
  if(ORDER_STOCK='a stock', K_ANZIANITA*GETTONE_VALORE,1*GETTONE_VALORE)),
  GETTONE_DIPENDENTE) AS PROV_V FERRO_GETTONE,
  IF(DIPENDENTE_AGENTE='AGENTE',
  [Prezzo di Vendita con IVA]*GETTONE_PERCENTUALE/100,
  [Prezzo di Vendita con IVA]*GETTONE_PERCENTUALE_DIPENDENTE/100) AS PROV_V FERRO_GETTONE_PERC,
```

e poi la provvigione Ferro sulla singola vendita in questo modo:

```
//PROVVIGIONE FERRO
IF(PERIMETRO='ESCLUSO', 0,
//agente,
IF(DIPENDENTE_AGENTE='AGENTE',
if(DRIVER_CANALE_VENDITA='COMMERCIANTE', $(v_provv_ferro_commerciante),
if(SUB_BU='USATO' OR ([KM0 SI NO]='SI' /*AND Km<=200*/),
[Prezzo di Vendita con IVA]/100* $(v_provv_usato_base_perc),

IF(TIPO_CALCULO_MARCA='MARGINE',[Margine 3]* $(v_perc_prov_veic_comm)/100,
IF(ISNULL(PROVV_FERRO_GETTONE),0,PROVV_FERRO_GETTONE)
+ IF(ISNULL(PROVV_FERRO_GETTONE_PERC),0,PROVV_FERRO_GETTONE_PERC))),
```



```

P:
load *,
    P_PROVV_FERRO + P_PROVV_FINANZIAMENTI + P_PROVV_ASSICURAZIONI + P_BONUS_USATO_OLD as P_PROVVIGIONI_DA_VENDITE,
    P_BASE_PROVV_FINANZIAMENTI + P_BASE_PROVV_ASSICURAZIONI AS P_BASE_FI ,
    P_PROVV_FINANZIAMENTI + P_PROVV_ASSICURAZIONI AS P_FI ;
load VENDITORE_LK,
    Count(Stock) as P_VENDITE_NUM,
    Sum([Prezzo di Vendita con IVA]) AS P_PREZZO_VENDITA_FATTURA_IVATO,
    Sum(PROVV_FERRO) AS P_PROVV_FERRO,
    Sum(PROVV_FINANZIAMENTI) as P_PROVV_FINANZIAMENTI,
    Sum([Provv Fin]) AS P_BASE_PROVV_FINANZIAMENTI,
    sum([Provv Pol]) AS P_BASE_PROVV_ASSICURAZIONI,
    Sum(PROVV_ASSICURAZIONI) AS P_PROVV_ASSICURAZIONI,
    Sum(BONUS_USATO_OLD) AS P_BONUS_USATO_OLD

RESIDENT VENDITE
GROUP BY VENDITORE_LK;

```

La chiave di questa tabella non è più lo stock della vendita ma è il Venditore, infatti ogni record rappresenta un venditore e i campi numerici delle Vendite vengono quindi raggruppati per Venditore e sommati.

Inoltre l'informazione del Team del Venditore viene fornita dalle mappature come visto nel *paragrafo 3.5.3*. Per vedere se il Venditore può accedere al bonus di Squadra c'è bisogno dell'informazione sul numero di Venditori nel Team perché permette di individuare il limite di accesso per un incremento nella Provvigione.

È interessante, dal punto di vista tecnico, osservare come si riesce ad aggiungere tale informazione sulla tabella [P], che ha un livello di aggregazione minore, attraverso il comando left join che crea un collegamento con sé stessa ma raggruppata per Team:

```

left join
load
    distinct P_TEAM_VENDITA,
    COUNT (DISTINCT VENDITORE_LK) AS NUM_VENDITORI_PER_TEAM_VENDITA
RESIDENT P
WHERE P_OBIETTIVO_PROVV_POL_SQUADRA_F=1 AND P_PERIMETRO='PERIMETRO' AND P_TIPO_RECORD='STD'
GROUP BY P_TEAM_VENDITA;

```

In questo modo ad ogni Venditore viene aggiunta l'informazione di quanti Venditori ci sono nel Team, escludendo quelli che non sono considerati nelle Provvigioni o sono Venditori che hanno ricevuto solo provvigioni sul ritiro permuta. Inoltre viene fatta un'ulteriore restrizione dovuta al periodo di validità delle Provvigioni, infatti per non penalizzare il Team, il cliente ha proposto che il periodo di validità rispettasse le seguenti regole:

```

LEFT JOIN
LOAD DISTINCT VENDITORE_LK,
    IF( DATA_FINE_VALIDITA < '$(v_vendite_period_d)+14', 0,
        IF(DATA_INIZIO_VALIDITA >= '$(v_vendite_period_d)', 0,
            IF(DATA_INIZIO_VALIDITA >= '$(v_vendite_period_prev_month_d)',
                IF(LEFT(DATA_INIZIO_VALIDITA,2) <= 15, 1, 0), 1))) AS OBIETTIVO_PROVV_POL_SQUADRA_F
RESIDENT VENDITE_1;

```

Cioè viene contato il Venditore che ha una fine validità superiore al mese corrente e che è stato assunto da almeno un mese. Per quest'ultima condizione bisogna prendere il giorno della data con il comando *left* che mi dà i primi due caratteri del campo 'DATA_INIZIO_VALIDITA' partendo da sinistra. In realtà questo controllo viene fatto già nel calcolo di primo livello per evitare di pagare la provvigione a coloro che hanno una data di fine validità inferiore al mese in corso, nella tabella [P] il campo 'OBIETTIVO_PROVV_POL_SQUADRA_F' viene rinominato con la lettera P davanti per evitare chiavi sintetiche.

Dal punto di vista strutturale il calcolo sul singolo venditore è molto simile al calcolo precedente e alla fine viene creata la seguente tabella [PROVVIGIONI]:

```

PROVVIGIONI:
LOAD VENDITORE_LK AS VENDITORE,
    VENDITORE_LK,
    P_VENDITE_NUM AS NUM_VENDITE,
    IF(P_OB_Pezzi_F=0, 'NO', 'SI') AS OBIETTIVO_VOLUME_VETTURE,
    P_PROVVIGIONI_FERRO_PRE_GAD AS PROVVIGIONE_FERRO_CALC,
    0 AS GAD_FERRO ,
    0 AS PROVVIGIONE_FERRO_TOT ,
    IF(P_OB_MARG_F=0, 'NO', 'SI') AS [OBIETTIVO_F&I],
    P_PROVV_FINANZIAMENTI_CALC AS PROVV_FINANZIAMENTI_CALC,
    P_PROVV_ASSICURAZIONI_CALC AS PROVV_ASSICURAZIONI_CALC,
    P_PROVVIGIONI_FI_PRE_GAD AS [PROVVIGIONE_F&I_CALC],
    0 AS GAD_FI,
    0 AS PROVVIGIONE_FI_TOT,
    P_BONUS_USATO AS BONUS_USATO
RESIDENT P1;

```

che prende solo i campi della tabella [P] relativi ai calcoli finali e a cui vengono aggiunti i campi di GAD, al momento settati a zero perché non ancora inseriti dall'utente, che serviranno per il passo successivo.

3.5.6 GESTIONE GAD DISCREZIONALE

Il cliente ha richiesto di poter eseguire delle possibili modifiche manuali alle Provvigioni maturate con il calcolo eseguito con Qlik. Poiché sulla piattaforma Qlik non è possibile a standard modificare le tabelle presenti in Presentation, la soluzione è stata di inserire un pulsante nella pagina di front-end "Calcolo Provvigioni Finale" (vedi *Figura 3.14*) che quando schiacciato apra un file xlsx che riproduca la stessa tabella di Qlik ma dove l'utente può scrivere le modifiche da apportare nelle colonne apposite (vedi *Figura 3.15*), in particolare ha la possibilità di aggiungere un valore che sarà sommato al valore predefinito oppure può inserire direttamente il valore che sostituirà quello di prima.

Calcolo Provvigioni Finale

DEF: utilizzo dati vendite FM 202011, GAD inserito

VENDITORE

Download file xlsx

VENDITORE	Q	Provvigione Ferro pre GAD	GAD Ferro compilato	Prov Ferro TOT	Provvigione Ferro Totale	Provvigione F&I pre GAD	GAD F&I compilato	Prov F&I Tot Compilato	Provvigione F&I Totale	Bonus Usato	Provvigione Totale
Totale											
Venditore 1											
Venditore 2											
Venditore 3											
Venditore 4											
Venditore 5											
Venditore 6											
Venditore 7											
Venditore 8											
Venditore 9											
Venditore 10											
Venditore 11											
Venditore 12											
Venditore 13											
Venditore 14											
Venditore 15											
Venditore 16											
Venditore 17											
Venditore 18											
Venditore 19											

Figura 3. 14 Foglio Calcolo Provvigioni Finale

VENDITORE	NUM_VENDITE	OBIETTIVO_VOLUME_VETTURE	PROVVIGIONE_FERRO_CALC	GAD_FERRO	PROVVIGIONE_FERRO_TOT	OBIETTIVO_F&I	PROVV_FINANZIAMENTI_CALC	PROVV_ASSICURAZIONI
Venditore 1	17	SI		50	0	SI		
Venditore 2	16	SI		0	0	SI		
Venditore 3	21	SI		0	0	SI		
Venditore 4	15	SI		0	0	SI		
Venditore 5	3	SI		0	0	NO		
Venditore 6	11	NO		0	0	NO		
Venditore 7	9	SI		0	0	NO		
Venditore 8	2	SI		0	0	NO		
Venditore 9	3	SI		0	0	NO		
Venditore 10	17	SI		0	0	NO		
Venditore 11	8	NO		0	0	NO		
Venditore 12	5	SI		0	0	NO		
Venditore 13	3	SI		0	0	NO		
Venditore 14	38	SI		0	0	NO		

Figura 3. 15 Foglio excel per inserire il GAD

Salvando poi il file e ricaricando i dati, essi saranno aggiornati e visibili in presentation.

La procedura che è stata applicata è la seguente. Nel caricamento dati, dopo aver creato la tabella [PROVVIGIONI] viene usato il comando

```
STORE PROVVIGIONI INTO [lib://OUTPUT/CALCOLO_PROVVIGIONI/CALCOLO_PROVVIGIONI.CSV](txt);
```

che crea un file .csv nella cartella OUTPUT del server del cliente. Quindi al pulsante in presentation viene data l'azione di aprire un file xls presente in una cartella nativa di Qlik che però punta al .csv creato da script. Il file si aprirà soltanto se l'account dell'utente è presente nella lista di permessi di accesso ad esso. Quindi, dopo aver effettuato le modifiche dovute, il file modificato deve essere salvato nella cartella INPUT, la collocazione del file è molto importante perché nello script vi è una sezione 'read GAD' in cui viene fatto un controllo che un file sia presente nella cartella:

```
LET v_Performance_file_exists =
isnull(FileTime(['lib://INPUT/CALCOLO_PROVVIGIONI/Provvigioni_']
& '$(v_period).' & 'xlsx'))); // if file exists then 0 else -1

if $(v_Performance_file_exists)<>0 then

EXIT Script;
end if;
```

e se il file non è presente allora non eseguirà i comandi successivi. Infatti la funzione *FileTime* mi restituisce la data e ora di ultima modifica del file scritto nelle parentesi (quindi il file delle Provvigioni modificate) ma questa può essere nulla nel caso non ci fosse il file poiché non c'è stato bisogno di generare un file di modifiche, quindi utilizzando il comando *isnull* restituirà 0 se la funzione precedente ho restituito un valore oppure -1 nel caso in cui avesse restituito *null* che andrà inserito nella variabile "v_Performance_file_exists". Il comando successivo è una condizione di *if* dove si va a testare se la variabile appena creata sia diversa da 0, in caso affermativo il caricamento dati si interromperà in quel punto poiché c'è il comando *EXIT Script*.

Nel caso in cui sia presente il file invece verranno aggiunte le colonne modificate alla tabella [PROVVIGIONI], leggendo tale file e utilizzando la *left join*:

```

left join(PROVVIGIONI)

LOAD
    VENDITORE,
//    NUM_VENDITE,
//    OBIETTIVO_VOLUME_VETTURE,
//    PROVVIGIONE_FERRO_CALC,
    GAD_FERRO AS GAD_FERRO_COMPILATO,
    PROVVIGIONE_FERRO_TOT AS PROVVIGIONE_FERRO_TOT_COMPILATO,
//    "OBIETTIVO_F&I",
//    PROVV_FINANZIAMENTI_CALC,
//    PROVV_ASSICURAZIONI_CALC,
//    "PROVVIGIONE_F&I_CALC",
    GAD_FI AS GAD_FI_COMPILATO,
    PROVVIGIONE_FI_TOT AS PROVVIGIONE_FI_TOT_COMPILATO
//    BONUS_USATO_OLD
FROM [lib://INPUT/CALCOLO_PROVVIGIONI/Provvigioni_$(v_period).xlsx]
(ooxml, embedded labels, table is Provvigioni);

```

Si noti che i campi da aggiungere vengono rinominati usando il comando **AS**, questo perché come chiave di join si deve avere solo il venditore, infatti ad esempio se si utilizzasse il nome uguale per tutti i campi allora nel caso in cui fosse stato inserito un valore al GAD_FERRO, quel venditore non andrebbe in join poiché nella tabella [PROVVIGIONI] il GAD_FERRO è zero mentre nel file di excel ha un valore diverso quindi avrebbe una chiave differente.

A questo punto verranno aggiornati i calcoli di provvigione considerando questi nuovi campi. Poiché l'utente ha due possibilità di modifica nel calcolo si deve testare in quale dei due casi si è:

```

PROVVIGIONI_1:
load *,
if(GAD_FERRO_COMPILATO>0,PROVVIGIONE_FERRO_CALC+GAD_FERRO_COMPILATO,
  if(PROVVIGIONE_FERRO_TOT_COMPILATO>0,
    PROVVIGIONE_FERRO_TOT_COMPILATO,PROVVIGIONE_FERRO_CALC)) as PROVVIGIONE_FERRO_FINALE,

if(GAD_FI_COMPILATO>0,[PROVVIGIONE_F&I_CALC]+GAD_FI_COMPILATO,
  if(PROVVIGIONE_FI_TOT_COMPILATO>0,
    PROVVIGIONE_FI_TOT_COMPILATO,[PROVVIGIONE_F&I_CALC])) as PROVVIGIONE_FI_FINALE

resident PROVVIGIONI;

DROP TABLE PROVVIGIONI;

RENAME TABLE PROVVIGIONI_1 TO PROVVIGIONI;

```

Si osservi che nel fare questa operazione viene creata una nuova tabella chiamata [PROVVIGIONI_1] che prende i campi dalla tabella precedente in cui ora sono stati aggiunti quelli modificati, usando il comando *resident*, e nel passo successivo viene eliminata la tabella [PROVVIGIONI] poiché, dato che non ha tutti i campi uguali a questa nuova tabella non si sarebbero concatenate ma si sarebbe creata una chiave sintetica con tutti i campi in comune che bisogna evitare come è stato descritto nel paragrafo 2.3. Infine per rendere più pulito lo schema dati viene rinominata la tabella rimasta come [PROVVIGIONI] attraverso il comando *RENAME TABLE*.

3.5.6 STORICIZZAZIONE PROVVIGIONI

Se l'analisi viene fatta nella modalità definitiva, dopo aver fatto tutti i passaggi precedenti viene salvata, con il comando *store*, la tabella [PROVVIGIONI] che contiene i dati definitivi in un file qvd chiamato provvigioni_YYYYMM.qvd. Questo ulteriore passaggio viene fatto per poter tenere tutte le informazioni delle provvigioni nei relativi periodi in modo tale da poter realizzare eventualmente un'altra applicazione che possa far vedere la storia delle provvigioni di ogni Venditore.

3.6 PERFORMANCE VENDITORE

La misurazione delle performance del venditore avviene a consuntivo in fase di chiusura mensile e segue la riunione M3 che convalida e ufficializza i valori del venduto del mese precedente. La Performance viene calcolata sulla base del margine del Ferro (cioè sulla

vendita delle vetture) e sulla Provvigione F&I (data dalla somma delle Provvigioni dei finanziamenti e delle assicurazioni vendute).

3.6.1 CARICAMENTO DATI

Come per l'applicazione Calcolo Provvigioni la base dati sono le Vendite e quindi i passaggi sono gli stessi. Inoltre le ulteriori informazioni che non sono presenti nelle vendite sono state recuperate utilizzando i file xls di Mapping, aggiunti dall'amministrazione nella cartella *PERFORMANCE_VENDITORE* e inserite nelle tabelle che si possono vedere evidenziate in *Figura 3.16*.

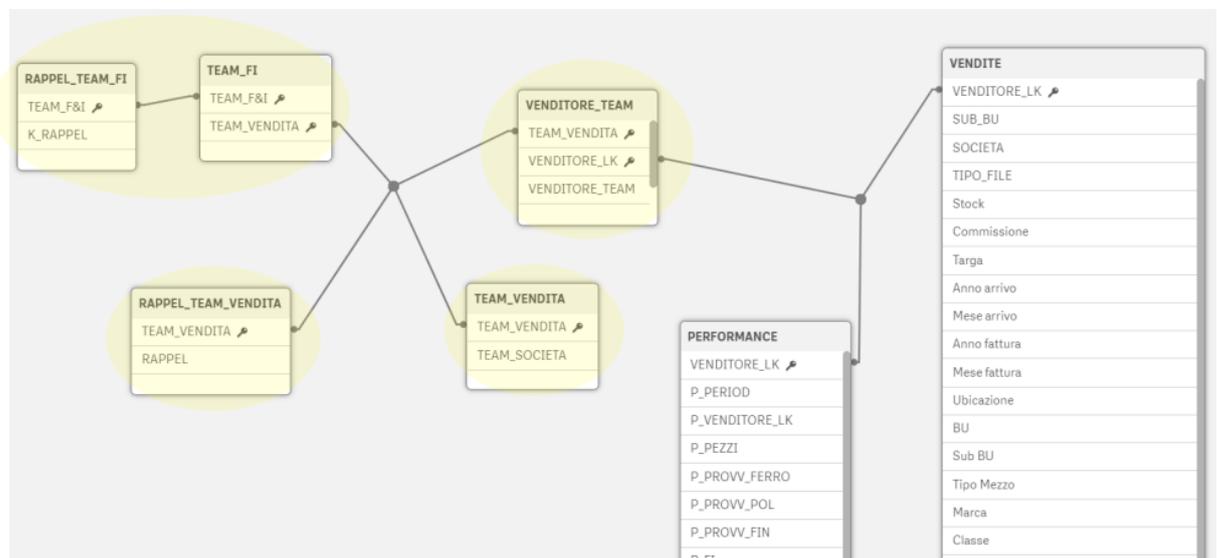


Figura 3. 16 Visualizzazione Tabellare dell'applicazione Perfomance Venditore

In particolare le informazioni necessarie sono le seguenti:

- **TEAM_VENDITA**: ad ogni Venditore è assegnato un team di Vendita.
- **TEAM_SOCIETA**: il Team di Vendita a sua volta è diviso nelle due Società Gino S.p.A. e Gino Auto e quindi, grazie al precedente campo, possiamo ricavare anche la Società per cui lavora il Venditore.
- **TEAM_FI**: è un altro tipo di raggruppamento che può essere assegnato al Team di Vendita.
- **K_RAPPEL/RAPPEL**: il primo definisce un coefficiente per ogni Team F&I mentre il secondo definisce un valore per ogni Team di Vendita. Entrambi serviranno per il calcolo della Performance sull'F&I. Poiché ogni mese l'amministrazione vuole scegliere quale dei due campi considerare, si è pensato di far creare all'utente incaricato un ulteriore file xls di Mapping nella cartella *RAPPEL* in cui l'utente incaricato dovrà inserire i valori di entrambi i campi (nel caso in cui viene

utilizzato il RAPPEL allora K_RAPPEL sarà settato a tutti 1, nel caso contrario il RAPPEL sarà settato a tutti 0; la logica sarà più chiara nei calcoli successivi).

Si crea quindi una tabella nuova [PERFORMANCE] raggruppata per Venditore che servirà come base per fare i calcoli e che contiene alcuni campi significativi della tabella [VENDITE] a cui vengono aggiunti quelli sopra descritti tramite il comando *left join* nello stesso modo in cui è stato fatto per la precedente applicazione. Inoltre vengono ulteriormente aggiunti i seguenti campi calcolati:

- P_PEZZI: rappresenta il numero di vendite eseguite per un determinato Venditore, calcolato contando tutti i record relativi allo stesso venditore.
- P_PROVV_FERRO: è il margine sulle vendite delle Vetture per singolo Venditore calcolata sommando solo il campo [Margine 3] ricavato dalla tabella [VENDITE].
- P_RAPPEL: è la percentuale o il valore da aggiungere all’F&I del Venditore, per calcolarlo si utilizza il seguente codice:
$$P_FI * (P_K_RAPPEL - 1) + P_RAPPEL_TOT_TEAM / P_PEZZI_TOT_TEAM * P_PEZZI$$
 AS P_RAPPEL,
in questo modo se in quel mese si considera il K_RAPPEL allora il P_RAPPEL sarà una percentuale dell’F&I altrimenti è una frazione delle vendite.
- P_FI_CON_RAPPEL: rappresenta la Performance finale di F&I del Venditore in cui viene sommato il suo F&I con il rappel calcolato sopra.
- P_TOT_MARGINE: è la somma dei campi P_PROVV_FERRO e P_FI_CON_RAPPEL, e rappresenta una metrica su cui il venditore viene valutato.

3.6.2 PRESENTATION

Prendendo lo stesso modello di Calcolo Provvigioni è stata realizzata una prima pagina in cui l’utente può scegliere il periodo in cui vedere l’analisi e in quale modalità (DEF o WIP). Inoltre è stato aggiunto un indicatore semaforico con il relativo dettaglio sulla mappatura dei Venditori che non sono presenti in alcuna parte del Mapping ma sono invece presente nelle Vendite (Vedi *Figura 3.17*).

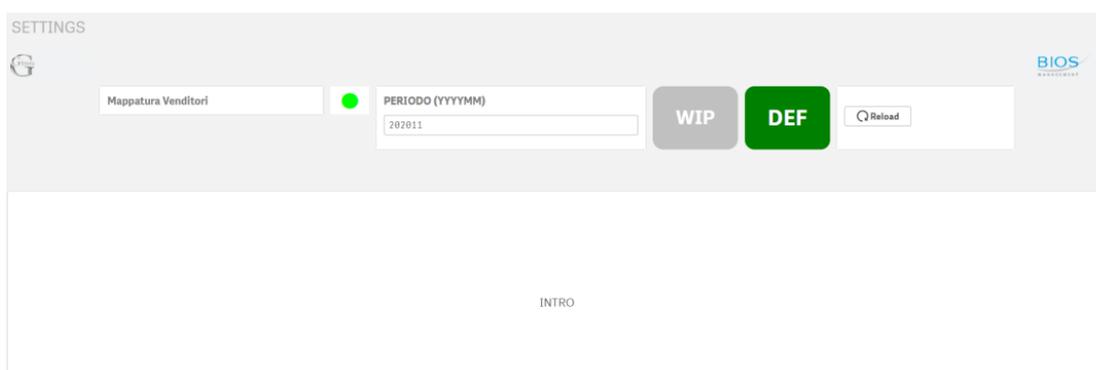


Figura 3. 17 Foglio di Introduzione di Performance Venditore

Per l'analisi il cliente ha richiesto delle Pivot in cui fossero presenti tutti i dettagli calcolati in script nella tabella [PERFORMANCE] per i Venditori suddividendoli per Team F&I, Team di Vendita e Filiale di Vendita (vedi *Figura 3.18*). Quindi la decisione è stata di inserire un'unica pagina in cui fossero presenti le tre tabelle Pivot in un contenitore dove l'utente può selezionare la pivot che vuole vedere.

Performance Team Vendita **Performance Team F&I** Performance Filiale di Vend...

Performance Team F&I

TEAM F&I Q VENDITORE Q Valori

		Pezzi	Prov Ferro	Prov Pol	Prov Fin	F&I	F&I/Pezzo	Rappel	Tot F&I con Rappel	F&I/Pezzo con Rappel	Margine Globale	Margine Globale per Pezzo
Totalli	Totalli	275	20.199	22.602	5.043	60489	219	6110	67219	244	2176,11	7,9
Team 1	Totalli	173	21.310	20.759	8.037	12.400	71	2110	23518	113	1371,01	7,7
	Venditore1	3	163	121	11	11	3,7	11	137	45,7	11,3	3,7
	Venditore2	1	113	213	14	12	12	11	137	126	2110	2,9
	Venditore3	1	111	213	11	12	12	11	137	126	2110	2,9
	Venditore4	1	111	213	11	12	12	11	137	126	2110	2,9
	Venditore5	1	111	213	11	12	12	11	137	126	2110	2,9
	Venditore6	1	111	213	11	12	12	11	137	126	2110	2,9
	Totalli	6	115	116	11	12	12	11	137	126	2110	2,9

Figura 3. 18 Foglio dettagli Calcolo delle Performance

Per rendere l'analisi più interessante è stata richiesta un'ulteriore parte in cui venga assegnato ad ogni Venditore una classe di Performance. Per realizzare questa parte sono stati aggiunti due fogli in cui vengono mostrate le Performance di F&I e di Margine Totale dei Venditori rendendo subito visibile in che classe di Performance fossero usando dei colori per distinguerli. Poiché i passaggi e il risultato è analogo viene descritto solo il foglio "Grafico Performance Venditore F&I/Pezzo Globale" (vedi *Figura 3.19*).

Per definire a che classe ogni venditore appartenga si è pensato di lasciare nel foglio la possibilità all'utente di definire i limiti che caratterizzano le classi in modo da rendere l'analisi il più libera possibile. Assegnando due limiti questi riempiono due variabili che servono nella creazione del nuovo campo 'PERFORMER_CLASS' nel caricamento dati nel seguente modo:

```
if(ROUND(P_FI_CON_RAPPEL/P_Pezzi)>$(v_up_performer_value),'high performer',
  if(ROUND(P_FI_CON_RAPPEL/P_Pezzi)>$(v_low_performer_value),'mid performer',
    'low performer')) AS PERFORMER_CLASS,
```

Questo campo avrà tre possibili valori (*low performer*, *mid performer*, *high performer*) e, lato presentation, viene definito come dimensione principale per poter assegnare un colore in base ai tre possibili valori (rosso per *low performer*, giallo per *mid performer*, verde per *high performer*).

Nel foglio è presente una tabella di dettaglio con dimensione il Venditore e con misure quelle principali di analisi evidenziando lo sfondo della colonna dell'F&I/Pezzo che mi definisce la Performance con i colori stabiliti per la dimensione PERFORMER_CLASS. L'utente può osservare in modo immediato la percentuale delle diverse classi con il grafico a torta a destra. L'analisi doveva essere svolta per le diverse società ed escludendo alcune Filiali; tuttavia si è pensato più efficiente aggiungere dei pulsanti nella

parte superiore del foglio che permettessero la scelta dell'analisi. Questa soluzione ha permesso di rendere più facile la navigazione dei fogli poiché in numero molto minore.

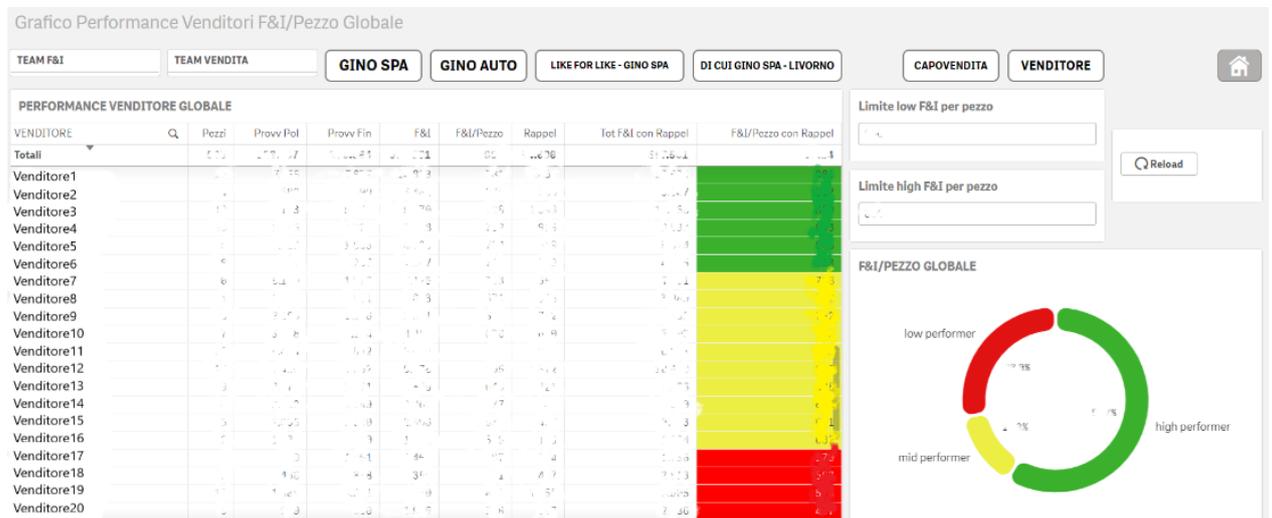


Figura 3. 19 Foglio Grafico Performance Venditore F&I/Pezzo Globale

3.7 TREND PERFORMANCE VENDITORE

L'applicazione *Trend Performance Venditore* è stata richiesta a fine dell'anno scorso per poter analizzare l'andamento delle Performance dei Venditori durante i mesi dell'anno. Infatti l'applicazione *Performance Venditore* può analizzare solo un mese per volta senza permettere di confrontare più periodi. Lato presentation, non ci è stata data una modalità di esposizione dei risultati e quindi è stata proposta un'analisi sul trend mensile medio e progressivo delle principali metriche presenti nell'applicazione *Performance Venditore*.

3.7.1 CARICAMENTO DEI DATI

Nella precedente applicazione è stato inserito il comando *store* per generare un file QVD che contenesse i campi della tabella [PERFORMANCE], salvato con il nome PERFORMANCE_YYYYMM.QVD. Esso viene generato ogni volta che i dati vengono caricati, sovrascrivendo il file se il periodo era già stato caricato. Poiché l'applicazione è stata creata nel mese di novembre gli altri periodi non erano stati storicizzati e quindi la soluzione è stata di settare ogni mese dell'anno e ricaricare i dati, verificando che le

mappature fossero complete, per poter generare i QVD mancanti. A questo punto si è potuto caricare nella nuova applicazione i dati prendendo tutti i QVD relativi alle Performance inserendo al posto del periodo l'asterisco:

LOAD

```
P_TEAM_VENDITA,
"P_TEAM_F&I",
VENDITORE_LK,
P_PERIOD,
P_VENDITORE_LK,
P_Pezzi,
P_PROVV_FERRO,
P_PROVV_POL,
P_PROVV_FIN,
P_F&I,
P_AVG_F&I,
ESCLUSO_F,
P_TIPO_VENDITORE,
P_TEAM_SOCIETA,
P_RAPPEL_TOT_TEAM,
P_K_RAPPEL,
P_Pezzi_TOT_TEAM,
P_RAPPEL_PERC,
P_RAPPEL_VALORE,
P_RAPPEL,
P_F&I_CON_RAPPEL,
P_AVG_F&I_CON_RAPPEL,
PERFORMER_CLASS,
P_TOT_MARGINE,
PERFORMER_CLASS_ORD,
PERFORMER_MARGINE_CLASS
```

```
FROM [lib://QVD_DWH/PERFORMANCE_VENDITORE/PERFORMANCE_*QVD]
(qvd);
```

3.7.2 PRESENTATION

Poiché questa applicazione è un'estensione delle Performance sono state riprodotte le pagine sui dettagli del Calcolo, tuttavia i valori si riferiscono alla somma su tutti i periodi presenti. Nel caso si volesse analizzare un periodo specifico sono presenti i filtri di Anno e Mese da poter inserire (Vedi Figura 3.20).

Pivot Performance Team F&I

SOCIETA TEAM_VENDITA VENDITORE MESE ANNO

Performance Team F&I

TEAM F&I VENDITORE

		Valori										
		Pezzi	Provv Ferro	Provv Pol	Provv Fin	F&I	F&I/Pezzo	Rappel	Tot F&I con Rappel	F&I/Pezzo con Rappel	Margine Globale	Margine Globale per Pezzo
Totali		2.704	4.871.795	2.111.115	1.115.619	8.113.362	655	1.111.2	1.111.115	111	1.111.115	1.111.115
Team1	Totali	2.704	4.871.795	2.111.115	1.115.619	8.113.362	655	1.111.2	1.111.115	111	1.111.115	1.111.115
	Venditore 1	1.111	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115
	Venditore 2	1.111	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115
	Venditore 3	1.111	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115
	Venditore 4	1.111	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115	1.111.115

Figura 3. 20 Foglio dettagli sul calcolo delle Performance

Nella pagina “Trend numero pezzi” sono stati inseriti due grafici lineari e dei filtri utili nell’analisi (vedi *Figura 3.21*).

Per la creazione dei grafici bisogna inserire delle dimensioni e delle misure. La dimensione, che sarà rappresentata sull’asse x, è il mese che è stato calcolato prendendo il campo P_PERIOD (nel formato ‘YYYYMM’) trasformato con la funzione *MONTHNAME* nel formato visibile nella Figura; questo perché Qlik può riconoscere tale dimensione come data e riesce anche a ordinarla in modo esatto oltre ad essere di migliore aspetto. Mentre le misure, che creano le linee del grafico, sono il numero di auto vendute nelle due Società. Per fare questa distinzione bisogna agire sulle misure attraverso l’uso della così detta *set analysis*, cioè nell’espressione delle misure viene aggiunta una parte di codice che mi prenda solo i record desiderati. Ad esempio per la prima bisogna sommare il campo P_PEZZI ma prendendo solo i record che hanno come P_TEAM_SOCIETA ‘GINO SPA’ scrivendo nell’espressione il seguente pezzo di codice dove la parte evidenziata corrisponde alla *set analysis*:

```
Sum({$<P_TEAM_SOCIETA={'GINO SPA'}>} P_PEZZI)
```

La differenza dei due grafici è che nel secondo viene scelta nelle proprietà del grafico l’opzione di *accumulo* per vedere l’andamento progressivo delle due misure.

Questo foglio è stato poi duplicato cambiando solo le misure con la Provvigione del Ferro. Un’altra analisi simile è stata quella sul Trend dell’F&I in cui viene realizzato un foglio dello stesso tipo degli altri ma in cui vengono inserite tre misure: la Provvigione sui finanziamenti, la Provvigione delle Polizze e infine l’F&I che è la somma delle prime due (vedi *Figura 3.22*).



Figura 3. 21 Foglio Trend numero Pezzi



Figura 3. 22 Foglio Trend F&I

Un'ulteriore analisi è stata fatta nel foglio "Trend medio Provv Ferro e F&I" in cui sono presenti due grafici lineari con dimensione sempre il periodo ma con misure il rapporto della somma delle Provvigioni del Ferro o di F&I sulla somma delle vendite. Nel secondo inoltre vengono aggiunte due linee di riferimento che prendono i valori decisi dall'utente e inseriti nelle caselle in basso a sinistra della *Figura 3.23* che rappresentano i limiti per le classi di Performance in modo da vedere nel complesso in quali mesi l'azienda ha avuto più profitti.

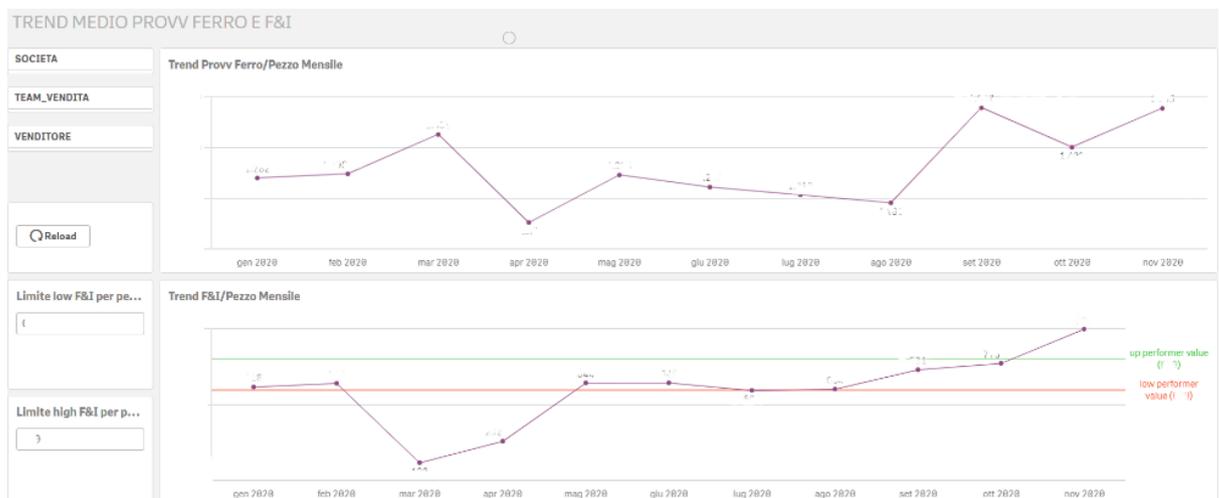


Figura 3. 23 Foglio Trend Medio Provv Ferro ed F&I

Infine nell'ultimo foglio "Trend F&I/Pezzo per Società" è stato scelto un grafico a dispersione che permette di individuare il rapporto che c'è tra F&I e le vetture vendute per ogni Venditore (vedi *Figura 3.24*). La dimensione (rappresentata dalle bolle) è data dal Venditore, mentre sull'asse x e sull'asse y vi sono rispettivamente le misure date

dalla somma delle vendite e dall’F&I. Nelle proprietà è stata anche aggiunta la condizione sul colore, cioè quando quel rapporto è minore del limite inferiore, lasciato a parametro all’utente, la bolla sarà rossa; se maggiore del limite superiore sarà verde altrimenti gialla. In questo modo la lettura del grafico è molto più immediata.



Figura 3. 24 foglio Trend F&I/Pezzo per Società

3.8 SCHEDULAZIONE TASK DI AGGIORNAMENTO E CARTELLE DEI PROGETTI

Per poter pubblicare le apps di Qlik Sense e inserirle in uno stream, è necessario accedere alla Qlik Management Console (qmc). Come è stato accennato nel *Capitolo 2* tramite la qmc è possibile gestire le licenze e le regole di accesso alle apps di tutti gli utenti dell’azienda.

Una volta pubblicate tutte le apps è possibile creare i task di caricamento e dare loro una schedulazione. In *Figura 3.25* è mostrato l’elenco dei task creati, uno per ciascuna app.

Name	Associated resource	Type	Enabled	Status	Last execution	Next execution	Tags
Reload task of ANALISI...	ANALISI VENDITE	Reload	Yes	Success	2021-01-23 07:40	On task event trig...	
Reload task of ETL VEND...	ETL VENDITE	Reload	Yes	Success	2021-01-23 07:40	2021-01-24 07:40	

Figura 3. 25 Task di aggiornamento delle applicazioni Qlik Sense

L'aggiornamento delle apps è giornaliero e inizia con l'app ETL VENDITE alle 7.40 del mattino, al termine dell'esecuzione dello script di ETL VENDITE, parte il task relativo all'app ANALISI VENDITE.

In realtà i task sono stati creati solo per le due applicazioni sulle vendite poiché per le provvigioni e le Performance l'utente può decidere in autonomia quando ricaricare i dati tramite i pulsanti di *reload*.

Per quanto riguarda le cartelle dei progetti si era già accennato lungo l'analisi delle varie applicazioni l'uso di numerosi file inseriti in determinate cartelle. In questo paragrafo si vuole dare maggiore chiarezza sulla posizione di tali file.

Le cartelle utilizzate sono le seguenti:

- FTP: dove sono inseriti tutti i file sorgenti reperiti dal gestionale.
- QVD_DWH: dove vengono inseriti tutti i file storicizzati di tipo qvd creati nelle varie applicazioni; tali file sono contenuti in sottocartelle che prendono il nome dalle applicazioni.
- MAPPING: in questa cartella sono presenti tutti quei file di Mapping che vengono compilati dal cliente per poter creare dei campi che dalla base dati non vengono forniti; in particolare abbiamo le seguenti sottocartelle specifiche per le applicazioni:
 - CALCOLO_PROVVIGIONI che contiene il Mapping CALCOLO_PROVVIGIONI_MAPPING.xlsx con i fogli relativi al tipo di venditore, piuttosto che al Team o agli obiettivi.
 - PERFORMANCE_VENDITORE che contiene il Mapping PERFORMANCE_VENDITORE_MAPPING.xlsx che caratterizza i vari Venditori, e il file RAPPEL_YYYYMM.xlsx che dovrà essere inserito ogni mese con il riferimento al periodo che conterrà i coefficienti per ogni Team che mi permettono di calcolare le Performance.
- INPUT: in questa cartella sono stati inseriti dei file esterni che vengono letti dalle applicazioni per integrare i dati presenti. In particolare abbiamo due sottocartelle, una che contiene un file relativo al recupero delle Provvigioni a fronte di ritiro Permuta (BACKLOG_PERMUTE_BIOS.xlsx) e l'altra che contiene il file con le correzioni discrezionali da apportare al calcolo delle Provvigioni (Provvigioni_YYYYMM.xlsx).
- OUTPUT: questa cartella è stata creata esclusivamente per inserire il file .csv salvato da Qlik descritto nel *paragrafo 3.5.6*.

Conclusioni

Bios Management ha realizzato numerosi progetti con Gruppo Gino, ho deciso di presentare questi in particolare perché sono nati proprio all'inizio del mio tirocinio curriculare.

Il primo progetto su cui ho lavorato è stato Performance Venditore. Il cliente prima di utilizzare la BI realizzava questo tipo di analisi su Excel mostrandoci le numerose tabelline accompagnate da grafici statici. Attraverso Qlik invece il cliente ci ha rivelato che le analisi sono molto più immediate e versatili.

Il progetto delle Provvigioni invece è stato un tema un po' più complesso perché molte informazioni non erano disponibili nelle vendite è quindi c'è stato bisogno di implementarle con numerosi fogli di Mapping che l'utente doveva man mano aggiornare.

Tuttavia il risultato a mio parere è stato soddisfacente perché, nonostante il progetto presentasse una buona parte di calcolo e la parte di analisi fosse molto limitata, ha permesso di rendere tutto il processo di Provvigioni automatico mentre in precedenza molte parti dovevano essere tenute sotto controllo in maniera manuale.

Ad esempio, attraverso Qlik, si è riuscito a realizzare il calcolo delle Provvigioni per ritiro Permuta in modo automatico; così facendo è possibile evitare errori dovuti alla parte manuale di correzioni a cui prima si era costretti. Inoltre il nostro progetto è stato usato anche come strumento per ottimizzare il processo potendo modificare tutti i parametri dalla dashboard e ricaricare in pochi secondi tutti i dati.

In conclusione ritengo che gli utenti dell'azienda che utilizzano questo nuovo strumento siano soddisfatti, infatti hanno limitato di molto le operazioni che dovevano fare manualmente e per cui perdevano molto tempo; le applicazioni contengono un numero modesto di fogli che limita la confusione ma che permette comunque di fare delle analisi mirate facendo selezioni in modo semplice e veloce; inoltre alcuni utenti hanno la possibilità di poter realizzare dei fogli propri ed effettuare analisi in modo autonomo per soddisfare delle richieste particolari in ogni occasione.

Dal punto di vista del consulente invece è affascinante scoprire il mondo che sta dietro ai fogli di presentation, dalle sorgenti da cui si prendono i dati a tutte le operazioni che si devono svolgere per normalizzare i dati affinché diventino puliti e legati tra di loro. Dopo ogni regola applicata bisogna controllare sui report se essa è stata applicata in modo esatto attraverso degli esempi; nel caso delle Provvigioni si sono presi numerosi casi per verificare che tutte le casistiche rispettassero le regole.

Se gli esempi nei dati caricati non sono presenti bisogna trovare il modo di crearseli. Infatti nel controllo sul periodo di validità del Venditore sono state inserite delle date temporanee di inizio validità nel Mapping per verificare che l'algoritmo escludesse quei venditori assunti da meno di un mese.

Lo strumento Qlik inoltre è molto facile da usare e permette di realizzare oggetti personalizzati in modo da soddisfare tutte le idee dei clienti, dal colore dello sfondo delle colonne di una tabella piuttosto che l'ordinamento dei dati o mostrare solo determinati valori.

Ringraziamenti

A conclusione di questo percorso di studi ritengo opportuno fare dei ringraziamenti alle persone che mi hanno accompagnato e guidato in questi anni.

In primo luogo alla mia *famiglia* che mi ha supportato e sostenuto in ogni mia scelta, in particolare a mia *madre* che mi ha trasmesso la passione per la Matematica e mi è sempre stata vicina, a mio *padre* che mi ha dato la possibilità di iniziare questo percorso, e infine a mia sorella *Paola* che mi ha sopportato anche nei momenti in cui ero maggiormente stressata perché in sessione esami.

A *Daniele*, che ha sempre creduto in me anche quando pensavo di non farcela dandomi la forza di continuare per raggiungere i miei obiettivi.

Alle mie amiche, in particolare ad *Alessia* che con la sua determinazione mi ha condizionato a dare il massimo in ogni situazione; ad *Alessandra*, mia compagna di studi fin dal primo giorno, con cui ho condiviso tutti gli esami e ci siamo sostenute a vicenda in ogni momento.

Infine vorrei ringraziare anche il mio Team di lavoro, in particolare *Sandro* e *Simona*, che, con la loro infinita pazienza, mi hanno seguito in questo nuovo percorso dandomi la possibilità fin da subito di lavorare sul campo.

Bibliografia

- [1] A.Rezzani, *Business Intelligence*, Maggioli Editore, 2017
- [2] «History of Business Intelligence» Available: <https://www.betterbuys.com/bi/history-of-business-intelligence/>
- [3] «Business intelligence v.03 » Available: <https://www.slideshare.net/lmauri/business-intelligence-v03>
- [4] «Che cos'è un database relazionale?» Disponibile; <https://www.oracle.com/it/database/what-is-a-relational-database/>
- [5] «Differenza tra Data Warehouse e Database Relazionale in informatica »Available:<https://vitolavecchia.altervista.org/differenza-tra-data-warehouse-e-database-relazionale-in-informatica/>
- [6] «Business Analytics: la BI che guarda avanti »Available:<https://www.zerounoweb.it/techtarget/searchsecurity/business-analytics-la-bi-che-guarda-avanti/>
- [7] <https://www.qlik.com/it-it/company>
- [8] «Dì addio ai limiti SQL. Dai il benvenuto alle intuizioni senza limiti. »Available: <https://www.qlik.com/it-it/products/associative-difference>
- [9] «Esecuzioni di selezioni» Available:https://help.qlik.com/it-IT/cloud-services/Subsystems/Hub/Content/Sense_Hub/Selections/make-selection-2.htm
- [10] «Stati delle Selezioni» Available:https://help.qlik.com/it-IT/cloud-services/Subsystems/Hub/Content/Sense_Hub/Selections/selection-states.htm
- [11] «Utilizzo dei file QVD» Available: https://help.qlik.com/it-IT/cloud-services/Subsystems/Hub/Content/Sense_Hub/Scripting/work-with-QVD-files.htm
- [12] «Chiave sintetiche» Available: https://help.qlik.com/it-IT/sense/September2020/Subsystems/Hub/Content/Sense_Hub/Scripting/synthetic-keys.htm

- [13] «Riferimenti circolari» Available: https://help.qlik.com/it-IT/cloud-services/Subsystems/Hub/Content/Sense_Hub/LoadData/circular-references.htm
- [14] «Ruoli di Qlik Sense» Available: https://help.qlik.com/it-IT/cloud-services/Subsystems/Hub/Content/Sense_Hub/Introduction/SaaS_roles.htm
- [15] «Dimensioni» Available: https://help.qlik.com/it-IT/sense/June2020/Subsystems/Hub/Content/Sense_Hub/Dimensions/dimensions.htm
- [16] «Misure» Available: https://help.qlik.com/it-IT/sense/June2020/Subsystems/Hub/Content/Sense_Hub/Measures/measures.htm
- [17] «Set Analysis ed espressioni set» Available: https://help.qlik.com/it-IT/qlikview/April2020/Subsystems/Client/Content/QV_QlikView/ChartFunctions/SetAnalysis/set-analysis-expressions.htm
- [18] «Snapshot library» Available: https://help.qlik.com/en-US/sense/November2020/Subsystems/Hub/Content/Sense_Hub/StoryTelling/Snapshots/snapshot-library.htm
- [19] «L'azienda – Chi siamo» Available: <https://www.ginospa.com/chi-siamo/>