

Politecnico di Torino

Dipartimento di Ingegneria Gestionale

Tesi di Laurea Magistrale



**Politecnico
di Torino**

**Progettazione e sviluppo di una dashboard per il
monitoraggio operativo di un parco veicolare**

Studente:
Francesco Giurato

Relatore accademico:
Prof. Daniele Apiletti

Relatore aziendale:
Dott.ssa Marzia Didio

Anno Accademico 2023/2024

Abstract

Con il termine Uptime si intende l'intervallo di tempo in cui un sistema, un dispositivo o un'applicazione è operativo e funzionante correttamente. Nel settore automotive il monitoraggio dell'efficienza operativa, misurata attraverso l'Uptime, riveste un ruolo cruciale poiché risulta essere uno strumento fondamentale per garantire la soddisfazione del cliente e lo svolgimento delle principali attività commerciali post-vendita. Il presente lavoro di tesi è il risultato finale di un progetto svolto all'interno di una società di consulenza e il progetto su cui è basato l'elaborato è stato commissionato da un'azienda cliente operante nel settore automotive che verrà identificata come ONHIGH.

VORlog Dossier Project, dove VOR sta per "Vehicle Off Road", è inserito all'interno di VORlog, un insieme di attività presso la società ONHIGH volte a migliorare l'esperienza del cliente nel programma aziendale di qualità e garanzia. La tesi ripercorre le tappe principali relative allo sviluppo di una dashboard in ambiente Qlik Sense; l'obiettivo alla base dell'app è quello di fornire uno strumento in grado di monitorare l'Uptime del parco veicolare e valutare la coerenza dei dati nei singoli dossiers, dei fascicoli digitali all'interno dei quali sono inserite tutte le informazioni dei veicoli che subiscono dei guasti e che vengono aperti proprio in corrispondenza di un fermo. Questo strumento supporterà l'identificazione tempestiva di problemi riducendo i tempi di inattività e migliorando l'efficienza complessiva dell'azienda poiché sarà in grado di fornire delle linee guida per poter intercettare e quindi risolvere eventuali problematiche evidenziate da dati fuori dai parametri prefissati. Il monitoraggio avverrà attraverso particolari KPI come, ad esempio, il numero di dossiers e la differenza tra il Time Of Repair (ovvero la data di rilascio del veicolo in circolazione) e la data di arrivo del veicolo in officina, questa differenza è l'intervallo temporale che identifica la durata di un fermo veicolo. Questi KPI, insieme ad altri,

sono inseriti in diversi fogli di analisi, i quali permetteranno di effettuare analisi con granularità diverse e con focus su diverse dimensioni di analisi.

Oltre all'attività di monitoraggio, l'app permetterà di valutare la consistenza dei dati contenuti nei dossieri grazie all'integrazione di dati esterni come il Digital Job Card, uno strumento utilizzato per mappare l'efficacia degli interventi in officina e il GeoFencing, uno strumento per ottenere la data d'ingresso e di uscita del veicolo dalla specifica officina. La presenza di queste informazioni permetterà di incrociare le diverse fonti per valutare eventuali mancanze e/o incongruenze dei dati contenuti nei dossieri.

Indice

1	Introduzione	10
1.1	Obiettivo del progetto	11
1.2	Panoramica del settore automotive	11
1.2.1	Global Value Chain del settore automotive	16
1.3	Contesto operativo aziendale	17
2	Data Analytics e Business Intelligence	19
2.1	Cenni di basi di dati	19
2.2	Modello dei dati	20
2.3	Progettazione di una base di dati	21
2.4	Data Warehouse	24
2.4.1	Architettura di un Data Warehouse	25
2.5	Business Intelligence	27
2.5.1	Tecnologia QlikSense	31
3	Business Issue	38

3.1	Time-line Progetto di Business Intelligence	40
3.2	Raccolta dei requisiti	41
3.3	Caratteristiche del caso studio	42
3.3.1	Struttura del progetto	42
3.3.2	File VorLog Dossier	43
3.3.3	Dimensioni di analisi	45
3.3.4	Key Performance Indicators (KPI)	47
3.3.5	GANTT CHART	49
3.3.6	Architettura della soluzione proposta	50
4	VORlog Dossier Project	52
4.1	Analisi e sviluppo	53
4.2	Progettazione del Data Model	54
4.3	Integrazione dei dati su Qlik Sense	56
4.3.1	Integrazione dei dati da NEWBI	57
4.3.2	Integrazione dei dati da Azure	61
4.4	Trasformazione dei dati - ETL2	64
4.4.1	ETL2 - Vehicle	64
4.4.2	ETL2 - Dossiers e Components	66
4.4.3	ETL2 - Contratti PMCH	73
4.4.4	ETL2 - Master Calendar	73

4.4.5	ETL2 - GeoFencing e DJC	76
4.5	Presentation	77
4.5.1	Summary	77
4.5.2	Dettaglio Dossiers	79
4.5.3	UpTime	80
4.5.4	Open Dossiers	83
4.5.5	Markets	85
4.5.6	Geofencing	85
4.5.7	Global Monthly	87
4.5.8	Weekly	88
5	Conclusioni e Sviluppi Futuri	89
6	Appendice	91
6.1	Sintassi dello script in Qlik	91
6.2	Set Analysis	96

Elenco delle figure

1.1	Global Car Production [3]	13
1.2	Global Commercial Vehicle Production [3]	14
1.3	Distribution of Global Car Production [3]	15
1.4	Distribution of Global Commercial Vehicle Production [3]	16
1.5	Global Value Chain (GVC) del settore automotive [4]	17
2.1	Esempio di un Modello Relazionale	21
2.2	Step per la progettazione di un Data Base	22
2.3	Dimensional Fact Model	23
2.4	Star schema vs snowflake schema	24
2.5	Cubo OLAP	25
2.6	Architettura di sistema di Business Intelligence	26
2.7	Processi in un progetto di Business Intelligence	27
2.8	Business Intelligence Services	29
2.9	Esempio di una dashboard sviluppata in ambiente Qlik Sense [11]	33

2.10	Schema di processo di Qlik Sense	37
3.1	Time line di un progetto di Business Intelligence	40
3.2	GANTT Chart progetto VORlog Dossier	49
3.3	Architettura dell'app proposta	51
4.1	Data Model sviluppato per la WAVE I	54
4.2	Data Model sviluppato per la WAVE II	55
4.3	Ingestion dati in ambiente Qlik Sense	57
4.4	Struttura del Data Warehouse NEWBI	58
4.5	Naming convention per le tabelle di Staging	60
4.6	Naming convention per tabelle di secondo livello	60
4.7	Naming convention per i campi	61
4.8	Summary Dashboard	78
4.9	KPIs Summary	78
4.10	Run Chart $TR_{\leq 24h}$ e $TR_{\leq 48h}$	79
4.11	Schermata Dettaglio Dossier	80
4.12	Schermata UpTime	81
4.13	KPIs UpTime	81
4.14	Confronto KPI Previous Year	81
4.15	Line chart Parco Circolante	83
4.16	Schermata Open Dossiers	84

4.17 Schermata Open Dossiers By Markets & Customer	84
4.18 Schermata Markets	85
4.19 Schermata Geofencing	86
4.20 Schermata Global Monthly	87
4.21 Schermata Weekly	88
6.1 Sintassi Set Analysis[14]	97

Capitolo 1

Introduzione

Il presente lavoro di tesi è il risultato finale di un progetto svolto all'interno di una società di consulenza che per motivi di privacy verrà chiamata da questo momento in poi KM. Il progetto, commissionato da un'azienda cliente che verrà chiamata ONHIGH, operante nel settore automotive, si pone come obiettivo finale quello di sviluppare una dashboard personalizzata in ambiente Qlik Sense. Questo documento tratterà il percorso che ha portato al risultato finale dal punto di vista gestionale, ponendo l'attenzione quindi a temi come il tempo impiegato per la realizzazione, e dal punto di vista dello sviluppo tecnico andando ad approfondire tutti gli step fondamentali per l'implementazione della dashboard. Inoltre, verrà proposta una breve descrizione della tecnologia utilizzata, Qlik Sense, a partire dai concetti basilari fino ad esplorare alcune delle funzionalità più importanti di tale piattaforma.

Pertanto, in questa prima parte verrà posta l'attenzione sul progetto e quindi sull'obiettivo finale da raggiungere e una panoramica dettagliata del mercato automobilistico attraverso una presentazione chiara e informativa della società committente. Questo approccio mirato fornirà una base solida per comprendere appieno il contesto in cui si inserisce il progetto di sviluppo della dashboard e apprezzare appieno il valore delle soluzioni proposte nel corso di questo lavoro di tesi.

1.1 Obiettivo del progetto

Il progetto VORlog Dossier si colloca all'interno dell'iniziativa più ampia denominata VORlog, focalizzata sull'ottimizzazione dell'affidabilità dei dati provenienti da diverse fonti che alimentano i dossier, tali documenti sono dei fascicoli digitali che raccolgono in modo esaustivo tutte le informazioni relative ai veicoli fermi in officina, come indicato dall'acronimo VOR che denota "Vehicles Off Road". L'obiettivo centrale del progetto è la creazione di una dashboard su Qlik Sense che, in primo luogo, permetta di estrarre indicatori di prestazione e tendenze rilevanti per quanto riguarda l'Uptime, ovvero la disponibilità operativa del parco veicolare in circolazione e, in secondo luogo, che consenta il monitoraggio della coerenza dei dati contenuti in ciascun dossier. In particolare, il secondo obiettivo, focalizzato sul consolidamento dell'accuratezza dei dati nei dossier, sarà conseguito tramite l'inclusione nella dashboard di due dati solitamente assenti nei documenti digitali, ovvero Digital Job Card (DJC) e il GeoFencing. Questo aspetto riveste un ruolo cruciale, poiché i dati attuali aggregati in VORlog non garantiscono con certezza la presenza dei veicoli in officina, a causa di alcune logiche di chiusura dei documenti applicate ai dossier VORlog, attualmente in fase di revisione.

1.2 Panoramica del settore automotive

Con l'espressione "settore automotive" ci si riferisce al settore commerciale dedicato alla progettazione e produzione di tutti i veicoli a motore, cioè automobili, autobus e veicoli commerciali come autocarri e mezzi da rimorchio. Questa definizione, tuttavia, risulta essere appartenente ad un perimetro molto più ampio poiché include anche tutte le aziende produttrici finali, le ditte che si occupano dei ricambi dei componenti, i rivenditori specializzati e le aziende che forniscono servizi di autonoleggio. Nonostante l'industria in questione sia nata in Europa, al giorno d'oggi le aziende leader nel settore si trovano in Cina; secondo i numeri dell'OICA (Organizzazione internazionale dei costruttori auto) il paese asiatico è al comando della

classifica per numero di veicoli prodotti con 27.020.615 veicoli nel 2022, seguito dagli Stati Uniti con 10.060.339 e dal Giappone con 7.835.519. Con i suoi considerevoli investimenti annuali pari a 59 miliardi di euro, il settore automobilistico mantiene il primato come principale investitore in ricerca e sviluppo all'interno dell'Unione Europea, costituendo un terzo della spesa totale della regione. Caratterizzata da un'elevata intensità di capitale e conoscenza, l'industria automobilistica svolge un ruolo chiave nello sviluppo socio-economico del paese. Attualmente, il settore è in fase di espansione, coinvolgendo un numero crescente di nazioni nella produzione di veicoli, mentre le dinamiche di mercato continuano a evolversi. La sua importanza nello sviluppo dell'economia moderna è strettamente correlata alla posizione del trasporto automobilistico all'interno dell'infrastruttura nazionale, determinando così le prospettive di sviluppo del settore. [1]

L'industria automobilistica è un notevole consumatore di risorse quali acciaio, ferro, alluminio, plastica, vetro, moquette, tessuti, chip per computer, gomma, e molti altri materiali. Secondo le statistiche, approssimativamente la metà del consumo globale di petrolio e gomma, circa un quarto della produzione di vetro e un sesto della produzione di acciaio sono imputabili all'industria automobilistica. Questa rappresenta il secondo settore, dopo la costruzione di aeromobili, in termini di volume di materiali consumati da altre industrie. Nei paesi sviluppati, una crescita dell'1% nel settore automobilistico si traduce in un aumento del 1,5% del PIL (Prodotto Interno Lordo). L'impatto indiretto dell'industria automobilistica sul PIL è ulteriormente amplificato attraverso le industrie collegate, che ricevono ordini direttamente dall'industria automobilistica. Quest'ultima è il principale ramo della costruzione di macchine nei paesi sviluppati, e praticamente ogni grande economia ospita un'industria automobilistica significativa. La quota di contributo dell'industria automobilistica al PIL varia dai 5 ai 10 punti percentuali in questi paesi. [2]

Negli ultimi anni il mercato ha dovuto fare i conti con alcuni “eventi estremi” quali pandemie, conflitti bellici a livello mondiale con conseguenti tensioni sui prezzi e crisi dei chip e delle materie prime; questi, sono eventi che si verificano con

una bassa probabilità, ma che possono avere un impatto significativo e a volte catastrofico quando si verificano. Il settore automotive in tal senso ha sofferto molto e le difficoltà globali hanno avuto come ripercussione la diminuzione della produzione (il cui massimo registrato risale al 2017). Nelle figure seguenti [(Fig.1.1) e (Fig.1.2)], sono riportati i trend di variazione della produzione per le automobili (Cars) e veicoli commerciali (commercial Vehicle):

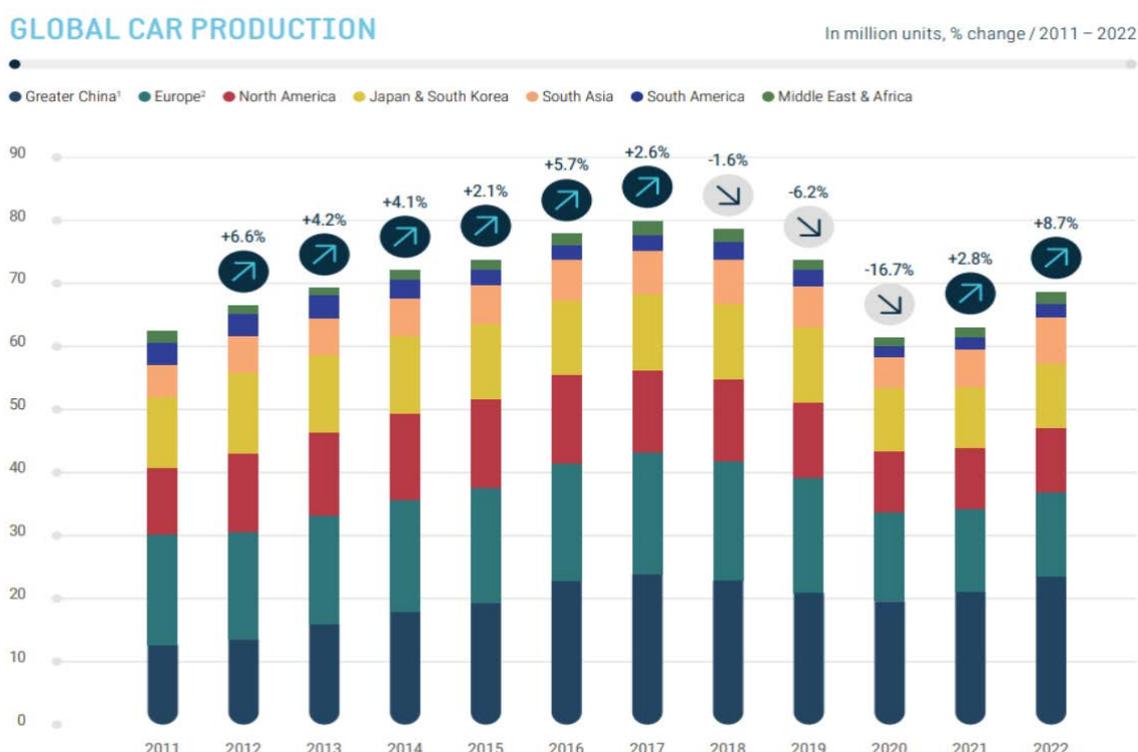


Figura 1.1: Global Car Production [3]

Prima del COVID-19 il settore automobilistico stava vivendo una fase di produzione e vendite crescenti, la domanda globale di veicoli era solida e alimentata da economie in crescita, accessibilità al credito e una crescente classe media in molte parti del mondo. Questo trend positivo è possibile notarlo dal grafico relativo alla variazione per la produzione di veicoli in [(Fig.1.1)]. In particolare, fino al 2019 il trend per quanto riguarda la produzione globale di auto risultava essere quasi sempre costante, solo nel 2020 si è assistito ad una diminuzione del $-16,7\%$, un dato preoccupante ma che riflette la difficoltà di un periodo storico drammatico che

ha avuto ripercussioni sul settore. Con l'arrivo della pandemia di COVID-19, il settore è stato drasticamente colpito dalle restrizioni alla mobilità, dalla chiusura temporanea delle catene di approvvigionamento e dalla diminuzione della domanda dei consumatori; per questi motivi le case automobilistiche hanno dovuto adattare le proprie operazioni e, in molti casi, sospendere temporaneamente la produzione.

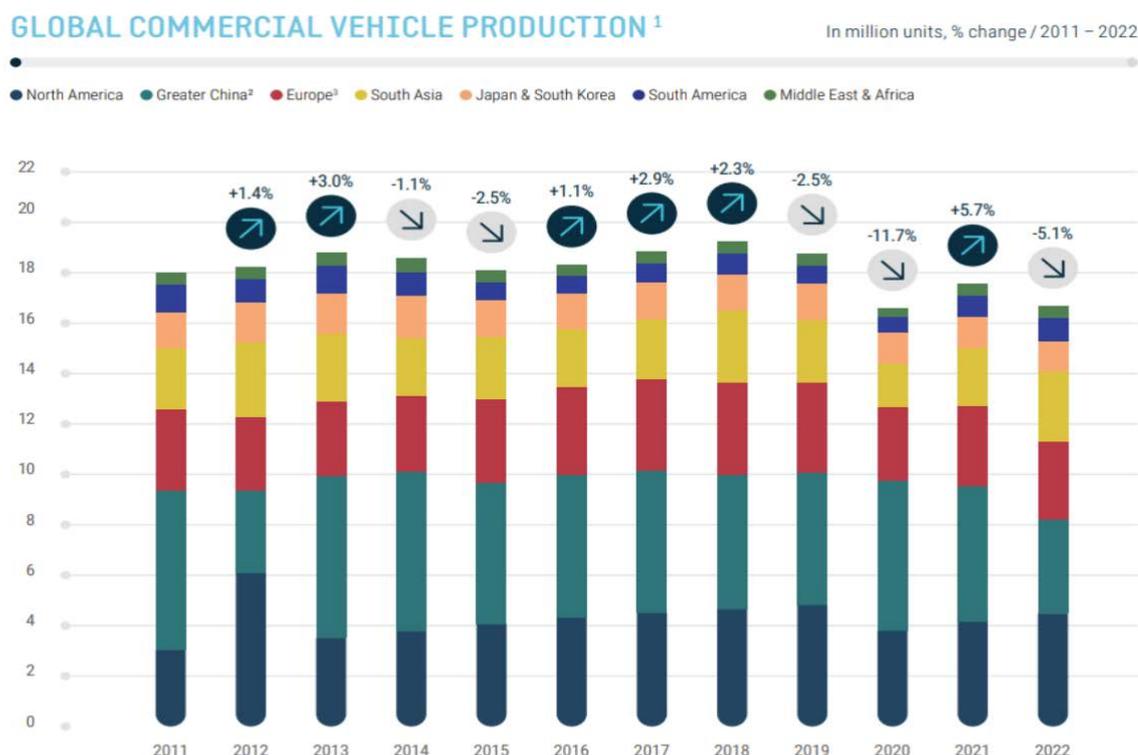


Figura 1.2: Global Commercial Vehicle Production [3]

La situazione è stata analoga anche per il segmento di mercato relativo alla produzione globale di veicoli commerciali per la produzione, anche per questo segmento la diminuzione in termini percentuali non è stata indifferente toccando il picco, nel 2020, del -11,7%.

Dopo il periodo più critico della pandemia, il settore automobilistico ha mostrato segni di ripresa; tuttavia, la velocità e la forza di questa ripresa hanno subito trend di variazione notevoli tra le regioni e i segmenti di mercato. Alcu-

ne aziende hanno sperimentato una forte domanda di veicoli, specialmente quelli orientati verso la sostenibilità, come i veicoli elettrici; altre, invece, hanno continuato ad affrontare sfide per la sopravvivenza specialmente quelle legate a problemi di approvvigionamento e alla scarsità di semiconduttori, che hanno rallentato la produzione.

In generale, la pandemia ha agito come catalizzatore di cambiamenti nella produzione di veicoli, molti produttori hanno ridefinito le loro strategie per affrontare le sfide post-COVID, con un maggiore focus sulla flessibilità operativa, la diversificazione delle catene di approvvigionamento e l’accelerazione della transizione verso veicoli più sostenibili. Il numero complessivo di veicoli prodotti è rimasto un indicatore chiave della salute del settore ma le dinamiche e le priorità sono cambiate in risposta agli impatti della pandemia e alle nuove esigenze dei consumatori.

Nelle due figure [(Fig.1.3) e (Fig.1.3)] che seguono, verrà illustrata la segmentazione del settore automotive, dividendo l’analisi per automobili e veicoli commerciali:

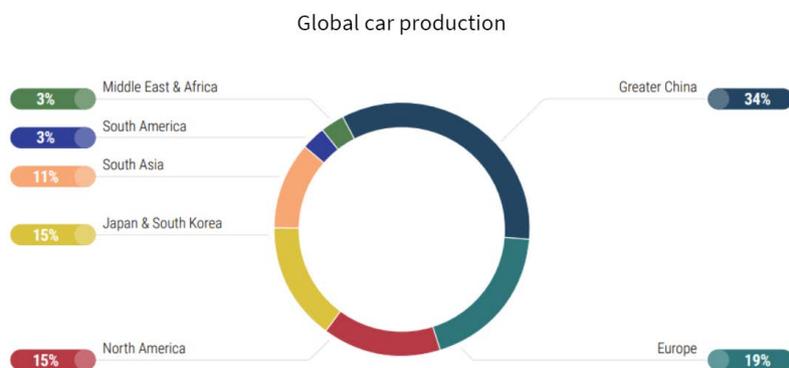


Figura 1.3: Distribution of Global Car Production [3]

Leader indiscusso per la produzione di autovetture è la Cina seguita da Europa, North America, Giappone e Corea del Sud e dal Sud Asia. La leadership della Cina nella produzione di autovetture è principalmente attribuibile alla sua ampia popolazione, che ha generato una significativa domanda interna. La crescita economica veloce ha contribuito all’incremento del reddito e alla crescente richiesta di

veicoli, per cui l'insieme di questi fattori, unito a una notevole capacità di produzione, ha consolidato la Cina come protagonista preminente nel settore automobilistico globale.

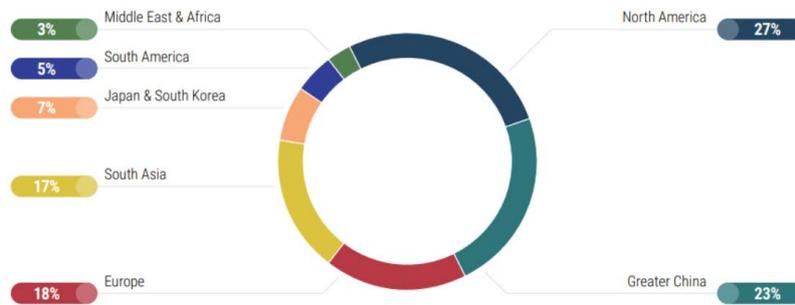


Figura 1.4: Distribution of Global Commercial Vehicle Production [3]

Gli Stati Uniti, la Cina, il Giappone, la Corea del Sud e alcuni stati dell'Europa sono paesi con notevole influenza nel settore automobilistico, coprendo sia il mercato delle automobili che quello dei veicoli commerciali. Queste nazioni ospitano importanti case automobilistiche, influenzano le dinamiche globali di produzione e vendita e sono spesso al centro delle innovazioni nel settore. La loro autorevolezza è determinata da fattori come la qualità dei veicoli prodotti, la forza economica, la dimensione del mercato interno e il ruolo nelle catene di approvvigionamento globali. [3]

1.2.1 Global Value Chain del settore automotive

La Global Value Chain (GVC) dell'industria automobilistica si presenta come un intricato tessuto di segmenti cooperanti tra loro. E' articolata principalmente in cinque parti fondamentali ([Fig.1.5]), includendo il segmento di ricerca e sviluppo (R&D), la produzione di componenti strategiche, la produzione e l'assemblaggio dei veicoli, il marketing e i servizi automobilistici. Di particolare rilievo è il differente contributo di valore aggiunto fornito da ciascun settore, con la ricerca e lo sviluppo automobilistico, la produzione di componenti chiave, il marketing e i servizi auto-

mobilitativi che emergono con un valore aggiunto più elevato rispetto alla produzione di componenti generici e la produzione/assemblaggio di veicoli.

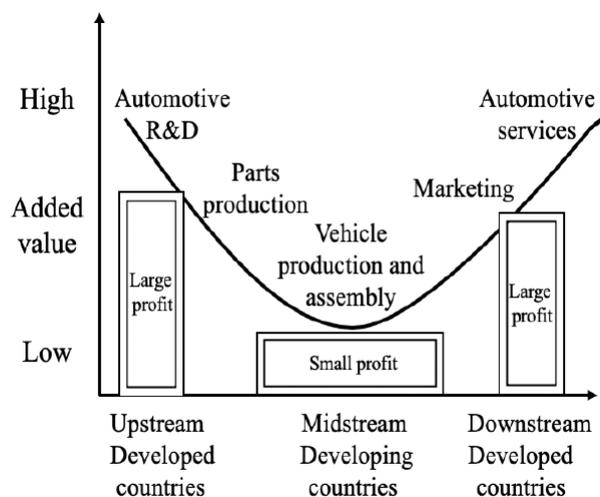


Figura 1.5: Global Value Chain (GVC) del settore automotive [4]

In secondo luogo, la distribuzione globale dei diversi segmenti della catena del valore dell'industria automobilistica rivela uno scenario stratificato. Le economie sviluppate, tra cui l'Europa, gli Stati Uniti e il Giappone, si posizionano in cima alla catena del valore globale, detenendo il controllo su risorse avanzate di ricerca e sviluppo, progettazione e tecnologie di produzione. Al contrario, molti Paesi in via di sviluppo occupano posizioni intermedie della catena del valore, trovandosi su livelli a basso valore aggiunto, come la produzione di componenti generici e l'assemblaggio di veicoli. La competizione futura nell'industria automobilistica, quindi, si prevede orientata principalmente verso i segmenti a più alto valore aggiunto, sottolineando l'importanza strategica di settori quali R&D, la produzione di componenti chiave, il marketing e i servizi automobilistici. [4]

1.3 Contesto operativo aziendale

ONHIGH è una società multinazionale leader nel settore dei veicoli commerciali e speciali, sistemi di propulsione e servizi finanziari correlati. L'azienda è famosa anche per la fornitura di servizi post-vendita di qualità, inclusa la manutenzione e

la fornitura di pezzi di ricambio; questa attenzione all'intero ciclo di vita dei veicoli contribuisce alla fidelizzazione della clientela e al mantenimento di una reputazione solida nel settore. Il core del business aziendale è rappresentato dalla progettazione e commercializzazione di una vasta gamma di veicoli:

- **Camion:** ONHIGH produce diverse tipologie di camion adatti per svariate esigenze e settori, queste includono i camion pesanti per il trasporto merci su lunghe distanze, veicoli specializzati per utilizzi specifici e camion leggeri per la distribuzione;
- **Autobus:** questi veicoli sono utilizzati per il contesto urbano e per il trasportare persone;
- **Veicoli industriali leggeri:** sono un esempio i furgoni commerciali e veicoli adatti per il trasporto di merci su distanze ridotte;
- **Veicoli speciali:** la società, oltre ai comuni veicoli, progetta e produce veicoli adatti per applicazioni particolari, come ad esempio i veicoli di difesa, veicoli per il fuoristrada e veicoli personalizzati su specifiche richieste del cliente;
- **Soluzioni sostenibili:** questa branca dell'azienda si occupa principalmente della produzione di soluzioni sostenibili come i veicoli a basse emissioni, veicoli elettrici o ibridi.

ONHIGH si distingue per la sua solida struttura organizzativa e la sua presenza consolidata nei mercati chiave di Europa, Asia, Africa e Sud America. La sua rinomanza è attribuibile alla costante ricerca di innovazione e alla dedizione per la qualità dei prodotti proposti. In un ambiente altamente competitivo; ONHIGH, inoltre, affronta con determinazione le sfide del settore, impegnandosi costantemente nella ricerca di opportunità per una crescita sostenibile. Nel corso di questo elaborato, il perimetro di analisi per la progettazione della dashboard sarà rispetto ai veicoli industriali leggeri e gli autobus.

Capitolo 2

Data Analytics e Business Intelligence

Per ottenere una comprensione approfondita della struttura del progetto oggetto dell'elaborato, in questo capitolo verrà fatta una panoramica sui concetti base dell'analisi dei dati e della Business Intelligence (BI) concentrando successivamente l'attenzione sullo strumento di analisi che verrà utilizzato per l'implementazione della dashboard: Qlik Sense, esplorando la sua struttura, il suo funzionamento e le sue caratteristiche chiave.

2.1 Cenni di basi di dati

Una base di dati (o database) è un'organizzazione strutturata e sistematica di dati gestita ed aggiornata; in generale, un database è progettato per contenere insiemi di dati correlati tra loro e per fornire un modo efficiente per archivarli, recuperarli e gestirli. Una base di dati è una collezione di dati gestita da un DBMS (Data Base Management System), ovvero un sistema di software in grado di gestire grandi collezioni di dati che siano: grandi, condivise, persistenti assicurando la loro affidabilità e privacy. Le caratteristiche principali di un sistema DBMS sono:

- **Creazione di Database:** Il DBMS consente la creazione di nuovi database e la definizione dei relativi schemi attraverso un linguaggio specializzato noto come linguaggio di definizione dei dati (Data Definition Language).
- **Query e Modifica dei Dati:** Gli utenti possono eseguire interrogazioni sui dati e apportare modifiche utilizzando un linguaggio dedicato denominato linguaggio di interrogazione o linguaggio di manipolazione dei dati.
- **Conservazione Sicura:** Il DBMS offre la capacità di archiviare un vasto volume di dati per lunghi periodi, garantendo la loro sicurezza.
- **Integrità dei Dati:** Garantisce l'integrità dei dati impedendo gli accessi simultanei a un dato specifico e controllando l'accesso concorrente da parte di diversi utenti. [5]

2.2 Modello dei dati

Un modello di dati in un data base è una rappresentazione astratta e organizzativa della struttura dei dati, questo modello definisce come i dati sono organizzati, memorizzati e manipolati. In letteratura ci sono diversi tipi di modelli di dati, ognuno con le proprie regole e convenzioni, però modelli di dati differiscono tra di loro e tra i principali troviamo, per esempio, il modello a documenti (NoSQL) e il modello a Serie Temporali ma in questo elaborato ci concentreremo sul modello Relazionale, ovvero il modello più diffuso e utilizzato in tutti i database moderni. Nel modello relazionale i dati sono organizzati in tabelle [(Fig.2.1)] e le relazioni tra le tabelle sono definite attraverso chiavi primarie e chiavi esterne.

Corsi	Codice	Nome	MatrDocente
	M2170	Fondamenti di informatica	D101
	M4880	Sistemi di elaborazione	D102
	F0410	Basi di dati	D321

Docenti	Matricola	Nome	Dipartimento	Telefono
	D101	Verdi	Informatica	123456
	D102	Bianchi	Elettronica	636363
	D321	Neri	Informatica	414243

Figura 2.1: Esempio di un Modello Relazionale

In un modello relazionale sono definiti lo schema che descrive la struttura dei dati ed è rappresentato dall'intestazione di ogni tabella e l'istanza, costituita dal contenuto di ogni tabella, cioè dei valori effettivi dei dati. La principale differenza tra schema e istanza è che il primo rimane invariato nel tempo, mentre l'istanza è variabile, anche molto rapidamente.

2.3 Progettazione di una base di dati

La progettazione di una base di dati è uno dei processi più critici all'interno di un progetto di Business Intelligence, che coinvolge la definizione della struttura e delle relazioni tra i dati con l'obiettivo finale di soddisfare i requisiti richiesti per l'applicazione che l'utente si aspetta. Il processo si suddivide in 4 momenti distinti: [(Fig.2.2)]

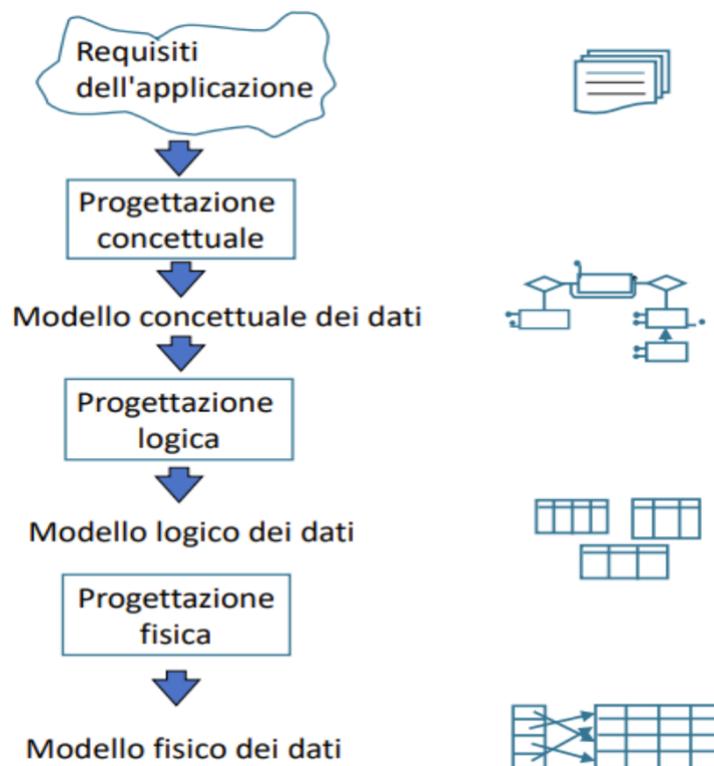


Figura 2.2: Step per la progettazione di un Data Base

- **Analisi dei requisiti:** Fase dedicata alla raccolta, filtraggio e analisi dei requisiti degli utenti finali da parte del progettista, al fine di comprenderne le informazioni di interesse strategico. Durante questa fase, vengono identificati i dati fondamentali da modellare e si ottengono indicazioni preliminari sul carico di lavoro. Una volta stabilito il carico di lavoro, si procede a definire la granularità nella rappresentazione dei dati. Questo passo è di cruciale importanza per il successo dell'intero progetto, poiché influisce sulla flessibilità delle interrogazioni all'interno del data mart. La granularità è determinata considerando la velocità di risposta desiderata e il livello di dettaglio richiesto nelle query, contribuendo così a plasmare la struttura complessiva del sistema. Nell'ambito specifico dei data warehouse, è essenziale introdurre concetti chiave:

- **Fatti:** I fatti rappresentano un insieme di eventi che si manifestano quando viene raggiunto un determinato obiettivo. Il fatto rappresenta

il problema del Business che si vuole modellare;

- **Attributi:** Gli attributi sono campi il cui valore è associato alla registrazione di un fatto generato da un obiettivo;
- **Dimensioni:** Una dimensione è una proprietà di un fatto che ne descrive una possibile coordinata di analisi;
- **Misure:** Le misure sono proprietà numeriche di un fatto che ne indicano un aspetto quantitativo.
- **Progettazione concettuale:** In questa fase, si procede alla creazione di uno schema concettuale per il data mart noto come Dimensional Fact Model (DFM) [(Fig.2.3)]. Per ciascun fatto di interesse identificato dagli utenti, si sviluppa uno schema che consente di rappresentare graficamente tutti i concetti fondamentali del modello multidimensionale, compresi fatti, misure, dimensioni e gerarchie.

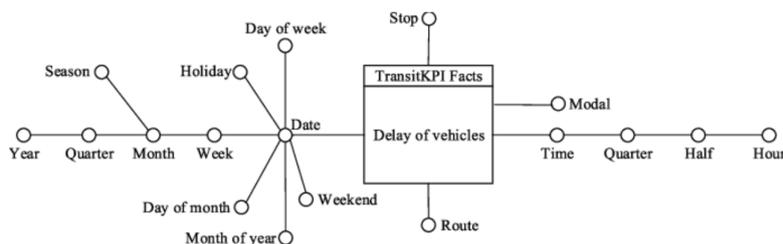


Figura 2.3: Dimensional Fact Model

- **Progettazione logica:** Nella progettazione logica vengono definite le procedure per sviluppare lo schema logico del data mart, partendo dallo schema concettuale precedentemente delineato. L'obiettivo primario di questa fase è massimizzare la velocità di recupero dei dati, consentendo l'uso efficiente di dati ridondanti e denormalizzati. In questa fase della progettazione viene definito lo schema logico [(Fig.2.4)] (che può essere a stella o snowflake) e le viste materializzate, ovvero strutture dati contenenti il risultato di una query precalcolata.

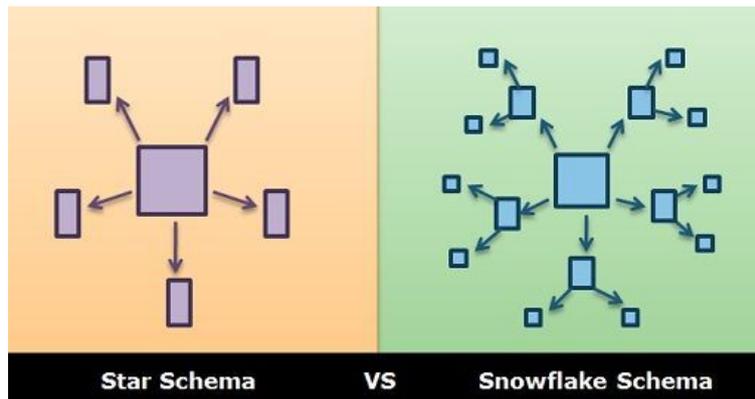


Figura 2.4: Star schema vs snowflake schema

Dalla progettazione logica ne deriva il modello logico dei dati, ovvero una rappresentazione astratta dell'organizzazione dei dati e le relazioni tra di essi; senza però, preoccuparsi dei dettagli tecnici del sistema di archiviazione o della sintassi SQL.

- **Progettazione fisica:** la progettazione fisica si concentra sugli aspetti tecnici e pratici dell'implementazione del database, tenendo conto delle caratteristiche specifiche del DBMS selezionato e delle esigenze di prestazioni dell'applicazione. Nel modello fisico dei dati viene definito come tali dati saranno memorizzati sul disco o in altri dispositivi di archiviazione. In corrispondenza di questo modello vengono definiti ad esempio quali colonne saranno utilizzate come chiavi primarie o la tipologia di dati che saranno utilizzati per rappresentare le informazioni all'interno del database.

2.4 Data Warehouse

Un Data Warehouse (DWH) costituisce un archivio organizzato dei dati aziendali necessario per l'analisi finalizzata ai processi decisionali; il Data Warehouse supporta i decisori descrivendo il processo di acquisizione, trasformazione e distribuzione di informazioni interne ed esterne alle aziende. Si distingue dai sistemi gestionali tradizionali che, a differenza dei primi, automatizzano operazioni di routine e vengono utilizzati per gestire applicazioni più ripetitive. Un concetto chiave del DWH

è la multidimensionalità, ciò significa che sfrutta uno spazio n-dimensionale per raggruppare ed analizzare gli eventi aziendali. Questa prospettiva multidimensionale è spesso rappresentata metaforicamente come un Cubo OLAP (On Line Transaction Processing) [(Fig.2.5)] , con gli assi che definiscono le dimensioni di analisi; ciascun asse rappresenta una possibile dimensione, con livelli di dettaglio identificati da gerarchie di attributi strutturati; ogni singolo cubo ottenuto dall'intersezione di tutte le dimensioni di analisi rappresenta un oggetto di dato con una granularità specifica. [6]

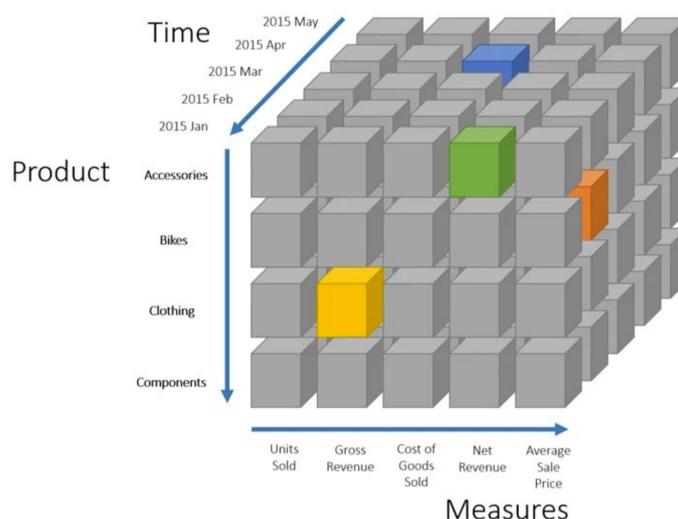


Figura 2.5: Cubo OLAP

2.4.1 Architettura di un Data Warehouse

L'architettura di un DWH è progettata per supportare la raccolta, la gestione e l'analisi di grandi quantità di dati aziendali e si inserisce nella parte centrale di un'architettura per l'analisi dei dati [(Fig.2.6)]. Un DWH è strutturato in modo da agevolare l'estrazione, la trasformazione e il caricamento dei dati (ETL), la memorizzazione organizzata delle informazioni e l'accesso da parte degli utenti attraverso strumenti di Business Intelligence.

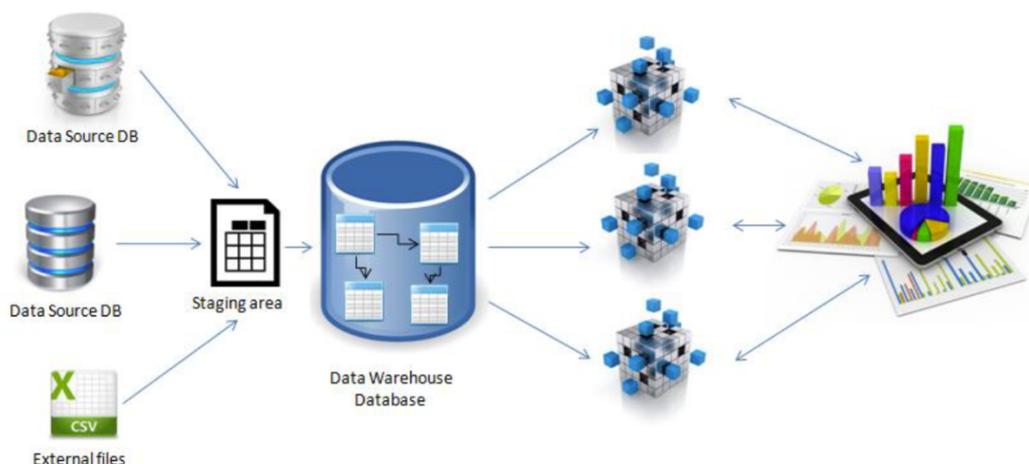


Figura 2.6: Architettura di sistema di Business Intelligence

A seconda della scelta dell'utente, l'architettura di un Data Warehouse può variare in base alle necessità del business. In questo elaborato, la struttura del DWH che verrà usato è di tipo three-layer:

- **Livello delle Sorgenti:** Questa componente coinvolge tutte le fonti da cui vengono estratti i dati, come database operativi, file, applicazioni esterne, ecc.
- **Livello di riconciliazione:** in questo livello, dopo il processo di ETL vengono sviluppate le logiche custom per il data warehouse;
- **Livello del Data Warehouse:** Questo livello è il cuore del DW e memorizza i dati in un formato organizzato per supportare l'analisi. Include: tabelle dimensionali e fattuali, i data marts, cioè un sottoinsieme dipartimentale del DW focalizzato su un settore prefissato, i metadati, ovvero informazioni dettagliate sulla struttura, la definizione e le relazioni dei dati all'interno del DS, gli strumenti di Business Intelligence, i quali consentono agli utenti di interagire con il DW per eseguire query, generare report e analizzare i dati.

2.5 Business Intelligence

La Business Intelligence (BI) negli anni è stata definita da diversi studiosi in modi molto simili. Business Intelligence è un termine introdotto nel 1989 da Howard Dresner, di Gartner Group, per descrivere un insieme di concetti e metodologie atti a sostenere il processo decisionale. Successivamente altri esponenti si sono pronunciati per dare una definizione alla BI; Singer fu uno dei primi, ad esempio, a definire la BI come una novità che apporta valore e che permette alle organizzazioni di ottenere informazioni decisionali che il normale reporting non fornisce. Noble, nel 2006, fornisce una definizione della BI come la capacità di fornire un vantaggio informativo per le aziende; in particolare, l'azienda fa quello che ha sempre fatto, ma nel modo più efficiente. Secondo Negash e Gray, invece, la BI è un processo guidato dai dati che combina l'archiviazione e la raccolta dei dati con la gestione della conoscenza per fornire input al processo decisionale aziendale. [7]

La BI è un insieme di tecnologie, processi, strumenti e metodologie che aiutano le organizzazioni a raccogliere, analizzare, presentare e monitorare i dati delle aziende per supportare la presa di decisioni informate. Dal punto di vista di implementazione, la BI include i seguenti processi [(Fig.2.7)] :

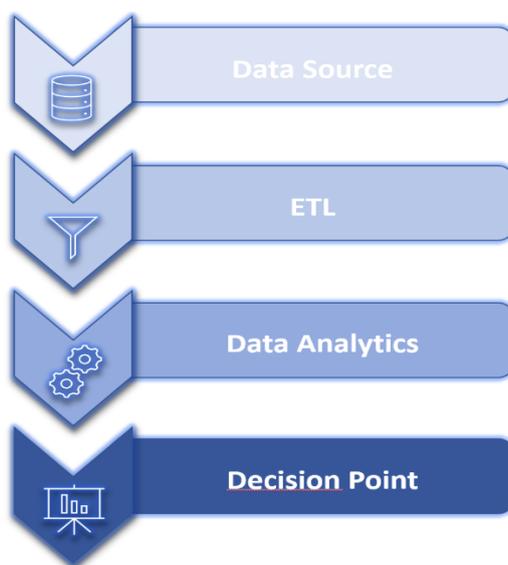


Figura 2.7: Processi in un progetto di Business Intelligence

La business intelligence comprende una serie di servizi essenziali [(Fig.2.8)] che consente, alle aziende che decidono di utilizzarla, di sfruttare al massimo il potenziale dei propri dati riuscendo così ad estrarre nuova conoscenza. Il processo inizia con la **Data Extraction**, dove le informazioni vengono estratte da diverse fonti, come database, file o applicazioni aziendali. Successivamente, il focus si sposta sulla **Data Cleansing**, un processo cruciale per garantire la qualità e l'integrità dei dati, eliminando errori, duplicati e inconsistenze. Una volta che i dati sono pronti, si procede con il **Data Storage**, che coinvolge la memorizzazione sicura e organizzata delle informazioni, spesso implementata attraverso data warehouses o data lakes. Il passo successivo è il **Data Modelling**, in questa fase si crea una rappresentazione strutturata e comprensibile dei dati, facilitando le analisi e le visualizzazioni. L'**Automating BI Dashboards** è un servizio che automatizza la creazione e l'aggiornamento di dashboard interattive, offrendo una panoramica visiva delle metriche chiave. La **Security & Governance** sono fondamentali per garantire la protezione dei dati sensibili e il rispetto delle normative. Questi servizi implementano politiche e procedure per controllare l'accesso ai dati e monitorare le attività. Infine, il **Self-Service BI** consente agli utenti di esplorare e analizzare i dati autonomamente, senza dipendere da team tecnici. Questo aspetto promuove l'empowerment degli utenti e accelera il processo decisionale. In sintesi, la BI fornisce una suite completa di servizi, dall'estrazione e pulizia dei dati fino alla creazione di modelli, dashboard automatizzate, sicurezza e la promozione dell'auto-servizio per massimizzare il valore derivato dai dati aziendali.



Figura 2.8: Business Intelligence Services

Lo scopo principale della BI è consentire ai decisori di tutti i livelli gestionali di prendere decisioni migliori e più tempestive. Si presuppone che la BI possa supportare il processo decisionale a tutti i livelli di gestione indipendentemente dal livello della loro strutturazione:

- A livello ESECUTIVO (o livello strategico) i sistemi di BI consentono di fissare con precisione gli obiettivi e di seguirne la realizzazione. La BI consente di eseguire diversi rapporti comparativi, ad esempio sui risultati storici, sulla redditività di particolari offerte, sull'efficacia dei canali di distribuzione, nonché di effettuare simulazioni di sviluppo o di prevedere i risultati futuri sulla base di alcune ipotesi;
- A livello di GESTIONE (o livello tattico) i sistemi di BI possono fornire una base per il processo decisionale nell'ambito del marketing, delle vendite, della finanza, della gestione del capitale, ecc. I sistemi consentono di ottimizzare le azioni future e di modificare opportunamente gli aspetti organizzativi, finan-

ziari o tecnologici delle prestazioni aziendali per aiutare le imprese a realizzare i propri obiettivi strategici in modo più efficace;

- A livello di PROCESSO (o livello operativo) i sistemi di BI vengono utilizzati per eseguire analisi ad hoc e rispondere a domande relative alle operazioni in corso dei reparti, alla situazione finanziaria aggiornata, alle vendite e alla cooperazione con fornitori e clienti, ecc. L'Operational Intelligence fornisce informazioni metriche in tempo quasi reale (o in tempo reale) sui processi, le attività e i risultati aziendali per supportare il processo decisionale operativo.

Da un punto di vista tecnico, l'architettura dei sistemi di BI permette l'analisi approfondita e la diffusione efficiente delle informazioni. L'analisi comprende soluzioni per esplorare in modo multidimensionale i dati economici reali, identificare le relazioni tra di essi e anticipare gli eventi futuri; questo livello si articola in strumenti analitici di base, quali query e report tradizionali, strumenti di elaborazione analitica in linea e visualizzazione dei dati, nonché strumenti analitici avanzati, tra cui strumenti statistici e data mining. Le applicazioni aziendali costituiscono un componente fondamentale, offrendo analisi specifiche per funzioni aziendali come produzione, vendite, monitoraggio della concorrenza e finanza, supportando così le decisioni in ambito aziendale. Un elemento cruciale nell'architettura della BI riguarda anche la distribuzione delle informazioni; in questo contesto, emergono due categorie principali di soluzioni: il metodo pull, che prevede la distribuzione dei risultati delle analisi attraverso portali aziendali all'interno di contesti intranet ed extranet, e il metodo push, che automatizza la diffusione dei risultati delle interrogazioni attraverso report inviati tramite canali come e-mail, SMS, fax o stampe automatiche. Particolare attenzione nei sistemi BI è data all'interfaccia utente che dovrebbe consentire all'utente tra l'altro: filtrare, ordinare e analizzare dati, formulare report ad hoc o predefiniti, generare scenari alternativi, produrre diagrammi e grafici. [8]

Data Visualization

La Data Visualization rappresenta un campo che trae contributi da diverse discipline. La psicologia, ad esempio, si occupa dello studio della percezione dei dati e dell'impatto di elementi come colori e forme sulla nostra interpretazione; parallelamente, l'informatica e la statistica hanno dato origine a nuove frontiere e strumenti, tra cui l'apprendimento automatico e le tecniche di data mining contribuendo così a un approccio più avanzato per l'analisi. Un ruolo cruciale è svolto anche dalla progettazione grafica e multimediale, che si rivela essenziale nella creazione di cruscotti infografici, i quali integrano diversi elementi quali dati, linee, barre e grafici. Few (2004) ha delineato chiaramente questi componenti sottolineando che i dati hanno il duplice scopo di quantificare le informazioni (dati quantitativi) e di classificarle (dati categorici). Gli strumenti di visualizzazione dei dati vengono impiegati sia nell'ambito industriale per facilitare il processo decisionale, sia nel contesto accademico. Nel campo dell'analisi aziendale, le visualizzazioni si rivelano estremamente utili per monitorare tutte le attività e per prendere decisioni in real time e svolge un ruolo fondamentale nel comprendere la posizione di mercato di un'azienda. A titolo di esempio, un buon tool di data Visualization misura le opinioni dei clienti, il mercato potenziale, e le azioni dei concorrenti, fornendo avvisi strategici precoci. [9] L'uso della data Visualization comprende tutti gli sforzi finalizzati a rendere più accessibile la comprensione dei dati mediante la loro rappresentazione visiva. La scoperta di modelli, tendenze e correlazioni, che potrebbero sfuggire durante l'analisi di dati testuali, diventa più evidente e facilmente identificabile grazie all'utilizzo di specifici strumenti software. Questa pratica implica la traduzione grafica di informazioni quantitative, trasformando sia serie di dati estese che più limitate in immagini agevolmente elaborate e comprensibili per il cervello umano. [10]

2.5.1 Tecnologia QlikSense

Qlik Sense è una piattaforma di Business Intelligence self-service, che permette agli utenti di creare report e dashboard senza la necessità di competenze tecniche com-

plesse. La sua struttura è stata concepita per fornire strumenti drag-and-drop che agevolano la creazione veloce e semplice di visualizzazioni dati. Qlik Sense, creata dalla compagnia svedese Qlik, fondata nel 1993 e lanciata nel 2014, rappresenta la progressione naturale di QlikView, una piattaforma di Business Intelligence precedentemente introdotta dall'azienda. Nel suo sviluppo, Qlik ha mirato a realizzare un prodotto più moderno e accessibile a tutti gli utenti, accentuando la sua user-friendliness.

Questo software di BI offre notevoli vantaggi attraverso la sua capacità di integrarsi con una vasta gamma di fonti dati. La piattaforma supporta vari formati di file, come Excel, CSV, XML e JSON, e può accedere a database SQL e NoSQL, servizi cloud e molte altre risorse dati. Ciò significa che gli utenti hanno la possibilità di analizzare dati provenienti da diverse fonti, inclusi quelli relativi a marketing, vendite, finanza e risorse umane. Con la sua architettura associativa, Qlik Sense consente un'analisi flessibile e dinamica dei dati. Gli utenti possono esplorare in modo intuitivo, approfondire i dettagli e stabilire connessioni tra i dati che potrebbero non essere state inizialmente previste, questo approccio consente, quindi, di rivelare nuove correlazioni e tendenze altrimenti difficilmente individuabili. L'adozione di questa tecnologia innovativa elimina la necessità di definire diverse gerarchie dimensionali nei dati, come avviene nei cubi OLAP tradizionali; e questo risulta essere un aspetto chiave nella scelta di questa soluzione poichè si ha una riduzione dei tempi nella gestione del progetto non indifferente.

La struttura dati di QlikSense è basata su un modello relazionale nel quale i record sono organizzati per minimizzare la ridondanza, e utilizza i campi chiave per riconciliare i record durante il loro utilizzo. Durante il processo di estrazione dei dati da una sorgente, QlikSense genera un modello dati in-memory che consente l'accesso rapido con maggiore velocità. L'interfaccia utente [(Fig.2.9)] del software risulta essere intuitiva e interattiva poichè presenta diversi curscotti, tabelle e grafici avanzati che possono variare a seconda delle esigenze dell'utente. Inoltre, è possibile scambiare i risultati ottenuti tra i diversi users attraverso funzioni di condivisione o creazioni di report. [11]

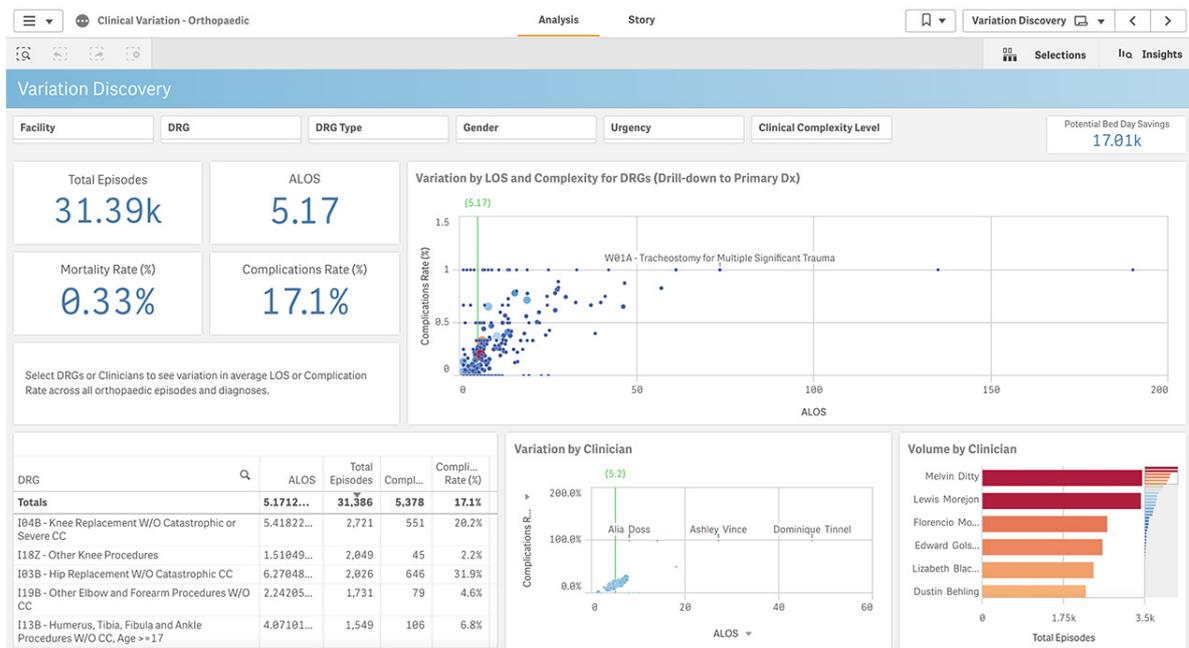


Figura 2.9: Esempio di una dashboard sviluppata in ambiente Qlik Sense [11]

Caratteristiche e funzionalità di Qlik Sense

In questo paragrafo verranno illustrate le principali caratteristiche e funzionalità che rendono Qlik Sense uno strumento di Business Intelligence self-service innovativo e performante rispetto alle altre soluzioni presenti sul mercato:

- Motore QIX:** il nucleo centrale di tutte le funzionalità di Qlik Sense è rappresentato dal motore Qlik Indexing Engine (QIX). Questo motore svolge un ruolo chiave ottimizzando l'utilizzo dei dati mediante avanzate tecniche di indicizzazione e compressione. Tale ottimizzazione consente l'elaborazione efficiente di ampie quantità di dati provenienti da diverse fonti. Inoltre, il motore QIX si distingue per la sua abilità nella rapida creazione di associazioni tra i dati e nell'esecuzione dinamica e altamente efficiente dei ricalcoli, caratteristiche di fondamentale importanza per la funzionalità complessiva della piattaforma;
- Modello associativo:** Il concetto di Modello Associativo in Qlik Sense impiega un approccio che consente il collegamento e l'associazione di tutti i dati

caricati, indipendentemente dalla loro provenienza. Quando un campo di una tabella dati presenta associazioni con altri campi presenti in diverse tabelle, l'utente può visualizzare tali connessioni in modo agevole mediante selezioni interattive o utilizzando la ricerca intelligente. Questa metodologia agevola l'estrazione di informazioni rilevanti, rendendole accessibili attraverso processi interattivi altrimenti difficili da dedurre;

- ***Visualizzazioni automatiche***: questa funzionalità è alla base del concetto di strumento user-friendly; Qlik Sense propone, in modo autonomo, visualizzazioni intelligenti e avanzate che aiutano ad analizzare in maniera più precisa i dati.
- ***Creazione di app self-service***: La creazione di app self-service su Qlik Sense è un processo agevolato dalle funzionalità di trascinamento e rilascio offerte dal software. La realizzazione di un'applicazione coinvolge pochi e semplici passaggi, che includono la preparazione e il caricamento dei dati, oltre alla composizione di dashboard, tutto senza la necessità di scrivere codice script o eseguire complesse interrogazioni. Questa funzionalità risulta essere fondamentale per tutti coloro che non possiedono delle skills nell'ambito del coding e permettono di manipolare dati senza dover utilizzare complessi script di codice.
- ***Qlik Management Console (QMC)***: Governance e dimensionamento sono gestiti attraverso la Qlik Management Console (QMC), consentendo un controllo centralizzato e monitoraggio dell'intera piattaforma. La QMC offre una serie di servizi dedicati a questo scopo, tra cui la sincronizzazione delle attività, la configurazione di regole di accesso e sicurezza, la gestione delle licenze e la creazione di connessioni dati. Inoltre, la scalabilità rappresenta un aspetto fondamentale in Qlik Sense, che sfrutta il suo motore dinamico QIX per abilitare un caricamento ed un'analisi efficienti di considerevoli quantità di dati. [12] I servizi disponibili nella QMC sono i seguenti:

- Connettività dei dati;
- Gestione delle applicazioni;
- Gestione dei task automatici;
- Monitoraggio dei dati;
- Amministrazione e sicurezza delle app.

Architettura di Qlik Sense

Nel processo di sviluppo di un'applicazione su Qlik Sense [(Fig.2.10)], la prima fase implica l'identificazione dei dati da importare. Tipicamente, tali dati sono estratti da un file, un foglio di calcolo o un database preesistente; la procedura prevede la possibilità di caricare i dati da una singola sorgente file o database, oltre a consentire il caricamento e l'integrazione simultanea di dati provenienti da diverse fonti. Una volta definiti i dati necessari per l'analisi, bisogna creare uno script di caricamento, grazie al quale viene semplificato il processo, all'interno del quale verrà definito il processo per integrare i dati all'interno del data lake nel quale avverrà lo storage, in cui le applicazioni e i report possono essere creati e condivisi dagli utenti ed è consentito anche l'utilizzo di più dispositivi contemporaneamente. [11]

Nel corso della fase di caricamento, Qlik Sense analizza ogni singola istruzione presente nello script e la esegue in successione; all'interno dello script di caricamento, è possibile incorporare l'istruzione "store", la quale consente la creazione di una duplicazione dei dati caricati. Questi dati vengono archiviati in file nel formato QVD (QlikView Data), che rappresenta un file contenente una tabella di dati esportati da Qlik Sense o Qlik View. Il QVD è un formato specifico di Qlik progettato per la lettura e la scrittura esclusiva da parte di Qlik Sense o Qlik View. Questo tipo di file è ottimizzato per garantire una rapida lettura dei dati da uno script di Qlik Sense, mantenendo comunque una struttura molto compatta. Comunemente, si segue un pattern [(Fig.2.10)] ben definito nella configurazione di un progetto Qlik, il quale include passaggi adattabili allo specifico progetto:

- **Sorgenti dati:** In questa prima fase vengono raccolti tutti i dati utili per l'implementazione del progetto; le informazioni possono provenire da data-warehouse esistenti, da file esterni o da file qvd utilizzati in altre applicazioni di Qlik Sense.
- **Estrazione o ETL1:** Questa fase è imputata al caricamento e lo storage dei file nel Data Lake predefinito. In Qlik Sense, questa parte, corrisponde all'estrazione dei dati dalle sorgenti e alla memorizzazione delle informazioni in file qvd. A questo livello, l'estrazione avviene senza apportare modifiche ai dati.
- **Trasformazione o ETL2:** A partire da questo step viene effettuata la manipolazione dell'informazione attraverso, ad esempio, l'implementazione di specifiche logiche. Le applicazioni QlikSense di questo secondo livello caricano i dati precedentemente salvati nei qvd di Staging e li elaborano attraverso trasformazioni e normalizzazioni; l'obiettivo è renderli fruibili secondo le necessità specifiche del progetto. I dati così elaborati sono poi memorizzati in file qvd di secondo livello.
- **Presentazione:** Corrisponde alla visualizzazione dell'app; in questa fase, avviene il caricamento delle tabelle dal livello precedente e la creazione di un modello relazionale, noto come "nuvola dati". A differenza delle fasi precedenti, i dati non vengono più storicizzati, ma rimangono all'interno dell'applicazione. Per utilizzarli, è necessario aggiungere elementi all'interfaccia che consentano di organizzare i dati in modo ordinato e specifico, in linea con le esigenze particolari del progetto.

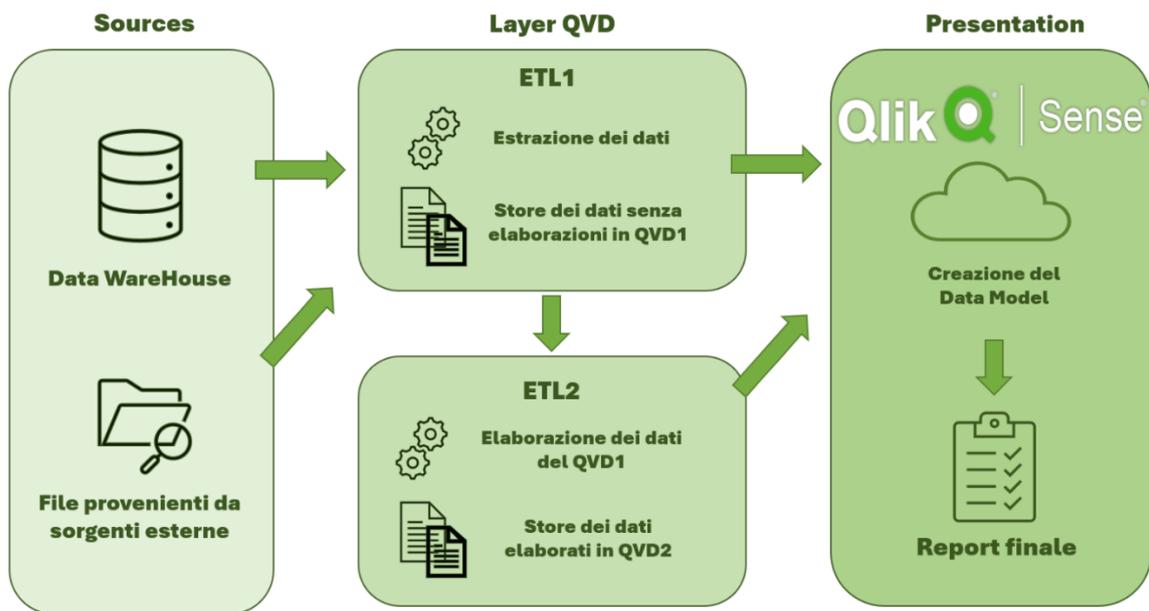


Figura 2.10: Schema di processo di Qlik Sense

Capitolo 3

Business Issue

Il progetto VORlog Dossier si inserisce all'interno di VORlog, un insieme di attività del programma di Quality and Warranty di ONHIGH, tese a migliorare l'affidabilità dei dati relativi a tutti gli aspetti inerenti la qualità dei veicoli. All'interno di questi fascicoli sono presenti tutte le informazioni relative ai veicoli effettivamente fermi in officina, come suggerito dall'acronimo VOR che sta per "Vehicles Off Road". L'obiettivo finale del progetto è quello di sviluppare una dashboard su Qlik Sense che permetta in primo luogo di monitorare l'UpTime, che è la disponibilità dei veicoli, dell'intero parco circolante e in secondo luogo di valutare la consistenza dei dati contenuti all'interno di ogni singolo dossier.

Più precisamente, il secondo obiettivo, ovvero quello del consolidamento della consistenza dei dati dei dossiers, verrà perseguito grazie all'integrazione all'interno della dashboard di due informazioni che normalmente non sono presenti all'interno del fascicolo digitale, la Digital Job Card (DJC) e GeoFencing. Questo aspetto è di fondamentale importanza, poiché i dati attuali che confluiscono in VORlog non garantiscono con certezza che i veicoli siano ancora presenti in officina, a causa di alcune logiche di chiusura applicate ai dossier VORlog, che al momento sono in fase di riformulazione. L'integrazione del geofencing rappresenta uno strumento estremamente affidabile, in grado di fornirci la data di uscita dei veicoli dalla zona di officina. Ciò consente di valutare con certezza se un determinato dossier

VORlog, apparentemente aperto per una lavorazione in corso, necessita ancora di attenzione o può essere chiuso, dal momento che è possibile avere la certezza che il veicolo non è più in officina. Le DJC, in maniera simile al GeoFencing permette di mappare tutti gli interventi avvenuti in officina fornendo uno strumento in più per integrare o verificare l'attendibilità del dato già presenta all'interno dei dossiers.

L'applicazione finale faciliterà l'individuazione tempestiva di eventuali problematiche, riducendo i tempi di inattività e migliorando l'efficienza globale dell'azienda. Questo sarà possibile grazie alla sua capacità di fornire linee guida per rilevare e risolvere prontamente i problemi identificati attraverso dati al di fuori dei parametri predefiniti. Il monitoraggio avverrà tramite indicatori chiave di prestazione (KPI) specifici, come ad esempio il numero di dossier e il tempo di riparazione (cioè il periodo che indica quanto tempo un veicolo è rimasto fermo) o l'UpTime. Questi KPI saranno integrati in vari fogli di analisi, consentendo analisi dettagliate con differenti livelli di granularità e concentrandosi su diverse dimensioni di analisi.

Con il termine Uptime si intende l'intervallo di tempo in cui un sistema, un dispositivo o un'applicazione è operativo e funzionante correttamente; ovvero rappresenta la durata durante la quale un preciso oggetto o componente è in grado di svolgere le funzioni, per cui è preposto, senza interruzioni o malfunzionamenti. Un alto Uptime indica una maggiore disponibilità operativa, mentre un basso Uptime può indicare periodi di fermo o manutenzione. Nell'industria automobilistica la disponibilità operativa dei veicoli è essenziale per garantire il corretto funzionamento delle attività commerciali e quindi, di conseguenza, la soddisfazione del cliente. Il monitoraggio del Uptime, e quindi delle informazioni relative agli eventuali guasti o malfunzionamenti di un veicolo, consente alle aziende di identificare e affrontare tempestivamente eventuali problemi, riducendo al minimo i tempi di fermo e migliorando l'efficienza.

3.1 Time-line Progetto di Business Intelligence

Al giorno d’oggi non tutte le realtà aziendali possono permettersi di sostenere i costi legati alla realizzazione di un progetto di Business Intelligence, per questo motivo l’attività di sviluppo viene interamente data in “outsourcing”, ad esempio, ad una società di consulenza specializzata. Una scelta di questo tipo risulta essere vantaggiosa, principalmente, per due motivi: il primo è relativo al forte livello di specializzazione della società di consulenza; la quale, grazie alle economie di apprendimento e all’economia di scala sviluppata nel settore, risulta essere più efficiente. In secondo luogo, perchè una volta instaurato il giusto coordinamento tra le parti, la società di consulenza sarà in grado di personalizzare i servizi erogati in base alle esigenze del cliente. Nella [(Fig.3.1)] che segue vengono illustrati tutti gli step che sono alla base di un progetto di Business Intelligence:



Figura 3.1: Time line di un progetto di Business Intelligence

Il punto di partenza è il “Business Issue”, ovvero l’espressione di una necessità o un bisogno da soddisfare da parte del cliente in termini di implementazione di una nuova soluzione BI. La fase immediatamente successiva è quella della Functional Analysis, la quale ha come principale obiettivo quello di raccogliere e analizzare in maniera più dettagliata i requisiti che il sistema dovrà soddisfare; in questa fase vengono organizzati meeting con gli Stakeholders chiave per la definizione per esempio delle fonti dati, della tipologia dei dati necessari, degli indicatori chiave di performance (o anche definiti KPI “Key Performance Indicator”) e delle assunzioni che vengono fatte a monte del progetto e che dovranno essere rispettate fino alla conclusione dello stesso. Dopo l’approvazione del documento di analisi funzionale si passa allo sviluppo vero e proprio della soluzione; nel caso di progetti BI questa fase potrebbe includere, per esempio, lo sviluppo di un nuovo Modello Dati, la creazione

di una nuova Dashboard, lo sviluppo di un Data Warehouse o l’implementazione di nuovi strumenti di analisi. Una volta terminato lo sviluppo, lo step successivo è il test e validazione dei risultati ottenuti con il cliente, in questa fase l’utente finale si avvicina al nuovo strumento e valuta in prima persona se il risultato ottenuto soddisfa le sue aspettative, superata la fase di test si prosegue con il rilascio in produzione e quindi, di conseguenza, l’immissione del nuovo strumento nell’azienda.

3.2 Raccolta dei requisiti

All’avvio di un progetto nell’ambito della Business Intelligence, una tappa cruciale è rappresentata dalla raccolta dei requisiti utente, un’attività di fondamentale rilevanza per assicurare il buon esito dell’intero progetto. Questo processo essenziale si basa su un attento ascolto degli obiettivi che il progetto deve raggiungere, i quali vengono formulati attraverso i requisiti definiti dagli utenti. Tali requisiti delineano le specifiche funzionalità che il software dovrà incorporare al fine di soddisfare le esigenze degli utenti. Questo processo di individuazione è essenziale per garantire che l’app sviluppata soddisfi in modo completo e accurato le effettive esigenze degli utenti, contribuendo così al successo complessivo del progetto. Nell’ambito del VorLog Dossier Project i requisiti principali che deve avere l’app sono:

- L’applicazione deve fornire dei dati consistenti per il monitoraggio dei dossiers aperti e chiusi;
- KPIs che permettano di monitorare l’Uptime del parco veicolare circolante;
- Sviluppo di una metodologia che permetta di identificare tutti I dossiers che per inconsistenza della sorgente dato non vengono chiusi e che quindi presentano una durata anomala;
- L’applicazione deve fornire un confronto year-to-date;
- Oltre ai fogli per la visualizzazione delle informazioni, all’interno dell’app saranno presenti due fogli chiamati “Raw data – Dossiers” e “Raw data – Com-

ponents” all’interno dei quali verranno inseriti tutti i dati provenienti dal flusso ASIST (cioè il flusso che permette di avere le informazioni sui dossiers) senza alcuna manipolazione o trasformazione principale.

3.3 Caratteristiche del caso studio

In questo paragrafo verranno illustrate le caratteristiche principali alla base del caso studio per la corretta comprensione del proseguo dell’elaborato. All’interno dell’azienda, oggetto del caso studio, il monitoraggio del tempo di fermo di un veicolo all’interno di un’officina viene monitorato grazie ad un Dossier dedicato. Il dossier rappresenta un fascicolo digitale all’interno del quale vengono inserite tutte le informazioni sul veicolo a partire dalla data di fermo fino al suo rilascio in circolazione. Ogni singolo dossier appartiene ad uno ed un solo veicolo, e contiene al suo interno alcune informazioni chiave come, ad esempio, la “Created on” ovvero la data in corrispondenza della quale è stato creato lo specifico Dossier, la “Breakdown date” che corrisponde alla data in cui si è verificato il guasto al veicolo, la “Time of Repair” che è la data in cui il veicolo è tornato a circolare normalmente, oltre alle informazioni relative ai componenti del veicolo che hanno subito dei guasti; l’apertura del primo component sancisce l’apertura del dossier associato allo specifico veicolo, mentre la chiusura dell’ultimo component in sospeso determina la chiusura del dossier.

3.3.1 Struttura del progetto

Il progetto è suddiviso in due macro wave che rappresenteranno due steps complementari, la prima WAVE riguarderà la creazione ex-novo dell’applicazione a partire dalla raccolta dei requisiti fino ad arrivare allo sviluppo, mentre la seconda WAVE avrà come obiettivo l’integrazione al risultato ottenuto precedentemente di alcuni campi specifici per effettuare delle analisi più approfondite. Nello specifico, le due WAVE sono strutturate nel seguente modo:

1. **WAVE 1:** In questa prima fase del progetto l'attenzione si concentra sull'elaborazione dei dati e sulla creazione di una dashboard Qlik Sense che mostra:
 - A) I dossier VORlog aperti e chiusi filtrabili in base a diverse dimensioni di analisi;
 - B) KPI relativi al contatore e alla lunghezza dei dossier (a livello di componente per capire quali sono i componenti che mantengono aperto il dossier VORlog).
2. **WAVE 2:** Lo strumento sviluppato nella WAVE 1 sarà integrato da 2 fonti di dati che sfruttano il Data Lake di ONHIGH (EDS), questi nuovi dati da inserire servono principalmente per valutare la consistenza delle informazioni contenute all'intero dei dossier. Le due fonti dato sono:
 - A) Digital Job Card (DJC): sono delle schede digitali che vengono aperte dalle officine per il cliente finale utili per mappare gli interventi attraverso un tool web. All'interno di questo tool è presente un data base all'interno del quale vengono inserite le informazioni relative al fermo dello specifico veicolo;
 - B) Geo Fencing: è un dato di origine telematica che permette di sapere se il veicolo è presente presso un determinato dealer (ovvero in officina).

3.3.2 File VorLog Dossier

Il "VORlog Dossier" è un fascicolo all'interno del quale vengono inserite tutte le informazioni relative ad un fermo veicolo. I file di log, nell'ambito informatico, costituiscono documenti elettronici che documentano dettagliatamente gli eventi, le azioni e le operazioni che si verificano all'interno di un sistema informatico; questi resoconti digitali sono generati da diversi componenti, a seconda di come si presenta l'architettura del sistema, tra cui: il sistema operativo, le applicazioni, i server web e i dispositivi di rete. Essenzialmente, i file di log svolgono il ruolo di testimoni digitali registrando in modo accurato e cronologico le attività svolte nel sistema. In questo modo, forniscono una traccia dettagliata e di facile consultazione che può

essere impiegata per monitorare, analizzare e risolvere eventuali problemi o anomalie nell'ambiente informatico. Un dossier può nascere da 4 diversi sistemi gestionali:

1. DMS (Dealer Management System), ovvero sistemi di gestione dei vari dealer, questo gestionale manda tramite interfaccia UDI o "Unique Device Identification" dei file .csv che vengono importati in un tool di ASIST;
2. CRM ANS24 (Customer Relationship Management Assistance No Stop H24): sono dossier che, generalmente, sono associati a delle riparazioni che richiedono una certa urgenza;
3. CSPA (Customer Product Support Reporting): in occasione di escalation di componenti Part Order;
4. Dealer tramite il front-end ASIST.

Ad ogni sorgente è associata una tipologia di ordine; in altre parole, lo specifico component inserito, a seconda della data source, può appartenere alla categoria NETWORK per i component inseriti dal sistema DMS, ANS24 per i sistemi CRM ANS24, VOR Parts e Part Order per il sistema CSPA e Request Report per quelli inseriti tramite il front-end di ASIST. Nonostante le fonti siano diverse, è possibile che un dossier contenga al suo interno più componenti diversi provenienti da diverse sorgenti. Inoltre, ogni dossier è specifico per un solo veicolo ed è identificato dalla chiave formata dal SAP Code, che è un codice univoco generato dal gestionale e dal VIN, ovvero il numero identificativo del telaio del veicolo. La chiusura di un dossier, che corrisponde all'uscita del veicolo dall'officina, avviene nel momento in cui anche l'ultimo componente ancora aperto viene chiuso; quest'ultimo aspetto risulta essere il più cruciale, poiché la non chiusura dei dossier è uno dei problemi alla base della commissione di questo progetto in quanto non permettono un'analisi veritiera dei dati nell'area di warranty and quality dell'azienda. Una delle finalità di questa dashboard è quella di riuscire ad individuare quali sono i dossier che non sono stati erroneamente chiusi e risalire, successivamente, al motivo per cui i sistemi sorgente non si comportano correttamente.

3.3.3 Dimensioni di analisi

Le dimensioni di analisi all'interno di una dashboard si riferiscono agli attributi o alle variabili utilizzate per esaminare e suddividere i dati; sono essenziali per ottenere una comprensione approfondita delle informazioni presentate e identificare correlazioni o modelli significativi nei dati. L'obiettivo è selezionare le dimensioni che meglio contribuiscono agli obiettivi di analisi e forniscono una visione completa e informativa del contesto in cui si sviluppano i dati, per cui la scelta delle dimensioni è influenzata dalla loro rilevanza rispetto agli obiettivi aziendali, dalla varianza significativa nei dati, dall'utilità nel guidare decisioni o azioni e dall'accessibilità dei dati necessari per supportarle. In definitiva, le dimensioni di analisi giocano un ruolo cruciale nel fornire un quadro comprensivo e significativo dei dati presentati attraverso una dashboard. La comprensione dettagliata delle fonti è essenziale per un'analisi accurata e per identificare meglio le dimensioni di analisi, le varie sorgenti dei dati contribuiscono in modo significativo alla formazione di una panoramica approfondita che verrà fatta nel capitolo seguente. Per quanto riguarda le informazioni specifiche sui singoli veicoli e sui contratti di garanzia, tali dati sono estratti dal Data Warehouse NEWBI all'interno di due tabelle distinte che chiameremo Vehicle e PMCH. Passando ai dossiers e ai componenti, i dati pertinenti emergono dai flussi ASIST sotto forma di due file QVD distinti. Infine, i dati necessari per la verifica relativa alla consistenza delle informazioni del dossier provengono da Digital Job Cards (DJC) e GeoFencing. Nel contesto del progetto "VORlog Dossier", le dimensioni selezionate dal business per l'analisi sono dei campi che provengono da specifiche tabelle e includono sia principali come ad esempio la Time of Repair, il Dealer il mercato di riferimento; sia informazioni accessorie come il rivenditore del veicolo, le officine di riparazione e il mercato del cliente finale. Di seguito l'elenco delle dimensioni di analisi:

Le dimensioni appena descritte verranno utilizzate per implementare i filtri di analisi all'interno della dashboard, utili per navigare i vari fogli dell'app e per ricercare le informazioni d'interesse.

Tabella 3.1: Dimensioni di analisi

<i>Dimensione</i>	<i>Descrizione</i>	<i>Fonte Dati</i>
Market	Rappresenta il mercato di riferimento in cui è stato preso in carico il fermo veicolo	Vehicle
Vehicle range	Tipologia del veicolo tra: Light/Medium/Heavy	Vehicle
Vehicle model	Modello specifico del veicolo	Vehicle
Vehicle age	Anzianità del veicolo al momento del fermo, corrisponde alla differenza tra la data di apertura del dossier e la data di inizio del contratto di garanzia	Vehicle
Number of days x component	Nemro di giorni per ogni componente associato allo specifico dossier	Components
Is NP vehicle	E' un campo che indica se il veicolo appartiene alla categoria Natural Power	Vehicle
Customer	Cliente finale	Vehicle
Dealer	Rivenditore di riferimento	Vehicle
Component	E' lo specifico component che è stato inserito all'interno del dossier	Components
Opened on	Data corrispondente al giorno in cui è stato aperto il dossier	Dossiers
VorLog dossier status	Flag sul dossier per identificare un dossier Open, Closed o Cancelled	Dossiers
VorLog dossier TR	Time of Repair, ovvero la data in cui il veicolo viene rilasciato in circolazione dopo il fermo	Dossiers
Opening and closure of Components	Data di apertura e chiusura relativa al singolo component	Components
Is in M&R (Y/N)	E' un filtro particolare del parco veicolare e permette di individuare quei veicoli che possiedono una polizza di copertura per eventuali guasti	PMCH
M&R validity dates	Sono le date di validità per i servizi di garanzia estesi	PMCH
Connected vehicle	Sono tutti i veicoli che possiedono dei contratti di garanzia	Connected Vehicle
Warranty start date	Data a partire dalla quale è stato attivato un servizio di garanzia per lo specifico veicolo	Vehicle
DJC	Dato utile per tracciare tutte le azioni svolte all'interno di un'officina	DJC
GeoFencing	Dato utile per tracciare tutte le azioni svolte all'interno di un'officina	GeoFencing

3.3.4 Key Performance Indicators (KPI)

I Key Performance Indicators (KPI), o Indicatori Chiave di Performance, sono metriche quantificabili utilizzate per valutare il successo o il rendimento di un'organizzazione, di un progetto o di un processo. Essi fungono da strumenti di misurazione strategica, consentendo di monitorare e valutare in modo efficace il raggiungimento degli obiettivi prefissati. I KPI servono a diversi scopi all'interno di un contesto aziendale:

- **Misurazione delle Prestazioni:** I KPI forniscono una misura chiara e tangibile delle prestazioni in relazione agli obiettivi strategici. Consentono alle organizzazioni di comprendere quanto stanno progredendo verso il conseguimento dei risultati desiderati.
- **Allineamento agli Obiettivi:** I KPI sono scelti in base agli obiettivi specifici dell'organizzazione o del processo. Questa selezione mirata assicura che le attività quotidiane siano allineate con gli obiettivi più ampi, mantenendo una chiara visione della direzione strategica.
- **Supporto alle Decisioni:** Forniscono un quadro oggettivo per prendere decisioni informate. Analizzando i KPI, i responsabili possono identificare aree di forza, debolezza o opportunità, facilitando il processo decisionale.
- **Comunicazione dell'Andamento:** I KPI sono strumenti efficaci per comunicare l'andamento delle prestazioni a tutti i livelli dell'organizzazione. Questa trasparenza promuove la consapevolezza e l'orientamento comune verso gli obiettivi condivisi.

L'implementazione dei KPI è fondamentale per garantire che le organizzazioni abbiano una comprensione chiara delle proprie performance e siano in grado

Tabella 3.2: Key Performance Indicators (KPIs)

<i>KPI</i>	<i>Descrizione</i>
<i>Numero di Dossiers</i>	Numero totale di dossiers creati da tutte le fonti dato.
<i>Numero di Dossiers aperti</i>	Numero di dossiers ancora aperti.
<i>Numero di Dossiers chiusi</i>	Numero di tutti i dossiers chiusi
<i>Numero dei Components</i>	Numero totale di tutti i components all'interno del dossier
<i>Durata del Dossier</i>	Intervallo di tempo che va dall'apertura del dossier fino al rilascio in circolazione del veicolo associato.
<i>Durata del Component</i>	Intervallo di tempo che identifica la durata di un component che va dalla sua apertura nel dossier specifico fino alla sua chiusura.
<i>Time of Repair</i>	Rappresenta la data in cui il veicolo viene rilasciato e quindi torna a far parte del parco veicolare circolante.
<i>Percentuale TR 24 and 48</i>	Percentuale di veicoli che hanno una Time of Repair minore o uguale a 24h o 48h.
<i>Total Time</i>	Total time per ogni dossier, corrisponde a time of repair – failure date.
<i>UpTime</i>	E' il tempo di disponibilità dei veicoli al netto delle ore di fermo
<i>Potential working hours</i>	Rappresenta il tempo di disponibilità del veicolo per il cliente
<i>Percentuale DMS</i>	Percentuale di dossiers aperti dalla fonte DMS

di adattarsi in modo flessibile alle mutevoli esigenze del mercato. La scelta accurata dei KPI, in linea con gli obiettivi strategici, contribuisce a creare un quadro di valutazione significativo, guidando l'efficacia operativa e la crescita sostenibile nel tempo. Nella tabella che segue sono stati identificati i KPIs implementati per permettere il monitoraggio dei dossiers:

3.3.5 GANTT CHART

Il GANTT chart è uno strumento grafico ampiamente utilizzato che offre una panoramica visiva del programma delle attività nel corso del tempo. Esso presenta le diverse attività del progetto disposte lungo l'asse temporale, consentendo di identificare facilmente le dipendenze tra le attività, le loro durate e i momenti chiave del processo decisionale. Questo strumento si rivela particolarmente prezioso nel contesto della business intelligence, poiché consente di coordinare le fasi di raccolta dati, analisi, implementazione delle soluzioni e verifica dei risultati, agevolando una gestione ottimizzata delle risorse e garantendo il rispetto dei tempi previsti.

Nella figura [(Fig.3.2)] che segue è rappresentato il diagramma GANTT del progetto, nel quale sono state riportate tutte le fasi e le tempistiche relative:

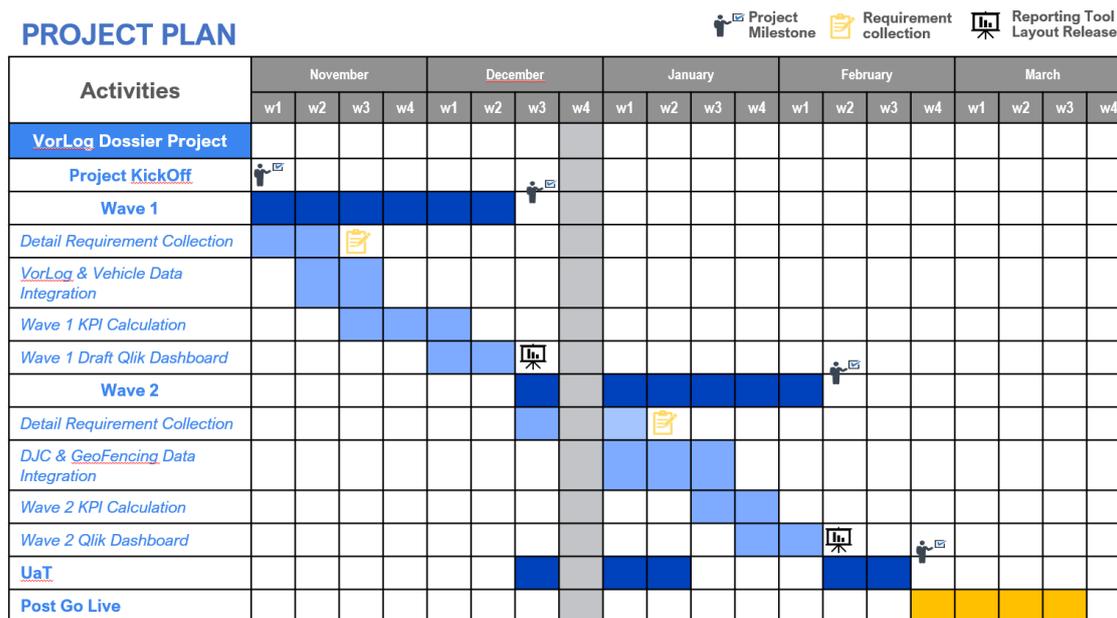


Figura 3.2: GANTT Chart progetto VORlog Dossier

Il periodo di tempo schedulato per il completamento dell'attività è di circa quattro mesi, considerando anche nella schedulazione il periodo di "UaT" e il "Post Go Live". Gli UAT, o User Acceptance Testing sono dei test che vengono effettuati a valle del rilascio applicativo; l'obiettivo principale degli UAT è valutare se il sistema soddisfa i requisiti funzionali e le esigenze degli utenti in modo efficace e in linea con

le aspettative. L'espressione "Post Go live", invece, in un'attività IT fa riferimento al periodo che segue immediatamente il completamento della fase di implementazione e all'attivazione effettiva del sistema o del servizio. Questo intervallo di tempo risulta essere cruciale e coinvolge una serie di attività e di processi volti a garantire il corretto funzionamento dell'applicazione, che gli utenti riescano ad utilizzare correttamente la dashboard e che in caso di problematiche ci sia un intervento tempestivo per ripristinare il funzionamento.

3.3.6 Architettura della soluzione proposta

L'architettura ad alto livello proposta [(Fig.3.3)] è quella inserita all'interno del documento di analisi funzionale redatto per il progetto e rappresenta la situazione relativamente alla WAVE 1; in particolare, in corrispondenza di questo momento temporale, nel Data Lake di Qlik Sense sono integrati tutti i dati relativi ai veicoli (come anagrafica e informazioni sui contratti connessi) e ai dossiers (e quindi anche i components presenti al loro interno). Successivamente, verranno integrati in Qlik anche i dati di DJC e GeoFencing provenienti da Azure. Gli elementi principali nella figura rappresentativa sono:

- I dati relativi alla DJC e Geo Fencing provengono dal cloud Azure e dovranno essere integrati successivamente in Qlik Sense;
- All'interno del Data Lake di Qlik Sense ci saranno i QVD con le informazioni relative ai Dossiers, e di conseguenza ai relativi components, al veicolo con tutte le informazioni riguardanti l'anagrafica e ai servizi di garanzia.
- Implementazione della soluzione in ambiente Qlik Sense.

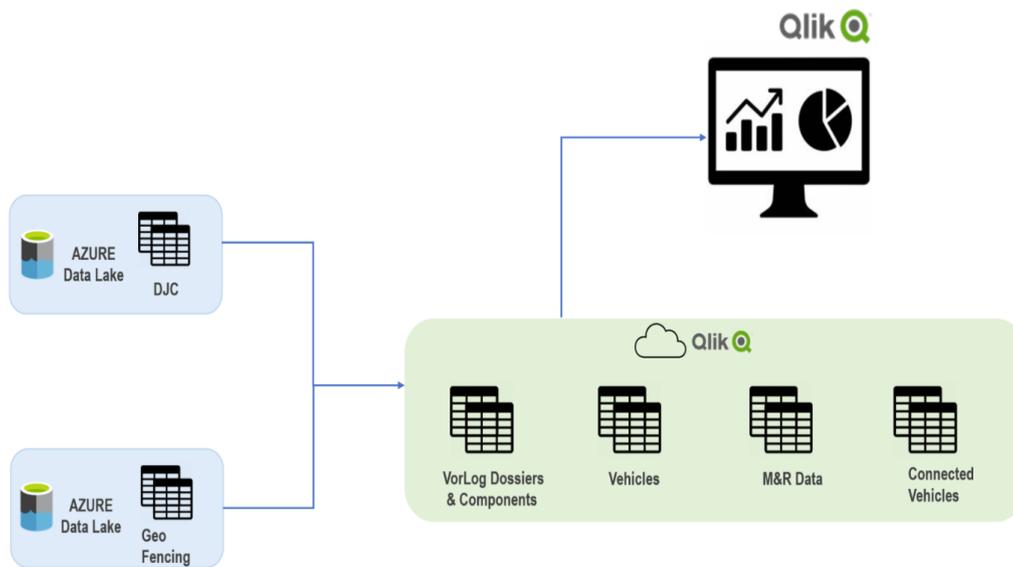


Figura 3.3: Architettura dell'app proposta

Capitolo 4

VORlog Dossier Project

Il capitolo centrale di questa tesi, intitolato "VORlog Dossier Project", costituisce il nucleo essenziale dell'elaborato, offrendo un dettagliato esame del processo di sviluppo e implementazione della dashboard commissionata da ONHIGH alla società di consulenza KM per il monitoraggio dei dossiers. In questo progetto la tecnologia di riferimento è Qlik Sense, una potente piattaforma di Business Intelligence utilizzata per analizzare e presentare in modo efficace le informazioni contenute nei fascicoli digitali che raccolgono dettagli su veicoli soggetti a guasti o fermi.

Il presente si propone di porre l'attenzione sulle fasi del processo di realizzazione della dashboard, a partire da quella iniziale di estrazione dei dati dai file sorgente, attraverso le attività di trasformazione e aggregazione, fino alla fase finale di presentazione dell'applicazione completa. In tal senso verrà esplorato il ruolo cruciale di Qlik Sense nel consentire una visualizzazione chiara e interattiva delle informazioni contenute nei dossier, consentendo un monitoraggio dettagliato delle attività legate ai veicoli, oltre allo sviluppo di KPIs fondamentali. L'analisi comprenderà un'overview delle sfide affrontate, delle scelte metodologiche adottate e dei risultati ottenuti, fornendo una panoramica completa sul significativo contributo che il "VORLog Dossier Project" apporta al contesto della tesi.

4.1 Analisi e sviluppo

Come precedentemente accennato nel paragrafo relativo a QlikSense, l'esecuzione del progetto comporta l'adempimento di passaggi chiaramente definiti, i quali devono essere osservati per garantire la realizzazione di un progetto robusto e automatizzato, capace di mantenere la propria integrità nel tempo senza richiedere ulteriori interventi, a meno che non siano necessari a causa di modifiche o sviluppi volontari. In linea generale, le fasi stabilite per sviluppare un'app Qlik Sense comprendono:

- **Progettazione del modello dati:** questa fase ha come obiettivo l'analisi delle fonti dati e dell'output finale, mirata a determinare la struttura tabellare più idonea per la buona riuscita del progetto.
- **Estrazione dei dati (ETL1):** è il processo di prelevamento dei dati da diverse fonti disponibili e successivo caricamento sul Data Lake.
- **Trasformazione dei dati (ETL2):** in questo step avviene l'attività di trasformazione, aggregazione e normalizzazione dei dati, al fine di sviluppare un modello dati conforme alle esigenze progettuali.
- **Presentazione:** consiste nella creazione dell'app Qlik Sense, la quale legge i dati precedentemente elaborati e li utilizza all'interno degli oggetti grafici per l'analisi.

Nell'ambito del progetto VorLog Dossier l'ETL1 non è previsto in quanto lo sviluppo commissionato alla società KM riguarda esclusivamente lo sviluppo dell'ETL2 e del livello di Presentation; questo perchè all'interno dell'azienda ONHIGH è presente un organo dedicato, chiamato CoE (Center of Excellence), che si occupa del caricamento dei dati da diverse fonti sul Data Lake che viene utilizzato da Qlik. Affinchè un dato possa essere utilizzato su Qlik, e quindi reso disponibile per la creazione di report e dashboard, è necessaria un'azione da parte del CoE, il quale dopo aver ricevuto il Solution Design, un documento in cui vengono specificate le

connessioni e i dati da trasferire, provvede al porting delle informazioni nell'ambiente richiesto. Per poter avere una visione chiara sull'origine dei dati, verrà fornita un'overview sulla provenienza dei diversi dati, analizzando le diverse fonti ad alto livello.

4.2 Progettazione del Data Model

Prima dell'inizio della parte relativa allo sviluppo dell'applicazione è stato necessario porre l'attenzione sulla struttura che il modello dati avrebbe dovuto assumere. Questa parte dell'analisi è cruciale affinché lo sviluppo della dashboard rispecchi le esigenze e i requisiti richiesti dal cliente; in tal senso, sono stati creati due data model che sono uno il proseguo dell'altro; il primo [(Fig.4.1)], è quello relativo alla WAVE I del progetto, e contiene le informazioni sui Dossiers, i Components, i veicoli, i contratti, i veicoli connessi e la modellazione di un calendario per definire e utilizzare in maniera appropriata i dati in formato DATA.

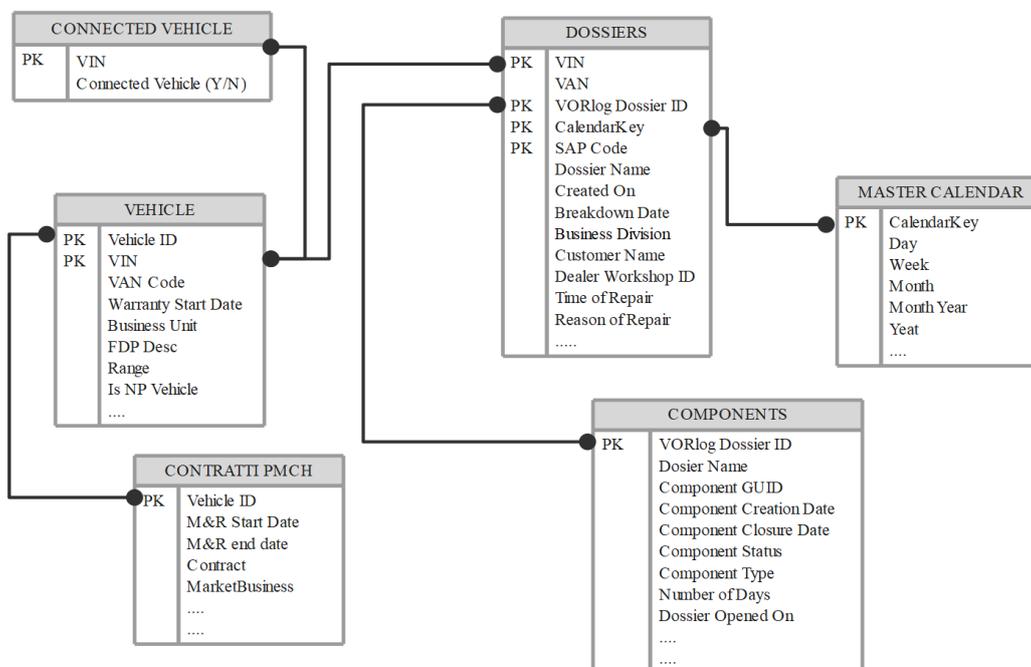


Figura 4.1: Data Model sviluppato per la WAVE I

Nella WAVE II, invece, sono stati integrati nel precedente data model le informazioni aggiuntive relative ai dati provenienti da DJC e GeoFencing [(Fig.4.2)].

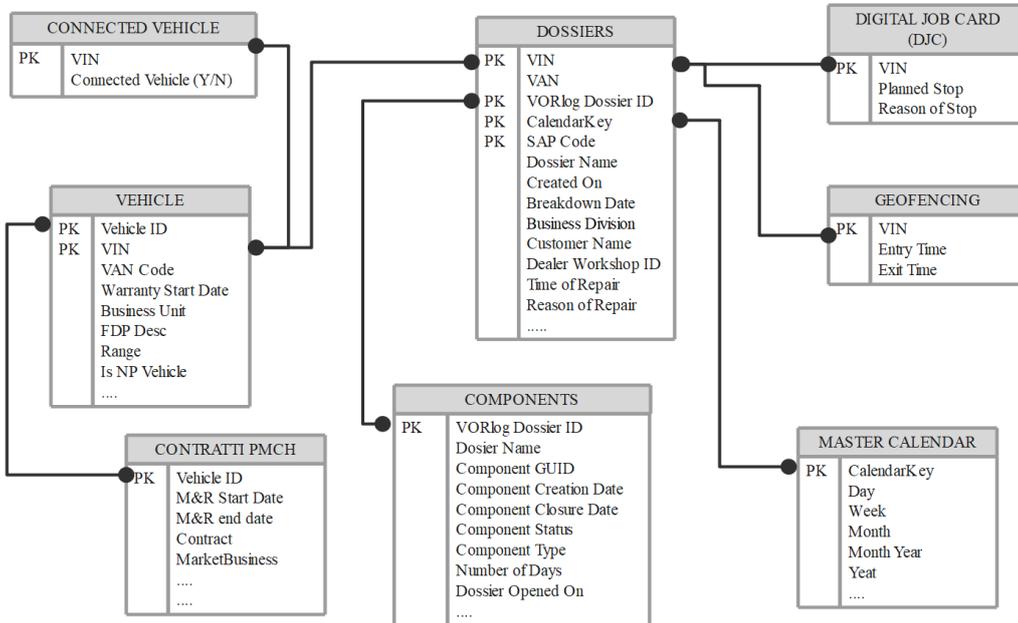


Figura 4.2: Data Model sviluppato per la WAVE II

4.3 Integrazione dei dati su Qlik Sense

La fase di ingestione dei dati costituisce il fondamento cruciale di qualsiasi progetto di business intelligence e data analytics. Rappresenta il punto di partenza vitale in cui vengono identificate, raccolte e integrate le sorgenti di dati eterogenee provenienti da diverse fonti aziendali. La definizione accurata delle sorgenti dati è un passo imprescindibile, poiché la qualità e la coerenza delle informazioni raccolte influenzano direttamente l'efficacia dell'intero sistema analitico. Tuttavia, questa fase è spesso accompagnata da notevoli sfide, come la gestione della complessità di dati provenienti da sistemi eterogenei, la varietà di formati e la necessità di garantire l'integrità e la sicurezza delle informazioni. L'integrazione dei dati richiede una strategia ben pianificata e l'adozione di strumenti tecnologici avanzati per assicurare un flusso efficiente e coerente di dati, garantendo così una base solida per le attività analitiche e decisionali nell'ambito del business intelligence. Nell'ambito del progetto VORlog Dossier l'ingestione dei dati [(Fig.4.3)] risulta essere particolarmente semplificata, poiché nonostante i dati provengono da fonti diverse, sono tutti disponibili nel Data Lake della società, ovvero una base di dati localizzata su una macchina fisica che consente agli utenti di accedere, condividere e collaborare su app e dashboard Qlik. Per questi motivi, il caricamento in Qlik Sense avverrà attraverso il comando di LOAD andando a prelevare le informazioni da dei QVD già presenti tramite una connessione. In questo paragrafo verrà illustrato il processo di ingestione dei dati ad alto livello, evidenziando la tipologia di sorgenti dato. I file contenenti i dossiers e i relativi components provengono da un altro Data Lake che chiameremo EDS attraverso il flusso ASIST che non viene gestito direttamente dalla società KM, i dati relativi all'anagrafica dei veicoli provengono dal Dataware House NEWBI in gestione a KM, mentre anche per i dati relativi al GeoFancing e alle DJC che provengono dal cloud di Azure la gestione non è in capo a KM. Nella figura che segue è raffigurata l'architettura dell'applicazione.

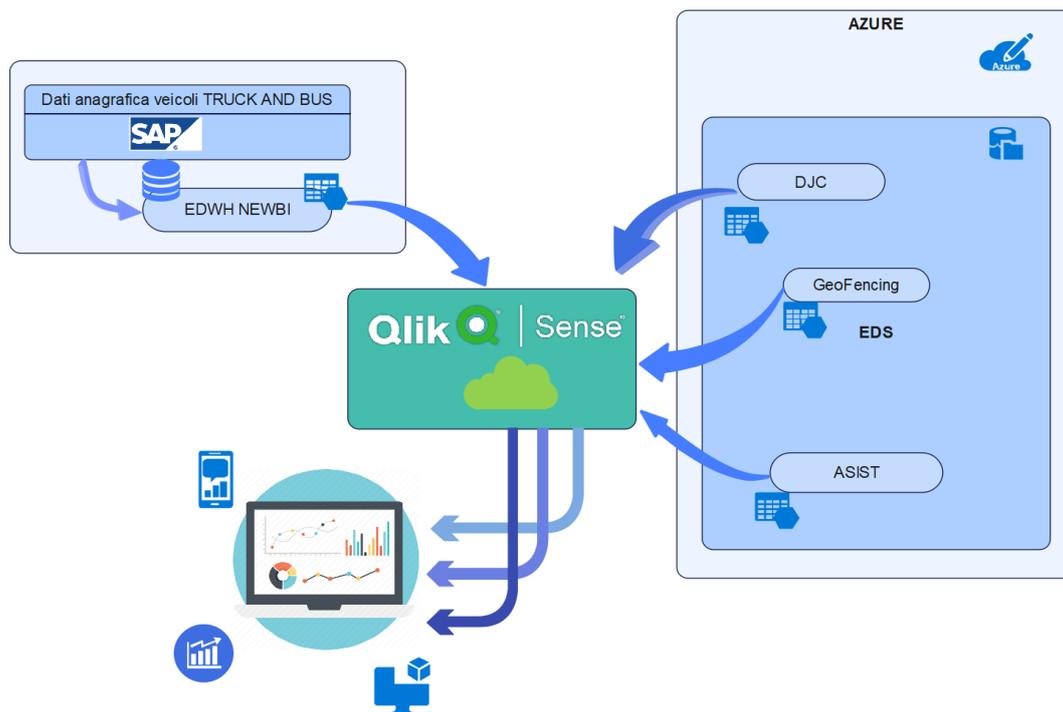


Figura 4.3: Ingestion dati in ambiente Qlik Sense

Come specificato nel paragrafo di Analisi, la parte relativa all'integrazione dei dati in ambiente Qlik non verrà trattata in questo elaborato poichè fuori il perimetro del progetto. La parte di caricamento dati, e quindi l'ETL1, sarà gestita da un team interno dell'azienda ONHIGH.

4.3.1 Integrazione dei dati da NEWBI

I dati relativi al singolo veicolo provengono da alcune tabelle contenute nel Data Warehouse NEWBI (New Business Intelligence), un DWH in gestione alla società KM per ONHIGH, in quanto tale viene fornita una descrizione dell'architettura e del funzionamento. In particolare, da questo database provengono le informazioni di anagrafica del veicolo e le informazioni sui contratti di garanzia, laddove fossero stati stipulati. Di seguito un'overview sul DWH, concentrando l'attenzione sul suo funzionamento e sulle principali caratteristiche.

Data Warehouse NEWBI

NEWBI è un Data Warehouse di tipo three-layer [(Fig.4.4)]: troviamo come fonte dati un sistema ERP basato su tecnologia SAP. Quest'ultimo costituisce la fonte primaria dei dati operativi di ONHIGH per le relative macro-region: EMEA (Europe, Middle East and Africa), APAC (Asia Pacific) e LATAM (Latin America). I dati operativi sono a loro volta collegati ai livelli superiori tramite i cosiddetti estrattori SAP. Questi ultimi agiscono parzialmente da ETL tools: solo una porzione dei dati, sia in termini di record, sia di struttura stessa delle tabelle dello schema relazionale da cui il Data Warehouse NEWBI attinge, viene importata tramite questi estrattori. Si noti anche l'ingestion a sistema di tabelle manuali con cui gli utenti hanno la possibilità di effettuare data entry: queste ultime sfruttano la tecnologia web Apparo. NEWBI è un DWH che risiede in una macchina fisica, per cui su un hardware della società ONHIGH.

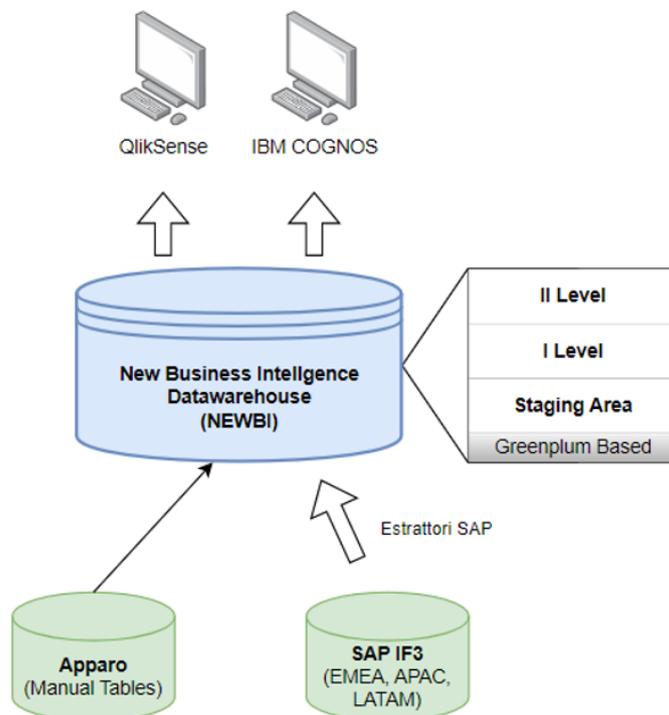


Figura 4.4: Struttura del Data Warehouse NEWBI

Come fase successiva, all'interno delle strutture del Data Warehouse, si

individua un livello di data staging che svolge una replicazione bidirezionale dei dati provenienti dalle fonti operative, ovvero tabelle esterne, estratte mediante specifici estrattori SAP. Le attività di pulizia e riconciliazione sono eseguite mediante stored procedures, ovvero istruzioni di compilazione PostgreSQL, atte a garantire una riconciliazione accurata dei dati al livello successivo. Tali procedure rappresentano il primo passo delle regole aziendali necessarie per elaborare in modo appropriato i dati.

Il primo livello (I Level) costituisce il Reconciled Data Layer di NEWBI. In questa fase, i dati vengono sottoposti a un processo di pulizia e conservazione, rispettando la granularità minima necessaria. Questo strato è di vitale importanza, poiché la struttura del database di SAP non agevolerebbe l'analisi dei dati aziendali. La naming convention adottata da SAP e il modello di dati stesso risulterebbero difficilmente utilizzabili per la creazione delle tabelle dei fatti e delle dimensioni ai livelli successivi.

Il secondo livello (II Level) costituisce il livello del Data Warehouse, dove i dati precedentemente riconciliati vengono sottoposti a un ulteriore processo di elaborazione attraverso stored procedures appositamente progettate. Successivamente, vengono memorizzati nelle tabelle dimensionali e dei fatti conformemente alla progettazione del modello dati. Il secondo livello, dunque, fornisce tutte le informazioni essenziali, incluse le tabelle dei fatti e delle dimensioni, per supportare la corretta generazione di analisi e report accessibili tramite le strati applicativi superiori. Due delle principali tecnologie impiegate per la creazione di report e dashboard sono IBM Cognos e QlikSense.

Da questo Data Warehouse vengono prelevate le tabelle contenenti informazioni relative all'anagrafica dei veicoli come ad esempio il VIN che rappresenta il codice identificativo del telaio, il VAN, ovvero il codice identificativo del motore, il mercato di riferimento del Dealer e del Customer, il modello del veicolo e altre informazioni chiave e la tabella contenente tutte le informazioni.

Naming convention NEWBI

Il database di NEWBI si distingue per l'implementazione di una naming convention che è stata definita durante la fase di sviluppo della soluzione. Di seguito, vengono presentate le scelte significative relative alla nomenclatura per i vari oggetti e istanze che compongono il database. Le tabelle di staging [(Fig.4.5)] sono collocate all'interno dello schema `dwrs00_sa`, rappresentante la zona di staging, e la loro denominazione segue uno schema organizzato secondo le seguenti convenzioni:

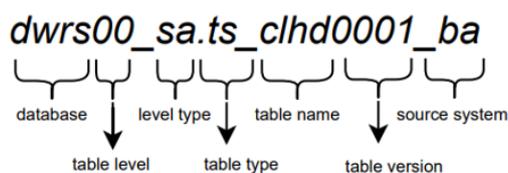


Figura 4.5: Naming convention per le tabelle di Staging

I primi quattro caratteri dello schema delineano il database di riferimento, mentre i due successivi indicano il livello del Data Warehouse a cui fa capo. La specifica tipologia di livello è rappresentata dai due caratteri successivi, chiaramente identificabili come Staging Area (SA). Il nome specifico di ciascuna tabella inizia con due caratteri che ne sanciscono la tipologia come "table of staging (ts)". Seguono poi quattro caratteri che indicano il nome della tabella, seguiti da ulteriori quattro caratteri numerici che specificano la versione della tabella stessa. Gli ultimi caratteri forniscono informazioni sul sistema sorgente di provenienza dei dati contenuti nella tabella. Per quanto riguarda le tabelle delle dimensioni (Fig.4.6), ospitate nello schema `dwrs02_dim`, la loro naming convention è più chiara e semplificata:

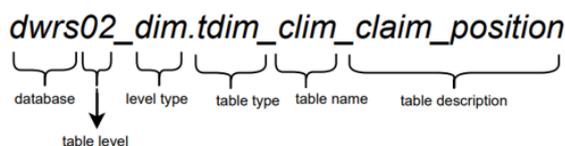
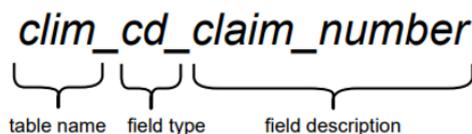


Figura 4.6: Naming convention per tabelle di secondo livello

I primi quattro caratteri dello schema identificano il database di appartenenza, mentre i due successivi indicano il livello specifico del Data Warehouse cui lo schema fa riferimento. La natura del livello è indicata dai successivi caratteri, identificando la tipologia come Dimension (DIM); il nome unico assegnato a ciascuna tabella, invece, inizia con quattro caratteri iniziali distintivi per la tipologia specifica della tabella stessa; ad esempio, "tabella delle dimensioni (tdim)". Oltre alle tabelle delle dimensioni è possibile trovare altri tipi come "tabella dei fatti (tfct)", "tabella di storicizzazione (thst)", "tabella di working (twrk)" e "tabella di relazione (trel)". Proseguendo la descrizione del nome, sono presenti quattro caratteri che indicano il nome specifico della tabella, seguiti da una serie variabile di caratteri che forniscono una descrizione dettagliata del contenuto della tabella. Infine, i nomi dei campi [(Fig.4.7)] all'interno delle tabelle rispecchiano appieno le convenzioni precedentemente delineate:



basati su cloud, tra cui elaborazione, analisi, archiviazione e infrastruttura di rete. Gli utenti hanno la flessibilità di selezionare e utilizzare questi servizi per lo sviluppo e la scalabilità di nuove applicazioni, nonché per l'esecuzione di applicazioni esistenti attraverso l'ambiente del cloud. [13]

L'obiettivo della nascita del Data Lake EDS in ONHIGH è quello di collezionare in un unico spazio i dati di tutte le sorgenti; all'interno di questo data lake sono presenti le informazioni chiave per l'implementazione della dashboard prime fra tutte le informazioni relative ai dossiers e agli specifici components contenuti al loro interno provengono dal flusso ASIST e vengono immagazzinate in EDS. Sempre all'interno di EDS sono presenti le informazioni che verranno utilizzate per la seconda parte del progetto, ovvero i dati relativi alle DJC e alla GeoFencing; questi dati verranno utilizzati principalmente per effettuare dei contro-check e valutare la consistenza delle informazioni contenute all'interno dei dossiers.

Integrazione dei dati Dossiers e Components

All'interno della società ONHIGH le informazioni sulla qualità del prodotto vengono raccolte da un CMS (Content Management System) e poi vengono inviate tramite il flusso ASIST al data lake EDS. Il flusso ASIST nonostante appartenesse ad un sistema esterno, all'interno dell'ambito ONHIGH, svolge un ruolo cruciale nella gestione dei diversi processi legati alla qualità. Esso racchiude e conserva tutte le informazioni relative ai difetti segnalati dai Product Specialist nell'area Quality. Questi dati rappresentano gli inconvenienti generati da veicoli in circolazione, che vengono registrati direttamente sul campo dal personale operativo o attraverso il processo di creazione delle segnalazioni da parte dei rivenditori. Coloro che documentano i difetti dei prodotti sono gli auditor interni, responsabili di inserire tali informazioni nel sistema durante le diverse fasi di audit interno. Lo scopo principale di questo sistema è creare un quadro accurato dei difetti e dei veicoli associati, che sia il più fedele possibile alla realtà. Il valore aggiunto si manifesta nella capacità dei colleghi del dipartimento Quality di identificare in anticipo i difetti e anticipare

potenziali flussi futuri di reclami, sia in termini di tipologia che di volumi. ASSIST invia estrazioni di file .csv contenenti le informazioni relative ai dossiers e ai components. I dati caricati verranno poi elaborati e trasformati nell'ETL2 successivamente per adattarli alle esigenze del Business.

Integrazione del dato Digital Job Card (DJC)

La Digital Job Card è una scheda tecnica digitale che viene aperta dal dealer per il cliente finale e che permette di mappare gli interventi di riparazione ai veicoli. Un fermo veicolo può essere causato da due tipologie di problematiche: la prima è relativa ad un guasto che si verifica mentre il veicolo è in movimento (tale casistica viene mappata grazie al campo `breakdownDate`, ovvero la data in corrispondenza della quale si è verificato il guasto), e che quindi preclude la possibilità dello stesso di continuare a circolare, e la seconda relativa ad un guasto che non preclude la circolazione per cui viene fissato un appuntamento in officina per la futura riparazione. A tal proposito, una delle funzionalità più rilevanti per l'analisi incrociata di consistenza dati nel progetto VORLog è quella relativa all'inserimento degli appuntamenti pianificati per la riparazione dei veicoli, le cui informazioni sono contenute nella Digital Job Card che, permettono attraverso la verifica degli stessi di valutare se effettivamente un veicolo è stato riparato presso l'officina e quanto tempo è rimasto fermo al suo interno. Alla base della DJC è presente un tool web che attraverso un'interfaccia grafica consente l'inserimento di tutti i dati. Lo storage di tutte queste informazioni viene eseguito su un data base Oracle presente nel tool web e successivamente vengono inviate al data lake EDS.

Integrazione del dato GeoFencing

Il dato relativo al GeoFencing fornisce delle informazioni di origine telematica che permettono di sapere se il veicolo è presente presso un determinato dealer, ovvero all'interno di un'officina; in particolare, fa riferimento a tutti i dati rilevati automaticamente in ingresso e uscita dall'officina. L'obiettivo alla base dell'utilizzo di

questa informazione è quello di integrare i Time of Repair quando il dato da VORlog è inattendibile o mancante, è importante poter risalire alla fonte della data di chiusura utilizzata

4.4 Trasformazione dei dati - ETL2

L'ETL2 assume un ruolo fondamentale nel progetto ed è un livello che si interpone tra l'ETL1 che è il layer in cui le tabelle utili al progetto vengono importate all'interno di Qlik senza subire alcuna elaborazione e il layer di Presentationn, nel quale viene riorganizzata la struttura del codice in maniera più snella. In questo layer, quindi, risulta esserci la fase più delicata dello sviluppo dell'app, poichè vengono messe a punto le trasformazioni e le elaborazioni dei dati precedentemente caricati per soddisfare e implementare le richieste del Business. Nel seguente paragrafo verranno spiegati i diversi ETL2 creati per la manipolazione dei dati a back-end, nello specifico sono stati creati diversi script a seconda del dato da elaborare. In alcuni casi si tratterà di portare il dato dal layer ETL1 al layer ETL2 lasciando l'informazione inalterata, mentre in altri casi verranno effettuate delle trasformazioni. L'analisi, a causa della mole di dati, verterà principalmente sulle porzioni di codice ritenute di particolare rilevanza ai fini dell'implementazione della dashboard, omettendo il caricamento di alcuni campi che non sono fondamentali per la comprensione.

Nel corso del seguente paragrafo sono stati utilizzati comandi Qlik per elaborare i dati. Per una comprensione dettagliata di questi comandi e delle relative applicazioni nel contesto del progetto si rimanda all'*Appendice* dell'elaborato, nel quale vengono fornite spiegazioni approfondite sul codice Qlik.

4.4.1 ETL2 - Vehicle

In questo ETL2 vengono effettuate diverse operazioni di manipolazione del dato, i dati vengono presi da diversi QVD attraverso delle JOIN su dei precisi campi che

rappresentano per le rispettive tabelle le chiavi primarie.

```

VAN_VHLS_tmp:
load
    pvan_cd_van_code ,
    pvan_cd_vin_code ,
    pvan_dt_warranty_start_date ,
    pvan_id_makt_count_mkt_transit ,
    pvan_id_res_customer ,
    pvan_id_warranty_start_date ,
    pvan_id_vcb ,
    pvan_id_vcl
from
[lib://FOLDER_XXX/ETL01_EDWH/St_TDIM_VEHCILE.qvd](qvd);

NoConcatenate
Vehicles_Source:
load
FirstValue(pvan_cd_van_code) as pvan_cd_van_code ,
FirstValue(pvan_cd_vin_code) as Vin ,
FirstValue(pvan_dt_warranty_start_date) as pvan_dt_warranty_start_date ,
FirstValue(pvan_id_res_customer) as pvan_id_res_customer ,
FirstValue(pvan_id_vcb) as pvan_id_vcb ,
FirstValue(pvan_id_vcl) as pvan_id_vcl

resident VAN_VHLS_tmp
group by pvan_cd_vin_code
order by pvan_id_makt_count_mkt_transit desc;

drop table VAN_VHLS_tmp;

join(Vehicles_Source)
VCB_VHLS:
load
    pvcb_cd_busu_business_unit ,
    if(pvcb_cd_busu_business_unit = 01,'Light',
        if(pvcb_cd_busu_business_unit = 02,'Medium',
            if(pvcb_cd_busu_business_unit = 03,'Heavy'))) as [Vhl_Range],
    pvcb_cd_fdp_code ,
    pvcb_ds_business_unit as [range],
    pvcb_ld_cmm_gm_ldes ,
    pvcb_ld_fdp_desc as [FDP desc],
    wildmatch(pvcb_ld_fdp_desc,'*NP*') as [is NP Vehicle],
    pvcb_id_vcb as pvan_id_vcb ,
    pvcb_ds_mdl_grp as [Model Group],
    pvcb_ld_vcb_ldes as [Model description],
    pvcb_cd_vcb_code as [VCB Code]
from

```

```
[lib://FOLDER_XXX/ETL01_EDWH/St_TDIM_VCB_VEHICLE.qvd] (qvd)
where match(pvcb_cd_language, 'EN')>0;

NoConcatenate
Customer_TMP:
load
    cust_cd_cust_code_isis,
    cust_cd_cust_code_sap,
    cust_ds_cust_nm as [Customer Name],
    cust_id_customer,
    cust_ds_cust_city,
    cust_cd_country as [Market],
from
[lib://FOLDER_XXX/ ETL01_EDWH/St_TDIM_CUSTOMER.qvd] (qvd);

join(Vehicles_Source)
load
    cust_cd_cust_code_isis,
    cust_cd_cust_code_sap,
    cust_cd_language,
    [Customer Name],
    [Market],
    cust_ds_cust_nm_2,
    cust_id_customer as pvan_id_res_customer

resident Customer_TMP;

drop table Customer_TMP;
```

4.4.2 ETL2 - Dossiers e Components

```
load *
from
[lib://FOLDER_XXX/ETL01_EDSINTEGGRATION/St_asist_vorlog_dossier.qvd] (qvd)
where
len(clu_vin)= 17;

Components_Source:
load *
from
[lib://FOLDER_XXX/ETL01_EDSINTEGGRATION/St_asist_vor_components.qvd] (qvd)
where
exists(clu_vorlogdossierid);
```

Questi due comandi permettono di prelevare tutti i dati contenuti nelle tabelle di Raw Data sia per i Dossiers che per i Components, ovvero il dato in formato grezzo proveniente dal flusso sorgente di ASIST con degli opportuni filtri inseriti nella clausola WHERE. Questi due caricamenti saranno utilizzati per creare nel livello di Presentation due fogli per la visualizzazione del dato grezzo, il motivo alla base di questa richiesta, da parte del Business, nella richiesta di voler visualizzare a front-end i dati provenienti dalla sorgente senza alcuna pre-manipolazione, questo sarà utile agli utenti per poter effettuare controlli di consistenza del dato. Per il caricamento dei campi relativi al QVD dei Components è stato scelto come approccio il seguente:

```

NoConcatenate
Components:
// ANS components
LOAD
    text(clu_vorlogdossierid) as clu_vorlogdossierid
    , clu_vorlogcomponentsid as component_guid
    , clu_clojuredate as component_closure_date
    , clu_componenttypecodename as component_type
    , clu_ansid_clu_creationdate as component_creation_date
    , clu_ansid as component_guid_2
    , statuscodename as component_status
    , clu_ansid_statuscodename as component_status_wo_canceled
    , clu_vorlogdossieridname as dossier_name
    , coalesce(clu_clojuredate, today()) - createdon as number_of_days
    , clu_networkid_clu_dateofopeningthedossier as dossier_opened_on
    , createdon as component_opened_on

Resident Components_Source
Where clu_componenttypecodename = 'ANS';
// Network components
Concatenate LOAD
    text(clu_vorlogdossierid) as clu_vorlogdossierid
    , clu_vorlogcomponentsid as component_guid
    , clu_clojuredate as component_closure_date
    , clu_componenttypecodename as component_type
    , clu_networkid_createdon as component_creation_date
    , clu_networkid as component_guid_2
    , statuscodename as component_status
    , clu_networkid_statuscodename as component_status_wo_canceled
    , clu_vorlogdossieridname as dossier_name
    , coalesce(clu_clojuredate, today()) - createdon as number_of_days
    , clu_networkid_clu_dateofopeningthedossier as dossier_opened_on

```

```

    , createdon as component_opened_on

Resident Components_Source
Where clu_componenttypecodename = 'Network';
// VOR PartOrder
Concatenate LOAD
    text(clu_vorlogdossierid) as clu_vorlogdossierid
    , clu_vorlogcomponentsid as component_guid
    , clu_closuredate as component_closure_date
    , clu_componenttypecodename as component_type
    , clu_vorpartorderid_createdon as component_creation_date
    , clu_vorpartorderid as component_guid_2
    , statuscodename as component_status
    , clu_vorpartorderid_statuscodename as component_status_wo_canceled
    , clu_vorlogdossieridname as dossier_name
    , coalesce(clu_closuredate, today()) - createdon as number_of_days
    , clu_networkkid_clu_dateofopeningthedossier as dossier_opened_on
    , createdon as component_opened_on

Resident Components_Source
Where clu_componenttypecodename = 'VOR Part Order';
// Request report
Concatenate LOAD
    text(clu_vorlogdossierid) as clu_vorlogdossierid
    , clu_vorlogcomponentsid as component_guid
    , clu_closuredate as component_closure_date
    , clu_componenttypecodename as component_type
    , Null() as component_creation_date
    , clu_requestreportid as component_guid_2
    , statuscodename as component_status
    , Null() as component_status_wo_canceled
    , clu_vorlogdossieridname as dossier_name
    , coalesce(clu_closuredate, today()) - createdon as number_of_days
    , clu_networkkid_clu_dateofopeningthedossier as dossier_opened_on
    , createdon as component_opened_on

Resident Components_Source
Where clu_componenttypecodename = 'Request report';
// PartOrder
Concatenate LOAD
    text(clu_vorlogdossierid) as clu_vorlogdossierid
    , clu_vorlogcomponentsid as component_guid
    , clu_closuredate as component_closure_date
    , clu_componenttypecodename as component_type
    , clu_partorderid_createdon as component_creation_date
    , clu_partorderid as component_guid_2
    , statuscodename as component_status

```

```

, clu_vorpartorderid_statuscodename as component_status_wo_canceled
, clu_vorlogdossieridname as dossier_name
, coalesce(clu_closuredate, today()) - createdon as number_of_days
, clu_networkid_clu_dateofopeningthedossier as dossier_opened_on
, createdon as component_opened_on

Resident Components_Source
Where clu_componenttypecodename = 'Part Order';

```

Utilizzare la funzione Concatenate per popolare la tabella dei Componentes ha agevolato l'ottimizzazione del processo di caricamento dati all'interno dell'app, poiché ad ogni iterazione Qlik leggeva i medesimi campi destinati alla tabella Components_Source adattando la clausola WHERE in base al component specifico.

```

NoConcatenate
Dossiers:
load
*,
if([TR-TA in hours]<= 48, 1) as [TR <= 48 H],
if([TR-TA in hours]<= 24, 1) as [TR <= 24 H],
if([TR-TA in hours]>48, 1) as [TR>48 H];
load
text(clu_vorlogdossierid) as clu_vorlogdossierid,
clu_brandidname,
ConvertToLocalTime(timestamp(clu_breakdowndate, 'DD/MM/YYYY hh:mm'), 'Rome') as
clu_breakdowndate,
clu_busflag,
clu_businessdivision,
clu_businessdivisionname,
clu_contactidname,
clu_customername,
ConvertToLocalTime(timestamp(clu_etr, 'DD/MM/YYYY hh:mm'), 'Rome') as clu_etr,
ConvertToLocalTime(timestamp(clu_lastmodifiedrunningvehicledate, 'DD/MM/YYYY hh:
mm'), 'Rome') as clu_lastmodifiedrunningvehicledate,
num(clu_name) as clu_name,
ConvertToLocalTime(timestamp(clu_repairstartdate, 'DD/MM/YYYY hh:mm'), 'Rome') as
clu_repairstartdate,
clu_runningvehicle,
clu_sourcecode,
clu_sourcecodename,
ConvertToLocalTime(timestamp(clu_submitteddate, 'DD/MM/YYYY hh:mm'), 'Rome') as
clu_submitteddate,
ConvertToLocalTime(timestamp(clu_tr, 'DD/MM/YYYY hh:mm'), 'Rome') as clu_tr,
if(UPPER(statuscodename) = 'CLOSED' and not(isnull(clu_tr)) and not(isnull(
clu_breakdowndate))),

```

```

        if(Timestamp(clu_tr,'DD/MM/YYYY hh:mm')-timestamp(clu_breakdowndate,'DD/MM/
            YYYY hh:mm')>0,
        (Timestamp(clu_tr,'DD/MM/YYYY hh:mm')-timestamp(clu_breakdowndate,'DD/MM/
            YYYY hh:mm'))* 24,null())      as [TR-TA in hours],
clu_unitid,
clu_unitidname,
ConvertToLocalTime(timestamp(clu_uptimeendvaliditydate,'DD/MM/YYYY hh:mm'),'
    Rome') as clu_uptimeendvaliditydate,
clu_uptimeguarantee,
if(isnull(clu_uptimeguarantee), null(),
    if(match(clu_uptimeguarantee,0),'N','Y')) as flgUptime,
    ConvertToLocalTime(timestamp(clu_uptimestartvaliditydate,'DD/MM/YYYY hh:mm'
        ),'Rome')      as clu_uptimestartvaliditydate,
clu_usage,
clu_van,
clu_vin as Vin,
clu_warrantystartdate,
clu_breakdowndate - clu_warrantystartdate as vehicle_age,
Num(ConvertToLocalTime(timestamp(createdon,'DD/MM/YYYY hh:mm'),'Rome')) as
    createdon,
floor(ConvertToLocalTime(timestamp(createdon,'DD/MM/YYYY hh:mm'),'Rome')) as
    createdon_calendarKey,
ConvertToLocalTime(timestamp(modifiedon,'DD/MM/YYYY hh:mm'),'Rome') as
    modifiedon,
ConvertToLocalTime(timestamp(overriddencreatedon,'DD/MM/YYYY hh:mm'),'Rome') as
    overriddencreatedon,
statuscode,
statuscodename,
clu_dealerworkshopidyominame,
if((mixmatch(clu_usageunitofmeasurementname,'KM')= 1 and num(clu_usage)<= 1000)
    or mixmatch(clu_usageunitofmeasurementname,'Miles')= 1 and num(clu_usage
        )<= 621.371),1) as [DoxLessThen1000Km],
statuscodename as [statusVorLogDossier],
year(date(clu_warrantystartdate,'DD/MM/YYYY')) as [WSD_Year]
Vin&date(clu_breakdowndate,'DD/MM/YYYY') as Key_Vorlog

resident Dossier_Source;

left join (Dossiers)
Load
    clu_vorlogdossierid,
    max(component_closure_date) as component_max_closure
Resident Components
group by clu_vorlogdossierid;

left join(Dossiers)
FlgANS:

```

```

LOAD
    text(clu_vorlogdossierid) as clu_vorlogdossierid,
    IF(mismatch(FirstValue(clu_ansid_statuscode),'1')=1, 'N',
        IF(mismatch(FirstValue(clu_ansid_statuscode),'2')=1, 'Y',
            IF(isnull(FirstValue(clu_ansid_statuscode)) or FirstValue(
                clu_ansid_statuscode) = '', null()))          as FlgStatusAns
resident Components_Source
Group By clu_vorlogdossierid

left join(Dossiers)
FlgVORParts:
LOAD
    text(clu_vorlogdossierid) as clu_vorlogdossierid,
    IF(mismatch(FirstValue(clu_vorpartorderid_statuscode),'1')=1, 'N',
        IF(mismatch(FirstValue(clu_vorpartorderid_statuscode),'2')=1, 'Y',
            if(isnull(FirstValue(clu_vorpartorderid_statuscode)) or FirstValue(
                clu_vorpartorderid_statuscode) = '', null()))          as
                FlgStatusVorParts
resident Components_Source
Group By clu_vorlogdossierid

left join (Dossiers)
Load
    Vin,
    Vhl_Range
resident Vehicles_Source;

drop field Vhl_Range from Vehicles_Source;

qualify *;

Dossier_RAW:
Load *
Resident Dossier_Source;

unqualify *;

qualify *;

Components_RAW:
Load *
Resident Components_Source;

unqualify *;

drop tables Dossier_Source;

```

```
Drop Table Components_Source;
```

Le funzioni `qualify *` e `unqualify *` sono state utilizzate per poter rinominare tutti i campi presenti nelle tabelle `Dossier_Source` e `Components_Source`, il motivo di tale scelta sta nella creazione dei fogli dedicati ai dati provenienti dal flusso sorgente. Questa funzione permette di rinominare tutti i campi della tabella selezionata aggiungendo un prefisso ad ogni campo, questo passaggio risulta essere essenziale, poichè Qlik in caso di più campi uguali in diverse tabelle crea delle chiavi sintetiche, le quali non permettono di gestire al meglio il Data Model. All'interno di questo ETL2 è stata anche implementata una logica per la creazione di due nuovi flag utili per applicare dei filtri (`FLAG_SP` e `FLAG_WSD`) nella dashboard, la logica utilizzata è stata quella di effettuare una `LEFT JOIN` con la tabella `Dossiers` inserendo tutti i records che obbediscono alla `WHERE` condition e per ognuno di essi aggiungere un campo relativo al flag valorizzato con '1', così facendo viene semplificata la creazione dei filtri.

```
left join(Dossiers)
VOR_SP:
load
    clu_name,
    1 as FLAG_SP
resident Dossiers
where wildmatch(Vhl_Range,'heavy','medium','light')>0 and (DoxLessThen1000Km
    <>'1' and (IsNull(FlgStatusAns) or [FlgStatusAns]='' or [FlgStatusAns]='NULL')
    and (IsNull(FlgStatusVorParts) or [FlgStatusVorParts]='' or [
        FlgStatusVorParts]='NULL')
    and coalesce([TR-TA in hours], 0) <= 10*24
    and ((wildmatch(statusVorLogDossier,'Closed')=1 and [TR-TA in hours]>0) or
        wildmatch(statusVorLogDossier,'Open')=1);

left join(Dossiers)
VOR_WSD:
load
    clu_name,
    1 as FLAG_WSD
resident Dossiers
where WSD_Year >= '2016';
```

4.4.3 ETL2 - Contratti PMCH

Nel seguente script sono riportate le informazioni di testata relative ai contratti ed è stato implementato il campo M&R (Maintenance and repair), il quale sarà utilizzato nella dashboard come filtro di analisi.

```

PMCH:
LOAD
    pmch_cd_stu,
    pmch_fl_pmc_extended_wrt,
    pmch_id_makt_contract_market,
    pmch_id_pvan_vcl      as pvan_id_vcl,
    text(date(pmch_id_time_end_date-'2415019','YYYY-MM-DD'))
                                     as pmch_end_date,
    text(date(pmch_id_time_start_date-'2415019','YYYY-MM-DD'))
                                     as pmch_start_date,
    if(not(wildmatch(pmch_fl_pmc_extended_wrt,'TCO')) and WildMatch(pmch_cd_stu,'
        CLOSED','OPEN'),'M&R','NO') as [M/R]

FROM [lib://FOLDER_XXX//ETL01_EDWH/St_TDIM_PMCH.qvd]
(qvd);

```

4.4.4 ETL2 - Master Calendar

Nel mondo della data analytics una delle sfide più ardue è quella di manipolare le informazioni relative alle date, il motivo di tale difficoltà risiede in diverse ragioni, tra le tante vengono riportate alcune delle più significanti:

- **Formati diversi:** Le date possono essere rappresentate in diversi formati (ad esempio MM/GG/AAAA o GG/MM/AAAA) e variano a seconda del contesto a livello internazionale; una cattiva gestione della data può complicare e compromettere la veridicità dell'informazione;
- **Fusi orari:** Le date e gli orari includono informazioni sui fusi orari, per cui la conversione tra fusi orari può essere complicata e può portare a errori se non viene gestita correttamente;

- **Operazioni aritmetiche complesse:** Calcolare la differenza tra due date, aggiungere o sottrarre giorni/mesi/anni può richiedere operazioni aritmetiche complesse, quindi è necessario svolgere attività di pre-processing a monte;
- **Rappresentazione informatica:** Spesso, nei sistemi informatici le date sono spesso rappresentate come numeri interi ad esempio con la conversione in formato giuliano, il che può portare a errori se non gestito correttamente durante le conversioni tra rappresentazioni e nel gestire precisione e limiti di rappresentazione;
- **Gestione date mancanti:** all'interno di un dataset non è singolare la situazione in cui ci sono delle date mancanti, questo può portare a problematiche di rilievo nella visualizzazione finale.

Per affrontare queste problematiche è importante utilizzare strumenti specifici per la gestione delle date, i quali permettono di evitare operazioni manuali complesse e consentono una buona comprensione delle regole e convenzioni utilizzate nel contesto specifico. Per i motivi elencati in Qlik è stato creato il Master Calendar, questo script crea un calendario centralizzato che facilita l'analisi dei dati temporali. Per la creazione del calendario viene preso come data di riferimento l'informazione contenuta nel campo `createdon_calendarKey` creato precedentemente nello script dei Dossiers.

```

QuartersMap:
MAPPING LOAD
    rowno() as Month_,
    'Q' & Ceil (rowno()/3) as Quarter
    AUTOGENERATE (12);

Temp:
Load
    min(createdon_calendarKey) as minDate,
    max(createdon_calendarKey) as maxDate

Resident Dossiers;

Let varMinDate = Num(Peek('minDate', 0, 'Temp'));
Let varMaxDate = Num(Peek('maxDate', 0, 'Temp'));
    
```

```

DROP Table Temp;

TempCalendar:
LOAD
    $(varMinDate) + Iterno()-1 As Num,
    Date($(varMinDate) + IterNo() - 1) as TempDate
    AutoGenerate 1 While $(varMinDate) + IterNo() -1 <= $(varMaxDate);

MasterCalendar_createdon:
Load
    TempDate AS createdon_calendarKey,
    week(TempDate) As Week_createdon,
    Year(TempDate) As Year_createdon,
    Month(TempDate) As Month_createdon,
    Month(TempDate) & '/' & Year(TempDate) as Month_year_createdon,
    Day(TempDate) As Day_createdon,
    ApplyMap('QuartersMap', month(TempDate), Null()) as Quarter_createdon,
    Week(weekstart(TempDate)) & '-' & WeekYear(TempDate) as WeekYear_createdon,
    WeekDay(TempDate) as WeekDay_createdon,
    if(Month(TempDate)*100 + Day(TempDate) <= Month(TempDate)*100 + Day(today())
        ,1,0) as YTD,
    If(today() - TempDate <= 5, '<=5',
        if (today() - TempDate <= 15, '5-15',
            if(today() - TempDate <= 30, '15-30',
                if(today() - TempDate <= 60, '30-60',
                    if( today() - TempDate <= 90, '60-90',
                        if( today() - TempDate <= 120, '90-120',
                            if(today() - TempDate <= 180, '120-180','>180')
                        )
                    )
                )
            )
        )
    ) as interval

Resident TempCalendar

Order By TempDate ASC;

Drop Table TempCalendar;

Workinghours:
load
RowNo() as Workinghours
Autogenerate(24);

```

```
WorkingDay:
Load
RowNo() as WorkingDay
Autogenerate(366);
```

Oltre alla definizione del calendario, in questo scrip sono stati creati degli intervalli di tempo che vengono aggiornati in tempo reale, ogni volta che viene aperta l'app, e che permettono di avere dei flag temporali utili per le analisi .

4.4.5 ETL2 - GeoFencing e DJC

Lo script di caricamento che segue illustra i campi che vengono portati sull'app relativi al GeoFencing e alla DJC e, che grazie al VIN vengono posti in RIGHT JOIN con la tabella principale dei Dossier. La presenza di campi come l'Entry e l'Exit Time del veicolo in officina nel Geofencing e del Planned Stop, coincidente con la data di un eventuale intervento di manutenzione prefissato, della DJC permetteranno di incrociare queste informazioni con quelle contenute nei dossier in modo da indagare e trovare possibili inconsistenze del dato contenuto nei Dossier.

```
GeoFencing:
LOAD
    Vin as Vin_Geofencing,
    date(entrytimestamp, 'DD/MM/YYYY hh:mm') as Entry_Time,
    date(exittimestamp, 'DD/MM/YYYY hh:mm') as Exit_Time,
    If(IsNull(exittimestamp), 'YES', 'NO') as [Into Workshop],
    If(IsNull(exittimestamp), today() - entrytimestamp, exittimestamp -
        entrytimestamp) as [Into Workshop Time],
    Vin&date(entrytimestamp, 'DD/MM/YYYY') as Key_Vorlog,

FROM [lib://FOLDER_XXX//ETL01_EDWH/St_GEOFENCING.qvd](qvd)
;

NoConcatenate
DJC:
LOAD
    Vin as clu_vin,
    Planned_Stop,
    Reason_ofStop
FROM [lib://FOLDER_XXX//ETL01_EDWH/ETL01_EDWH/St_DJC.qvd](qvd)
```

;

4.5 Presentation

Il Presentation Layer in Qlik Sense consente di visualizzare le informazioni caricate ed elaborate nelle fasi precedenti dell'implementazione. Le principali funzionalità di questo layer sono:

- **Visualizzazione dei dati:** Il layer Presentation permette agli utenti di creare dashboard, report e visualizzazioni interattive dei dati e gli utenti possono selezionare diversi tipi oggetti per rappresentare i dati;
- **Interattività:** Qlik Sense offre una User Experience interattiva che consente agli utenti di esplorare i dati, filtrarli, selezionarli e analizzarli in tempo reale, questo layer facilita questa funzione consentendo agli utenti di eseguire analisi ad hoc e di rispondere rapidamente alle richieste del Business;
- **Personalizzazione dell'interfaccia utente:** Gli sviluppatori e gli amministratori possono personalizzare l'interfaccia per adattarla alle esigenze dell'utente finale; questa personalizzazione può includere la creazione di layout, temi grafici e stili di visualizzazione.

In questo paragrafo verranno visualizzate le diverse schermate dell'app finale focalizzando l'attenzione sui grafici e le tabelle più importanti. Inoltre, saranno spiegate le principali logiche applicate a front-end.

4.5.1 Summary

Il foglio principale dell'app è il "Summary" [(Fig.4.8)]; questo foglio offre un sommario di tutte le principali misure e viene mostrato un'overview della situazione

generale relativa ai Dossier al variare di alcune dimensioni come, per esempio, l'anno di riferimento. Nello specifico, all'interno di questa schermata vengono riportati diversi grafici, a partire dal trend del Time Of Repair (TR) differenziato per $TR \leq 24h$ e $TR \leq 48h$, i grafici a torta per evidenziare la distribuzione dei dossier per Range e Sorgente dati, il barchart per focalizzare l'attenzione sui mercati più interessanti dall'apertura dei fascicoli e il numero di Dossier aperti per mese.

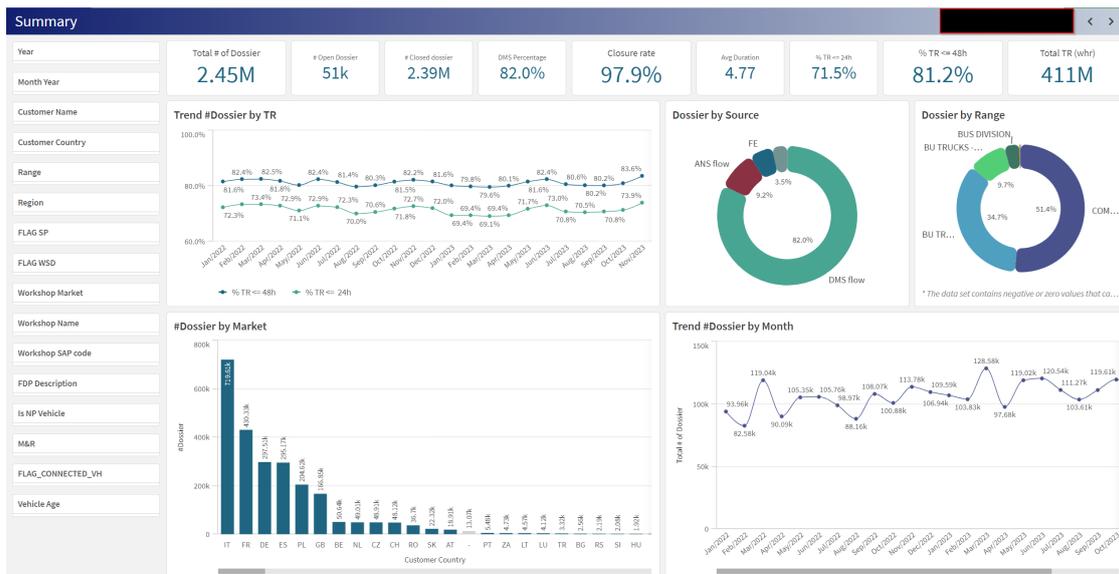


Figura 4.8: Summary Dashboard

Al di sopra dei grafici sono stati posizionati i KPIs principali [(Fig.4.9)], tra questi il Total # of Dossiers, ovvero il numero totale di fascicoli presenti nei Data Base, il numero di Dossiers aperti e chiusi, la percentuale di Dossiers aperti tramite il sistema DMS, la percentuale di fascicoli chiusi sul totale e indicatori relativi ai $TR \leq 24h$ e $TR \leq 48h$ in termini percentuali.



Figura 4.9: KPIs Summary

Il run chart presente in questa schermata mostra la variazione annuale del Time Of Repair (TR) [(Fig.4.10)]. Per ottenere questo grafico a front-end è stata implementata una logica caratterizzata dalla presenza della set analysis, la quale

permette di calcolare funzioni di aggregazione mantenendo fissi alcuni filtri; nello specifico, a numeratore di entrambe le formule la set expression include i due filtri $[TR \leq 24h] = \{ '1' \}$ e $[TR \leq 48h] = \{ '1' \}$, i quali restano fissi rispetto al denominatore che viene calcolato sul totale, quindi includendo tutti i diversi TR. Nel caso si desiderino approfondimenti su come utilizzare la set analysis, si consiglia di consultare il capitolo **Appendice** dove viene fornita una dettagliata spiegazione sul suo utilizzo e le sue applicazioni.

```
Count ({<statuscodename = {'Closed'},[TR<=24H] = {'1'}, FLAG_SP= {'1'}, FLAG_WSD = {'1'}>> distinct clu_vorlogdossierid)
/
Count ({<statuscodename = {'Closed'}, FLAG_SP= {'1'}, FLAG_WSD = {'1'}>> distinct clu_vorlogdossierid)

Count ({<statuscodename = {'Closed'},[TR<=48H] = {'1'}, FLAG_SP= {'1'}, FLAG_WSD = {'1'}>> distinct clu_vorlogdossierid)
/
Count ({<statuscodename = {'Closed'}, FLAG_SP= {'1'}, FLAG_WSD = {'1'}>> distinct clu_vorlogdossierid)
```

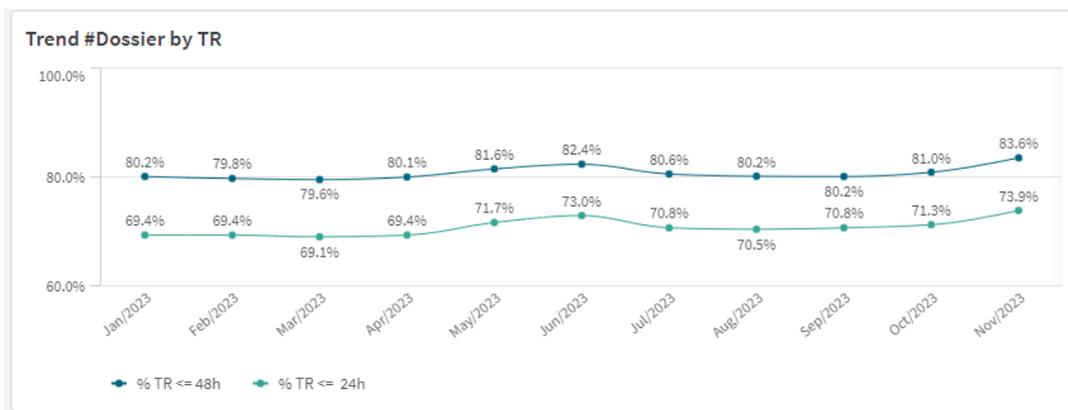


Figura 4.10: Run Chart TR \leq 24h e TR \leq 48h

4.5.2 Dettaglio Dossiers

Nella seconda pagina dell'app chiamata "Dettaglio Dossiers" [(Fig.4.11)] vengono mostrate tutte le informazioni di dettaglio relative al Dossiers. I KPIs mostrati in questa schermata sono il numero totale de Dossiers, di quelli aperti e di quelli chiusi, il Closure Rate e la percentuale di Dossiers DMS rispetto al totale. Nella tabella,

invece, vengono riportati i dettagli a partire dal numero identificativo de Dossier, la data di breakdown (corrispondente alla data in cui si è verificato il guasto), la data di apertura del Dossier, la data stimata per il rilascio del veicolo, il TR, il nome dell'officina, il market e tanti altri. Questo foglio di analisi è importante per poter approfondire le anomalie o assenza di dati presenti sugli altri fogli poichè presenta al suo interno tutte le informazioni.

The screenshot shows a dashboard titled 'Dossiers - Dettaglio'. At the top, there are five summary cards: 'Total # of Dossier' (2.45M), '# Open Dossier' (51k), '# Closed dossier' (2.39M), 'Closure rate' (97.9%), and 'DMS Percentage' (82.0%). Below these is a filter sidebar on the left with categories like Customer Name, Country, Range, Region, FLAG SP, FLAG WSD, Workshop Market, Workshop Name, Workshop SAP code, FDP Description, Is NP Vehicle, M&R, FLAG_CONNECTED_VH, and Vehicle Age. The main area contains a table with columns: Dossier Number, Source Code, Status Code - Dossier, Opened on - Dossier, Last Modified - Dossier, Failure Date, Estimated Time of Repair, TR, TR TA in Hours, Warranty Start Date, and Product Gr. The table lists various dossier entries with their respective dates and statuses.

Figura 4.11: Schermata Dettaglio Dossier

4.5.3 UpTime

La terza schermata è, insieme al Summary, il cuore della dashboard poichè è incentrata sul monitoraggio dell'UpTime[(Fig.4.12)], ovvero la disponibilità operativa dei veicoli. Nella tabella Pivot vengono mostrate le informazioni totali e le informazioni con granularità mensile. I KPI principali sono il numero di Dossier, la percentuale di Dossier aperti tramite sistema DMS e le percentuali di dossier con TR minore di 24 e 48 ore[(Fig.4.13)].

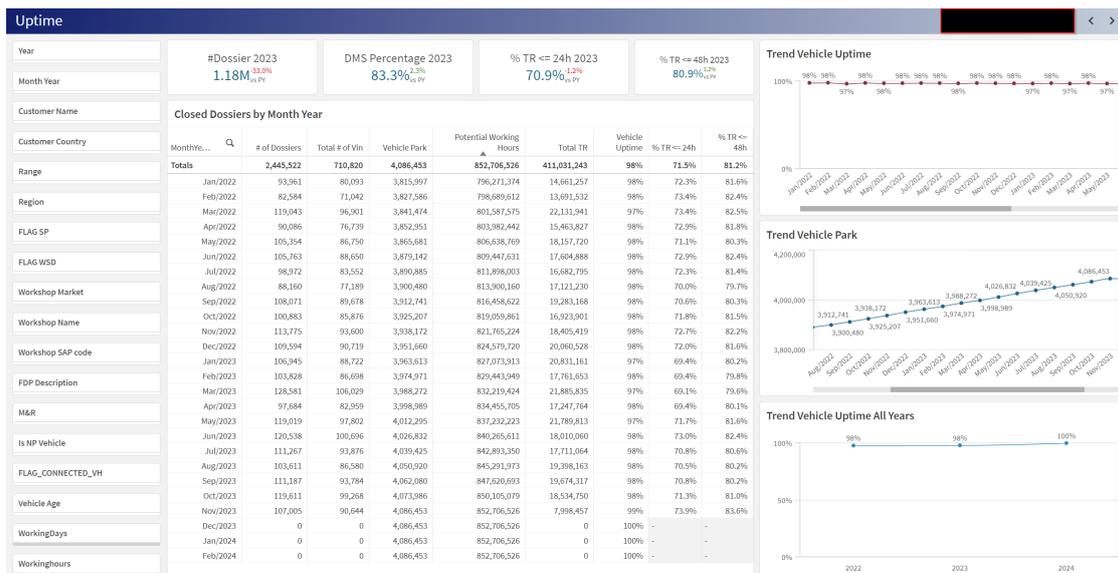


Figura 4.12: Schermata UpTime

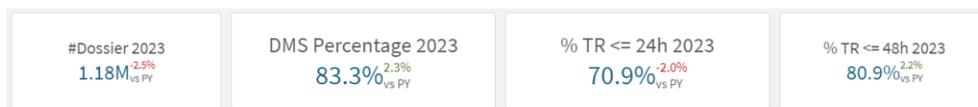


Figura 4.13: KPIs UpTime

In questo foglio di analisi, come si vede dalla figura [(Fig.4.13)], i KPI presentano un'ulteriore valore all'interno dei vari box [(Fig.4.14)]; questo valore fa riferimento alla variazione dello specifico indicatore di prestazione rispetto all'anno precedente (o Previous Year PY).



Figura 4.14: Confronto KPI Previous Year

Per calcolare la variazione in numero del totale dei dossier dal 2022 al 2023, a front-end è stata implementata la seguente logica:

```
(Count ({{<statuscodename = {'Closed'}, Year_createdon = {"$(= $(v_CY))"}, [YTD]={ '1' }>>}}distinct clu_vorlogdossierid)
```

```
Count ({<statuscodename = {'Closed'}, Year_createdon = {"$(= $(v_PY))"}, [YTD]={'1'
    }>>distinct clu_vorlogdossierid))
/
Count ({<statuscodename = {'Closed'}, Year_createdon = {"$(= $(v_PY))"}, [YTD]={'1'
    }>>distinct clu_vorlogdossierid)
```

Il codice permette di ottenere il Distinct Count del campo clu_vorlogdossierid calcolato solo sulla base dei filtri implementati nella set analysis. [YTD] è un campo che viene caricato nell'ETL2 Master Calendar, "Year-To-Date" è un'espressione utilizzata per riferirsi al periodo di tempo che inizia dall'inizio dell'anno corrente fino alla data odierna. In altre parole, indica l'intervallo di tempo che va dall'inizio dell'anno fino alla data attuale, compresi. Nel caso specifico, questo filtro permette di tenere in considerazione, nel conteggio, solo i vin associati ad una data di creazione antecedente, o uguale, alla data corrente e con punto di partenza per il conteggio il primo giorno dell'anno corrente.

La schermata UpTime assume, in termini strategici, un ruolo fondamentale nella società ONHIGH proprio per la possibilità che offre di visualizzare la variazione di alcuni indicatori chiave da un anno all'altro; risulta, quindi, utile per prendere decisioni in merito ad azioni correttive. In questa sezione, oltre ai classici filtri posti sulla sinistra sono presenti altri due bottoni: WorkingHours e WorkingDay, questi vengono utilizzati per modificare i parametri dati in input per il calcolo dell'UpTime, il quale risulta essere calcolato con la seguente formula:

```
1 - (sum([TR-TA in hours])
/
[Potential Working Hours])
```

Questa formula permette di ottenere la percentuale in ore di disponibilità dei veicoli sul totale del parco circolante. Infatti, la percentuale complementare mostra proprio la disponibilità a circolare dei veicoli rispetto al totale che include anche quelli fermi per guasti o riparazioni. Il Potential Working Hours è il numero di ore totali di lavoro che viene calcolato nel seguente modo:

```
[Vehicle Park] * v_workingDays * v_workingHours / 12
```

Il Vehicle Park, invece, viene calcolato come una somma cumulativa di tutti i veicoli; oltre al valore numerico presente in tabella viene mostrato il line chart con il cumulativo totale [(Fig.4.15)]. Il codice per il calcolo è il seguente:

```
rangesum(above(Sum(Vin), 0, RowNo()))
```

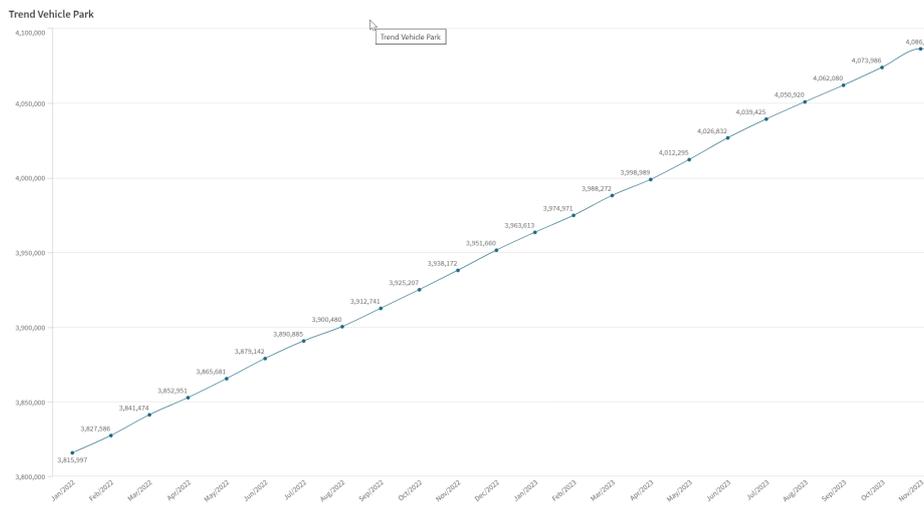


Figura 4.15: Line chart Parco Circolante

4.5.4 Open Dossiers

Per il monitoraggio dei Dossiers aperti sono state sviluppate due schermate; in entrambe sono state create delle tabelle pivot con differenti finalità. La prima schermata [(Fig.4.16)] visualizza il numero di dossier aperti da ogni sorgente dato categorizzandoli in diversi range di interesse che identificano i giorni totali per cui il Dossier è rimasto aperto; per ogni informazione viene visualizzato il valore assoluto e quello percentuale.

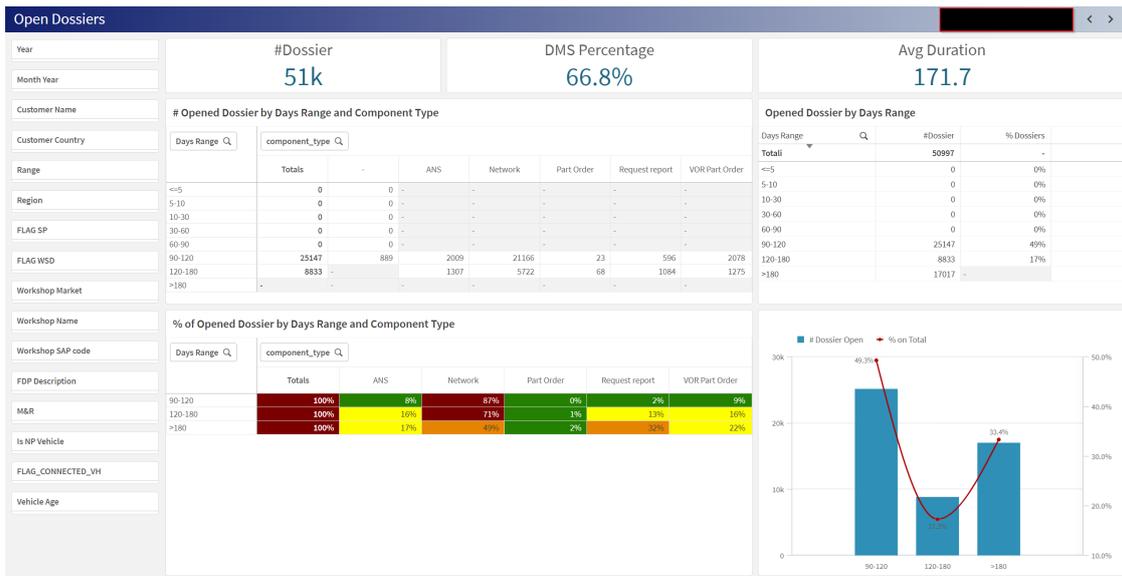


Figura 4.16: Schermata Open Dossiers

Nella seconda schermata [(Fig.4.17)], invece, sono state implementate le tabella pivot riferite ai diversi mercati e ai clienti, sempre in relazione ai diversi intervalli in giorni relativi al lasso di tempo in cui i Dossiers sono rimasti aperti.

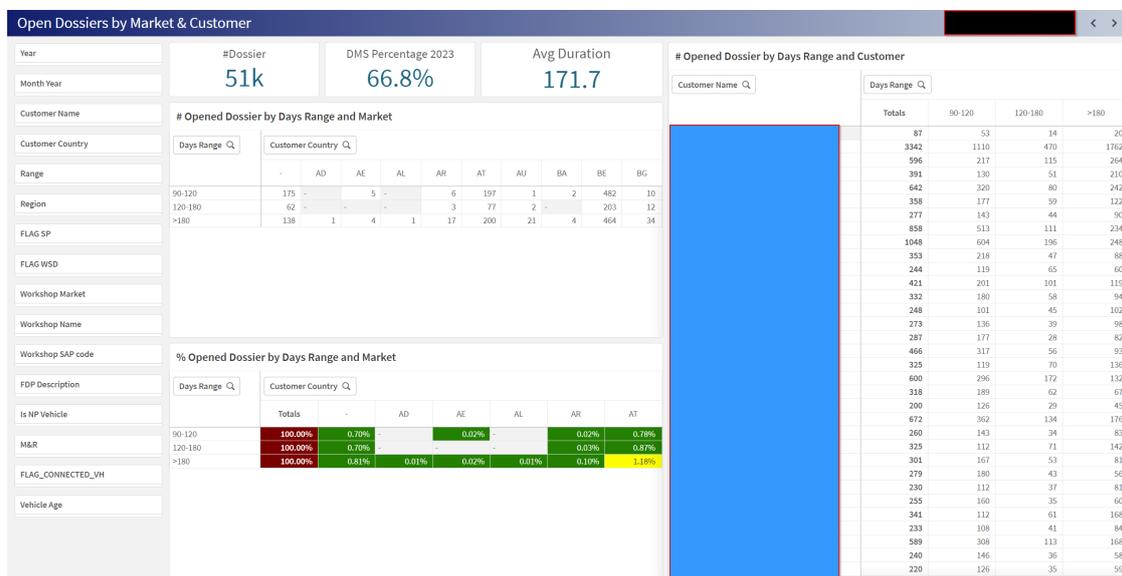


Figura 4.17: Schermata Open Dossiers By Markets & Customer

4.5.5 Markets

Tra le varie richieste del Business, una delle principali è stata quella di avere una schermata che riportasse in dettaglio le informazioni concernenti la diffusione dei Dossiers per i diversi mercati [(Fig.4.18)]. Questa esigenza nasce dalla volontà di voler monitorare i mercati in cui l’apertura di un Dossier (e quindi implicitamente di un fermo veicolo) avviene più spesso. Nella tabella principale è possibile selezionare lo specifico mercato e visualizzare sia il dato aggregato per anno che il dato per ogni singolo Dealer del mercato, grazie alla selezione specifica. Inoltre, sempre in questa schermata, è stato inserito un bar chart orizzontale che consentisse di visualizzare il nome del Dealer e il numero di Dossiers aperti presso lo stesso.

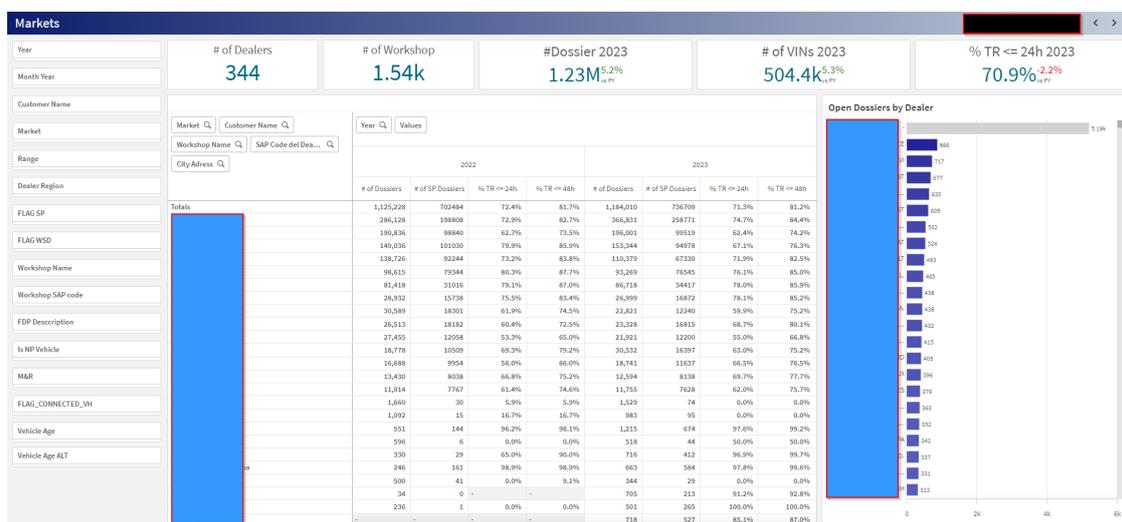


Figura 4.18: Schermata Markets

4.5.6 Geofencing

Questo foglio dell’applicazione è uno di quelli preposto per il controllo relativo alla veridicità dei dati. Uno degli scope principali alla base del progetto è quello di monitorare il livello di accuratezza dei dati provenienti dai dossier; questo aspetto è dovuto, principalmente, alla differenziazione delle varie fonti da cui provengono i dati dei fascicoli. Il fatto che non ci sia una procedura standardizzata ma, al contrario, ci siano diversi software gestionali sorgente (ad esempio il DMS, il CRM

ANS24 o il tool manuale APPARO) che permettono di creare un dossier è una specifica non di poco conto e poichè non c'è una regola universale per la gestione degli stessi ed è necessario, quindi, avere strumenti di supporto per intercettare e gestire situazioni anomale. Per questo motivo vengono utilizzati i dati provenienti dal Geofencing, un tool in grado di acquisire informazioni relative alla presenza di uno specifico veicolo, che viene identificato attraverso il VIN, all'interno di un'officina (o Workshop); inoltre, la chiave utilizzata per collegare i dati è il Vin e la data di Breakdown per i dossier e il Vin e la data di Entry Time per il Geofencing chiamata Key Vorlog. Nel foglio dedicato [(Fig.4.19)] vengono illustrati due KPI che sono il numero di dossier aperti e chiusi che permettono di effettuare il confronto tra il risultato che fa fede alle informazioni contenute nei dossier e lo stesso KPI calcolato con il dato del Geofencing. E' importante sottolineare che questo strumento, come il DJC, risulta avere delle debolezze: in primo luogo non tutte le officine affiliate a ONHIGH possiedono questa tecnologia, motivo per cui un'analisi corretta può essere effettuata solo considerando le officine utilizzatrici di tali strumenti e, in secondo luogo bisogna tenere in considerazione anche di possibili dati errati o anomali dovuti a errori di acquisizione dei dati.

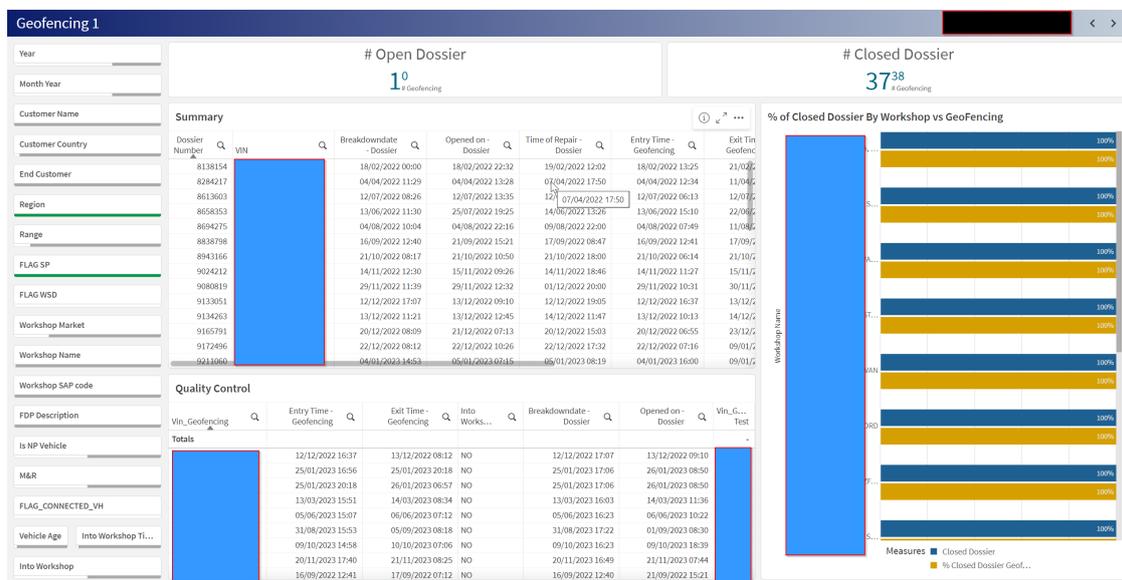


Figura 4.19: Schermata Geofencing

Nella tabella Summary vengono riportate la date più importanti per ef-

fettuare il controllo sulla consistenza del dato: la Breakdown Date che nei Dossier corrisponde verosimilmente al giorno in cui è avvenuto il fermo del veicolo e la Time of Repair che corrisponde, invece, alla data i cui il veicolo è tornato a circolare. Queste due date vengono confrontate, rispettivamente, con l'Entry Time e l'Exit Time, ovvero data di ingresso e di uscita del veicolo dall'officina. In situazioni di coerenza del dato, queste due informazioni sono pressocchè coincidenti, anomalie su queste informazioni sarebbero campanelli di allarme. Il grafico a destra mostra la percentuale di dossier chiusi per Dossier e Geofencing per ogni Workshop mentre la tabella in basso è uno strumento secondario per l'utente che permette di controllare le informazioni proveniente dal Geofencing indipendenti da quelle dei Dossier.

4.5.7 Global Monthly

In questa sezione dell'app [(Fig.4.20)] la visualizzazione è focalizzata sui valori globali con focus sull'anno corrente e mostra i principali grafici visti fino a questo momento. Per ognuno di questi grafici è stata inserita una set analysis che consente di considerare i dati di uno specifico anno, per fare ciò a front-end è stata definita una variabile dipendente dall'anno corrente.



Figura 4.20: Schermata Global Monthly

4.5.8 Weekly

Un ulteriore livello di dettaglio viene aggiunto nella schermata "Weekly", all'interno della quale la granularità delle informazioni è a livello di singola settimana di un determinato anno. In particolare, il grafico centrale per ogni settimana evidenzia la differenza del numero di Dossiers per l'anno corrente e il precedente; mentre i due run charts mostrano i trend per i TR $\leq 24h$ e $48h$ per settimana.

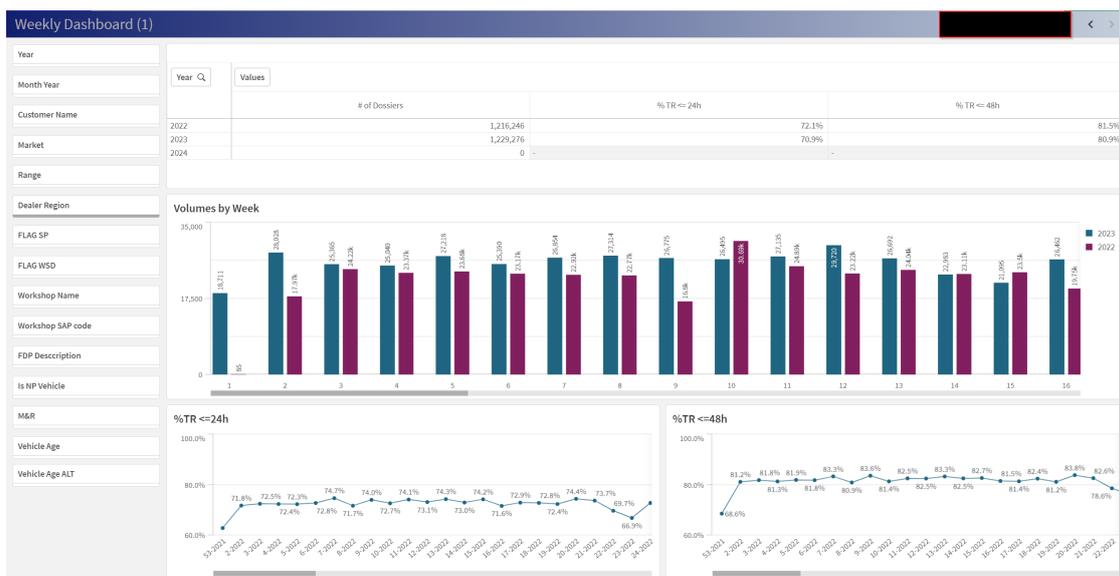


Figura 4.21: Schermata Weekly

Capitolo 5

Conclusioni e Sviluppi Futuri

La dashboard presentata in questo elaborato per il progetto VORlog Dossier è stata progettata e sviluppata in risposta a delle esigenze ben precise che sono state poste in essere dalla società ONHIGH, una multinazionale che opera nel settore automotive come principale player in tutto il mondo; tra tutte, in primo luogo la necessità monitorare la disponibilità del parco veicolare e in secondo luogo cercare di monitorare la veridicità dei dati inerenti i guasti dei veicoli già acquisiti attraverso ulteriori strumenti di acquisizione dati che fungeranno da supporto per il sistema principale. Il progetto si inserisce all'interno di un programma più ampio mirato al miglioramento del sistema di reportistica della function aziendale di Qualità e Garanzia del prodotto. La nuova app, sviluppata in ambiente Qlik Sense da KM, ha permesso di ottenere un quadro generale rispetto alla gestione dei veicoli che subiscono dei guasti o dei fermi dovuti a interventi di manutenzione programmati; uno strumento di questo genere, all'interno di una società che produce e distribuisce veicoli di vario tipo, risulta essere fondamentale per il miglioramento dei servizi offerti post vendita come programmi di garanzia, investimenti in piani di ricerca e sviluppo e supporto continuo al cliente.

Il contesto operativo e la coordinazione del lavoro in squadra hanno permesso a me, e quindi a tutto il team con cui ho lavorato, di portare a conclusione il progetto raggiungendo tutti gli obiettivi e realizzando i desiderata del cliente in

termini di funzionalità dell'app e mantenimento di standard user-friendly. Quest'ultimo risulta essere il principale requisito in quanto gli utenti finali, non essendo esperti di sviluppo e progettazione di strumenti di Business Intelligence, prediligono l'utilizzo di software semplici e intuitivi che non richiedano, quindi, eccessiva conoscenza e formazione specifica per l'utilizzo dell'app. Il lavoro svolto mi ha permesso di acquisire nuove competenze sia in ambito tecnico, ponendo l'attenzione sulle principali metodologie di data quality, data mining, analisi statistiche e modellazione dei dati in Qlik Sense, sia dal punto di vista personale per quanto riguarda la gestione dei tempi e delle attività in un progetto di Business Intelligence all'interno di una società di consulenza.

In termini di proiezione al futuro, come già è emerso da alcuni incontri con il Business, esistono delle migliorie e delle integrazioni che potranno esser fatte per apportare valore aggiunto al nuovo strumento. Come integrazione verrà discussa la possibilità di apportare nuove informazioni attraverso dei flussi dati inerenti alcune teleservizi offerti da ONHIGH per il monitoraggio dello stato di salute dei veicoli. Per le migliorie dell'app, invece, tra le richieste del Business c'è l'implementazione di un meshup per rendere fruibile l'applicazione attraverso pagine web altamente personalizzate e quella di creare un unico sistema di generazione dei dossier; questo potrebbe essere un modo per garantire l'efficienza e la standardizzazione di alcune regole riguardanti la gestione del singolo dossier, come ad esempio le logiche utilizzate per la chiusura automatizzata degli stessi dopo un determinato intervallo di tempo, che permetterebbero di avere dei "Time of Repair" coerenti con la realtà e quindi, di conseguenza, avere delle numeriche più consistenti.

Capitolo 6

Appendice

6.1 Sintassi dello script in Qlik

LOAD - FROM/RESIDENT

L'istruzione LOAD ha la capacità di caricare dati da diverse fonti, tra cui file esterni, dati definiti all'interno dello script, tabelle precedentemente caricate, pagine web, risultati di query SELECT successive o attraverso la generazione automatica di dati. Per il caricamento da file esterni vengono usati i comandi FROM o RESIDENT, il primo richiama uno specifico file mentre il secondo carica i dati da una tabella caricata in un passaggio precedente dello script e che quindi è già presente nel data model di Qlik. [14]

```
LOAD
    fieldlist
from file
```

```
LOAD
    fieldList
resident tableName
```

Per definire i dati all'intero dell' script stesso, invece, è necessario usare il comando IN LINE.

```
Load *
  Inline [
    ID, Name, Age, Title
    1, Luca, 31, Team Leader
    2, Paolo, 22, Technical Support Engineer
    3, Maro, 40, Customer Engineer
    4, Jannifer, 27, IT Developer
  ];
```

JOIN

Il comando join in Qlik Sense unisce la tabella caricata con una tabella esistente o con l'ultima tabella di dati creata in precedenza. Esistono quattro tipi di join:

1. Left join: È il tipo più comune di join e consente di mantenere tutte le transazioni così come sono e di aggiungere i campi dati di riferimento supplementari dove viene trovata una corrispondenza relativi alla tabella che viene messa in join.
2. Inner join: Utilizzato quando sono importanti solo i risultati in cui è presente una corrispondenza di associazione, questa elimina tutti i record sia dai dati di origine caricati che dalla tabella di destinazione se non viene trovata alcuna corrispondenza, riducendo così il numero di record nella tabella di destinazione.
3. Outer join: Conserva sia i record di destinazione che tutti i record in ingresso, mantenendo ogni set di record anche se non viene trovata alcuna corrispondenza. I campi per i quali non viene trovata una corrispondenza non verranno popolati e rimarranno vuoti (null).
4. Right join: Questo tipo di join mantiene tutti i record che appartengono alla tabella in entrata, riducendo i record nella tabella iniziale tramite l'operazione di join solo ai record in cui è presente una corrispondenza di associazione nei record in entrata. [14]

```
left join (tableIniziale)
load
    field
    field
    ...
from tableEntrata
```

CONCATENATE - NOCONCATENATE

Concatenate nel caricamento dello script consente di aggiungere un set di dati a una tabella in memoria già esistente. È comunemente utilizzato per unire diversi set di dati transazionali in una tabella dei fatti centrale o per costruire insiemi di dati di riferimento comuni provenienti da più fonti, la sua funzionalità è simile a quella dell'operatore UNION di SQL. NoCONCATENATE, invece, viene utilizzato quando si vuole creare una tabella che contiene campi identici a quelli di un'altra tabella.

SET - LET

L'istruzione set è impiegata per definire le variabili nello script, queste variabili possono essere utilizzate per sostituire stringhe, unità o percorsi. L'istruzione, invece, let funge da complemento all'istruzione set e anch'essa viene utilizzata per definire delle variabili negli script. Rispetto a set, l'istruzione let valuta l'espressione sul lato destro del simbolo '=' al momento dell'esecuzione dello script, prima di assegnarla alla variabile. [14]

```
Set variablename=string
```

```
Let variablename=expression
```

DROP

Il comando DROP può essere utilizzato per eliminare un campo o una tabella dal database. [14]

```
Drop FileName
from Table

drop table tablename/tablenames
```

QUALIFY e UNQUALIFY

L'istruzione Qualify offre la possibilità di modificare la qualificazione dei nomi dei campi, in particolare permette di aggiungere come prefisso ad ogni campo il nome della tabella di provenienza. [14]

```
qualify *;

Table_name:
Load *
Resident Table_Source;

unqualify *;
```

Funzioni Data e Ora

Le funzioni data e ora di Qlik Sense offrono la possibilità di manipolare e convertire i valori relativi a data e ora; queste funzioni possono essere impiegate sia nello script di caricamento dei dati che nelle espressioni grafiche a front-end e consentendo agli utenti di adattare e gestire le informazioni temporali secondo le proprie esigenze. Per i comandi che verranno descritti in questo paragrafo il formato di partenza di ora e data hh:mm:ss e YYYY-MM-DD (ISO 8601). Le funzioni riportate permettono di ottenere, in ordine: i secondi, i minuti, le ore, il giorno, la settimana, il mese, l'anno, la settimana all'interno di un anno.[14]

```
second(fieldDate)
minute(fieldDate)
```

```
hours(fieldDate)
day(fieldDate)
week(fieldDate)
month(fieldDate)
year(fieldDate)
weekyear(fieldDate)
```

In Qlik è possibile anche convertire un indicatore temporale in ora locale:

```
ConvertToLocalTime(timestamp(clu_etr, 'DD/MM/YYYY hh:mm'), 'Rome')
```

Funzioni di formattazione

Per una corretta manipolazione dei dati è possibile effettuare conversioni dei dati in diversi formati, nello script che segue verranno citate le più importanti. La funzione `TimeStamp()` di Qlik Sense permette di formattare un'espressione come un valore di data e ora, seguendo il formato dell'indicatore temporale impostato nelle variabili di sistema nello script di caricamento, però è possibile fornire una stringa di formattazione per specificare un formato personalizzato, se necessario.[14]

```
timestamp(fieldDate, 'DD/MM/YYYY hh:mm')
```

Il comando `Date()` restituisce la data nel formato specificato nel secondo parametro.[14]

```
date(fieldDate, 'DD/MM/YYYY')
```

Il comando `text()` restituisce il campo in una stringa di caratteri[14]

```
text(fieldName)
```

Match

La funzione `Match()` in Qlik Sense confronta il primo parametro con tutti i parametri successivi e restituisce la posizione numerica delle espressioni corrispondenti.

È importante notare che questo confronto rispetta la distinzione tra maiuscole e minuscole.[14]

```
match( str, expr1 [ , expr2, ...exprN ])
```

MixMatch

Il comando MixMatch() ha la stessa funziona di Match() con la differenza che questo comando non è case sensitive per cui non tiene conto della distinzione tra maiuscole e minuscole.[14]

```
mixmatch( str, expr1 [ , expr2, ...exprN ])
```

WildMatch

Questo comando è un'ulteriore alternativa ai due precedenti con la particolarità che consente l'utilizzo di caratteri jolly, come * e ? nelle stringhe di confronto: * corrisponde a qualsiasi sequenza di caratteri, mentre ? corrisponde a qualsiasi carattere singolo. [14]

```
wildmatch( str, expr1 [ , expr2, ...exprN ])
```

6.2 Set Analysis

La Set Analysis è una funzionalità di Qlik Sense utile per effettuare analisi approfondite. Questo strumento consente agli utilizzatori di creare delle espressioni che filtrano i dati in base a condizioni specifiche quali intervalli di tempo, campi o altre caratteristiche del dato. La sintassi per utilizzare questa funzione è la seguente [(Fig.6.1)]:

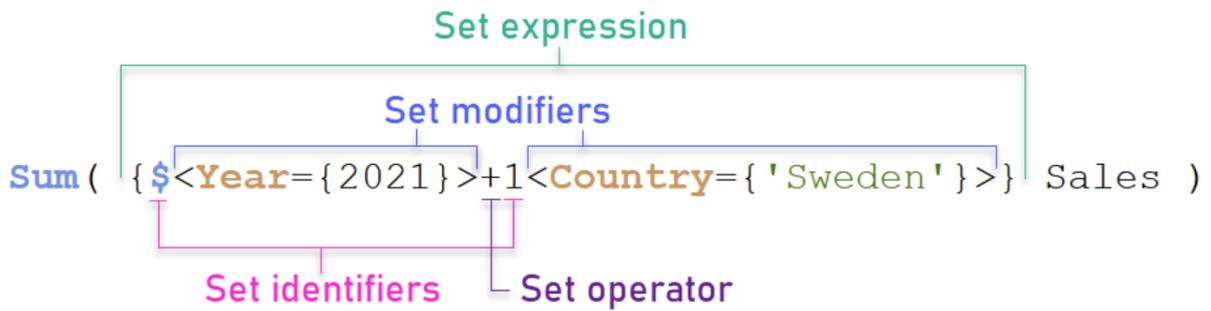


Figura 6.1: Sintassi Set Analysis[14]

Nell'esempio in figura viene calcolata la somma di Sales considerando tutti i campi che hanno rispettivamente l'anno 2022 (applicato solo al contesto di selezione corrente) e Country Sweden. La set Analysis contiene:

1. Identifiers: un identificatore rappresenta un set di dati, tutti i dati o un sottogruppo di essi;
2. Operators: sono degli operatori utilizzati per creare unioni, differenze o intersezioni, nell'esempio in figura;
3. Modifiers: è composto da uno o più nomi di campo per impostare il filtro; il modificatore è racchiuso tra "<>".[14]

Bibliografia

- [1] European Automobile Manufacturers Association (ACEA). *ACEA - European Automobile Manufacturers Association*. URL: <https://www.acea.auto/>.
- [2] Saberi. «The Role of the Automobile Industry in the Economy of Developed Countries». In: *International Robotics & Automation Journal* 4.3 (2018), pp. 179–180. DOI: 10.15406/iratj.2018.04.00119.
- [3] European Automobile Manufacturers Association (ACEA). *ACEA Pocket Guide 2023-2024*. 2023. URL: <https://www.acea.auto/files/ACEA-Pocket-Guide-2023-2024.pdf>.
- [4] Yu-Ni Tang. «Research on the Transformation and Upgrading of China’s Automobile Industry From the Perspective of Global Value Chain». In: *International Academic Conference on Frontiers in Social Sciences and Management Innovation (IAFSM 2019)*. Atlantis Press. 2020, pp. 415–420.
- [5] R. Ramakrishnan e J. Gehrke. *Sistemi di basi di dati*. McGraw-Hill, 2004.
- [6] Matteo Golfarelli e Stefano Rizzi. *Data Warehouse – teoria e pratica della progettazione*. McGraw-Hill, 2006.
- [7] Deanne Larson e Victor Chang. «A review and future direction of agile, business intelligence, analytics and data science». In: *International Journal of Information Management* 36.5 (2016), pp. 700–710. ISSN: 0268-4012.
- [8] Helena Dudycz. «Visualization methods in Business Intelligence systems—an overview». In: *Business Informatics. Data Mining and Business Intelligence* 16 (2010), pp. 9–24.

- [9] M. Aparicio e C. J. Costa. «Data Visualization». In: *Communication Design Quarterly Review* 3.1 (2015), pp. 7–11.
- [10] M. Islam e S. Jin. «An Overview of Data Visualization». In: *2019 International Conference on Information Science and Communications Technologies (ICISCT)*. Tashkent, Uzbekistan, 2019, pp. 1–7. DOI: 10.1109/ICISCT47635.2019.9012031.
- [11] Qlik. *Qlik*. URL: <https://www.qlik.com/>.
- [12] DataFlair. *Qlik Sense Features*. URL: <https://data-flair.training/blogs/qlik-sense-features/>.
- [13] Net Informatica. *Microsoft Azure: cos'è e a cosa serve*. Accesso al sito: inserire la data di accesso. inserire l'anno. URL: <https://www.net-informatica.it/microsoft-azure-e-a-cosa-serve/>.
- [14] Qlik. *Scripting Statements and Keywords*. 2023. URL: https://help.qlik.com/it-IT/sense/November2023/Subsystems/Hub/Content/Sense_Hub/Scripting/script-statements-keywords.htm (visitato il 07/02/2024).