

**Università Politecnica delle Marche**

Facoltà di Ingegneria

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione

---



**Tesi di Laurea**

**Progettazione e implementazione della componente  
“KPI-Pedia” relativa ad un sistema di data analytics per  
un’azienda di apparecchi medicali**

**Design and implementation of the “KPI-Pedia” component  
of a data analytics system for a company producing medical  
equipment**

Relatore

Prof. Domenico Ursino

Candidato

Simone Miccoli

---

**Anno Accademico 2020-2021**



---

# Indice

<b>Introduzione</b> .....	7
<b>1 Introduzione ai KPI</b> .....	9
1.1 I KPI nella Business Intelligence .....	9
1.1.1 I KPI .....	9
1.1.2 I CPI .....	10
1.2 Overview dei principali KPI utilizzati .....	10
1.2.1 CRM .....	11
1.2.2 KPI-Pedia .....	11
<b>2 Descrizione del progetto complessivo</b> .....	15
2.1 Caratteristiche generali e obiettivi del progetto .....	15
2.2 Back-End .....	16
2.2.1 Le principali tabelle sorgente .....	17
2.2.2 Gestione e integrazione dei dati con Oracle Data Integrator ..	18
2.2.3 Le principali tabelle target .....	23
2.2.4 Creazione delle misure con Tabular Editor .....	23
2.3 Front-End .....	24
2.3.1 Creazione delle dashboard .....	25
2.3.2 Overview delle principali dashboard .....	25
<b>3 Analisi dei requisiti relativi alla componente KPI-Pedia</b> .....	27
3.1 Contactability: regole di validità .....	27
3.1.1 Le RegExp in SQL .....	28
3.1.2 Canali di comunicazione validi .....	28
3.2 Requisiti dei principali KPI .....	31
3.3 Requisiti delle dashboard .....	32
<b>4 Progettazione della componente KPI-Pedia</b> .....	33
4.1 Progettazione dei principali mapping in ambiente di test ODI .....	33
4.2 Progettazione dei principali package in ambiente di test ODI .....	37
4.2.1 Package CUSTOMER BASE ALL EVENTS .....	37
4.2.2 Package ENGINE .....	38

<b>5</b>	<b>Implementazione della componente KPI-Pedia</b> .....	41
5.1	Export da ambiente di test .....	41
5.2	Rilasci .....	42
5.3	Implementazione misure su Tabular Editor .....	43
<b>6</b>	<b>Uno sguardo alle componenti realizzate</b> .....	45
6.1	Dashboard KPI-Pedia.....	45
6.2	Dashboard Contactability.....	46
6.3	Altre dashboard realizzate.....	47
<b>7</b>	<b>Discussione in merito al lavoro svolto</b> .....	51
7.1	Lezioni apprese.....	51
7.1.1	L'importanza della Business Intelligence e dei KPI.....	51
7.1.2	L'importanza dell'analisi dei requisiti.....	52
7.1.3	L'importanza delle sessioni di UAT.....	52
7.1.4	L'importanza dei rilasci.....	53
7.2	Analisi SWOT.....	53
7.2.1	Punti di forza.....	54
7.2.2	Punti di debolezza.....	54
7.2.3	Opportunità.....	55
7.2.4	Minacce.....	55
<b>8</b>	<b>Conclusioni</b> .....	57
	<b>Riferimenti bibliografici</b> .....	59
	<b>Ringraziamenti</b> .....	61

---

## Elenco delle figure

2.1 Schema logico dell'infrastruttura realizzata	16
2.2 Le fasi di ETL	18
2.3 ODI - Logo	19
2.4 Esempio di mapping in ODI	20
2.5 Insieme degli operatori in ODI	21
2.6 Esempio di package in ODI	21
2.7 Visual ODI	22
2.8 Tabular Editor - Logo	24
2.9 Esempio di misura realizzata su Tabular Editor	24
2.10 Microsoft Power BI - Logo	24
2.11 Customer Overview	25
2.12 Show in Store	26
2.13 Called by Call Center	26
3.1 Regole di validità per Direct Mail	28
3.2 Regole di validità per Email	29
3.3 Regole di validità per Phone	30
3.4 Regole di validità per Mobile	30
4.1 Flusso del mapping FT DATA ENRICHMENT	34
4.2 Lista dei mapping della KPI-Pedia	35
4.3 Flusso del mapping HA SALE	36
4.4 Package CB All Events	38
4.5 Package ENGINE	39
4.6 Espansione package CUSTOMER BASE COMPUTATION	39
5.1 Lista di misure percentuali su <i>Tabular Editor</i>	43
5.2 Misura update validi con consenso su canale MOBILE	43
6.1 Pagina con KPI di natura <i>Commercial</i>	45
6.2 Trend Overview - Contactability	47
6.3 Historical Customer Base	48
6.4 Historical Customer Base	48
6.5 Dashboard - CLC	49



---

## Introduzione

Nel corso degli anni, lo sviluppo di molteplici tecnologie, come smart device e applicazioni software, ha comportato una crescita esponenziale del volume di dati generato all'interno delle organizzazioni. Le realtà aziendali che operano nel settore retail sono state le prime ad essere coinvolte da tale fenomeno, in quanto guidate dalla *Digital Transformation*. Con la stessa velocità con cui è cresciuta la generazione di dati sono cresciute anche le sfide e le opportunità di utilizzo della cosiddetta *data analytics*. Le organizzazioni riconoscono il valore posseduto dai dati e si adoperano per una loro gestione efficiente ed efficace, consolidandosi sempre più in modelli di business di tipo data-driven. I dati vengono generati durante ogni attività quotidiana, da diverse sorgenti e in differenti formati, e le aziende necessitano di strumenti e competenze adeguate per analizzarli correttamente.

“Tortura abbastanza a lungo i dati ed essi confesseranno qualunque cosa.” Con questa citazione anonima si capisce che avere a disposizione una grande mole di dati non porta alcun vantaggio competitivo all'organizzazione che li possiede, a meno che venga realizzato un sistema ad hoc che permetta di “lavorare” il dato fino all'ottenimento di conoscenza. È proprio la conoscenza che permette ad un'azienda di avere una visione più smart sugli obiettivi di business prefissati. Capire, dunque, come poter collezionare dati, gestirli e integrarli è una strategia fondamentale per migliorare gli output delle analitiche, potenziare le attività di business intelligence, e offrire al management un supporto più robusto nei processi decisionali.

Il noto manager Vincenzo Cosenza afferma che: “Nei soli Stati Uniti, per poter sfruttare efficacemente le potenzialità dei big data nei processi decisionali occorrerebbero dalle 140 alle 190 mila persone”. Ciò fa capire quanto questo mercato sia in crescita ed a beneficiarne non sono solo le aziende che utilizzano sistemi di business intelligence, ma anche l'analista che si avvicina a questo settore. Tutto ciò per marcare il fatto che la data analytics si sta evolvendo verso il raggiungimento di un vero e proprio stato di ecosistema a beneficio dell'intero mercato.

La presente tesi è collocata in questo contesto culturale. Essa, infatti, si occupa della realizzazione di un sistema ad hoc di data analytics per un'azienda che opera nel settore retail di apparecchi medicali. In particolare, si pone l'attenzione su un particolare stream, ovvero quello della KPI-Pedia. Con questo nome si indica esplicitamente la volontà di raccogliere quanti più indicatori di performance possibili, su richiesta dell'azienda committente, realizzando una vera e propria “Enciclopedia di

KPI”.

Proprio a consolidare il concetto che i dati hanno di per sè valenza nulla, se non “addomesticati” per giungere alla conoscenza, viene affrontata sia la realizzazione del back-end che del front-end. Il primo è costituito da un processo di trasformazione e caricamento dei dati, realizzati con il supporto di diversi tool, come Oracle Data Integrator e Tabular Editor. In questa fase si prelevano in primis i dati contenenti informazioni relative alla clientela dell’azienda da appositi database sorgente. Successivamente sarà avviato un complesso processo di trasformazione del dato che culminerà con il popolamento di apposite tabelle all’interno di database target, uno per ogni nazione coinvolta all’interno del progetto. Il front-end è, invece, costituito da un insieme di report all’interno dei quali vengono visualizzati gli indicatori di performance, precedentemente calcolati, in diverse modalità.

L’obiettivo finale dell’azienda presso la quale è stata realizzata la presente tesi, è quello di consegnare dei dashboard ad hoc richiesti dal cliente, che contengano, e quindi che permettano di visualizzare in maniera corretta, tutti gli indicatori richiesti in fase di analisi dei requisiti. L’azienda farà, poi, uso della reportistica per capire la direzione strategica da intraprendere al fine di ottenere vantaggio competitivo e affermarsi come leader di mercato nel proprio settore.

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

- Nel Capitolo 1 verrà introdotto il concetto di KPI nel mondo della business intelligence; si cercherà di capire come questo possa aiutare l’azienda al raggiungimento dei propri obiettivi e ci sarà un piccolo excursus sui principali indicatori realizzati nel progetto.
- Nel Capitolo 2, invece, si troverà la descrizione del progetto complessivo, sia lato back-end che front-end, con uno sguardo ai principali tool utilizzati e a come questi supportano la realizzazione di un sistema di data analytics. Verranno, anche, richiamate le principali tabelle utilizzate, nonché alcuni visual presenti nei report realizzati.
- Nel Capitolo 3 verrà affrontata l’analisi dei requisiti, dando uno sguardo alle principali richieste dell’azienda committente.
- Nel Capitolo 4 si vedrà come progettare dei flussi di mapping e pacchetti utili alla trasformazione dei dati sorgente con il tool Oracle Data Integrator.
- Nel Capitolo 5, invece, si vedrà come esportare tutto ciò che è stato fatto in fase di progettazione in ambiente di produzione, dove si trovano le strutture che popoleranno i visual della reportistica.
- Nel Capitolo 6 verrà dato uno sguardo ai principali dashboard realizzati.
- Nel Capitolo 7 si discuterà in merito al lavoro svolto, quindi verranno presentate le lezioni apprese e si condurrà un’analisi SWOT per capire punti di forza e debolezza, ma anche opportunità e minacce, all’interno del progetto.
- Nel Capitolo 8 verranno tratte le conclusioni e delineati eventuali sviluppi futuri.



## Introduzione ai KPI

*In questo primo capitolo sarà introdotta l'importanza dei KPI in ambito di Business Intelligence, successivamente verrà proposto un piccolo excursus sui principali KPI utilizzati all'interno del progetto oggetto della presente tesi, che saranno meglio discussi in seguito.*

### 1.1 I KPI nella Business Intelligence

#### 1.1.1 I KPI

KPI risulta essere l'acronimo dell'espressione *Key Performance Indicator*, in italiano conosciuta anche come *Indicatore Chiave di Prestazione* (ICP). In ambito di Business Intelligence si tratta di uno strumento molto potente ed importante perchè permette alle imprese di verificare in quale misura vengano raggiunti gli obiettivi aziendali, in generale diversi in base alla tipologia di organizzazione. Nella pratica non vi è un numero di indicatori fissi disponibili per la consultazione, in quanto il tutto dipende dal settore di utilizzo, nonchè dagli obiettivi di business.

Chiaro è che non tutti gli indicatori possono essere definiti banalmente KPI; questo perchè i KPI devono rispettare una serie di requisiti sotto elencati:

- *essere quantificabili*: ciascun KPI deve poter essere rappresentato numericamente;
- *essere allineati con gli obiettivi aziendali*: i KPI devono rappresentare in maniera quantitativa gli obiettivi dell'impresa;
- *poter dimostrare l'andamento aziendale*: gli indicatori devono essere in grado di mostrare la direzione strategica che sta prendendo l'azienda;
- *essere operativi*: devono poter essere messi in pratica per valutare i risultati effettivi.

In ambito di Business Intelligence i KPI hanno molto seguito soprattutto quando vengono riportati all'interno di apposite *Dashboard*, ovvero dei veri e propri Cruscotti aziendali. In essi i KPI vengono riportati seguendo una particolare linea estetica e impressiva che ne migliora la lettura da parte dei vertici aziendali che non seguono l'intero percorso di progettazione e calcolo.

Un aspetto molto importante in tal senso è la corretta scelta dei KPI più significativi e maggiormente adattabili alla causa e agli obiettivi aziendali. L'utilizzo di KPI poco significativi, o comunque difficili da leggere, renderebbe più complesso il processo di crescita aziendale, ed è proprio da questo che scaturisce la necessità di avere un particolare focus su questa tematica. È proprio l'analisi dei KPI che potrebbe decretare la crescita o la decrescita di un'impresa.

### 1.1.2 I CPI

Se, da una parte, i KPI permettono di avere una visione globale della direzione strategica intrapresa dall'azienda, di certo non si possono trascurare le aspettative del singolo cliente. È proprio da questa necessità che nasce il concetto di CPI, ovvero *Customer Performance Indicator*, che è possibile considerare come una sorta di KPI ma a livello più basso.

Con l'avvento delle nuove tecnologie, dei Big Data e della Data Science, ad oggi i dati delle aziende sono moltissimi e riescono a descrivere nel dettaglio il pubblico. L'obiettivo è conoscere al meglio il proprio cliente e ciò che sta cercando, in modo tale che questo gli possa essere offerto nel momento migliore. Avere soltanto i dati del cliente non è sufficiente a tale scopo ma è necessario operare con un'analisi strategica delle informazioni. È proprio per questo che la Market Analysis sta acquistando un peso sempre maggiore all'interno delle aziende.

Avere dei reparti che si occupano solo ed esclusivamente di questo job permette di avere una panoramica assoluta della tipologia di clienti ai quali si sta cercando di vendere un prodotto e/o servizio. Puntualmente si riesce a carpire quali sono le necessità del cliente in un preciso istante. Tutto questo viene supportato dalla miriade di strumenti di analisi messi a disposizione dalla Data Science, che alla fine, permettono di avviare campagne di vendita mirate al cliente specifico per fare breccia in quella che è la reale necessità di quest'ultimo e in modo tale da far acquisire valore all'azienda.

È evidente che la corretta analisi di questi CPI porti al miglioramento dei KPI e, di conseguenza, avvicina sempre più l'organizzazione al pieno raggiungimento degli obiettivi prefissati. Il concetto fondamentale è, dunque, che il valore dei KPI e quello dei CPI sono direttamente proporzionali; al crescere del secondo inevitabilmente, crescerà anche il primo.

## 1.2 Overview dei principali KPI utilizzati

In questa sezione viene introdotta una panoramica sui principali KPI impiegati nel progetto oggetto della presente tesi, avendo cura, nei capitoli successivi, di esplicitare tutto il processo di analisi dei requisiti, progettazione ed implementazione.

All'interno dell'azienda di apparecchi medicali al quale questa tesi fa riferimento si ha un'attenzione quasi maniacale ai KPI nonché ai CPI che vengono elaborati. Si premette che tutti gli indicatori vengono riportati all'interno di *Dashboard* specifiche, che mostrano le rispettive analisi numeriche e i rispettivi andamenti temporali: ma tutto ciò sarà meglio discusso nei capitoli successivi.

### 1.2.1 CRM

Prima di esporre i principali KPI è doveroso fare un escursus su una componente molto utilizzata in ambito di Marketing Analytics, ovvero il CRM. Acronimo di *Customer Relationship Management*, esso fa riferimento ad una strategia per la gestione di tutti i rapporti e le interazioni di un'azienda che hanno luogo con i clienti potenziali ed esistenti (da qui vedremo la classificazione di tutta la *Base Clienti* in tre cluster diversi).

L'implementazione di un ottimo CRM aiuta le aziende a rimanere in contatto con i clienti, a semplificare i processi e a migliorare la redditività. Inoltre, queste soluzioni aiutano ad offrire una customer experience eccellente lungo tutto il ciclo di vita del cliente, a ogni interazione di marketing, vendita, e-commerce e servizio clienti.

I KPI di cui si parlerà sono tutti allocati in un sistema di questo tipo, in quanto permettono di avere sia una panoramica globale delle caratteristiche di tutta la Base Clienti nonché una conoscenza puntuale delle peculiarità di uno specifico cliente.

### 1.2.2 KPI-Pedia

La KPI-Pedia può essere considerata un po' il fulcro del sistema di CRM perchè raccoglie al suo interno (da qui il termine *Pedia*, quasi ad intendere una sorta di enciclopedia dei KPI) un vasto elenco di indicatori che permettono all'impresa di raggiungere gli obiettivi prefissati, nonché di soddisfare le esigenze del cliente.

#### Contactability

Se, da una parte, è importante conoscere le caratteristiche del singolo cliente e la sua storia all'interno dell'azienda, altrettanto lo è avere una visione complessiva della Base Clienti, che, da ora in poi, verrà identificata come *Customer Base*. La *Contactability* non è altro che una componente dell'intera infrastruttura all'interno della quale vengono collocati i KPI di cui sopra e; permette di avere idea delle modalità specifiche con le quali i clienti dell'azienda di apparecchi medicali possono essere contattati.

I canali di comunicazione diretta che vengono considerati per le interazioni con il cliente sono le seguenti:

- *DM* Direct Mail: consiste nell'indirizzo della residenza del cliente;
- *Email*: consiste nell'indirizzo di posta elettronica del cliente;
- *Phone*: corrisponde al numero di telefono fisso del cliente;
- *Mobile*: consiste nel numero di telefono di un dispositivo mobile del cliente.

Inoltre i clienti possono essere contattati da due entità diverse che sono le seguenti:

- *Shop*: è il negozio fisico di riferimento del cliente.
- *Call Center*: è la centrale telefonica alla quale i clienti possono rivolgersi per informazioni e da cui possono ricevere campagne pubblicitarie.

Conoscere lo storico dei clienti che vengono contattati da uno Shop piuttosto che da un Call Center, e soprattutto il canale sul quale il contatto è stato protratto è fondamentale in termini di business per adattarsi alle necessità dell'acquirente. Questo è tanto più importante se si considera che l'azienda di apparecchi medicali offre, in generale, prodotti per persone di una fascia d'età particolarmente avanzata, per la quale avere dei contatti funzionanti è di notevole rilevanza. Proprio da qui deriva la necessità di avere KPI come quelli sotto riportati:

- *Customer w/ at least one valid channel*: indica il numero dei clienti che sono presenti all'interno della Customer Base con almeno un canale di comunicazione valido. Questo KPI è di fondamentale importanza perchè avere una numerica eccessivamente bassa sull'intero totale lascia intendere che ci sono molti clienti che, in fase di interazione, non comunicano alcun contatto valido, rendendo complesso l'inoltro di campagne pubblicitarie e informazioni. Come si vedrà in seguito, la validità di un canale di contatto dipende da alcune regole di validità (*RegExp*) implementate a livello Back-End, suggerendo come, magari, l'assenza di canali validi possa essere attribuita a negligenze dei dipendenti che non annotano gli estremi di contatto in maniera opportuna, o a volte anche issue/bug lato Team Deloitte.
- *Customer Updated*: indica il numero di clienti che hanno subito un update su uno dei canali di comunicazione in seguito ad un'interazione avvenuta con uno Shop o un Call Center.
- *Customer Valid Updated*: indica il numero di clienti che hanno effettuato un update su uno dei canali che, però, sia valido secondo le regole di validità stabilite in fase di analisi dei requisiti.
- *Customer Useful Updated*: indica il numero di clienti che hanno effettuato un update valido su uno dei canali che, però, risulta essere utile rispetto alla situazione subito precedente a quella in cui viene fatta questa interazione. L'utilità si ha se il canale permette l'arricchimento delle informazioni che si hanno su quel particolare cliente.
- *New Customer Updated*: indica il numero di nuovi clienti, che sono cioè alla prima registrazione all'interno del database, che entrano con almeno un canale di comunicazione valido. Esso permette dunque, di avere una panoramica sulla situazione dei clienti alla loro registrazione.
- *% of Customer Valid/Useful Updated*: indica, in termini percentuali, quanti clienti hanno ricevuto un update valido/utile sulla totalità dei clienti presenti nella Customer Base. Questo KPI viene calcolato per ciascun canale di comunicazione.
- *Customer Valid Privacy*: indica il numero di clienti che forniscono in maniera valida il consenso al trattamento dei dati e della privacy. Viene calcolato sulla base dei consensi forniti dal cliente su ciascun canale di riferimento.

Anche se sarà meglio affrontato nei capitoli successivi, si anticipa che queste tipologie di indicatori vengono calcolati in maniera diversa in base ai requisiti che vengono forniti dai vari Paesi. Questo perchè l'azienda di apparecchi medicali ha optato per una soluzione *Cross Country*, in cui ognuna delle "Country" coinvolte comunica al Team Deloitte i requisiti da rispettare (come, ad esempio, le regole di validità sopracitate) in base anche ai regolamenti interni del Paese stesso.

## Customer Specific KPI

Il calcolo di indicatori *Customer Specific* permette di avere la conoscenza necessaria per ciascun cliente e di indirizzare in maniera corretta delle campagne di marketing mirate; ciò rappresenta uno degli obiettivi principali di un sistema CRM.

Di seguito un elenco dei principali KPI che saranno meglio discussi nei capitoli successivi:

- *COM LAST HA SALE DATE*: indica la data di ultima vendita del cliente. Il prefisso *COM* sta ad indicare la categoria di appartenenza del KPI, in questo caso Commercial.
- *COM LAST HA SALE PRODUCT DS*: corrisponde alla descrizione del prodotto che è stato consegnato al cliente nell'ultima vendita effettuata. Nella maggior parte dei casi si tratta di *HA*, ovvero Hearing Aids.
- *COM LAST HA SALE RECENCY DS*: indica la fascia temporale all'interno della quale è avvenuta l'ultima vendita. In generale i cluster si dividono di trimestre in trimestre per le vendite effettuate nell'ultimo anno (0-3, 3-6, 6-9, 9-12 month) e di anno in anno per le vendite successive al primo anno passato (1-2, 2-3, 3-4, 4-5, 5-6, 6-7, 7-8, 8-9, 9-10, 10+ year).
- *COM LAST HA SALE QUOTATION*: indica il prezzo del dispositivo acquistato nell'ultima vendita dal cliente. Una vendita potrebbe avere anche più di un prodotto; quindi, questo KPI permette di capire qual è stata, al netto della eventuale scontistica applicata, la spesa dell'acquirente.
- *INT LAST INTERACTION DATE*: indica qual è stata la data in cui il cliente ha effettuato la sua ultima interazione, in uno shop o chiamando un call center. In questo caso il prefisso è riferito al termine Interaction.
- *INT LAST APPOINTMENT DATE*: indica qual è stata l'ultima data in cui il cliente ha preso un appuntamento presso uno shop.
- *TRI LAST TRIAL DATE*: indica quando è avvenuto l'ultimo trial, e quindi l'ultima prova da parte del cliente. Questo perchè l'azienda offre la possibilità di completare un periodo di prova prima dell'effettivo acquisto di un dispositivo. Il prefisso *TRI* sta per Trial.
- *TRI LAST TRIAL STATUS*: in merito a quanto sopra riportato, fornisce informazioni sullo stato dell'ultimo trial, ovvero se è ancora in corso, se si è concluso con l'effettiva vendita o con un nulla di fatto.
- *TST LAST TEST DATE*: indica la data dell'ultimo test che è stato effettuato in sede dal cliente. Il prefisso indica l'abbreviativo di Test.
- *TST LAST TEST RESULT DS*: è una descrizione del risultato dell'ultimo test condotto sul cliente. In base al risultato, viene proposta a quest'ultimo una particolare soluzione medicale.
- *TST LAST TEST STATUS*: indica l'azione che è stata effettuata dal cliente in seguito all'ultimo test effettuato. Un esempio potrebbe essere l'acquisto di un apparecchio in seguito alla degradazione di una caratteristica fisica del paziente oppure un semplice trial di prova.

Il termine *LAST*, ripetuto nei KPI sopra, indica la situazione ad oggi dell'individuo come cliente dell'azienda. Come si vedrà successivamente, i KPI che contengono questo termine, sono sottoposti a ricalcolo ogni giorno. Se, da una parte, questi KPI

hanno la tendenza a cambiare, in base alle attività svolte dal cliente e registrate all'interno del database, è presente anche uno storico dove sono collezionati tutti i record di uno specifico cliente. Questa componente permette di risalire alla storia dello stesso all'interno dell'azienda e comprendere, dunque, per ogni giorno passato, qual è stata la situazione del cliente in esame.

## Descrizione del progetto complessivo

*Questa sezione è dedicata alla descrizione generale del progetto in un'ottica globale ponendo particolare attenzione al processo realizzativo dei KPI e al loro utilizzo all'interno del Front-End.*

### 2.1 Caratteristiche generali e obiettivi del progetto

Il progetto nasce dalla necessità dell'azienda di apparecchi medicali di avere un'unica grande infrastruttura in grado di poter non solo gestire i dati di tutti i clienti ma anche di sfruttare questi ultimi in maniera strategica, in modo tale da raggiungere una posizione da leader all'interno del mercato in questione.

Quindi non si tratta della sola realizzazione e gestione di un ampio database, ma anche dello sviluppo di un sistema di *Data Analytics* in grado di estrarre informazioni e conoscenza utile al business e alla crescita dell'azienda. Quest'ultima non opera soltanto all'interno del territorio italiano, ma a livello internazionale. Le principali sedi risultano essere dislocate nella maggior parte dei paesi europei, negli Stati Uniti d'America e in Oceania.

È proprio da qui che nasce la necessità di sviluppare una soluzione cosiddetta *Cross Country*, ovvero un sistema che non sia lo stesso per ciascun paese, ma che, partendo da una base progettuale identica, sia adattabile a ciascuna nazione interessata. Il motivo di tale scelta è legato anche alle diverse policy di gestione dei dati sensibili che si hanno nei vari paesi, il che comporta inevitabilmente, diverse regole di validità, nonché di calcolo dei KPI. Le nazioni per le quali è stata sviluppata una soluzione di Data Analytics risultano essere: Italia, Francia, Spagna, Regno Unito, Belgio, Svizzera, Germania, Portogallo, Olanda, Stati Uniti d'America, Nuova Zelanda e Australia. Tutta la trattazione che farà seguito a questa sezione fa riferimento ai paesi europei, il cui progetto viene identificato con la sigla *OneBigData - EMEA*.

In linea di massima lo scopo del progetto è quello di effettuare operazioni di *ETL* (Extract/Transform/Load) sui dati sorgente in modo tale da ottenere tabelle contenenti dati ripuliti da record errati ed eccessivi valori *null*, e quindi una qualità del dato generalmente superiore. Tutte queste tabelle vengono, poi, utilizzate all'interno di un sistema che risulta essere comune a tutti i paesi europei e all'interno

del quale sono implementate regole per il calcolo dei principali KPI, o comunque di campi che vengono manipolati con un ulteriore step aggiuntivo per ottenere le misure desiderate. Una volta che si hanno a disposizione tutte le informazioni, è possibile processare le tabelle all'interno del database. Il software di Business Intelligence prescelto sarà, dunque, in grado di comunicare e attingere i dati dal database al quale punta e, quindi, utilizzare i dati per la realizzazione di cruscotti ad hoc.

Le dashboard rappresentano il prodotto finale di questa catena progettuale e vengono utilizzate dal management per:

- avere una visione generale di quello che sta succedendo allo stato attuale all'interno di un particolare paese, in termini di contattabilità dei clienti;
- vedere lo storico delle azioni che un particolare cliente ha intrapreso all'interno dell'azienda;
- inviare i KPI ad un team che si occupa solo ed esclusivamente dell'erogazione di campagne specifiche nei confronti dei clienti, sulla base delle analisi condotte sugli indicatori ricevuti.

L'obiettivo del progetto è, dunque, quello di realizzare questo sistema complessivo di *Data Analytics* in modo che l'azienda riesca a gestire, nel migliore dei modi, le relazioni con i propri clienti cercando di offrire loro un servizio adeguato e quanto più personalizzato sulla base delle loro attuali necessità. Riuscire a fare questo significa, inevitabilmente, un maggiore appeal nel settore di mercato interessato e un vantaggio strategico sui principali concorrenti.

Nella Figura 2.1 viene riportata l'infrastruttura che costituisce il Back-End e il Front-End del progetto che saranno meglio discussi nelle sezioni successive.

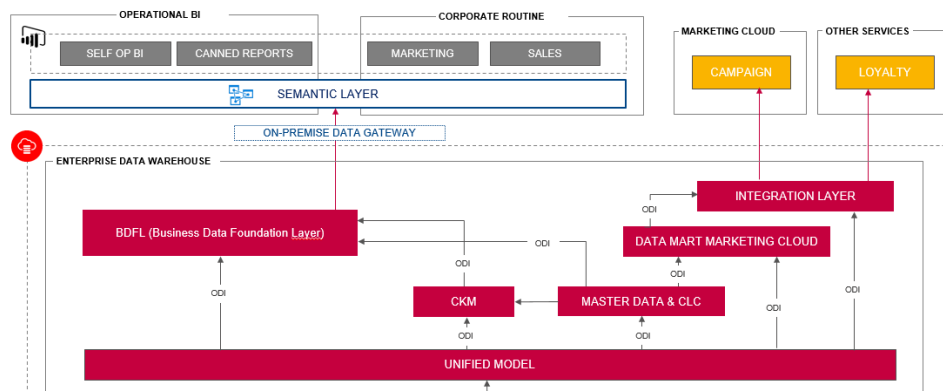


Figura 2.1. Schema logico dell'infrastruttura realizzata

## 2.2 Back-End

A partire dalla Figura 2.1 si evidenzia come la parte di back-end risulta essere tutta quella costituita da blocchi in rosso, rinominata *Enterprise Data Warehouse*.



Si evidenziano le seguenti componenti:

- *Unified Model*: comunemente indicato con l'acronimo *UM*, si tratta di un modello in grado di storicizzare i record dopo aver unificato tutti i sistemi sorgente sottostanti all'interno dei quali vengono raccolti tutti i dati del cliente. Le tabelle sorgente alle quali si farà riferimento sono *partizionate* per nazioni e fanno parte di questo modello.
- *Customer Knowledge Management*: indicato come *CKM*, risulta essere una struttura intermedia in grado di supportare la creazione delle misure e dei KPI.
- *CLC*: rappresenta il *Customer Lyfecycle*, ovvero una struttura in grado di registrare lo stato di avanzamento del cliente. Gli step risultano essere tre:
  - *Lead*: è un customer che non ha mai effettuato un acquisto o una prova di nessun apparecchio medicale. Rientrano in questo cluster tutti i clienti che hanno preso un appuntamento.
  - *Prospect*: risultano essere dei prospecti, ovvero individui che hanno effettuato la prova di un apparecchio, e quindi, sono potenziali clienti.
  - *Customer*: è un cliente vero e proprio, ovvero un individuo che abbia fatto l'acquisto di almeno un apparecchio medicale.
- *BDFL*: acronimo di *Business Data Foundation Layer*, è un DataMart in cui sono consolidate le più importanti informazioni ed entità di business, come vendite, trial, test ed appuntamenti.

### 2.2.1 Le principali tabelle sorgente

Come anticipato, le principali tabelle sorgente fanno parte dello *Unified Model*. Di seguito un elenco delle più importanti:

- *UM C U B ADDRESS*: è una sorta di rubrica all'interno della quale sono riportate informazioni come il codice univoco che indentifica il customer, l'indirizzo di residenza (sono previsti più campi in modo tale che il cliente possa comunicare anche più di un indirizzo), il numero di telefono fisso, il numero di telefono mobile, l'email. Inoltre, sono presenti diversi timestamp che indicano, ad esempio, la data in cui è avvenuto l'inserimento del record sul database, la data fin cui il record risulta essere valido, etc.
- *UM C U B ADDRESS BOOK*: molto simile alla precedente, ma in questa tabella vengono riportate informazioni anagrafiche come il nome e l'età del cliente, a cui fanno seguito i consensi che sono stati dati da quest'ultimo in fase di registrazione.
- *UM C U B ITEM HISTORY*: è una sorta di storico che contiene tutti gli ordini che sono stati effettuati dai clienti dell'azienda.
- *UM C U B ACTIVITY*: contiene uno storico di tutte le attività, e quindi delle interazioni, che un cliente ha avuto con shop e call center. Presenta, quindi, dei campi che indicano il tipo di attività che è stata svolta, quando è stata svolta, qual è stato l'esito etc.

Ci sono, poi, tabelle che non sono delle vere e proprie sorgenti, cioè che non presentano record che vengono direttamente storicizzati all'interno dello *Unified Model*, ma che vengono popolate a partire da quest'ultimo attraverso operazioni di ETL particolarmente complessi. Il motivo per il quale vengono inserite in questa

sezione, è comunque, legato al fatto che sono tabelle che costituiranno delle sorgenti per alcuni *mapping* presenti nell'ambiente *Oracle Data Integrator*. Le principali risultano essere:

- *FT SALES*: è una tabella in cui sono riportate tutte le informazioni relative agli acquisti effettuati dai clienti dell'azienda, come la tipologia di prodotto acquistato, la quantità, il prezzo, la data di vendita, etc. La particolarità di questa tabella è che non è ordinata per codice cliente, bensì per codice HK (Hash Key); quindi è sempre necessario collegarla con un'altra tabella che permette di associare in maniera univoca un HK ad un codice cliente, in modo tale da rendere più semplice la lettura della tabella stessa.
- *FT ALL APPOINTMENTS*: contiene tutte le informazioni relative agli appuntamenti presi dai clienti.
- *FT TEST ACTIVITY*: riporta all'interno tutte le informazioni sulle attività di test svolte dai clienti.
- *FT TRIAL*: contiene tutti i dati relativi ai trial condotti dai clienti, comprese le informazioni legate all'esito della prova, necessarie a capire se i clienti sono più propensi all'acquisto dopo aver fatto una prova.

Anche queste tabelle, come quelle dell'UM, risultano essere partizionate per nazione.

## 2.2.2 Gestione e integrazione dei dati con Oracle Data Integrator

### ETL

Per poter portare all'interno delle dashboard un dato di alta qualità, è inevitabile condurre delle operazioni di *ETL*. Nella Figura 2.2 viene illustrato uno schema che riassume tutte le fasi di ETL.

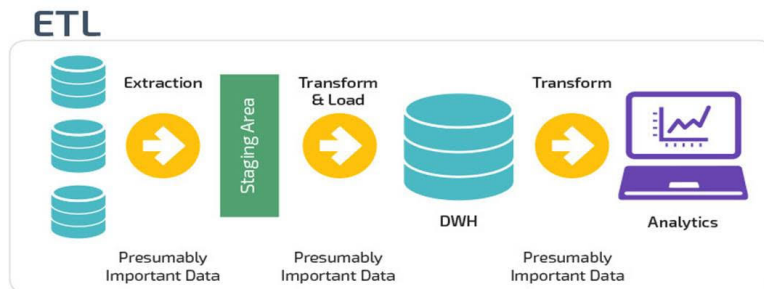


Figura 2.2. Le fasi di ETL

Riassumendo brevemente un processo di ETL comprende:

- *Estrazione*: i dati estratti vengono collocati in destinazioni quali Data Lake o Data Warehouse.

- *Trasformazione*: i dati sporchi e ammassati subiscono una serie di processi e attività in modo tale che vengano plasmati in un set pronto per la fase successiva di caricamento.
- *Caricamento*: prevede il caricamento dei dati estratti e trasformati in una nuova destinazione.

### ODI - Oracle Data Integrator

*Oracle Data Integrator* (Figura 2.3) più semplicemente *ODI*, è un tool di ETL prodotto da ORACLE. La principale caratteristica di questo tool è che offre un ambiente grafico per integrare, gestire e mantenere una base di dati all'interno di un sistema di Business Intelligence.



**Figura 2.3.** ODI - Logo

Gli elementi più importanti di questo tool sono i *mapping* e i *package*. I primi esprimono come il dato viene trasformato, a partire dalla tabella sorgente, in delle tabelle di destinazione. Nella Figura 2.4 viene riportato un esempio di mapping; le tabelle che non hanno collegamenti entranti rappresentano le sorgenti, mentre quelle che non hanno collegamenti uscenti rappresentano le destinazioni.

All'interno di un mapping è possibile utilizzare una serie di operatori che permettono di integrare, filtrare e modificare i dati a monte creando un vero e proprio flusso che termina all'interno della tabella di destinazione. Di seguito, vengono riportate tutte le operazioni che possono essere effettuate, (Figura 2.5), tra cui:

- *Join*: corrispondono ad operazioni insiemistiche tra le tabelle; sono possibili Left, Right, Inner e Full join.
- *Aggregate*: è l'equivalente di una Group By; raggruppa record e valori per campi.
- *Filter*: permette di filtrare i dati secondo particolari condizioni.
- *Expression*: sono utili per creare delle condizioni come, ad esempio, le regole di validità.

I *package*, invece, definiscono il modo con il quale i mapping vengono eseguiti, fungendo come dei veri e propri “vigili” che dirigono il “traffico” dei dati. Un package può essere, a sua volta, formato da altri package, e così via, fino ad arrivare allo step più in basso in corrispondenza del quale si trovano i mapping. I package, infine, stabiliscono se le esecuzioni dei mapping avvengono in maniera sincrona o asincrona. Nella Figura 2.6 viene riportato un esempio di package.

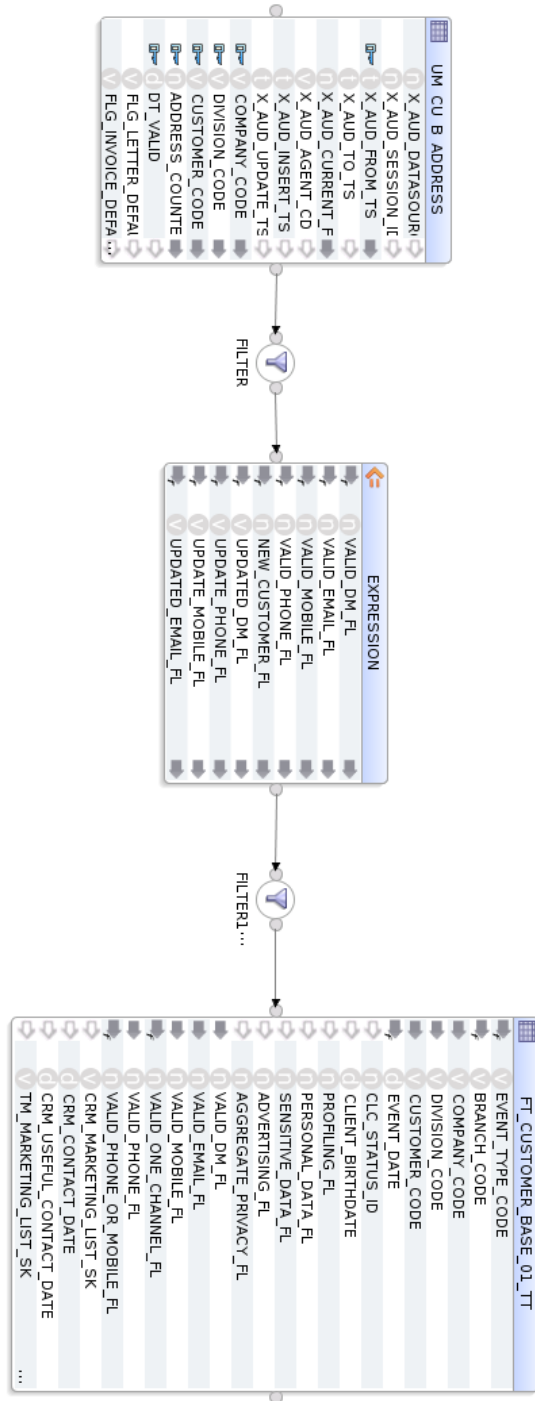
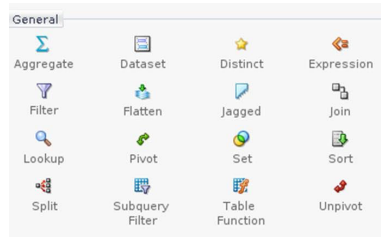
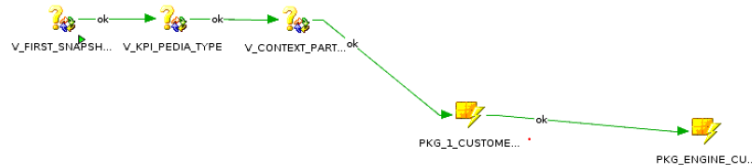


Figura 2.4. Esempio di mapping in ODI



**Figura 2.5.** Insieme degli operatori in ODI



**Figura 2.6.** Esempio di package in ODI

Nei package è possibile anche inserire delle variabili che stabiliscono quando debba essere eseguito un flusso piuttosto che un altro.

All'interno del tool è possibile sostare in una finestra di *Designer*, che è quella che permette di vedere tutti i mapping e i package presenti all'interno del progetto, e una finestra di *Operator*, che invece, mostra, in seguito al lancio di mapping o package, l'esito del run; questo può essere positivo (i dati vengono scritti correttamente all'interno delle tabelle target) o negativo (il mapping/package si “rompe” a causa di qualche particolare errore interno del tool, come, ad esempio, una sintassi SQL errata all'interno di qualche condizione in una espressione).

All'atto dell'esecuzione di un mapping possono essere intraprese delle azioni standard definite in fase di progettazione e che vanno sotto il nome di *Knowledge Module* note anche come *KM*. Queste azioni possono essere di pre-esecuzione (vengono, cioè, effettuate prima che il mapping venga effettivamente eseguito), intra-esecuzione (intraprese durante l'esecuzione) e post-esecuzione (effettuate dopo l'esecuzione del mapping). Un classico esempio di KM è il troncamento della tabella target, effettuato prima o durante il run del mapping.

Per completezza, nella Figura [2.7](#) è riportata una visuale complessiva del tool ODI.

### ODI nel progetto

Nella Figura [2.1](#) si nota come tutte le componenti all'interno dell'Enterprise Data Warehouse vengano implementate in ODI. Tutte le tabelle sorgenti dello *Unified Model* sono contenute tra i *Model* di questo tool. A partire da queste, insieme alle informazioni del *BDFL*, è possibile realizzare mapping e, di conseguenza, package ad hoc che permettono di popolare delle tabelle target in cui la qualità del dato risulta essere di gran lunga superiore a quello di partenza.

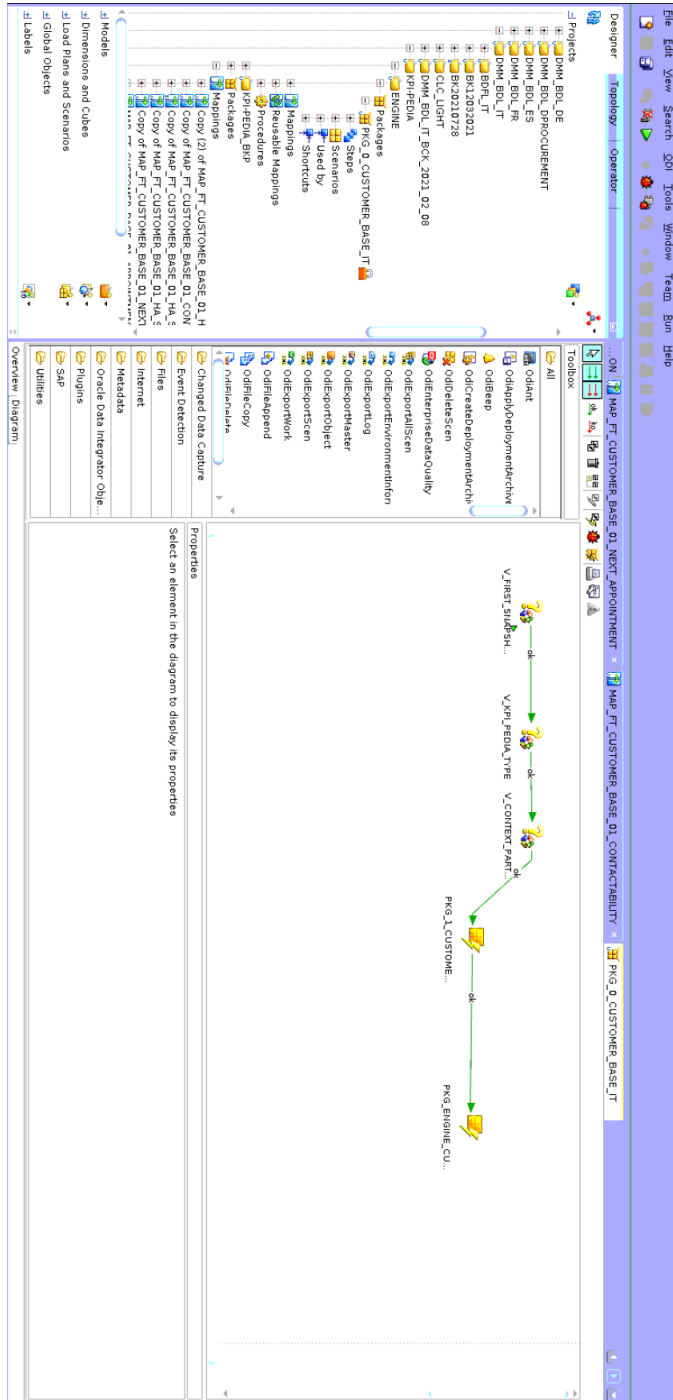


Figura 2.7. Visual ODI

Le modalità con cui i dati dalla sorgente risalgono a destinazione, e quindi come si comportano i vari mapping e package, saranno discusse nei capitoli successivi.

Il dato pronto può essere, dunque, manipolato per la creazione di misure e per il calcolo di KPI che saranno, poi, visualizzati all'interno di un tool di BI.

### 2.2.3 Le principali tabelle target

Di seguito sono riportate le principali tabelle di destinazione che vengono popolate dai mapping e package ODI.

- *FT DATA ENRICHMENT*: è una delle tabelle più importanti all'interno dell'intero sistema, perchè è quella attorno alla quale ruota l'intera *Contactibility* dell'azienda. All'interno di essa si trovano, per ciascun cliente tutti gli arricchimenti che sono stati effettuati. La tabella, quindi, memorizza se sono stati effettuati update su uno o più canali di comunicazione (DM, Email, Phone, Mobile, Privacy) e se questi aggiornamenti sono validi e utili sulla base delle regole di validità implementate in precedenza. Ovviamente, sono presenti anche tutta una serie di date che indicano il giorno in cui l'update è stato fatto, quando è stato registrato nel database, chi ha effettuato l'update e da dove (shop o call center con relativo codice identificativo), etc.
- *FT CUSTOMER BASE*: è la tabella che contiene lo storico di ciascuna attività intrapresa dal cliente a partire dal primo contatto. Al suo interno sono riportati tutti i KPI specifici dei clienti che fanno riferimento ad una precisa data del passato. Questa tabella permette, quindi, di avere una foto in qualsiasi periodo del passato del cliente in esame. Il nome risulta essere abbastanza parlante; esso, infatti, sta ad indicare una struttura che contiene l'intera base clienti dell'azienda.
- *FT CUSTOMER BASE ACTUAL*: molto simile alla precedente offre, però, uno snap della situazione attuale di ciascun cliente registrato all'interno del database aziendale. Esso presenta dei KPI che vengono aggiunti in un secondo momento con altri mapping a partire dalla FT CUSTOMER BASE, che funge, in questo caso, da sorgente.

### 2.2.4 Creazione delle misure con Tabular Editor

*Tabular Editor* (Figura 2.8) è uno strumento esterno di *Microsoft Power BI* che permette di creare e gestire facilmente dei modelli tabulari attraverso un editor intuitivo e leggero. Una visualizzazione gerarchica mostra tutti gli oggetti del modello tabulare organizzati in cartelle di visualizzazione, con il supporto alla modifica delle proprietà a selezione multipla e l'evidenziazione della sintassi DAX.

Attraverso l'utilizzo del linguaggio DAX, *Tabular* permette di creare, in maniera molto rapida, delle misure a partire dai campi presenti nelle principali tabelle target.

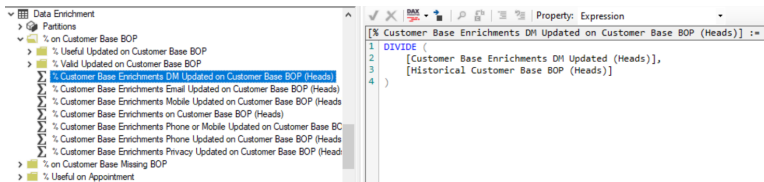
In questo modo è possibile arricchire di misure il modello già esistente generato in precedenza con le fasi di ETL e produrre un file .bim che può essere installato in modo tale da allineare il tutto all'interno di un repository condiviso al quale tutti gli utenti del progetto hanno accesso.



**Figura 2.8.** Tabular Editor - Logo

Il modello creato, inoltre, corrisponde a quello che viene utilizzato da Power BI come *Data Model*. Essendo, infatti, Tabular Editor uno strumento esterno di Power BI, la comunicazione tra i due tool risulta essere molto efficiente.

Nella Figura 2.9, viene riportato un esempio di misura realizzata in linguaggio DAX all'interno di Tabular Editor. Nello specifico, si tratta della misura percentuale di update effettuati sul canale DM (Direct Mail) rispetto alla totalità dei clienti presenti all'interno della Customer Base.



**Figura 2.9.** Esempio di misura realizzata su Tabular Editor

Tutte le misure seguono una sintassi di questo tipo e sono fondamentali per la realizzazione di dashboard semplici ed intuitive.

## 2.3 Front-End

L'azienda di apparecchi medicali ha scelto Microsoft Power BI (Figura 2.10) come tool standard di Business Intelligence da utilizzare per la realizzazione di tutte le dashboard. Il tool è una raccolta di servizi software, app e connettori che interagiscono per trasformare le origini dei dati non correlate in un insieme di informazioni coerenti, visivamente accattivanti e interattive. I dati potrebbero essere un foglio di calcolo Excel o una raccolta di data warehouse basati sul cloud o ibridi locali. Power BI consente di connettersi facilmente alle origini dei dati, di visualizzare e scoprire le informazioni importanti e di condividerle con tutti gli utenti o con quelli necessari.



**Figura 2.10.** Microsoft Power BI - Logo



### 2.3.1 Creazione delle dashboard

Le dashboard vengono realizzate a partire dall'applicazione desktop, avendo cura di avere il puntamento verso il corretto database a seconda della nazione che si sta trattando. Si ricorda, infatti, che la quasi totalità delle tabelle di back-end sono partizionate per paese. Questo fa sì che venga stanziato un database diverso per ciascuna nazione, come contenitore dei dati esclusivi per quella partizione, mentre il modello dei dati rimane sempre il medesimo.

Tutte le misure realizzate a livello Tabular sono visibili tra i campi del report e possono essere trascinati all'interno di oggetti visivi per la loro visualizzazione.

Una volta che la dashboard risulta essere ultimata, essa viene pubblicata all'interno di un workspace creato nell'account PBI del corporate aziendale. Dal workspace può essere generata un'App (App Power BI) il cui accesso è fornito e riservato ai membri dell'organizzazione. Sono, pertanto, i report presenti nelle App ad essere consegnati come deliverable nei confronti dell'azienda.

### 2.3.2 Overview delle principali dashboard

In questa sezione viene presentata una breve overview sui principali dashboard consultati dall'azienda e che riguardano l'ambito *Contactability*.

Nella Figura 2.11 viene riportata una dashboard nella quale è presente una overview dei customer all'interno di uno specifico periodo temporale. Nella tabella in basso a destra sono riportate tutte misure in percentuali, che suggeriscono il numero di update per ciascun canale rispetto alla totalità della Customer Base. Il plot in basso a sinistra, invece, non rappresenta altro che l'andamento degli arricchimenti effettuati seguendo un'aggregazione mensile nei rispettivi anni. In alto ci sono, invece, sei semplici indicatori percentuali che rendono queste misure più immediate ed impressive.

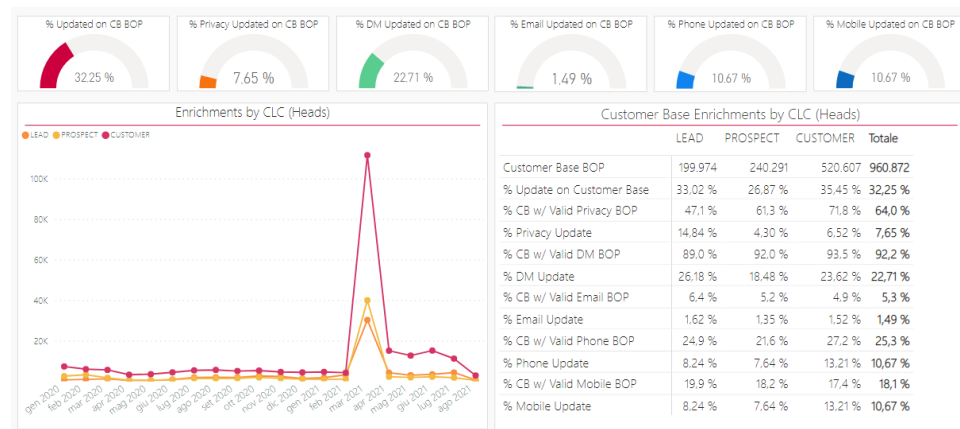
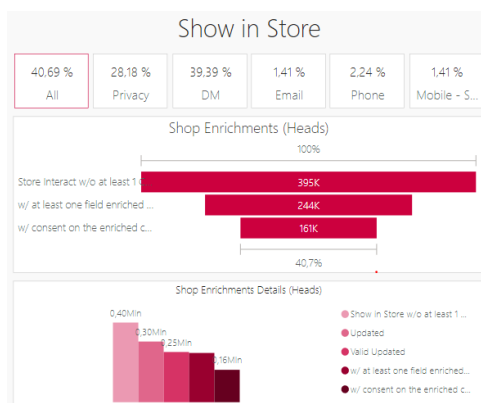


Figura 2.11. Customer Overview

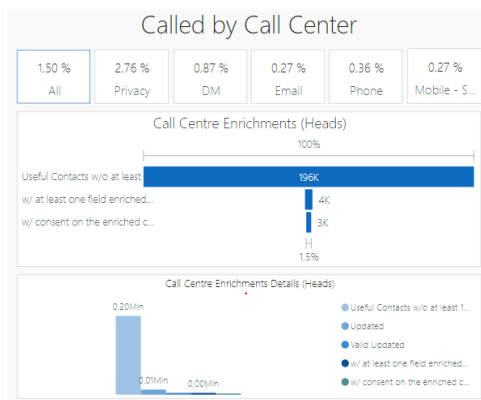
Nella Figura 2.12 sono riportate informazioni in relazione alle interazioni che sono avvenute per mezzo di uno Store e, quindi, in un negozio. Nella parte centrale

si trovano tre “gradini”. Il primo indica il numero complessivo di interazioni senza almeno un canale valido di comunicazione. Il secondo, invece, denota il numero di interazioni che ha portato all’arricchimento di almeno un campo. Il terzo, infine, indica quanti di questi ultimi hanno ricevuto anche il consenso. Nella parte inferiore si ha, invece, una visualizzazione diversa delle stesse misure con l’aggiunta di update e update validi. Sopra si trovano, invece, delle “cards” che mostrano, in percentuale, il numero di appuntamenti utili fissati senza quel canale di comunicazione valido. Anche in questo caso tutte le misure dipendono dai periodi temporali che vengono selezionati.



**Figura 2.12.** Show in Store

Le stesse misure sono ripetute per le interazioni avvenute tramite Call Center; la relativa visualizzazione è riportata nella [Figura 2.13](#)



**Figura 2.13.** Called by Call Center

Quelle analizzate sono solo una parte della totalità di dashboard realizzate; esse saranno meglio discusse nei capitoli successivi.

## Analisi dei requisiti relativi alla componente KPI-Pedia

*Questo capitolo è dedicato all'analisi dei requisiti per definire le funzionalità che il prodotto deve offrire, e quindi i requisiti che devono essere rispettati, stabiliti dall'azienda committente.*

### 3.1 Contactability: regole di validità

Come discusso in precedenza, la *Contactability* è una componente di elevata importanza all'interno del progetto, perchè è quella che conta al proprio interno tutta una serie di KPI che permettono di avere una overview completa sulla gestione delle relazioni con i clienti. Si ricorda, ad esempio, la presenza di indicatori che esprimono in percentuale il numero di clienti che ha effettuato degli update su un particolare canale di comunicazione (DM, Email, Phone, Mobile). Essa, inoltre, permette di conoscere se questi update siano solo validi o validi e utili.

La validità dei canali di comunicazione è un aspetto fondamentale in ambito *CRM* (Customer Relationship Management) soprattutto quando le interazioni col cliente, come semplici chiamate o l'inoltro di campagne pubblicitarie, vengono fatte tramite sistemi automatizzati, che inviano le informazioni ai contatti prelevati dal database.

In tal senso per il business è estremamente importante comunicare dei requisiti chiari e specifici che permettano di stabilire la validità o meno dei canali di comunicazione. Avere un numero troppo elevato di contatti non validi aumenta le probabilità che i clienti non vengano correttamente contattati, facendo venire meno l'interesse da parte di questi ultimi nei confronti dell'azienda, rallentando di gran lunga il processo di raggiungimento degli obiettivi di business prefissati.

Da qui deriva proprio l'interesse dell'azienda di apparecchi medicali di fornire delle *regole di validità* oggettive e quanto più sensitive ad eventuali errori di registrazione all'interno dei database sorgente.

Essendo il progetto di natura cross country queste regole non sono identiche per ciascun paese; come esempio banale, basti pensare che il formato dei numeri di telefono e degli indirizzi risulta essere diverso da paese a paese. Nelle sezioni successive si farà riferimento al paese Italia, in modo anche da avere una maggior comprensione di queste regole.

### 3.1.1 Le RegExp in SQL

SQL (Structured Query Language) è un linguaggio standardizzato per database basati sul *modello relazionale*, progettato per le seguenti operazioni:

- creare e modificare schemi di database;
- inserire, modificare e gestire dati memorizzati;
- interrogare i dati memorizzati;
- creare e gestire strumenti di controllo e accesso ai dati.

Contrariamente a quanto si pensa, SQL non è un linguaggio dedicato alla sola interrogazione, in quanto alcuni operatori permettono di creare, gestire e amministrare database.

Una *regular expression* (abbreviato in regex o regexp, talvolta chiamata rational expression) è una sequenza di caratteri che forma uno schema di ricerca, principalmente usato per trovare corrispondenze e nelle funzioni di “trova e sostituisci”. Si può anche usare come generatore di dati, seguendo il concetto delle regular expression inverse e per produrre dati di test casuali da usare in database di test.

A livello logico, nel progetto in esame, le regexp sono utilizzate per generare un insieme di “stringhe” valide secondo le regole stabilite con l’azienda. Attraverso un operatore di confronto si stabilisce se il dato interessato corrisponda o meno con le caratteristiche di quell’insieme. L’operatore in questione è *REGEXP\_LIKE* che consente di verificare la corrispondenza di espressioni regolari nella clausola WHERE di un’istruzione SELECT, INSERT, UPDATE o DELETE.

### 3.1.2 Canali di comunicazione validi

Di seguito si vedranno le regole di validità per i quattro canali di comunicazione facendo riferimento alla nazione Italia.

#### Valid DM

Nella Figura 3.1 viene esaminato l’indirizzo principale (DM), per il quale non si hanno regexp, essendo la condizione molto più rilassata.

```

CASE
  WHEN
    (
      UH_CU_B_ADDRESS.ADDRESS_LINE1 IS NOT NULL
      OR
      UH_CU_B_ADDRESS.ADDRESS_LINE2 IS NOT NULL
      OR
      UH_CU_B_ADDRESS.ADDRESS_LINE3 IS NOT NULL
      OR
      UH_CU_B_ADDRESS.ADDRESS_LINE4 IS NOT NULL
    )
  THEN 1
  ELSE 0
END

```

Figura 3.1. Regole di validità per Direct Mail

Il *valid dm* è, chiaramente, un semplice flag che per ciascun record, sarà popolato dal valore 1 nel caso in cui almeno uno dei campi della tabella sorgente contenente gli indirizzi (ADDRESS\_LINE1, ADDRESS\_LINE2, ADDRESS\_LINE3, ADDRESS\_LINE4) non sia nullo; altrimenti il flag sarà impostato a 0. Si tratta di una condizione rilassata in quanto anche la sola presenza di un carattere spurio determina che il flag sia pari a 1.

### Valid Email

Nella Figura 3.2 è riportata la regola di validità per l'indirizzo di posta elettronica. In questo caso è presente la REGEXP\_LIKE.

```
CASE
WHEN REGEXP_LIKE (UH_CU_B_ADDRESS.EMAIL , '^[a-zA-Z0-9_\-\.]+@[a-zA-Z0-9_\-\.]+\.[a-zA-Z]{2,5}$' )
THEN 1
ELSE 0
END
```

**Figura 3.2.** Regole di validità per Email

Qualora il record, nella tabella sorgente, abbia campo EMAIL, popolato da una stringa formata da un insieme qualsiasi di lettere (sia maiuscole che minuscole), compresi caratteri speciali come '-', ',', ':', seguito da una chiocciola e subito dopo da un nuovo insieme come il precedente seguito da un punto e una stringa di lettere di lunghezza compresa tra due e cinque, allora il flag del *valid email* sarà posto a 1, altrimenti sarà posto a 0.

### Valid Phone

Si ricorda che Phone fa riferimento al numero di telefono fisso; nella Figura 3.3 è riportata la corrispettiva regola di validità.

Nella tabella sorgente sono presenti, per ciascun record, i campi PHONE1, PHONE2, PHONE3 e MOBILE. I tre campi PHONE sono dovuti al fatto che, in fase di registrazione, potenzialmente, un cliente potrebbe comunicare anche più di un numero di telefono.

Il flag *valid phone* sarà impostato ad 1 quando almeno uno dei quattro campi soddisfa le condizioni espresse dalla REGEXP\_LIKE. Per PHONE1, PHONE2 e PHONE3 le condizioni sono le stesse. Se il campo ha una lunghezza maggiore o uguale a 7, non inizia con una stringa "0039" e "0000", ma con una stringa "+39" o "00", seguita da un 0 e da uno a tre caratteri compresi tra 0 e 9 (a costituire il prefisso), a sua volta seguito da quattro a sette numeri compresi tra 0 e 9 (compresi eventuali caratteri speciali come '-'), allora il numero di telefono fisso è valido.

In realtà tale regola persiste uguale anche per il campo MOBILE; questo perchè, in fase di registrazione, un numero fisso potrebbe essere erroneamente inserito nel campo dedicato ai telefoni mobili. In questo modo si riuscirebbero a contattare anche i clienti che rientrano in questa casistica avendo una mappatura completa.

### Valid Mobile

Nella Figura 3.4 sono riportate le regole di validità per il Mobile che corrisponde al numero di telefono cellulare.

```

CASE
WHEN
(
LENGTH(UH_CU_B_ADDRESS.PHONE1) >= 7 AND
UH_CU_B_ADDRESS.PHONE1 NOT LIKE '0039%' AND
UH_CU_B_ADDRESS.PHONE1 NOT LIKE '0000%' AND
REGEXP_LIKE (UH_CU_B_ADDRESS.PHONE1 , '^((00|\+39[\. ]??)?(0{1}[0-9]{1,3})[\s|\.\|-]?\d{4,7})$' )
)
OR
(
LENGTH(UH_CU_B_ADDRESS.PHONE2) >= 7 AND
UH_CU_B_ADDRESS.PHONE2 NOT LIKE '0039%' AND
UH_CU_B_ADDRESS.PHONE2 NOT LIKE '0000%' AND
REGEXP_LIKE (UH_CU_B_ADDRESS.PHONE2 , '^((00|\+39[\. ]??)?(0{1}[0-9]{1,3})[\s|\.\|-]?\d{4,7})$' )
)
OR
(
LENGTH(UH_CU_B_ADDRESS.PHONE3) >= 7 AND
UH_CU_B_ADDRESS.PHONE3 NOT LIKE '0039%' AND
UH_CU_B_ADDRESS.PHONE3 NOT LIKE '0000%' AND
REGEXP_LIKE (UH_CU_B_ADDRESS.PHONE3 , '^((00|\+39[\. ]??)?(0{1}[0-9]{1,3})[\s|\.\|-]?\d{4,7})$' )
)
OR
(
LENGTH(UH_CU_B_ADDRESS.MOBILE) >= 7 AND
UH_CU_B_ADDRESS.MOBILE NOT LIKE '0039%' AND
UH_CU_B_ADDRESS.MOBILE NOT LIKE '0000%' AND
REGEXP_LIKE (UH_CU_B_ADDRESS.MOBILE , '^((00|\+39[\. ]??)?(0{1}[0-9]{1,3})[\s|\.\|-]?\d{4,7})$' )
)
THEN 1
ELSE 0
END

```

Figura 3.3. Regole di validità per Phone

```

CASE
WHEN
(
REGEXP_LIKE (UH_CU_B_ADDRESS.MOBILE , '^((00|\+39[\. ]??)?3\d{2}[\. ]??\d{6,7})$' )
)
OR
(
REGEXP_LIKE (UH_CU_B_ADDRESS.PHONE1 , '^((00|\+39[\. ]??)?3\d{2}[\. ]??\d{6,7})$' )
)
OR
(
REGEXP_LIKE (UH_CU_B_ADDRESS.PHONE2 , '^((00|\+39[\. ]??)?3\d{2}[\. ]??\d{6,7})$' )
)
OR
(
REGEXP_LIKE (UH_CU_B_ADDRESS.PHONE3 , '^((00|\+39[\. ]??)?3\d{2}[\. ]??\d{6,7})$' )
)
THEN 1
ELSE 0
END

```

Figura 3.4. Regole di validità per Mobile

Anche in questo caso si fa utilizzo della REGEXP LIKE che agisce sui campi MOBILE, PHONE1, PHONE2, PHONE3 dei record della tabella sorgente. Se almeno uno di questi campi presenta un primo carattere equivalente a “3”, seguito da due cifre, un’eventuale carattere speciale e, subito dopo, una stringa formata da sei o sette cifre, e in assenza di un prefisso iniziale, allora il numero è valido. In caso di presenza di prefisso, la stringa iniziale deve essere “+39” o “00”; altrimenti il numero ha validità nulla.

Anche in questo caso il motivo per cui si fa riferimento anche ai campi PHONE1, PHONE2 e PHONE3, nonostante si tratti di Mobile, è legato al possibile inserimento errato di questo numero nei campi adibiti a ricevere il numero fisso.

Come anticipato, queste regole di validità sono diverse nazione per nazione; questo perchè il prefisso che si ha in Italia, ad esempio, è diverso da quello che si ha in Francia, piuttosto che in Germania. Pertanto, queste regole sono anche dette “country specific”. Nonostante il contenuto sia sostanzialmente diverso, la struttura della regexp rimane sempre la stessa.

## 3.2 Requisiti dei principali KPI

I principali KPI customer specific sono interessati nella loro nomenclatura da un prefisso che ne indica la categoria di appartenenza; ad esempio *COM* sta per Commercial, *INT* per Interaction, *TRI* per Trial e *TST* per Test.

Si ricorda, inoltre, che tutta la base clienti risulta essere clusterizzata in tre categorie: Lead, Prospect e Customer.

È proprio da qui che viene uno dei più importanti requisiti chiesti dall'azienda committente, ovvero la coerenza dei KPI rispetto allo status del cliente in esame.

Essendo i KPI commercial legati all'ambito commerciale e, di conseguenza, alle vendite, è chiaro che un Lead (cliente che ha preso un appuntamento ma che non ancora compie alcuna prova e, tantomeno, alcun acquisto) non possa avere questi KPI valorizzati. Al tempo stesso, anche un Prospect (cliente che ha fatto una prova ma ancora nessun acquisto) non può averli. Inoltre, anche i Customer (clienti che hanno effettuato almeno un acquisto) possono essere soggetti a problematiche come l'assenza dei KPI commerciali.

I casi più eclatanti che si verificano in tal senso sono la presenza di:

- *Customer senza Last Sale Date*: un Customer non può non avere una data di ultima vendita, perchè da requisito è definito tale se, nel suo storico, ha effettuato almeno un acquisto. Quando si verifica un problema del genere è molto probabile che non sia sbagliata la classificazione del cliente (Customer) bensì, ad essere errata, è qualche condizione all'interno dei mapping che "blocca" la data di ultima vendita non facendola risalire dove desiderato.
- *Prospect senza Last Trial Date*: un cliente prospetto non può non avere una data di ultimo trial, perchè esso è definito tale se e solo se ha effettuato almeno una prova, ma non ha ancora acquistato nulla. Anche in questo caso l'errore potrebbe essere riconducibile ad una data che viene bloccata all'interno del mapping per qualche particolare condizione errata.
- *Prospect con Last Sale Date*: un Prospect non può avere una data di ultima vendita, perchè è soltanto un potenziale Customer futuro. In questo caso potrebbe essere errata la classificazione del cliente, perchè difficilmente il flusso dei mapping tende ad aggiungere dati piuttosto che filtrarli.
- *Lead senza Last Appointment Date*: un Lead è, da requisito, un soggetto che ha preso almeno un appuntamento, ma ancora esegue una prova che lo classificherebbe come prospetto. Ne segue che, per definizione, egli non possa non avere la data dell'ultimo appuntamento. Anche quando si verificano questi casi è molto probabile che la data in questione venga erroneamente filtrata.
- *Lead con Last Sale Date*: è fondamentalmente assimilabile alla casistica del Prospect senza Last Sale Date, con la differenza che il Lead non può mai aver effettuato un trial, come da requisito.
- *Lead con Last Trial Date*: da requisito il Lead non può aver fatto alcuna prova e, quindi, non può avere una data di ultimo trial.

Un altro importante requisito in ambito KPI-pedia è legato al tema degli Shop. Tra i principali KPI Customer Specific ve ne è, infatti, uno che indica lo shop, e quindi il negozio, di riferimento per quel cliente. Questo KPI è di fondamentale importanza, perchè filtrando una dashboard per Nazione-Regione-Area-Shop, è

possibile risalire in maniera puntuale a tutti i clienti che hanno interazioni con quel particolare negozio. Se alcuni clienti non presentano questo indicatore, rimangono tagliati fuori dal filtro e, quindi, non possono essere correttamente classificati in uno shop di riferimento.

Da qui la particolare attenzione, in fase di analisi dei requisiti, per la tematica degli shop.

La mancanza di questa informazione per un particolare cliente può essere dovuta all'errato aggancio di tabelle durante il flusso di esecuzione dei mapping e package, a particolari condizioni che inconsapevolmente vanno a filtrare il dato oppure all'effettiva assenza di quest'ultimo nelle tabelle sorgente.

### 3.3 Requisiti delle dashboard

Supposto che tutti i requisiti relativi al Back-End vengano soddisfatti, lato Front-End non si può che avere un dato pulito pronto ad essere visualizzato.

I requisiti che si hanno, pertanto, sulle dashboard si limitano allo stile di questi ultimi nonché alla loro prontezza in fase di selezione dei filtri.

Le principali dashboard non devono avere pagine troppo complesse, piene ed articolate, ma devono seguire una linea quanto più snella possibile senza sminuire l'evidenza dei dati. Gli oggetti visivi da utilizzare si limitano ad andamenti temporali, tabelle, tabelle pivot, istogrammi semplici, indicatori percentuali e filtri dei dati. In molti report è necessario avere la possibilità di effettuare drill-down sulle aree di riferimento e, di conseguenza, sugli shop, avere una selezione multipla e non aggregata (da ... a ...) per i mesi degli anni, passare da dati che riguardano Customer a quelli che interessano Lead e Prospect, e scegliere di visualizzare le interazioni da store piuttosto che da call center.

Un aspetto importante è la reattività delle dashboard; più esse rispondono velocemente alla selezione dei filtri, tanto più alto sarà il valore della user experience. Dal momento che, nella maggior parte dei report, le visualizzazioni rimangono le medesime, e quelli che cambiano sono i dati all'interno degli oggetti visivi, è fondamentale scegliere una soluzione che renda quanto più leggera possibile la dashboard. I bookmark sono pulsanti che permettono di navigare tra pagine del report nascoste soggette a duplicazione e sostituzione delle misure all'interno delle visual interessate. L'utilizzo dei bookmark comporta un aumento del numero di pagine e la necessità di dover propagare eventuali modifiche manualmente ad ogni pagina, ma tende a migliorare la prontezza della dashboard nello scambio delle misure. Un'altra soluzione consiste nell'utilizzo di un meccanismo di switch vero e proprio che vada in maniera istantanea a sostituire le misure della visual interessata. Questa tecnica è più complessa da mettere in pratica; essa però, diminuisce il numero di pagine complessivo del report ma rischia di renderlo meno pronto e veloce.

Le dashboard realizzate presentano tutte l'utilizzo di bookmark.



## Progettazione della componente KPI-Pedia

*In questo capitolo viene affrontata la parte di progettazione dei principali mapping e package in ambiente di test, fulcro dell'intero Back-End nonchè del progetto stesso.*

### 4.1 Progettazione dei principali mapping in ambiente di test ODI

L'intera parte di progettazione viene realizzata all'interno del software *ODI* (Oracle Data Integrator) in ambiente di test. Questo ambiente permette di progettare tutti i mapping e package necessari per popolare in maniera corretta le tabelle target necessarie. In questo modo è possibile individuare errori di diversa tipologia prima che le componenti progettate vengano rilasciate in ambiente di produzione causando non pochi problemi nei deliverable consegnati all'azienda committente.

Tra le principali tabelle target si ricorda la *FT DATA ENRICHMENT*, che contiene, al proprio interno, tutte le informazioni relative agli update che sono stati effettuati da ciascun cliente, la validità degli update e la loro eventuale utilità. Questa viene popolata dall'omonimo mapping, di cui la parte iniziale viene riportata all'interno della Figura [4.1](#).

Le tabelle sorgente di questo mapping risultano essere le già discusse *UM CU B ADDRESS* e *UM CU B ADDRESS BOOK*. La prima consiste in una sorta di rubrica all'interno della quale, per ciascun cliente, sono riportati numeri di telefono, sia fisso che mobile, indirizzo di posta elettronica, indirizzo di casa, e così via. La seconda, invece, è un'anagrafica, e quindi vi sono dati come il nome del cliente, l'età ed i consensi che quest'ultimo in fase di registrazione ha garantito in corrispondenza dei canali di comunicazione (ad esempio; consenso sul PHONE pari a 'Y' equivale alla possibilità di contattare il cliente sul suo numero di telefono fisso, purchè quest'ultimo sia presente e valido).

Nel mapping si distinguono, dunque, due rami principali, quello inferiore guidato dalla *ADDRESS* e quello superiore guidato dalla *ADDRESS BOOK*.

Partendo da quello inferiore, si trova un filtro che permette la risalita a tutti i record "validi", secondo un particolare campo della tabella sorgente denominato *X\_AUD\_CURRENT\_FL*. Questo perchè nella sorgente potrebbero essere scritti dei record

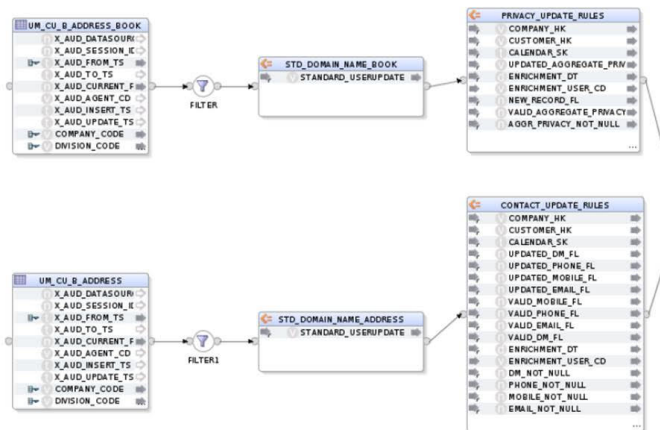


Figura 4.1. Flusso del mapping FT DATA ENRICHMENT

che non hanno validità in base a determinate policy di business, e quindi hanno l'identificatore sopraccitato *null*, e dunque devono essere bloccati in partenza. Successivamente si trova un'espressione che, attraverso un'operazione di *REPLACE( TRIM() )* permette di correggere il campo dedicato all'informazione sull'utente che ha effettuato la registrazione di quel record (*USER\_CD*). Di seguito una serie di ulteriori espressioni che permettono, innanzitutto, di decretare se sia stato effettivamente effettuato un update di un canale da quel cliente. In generale, le regole che determinano se sia stato effettuato un aggiornamento su un canale sono molto semplici, perchè verificano, semplicemente, se il campo sia null o meno. Nel primo caso l'update non è stato effettuato, mentre secondo caso l'update è stato effettuato. In questa parte sono anche presenti tutte le regole di validità discusse nel capitolo precedente, ovvero tutte le *regexp* che determinano la validità dei canali di comunicazione.

Per il ramo superiore, in maniera del tutto analoga, si trovano filtro ed espressione iniziali. Cambiano, però, le condizioni successive. Qualora il cliente fornisca almeno un consenso positivo in uno dei campi *FLG\_PRIVACYPERSDATA*, *FLG\_ADVERTISING*, *FLG\_SENCDATA*, *FLG\_PROFILING* (tutti riferiti alla gestione dei dati personali), allora l'update sul flag della privacy è posto a 1. Sotto una particolare combinazione di consensi dati, l'update può anche essere considerato valido da parte dell'azienda che ha il via libera nel trattamento dei dati del cliente. In questa parte viene popolato, anche, un campo molto importante che viene utilizzato in una particolare pagina di dashboard, ovvero *NEW\_RECORD\_FL*, un flag che determina se il record registrato all'interno del database sia riconducibile alla prima registrazione del cliente, che, dunque, verrà successivamente etichettato come *NEW CUSTOMER*.

Successivamente a questo flusso vi sono diversi operatori, come una *Union* e una *Aggregate*, che permettono di "schiacciare" tutti i record relativi a ciascun codice cliente, in modo tale da avere nella destinazione una sola riga per ogni customer. Di fatto viene popolata la tabella *FT DATA ENRICHMENT*.

Essendo tale mapping sviluppato in ambiente di test, una volta realizzato deve essere lanciato. Al termine dell'esecuzione (in generale l'esecuzione termina entro pochi minuti), per verificare il corretto popolamento della tabella, si eseguono dei

check sul database attraverso il tool *SQL Developer* che permette di stabilire una connessione al repository dove sono contenute le tabelle scritte dai mapping, offrendo la possibilità di aprire fogli SQL dove poter eseguire query per interrogare le strutture interessate.

In linea di massima questi check possono essere condotti in maniera generale attraverso l'utilizzo di numeri o in maniera puntuale esaminando, nello specifico e per intero, un pool predefinito di customer. Nel primo caso vengono eseguite query di tipo *SELECT COUNT* che restituiscono dei conteggi che possono essere confrontati con le informazioni estrapolate dalle tabelle sorgente. Nel secondo caso, invece, si cerca di individuare dei record interessanti che permettono di analizzare il corretto funzionamento della logica di popolamento del mapping.

Tutti i mapping che riguardano, invece, i KPI Customer Specific sono presenti all'interno di una particolare cartella nell'ambiente ODI. Nella Figura 4.2 è riportata la lista di tutti questi mapping.

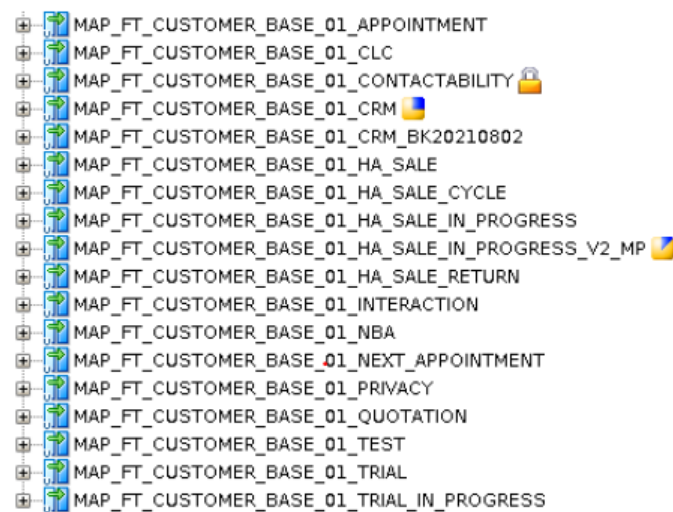


Figura 4.2. Lista dei mapping della KPI-Pedia

Tutti questi mapping, partendo da diverse tabelle sorgente, popolano la stessa tabella target *FT CUSTOMER BASE 01 TT*. Si tratta di una tabella di *staging*, ovvero una struttura necessaria all'archiviazione temporanea dei dati. Come si vedrà nella sezione successiva, infatti, questa tabella risulterà la sorgente di alcuni mapping presenti all'interno di package necessari al popolamento delle target effettive.

Nella Figura 4.3 viene riportato il flusso iniziale del mapping *HA SALE*, uno dei più importanti e delicati di questa cartella.

Nella tabella *FT-SALES* sono presenti tutte le informazioni relative agli ordini effettuati dai clienti dell'azienda, come la data di vendita, se il record è riferito ad un ordine di acquisto, lo shop di riferimento, l'impiegato che ha portato a termine l'ordine, la tipologia di prodotto, e così via. Poichè in questa tabella non sono presenti soltanto ordini di vendita (*SALES*) è presente un filtro denominato *SA* che consente di estrarre i soli record che abbiano come tipologia d'ordine, appunto, una

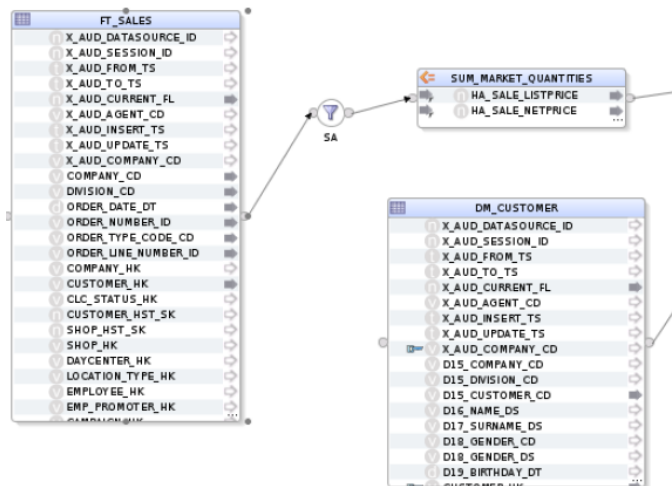


Figura 4.3. Flusso del mapping HA SALE

sale. Nell'espressione, invece, vengono, rispettivamente, calcolati, sulla base della quantità di prodotti inseriti in un ordine di vendita, il prezzo di listino e il prezzo al netto di eventuale scontistica applicata ai dispositivi.

Poiché la *FT.SALES* presenta soltanto gli hash key univoci attribuibili ai rispettivi customer, la tabella *DM\_CUSTOMER* viene utilizzata per essere collegata con la prima, in modo tale, che agganciando sull'hash key, si può attribuire anche per le sale un codice cliente a ciascun record, consentendo una più facile lettura della tabella target.

Nella parte successiva del mapping vengono utilizzate altre tabelle in join con *FT\_SALES* (che rimane sempre quella che guida) come la *UM\_CU\_B\_ITEM\_HISTORY* che è una sorta di storico degli ordini passati effettuati dai clienti dell'azienda e serve sostanzialmente a recuperare eventuali informazioni non contenute nella *FT\_SALES*. Infine, i dati vengono selezionati all'interno della *FT\_CUSTOMER\_BASE\_01 TT*.

Di seguito viene riportata una overview dei principali mapping presenti all'interno della Figura [4.2](#):

- **HA SALE IN PROGRESS**: in generale, quando un cliente realizza un acquisto, l'operazione è classificata come una sale, ma non è detto che questa non possa essere soggetta a cambiamenti. Una delle azioni più comuni, infatti, è lo storno di una vendita, ovvero la restituzione del prodotto da parte del cliente che, dopo un determinato periodo, non si trova più a suo agio con quel dispositivo. Da qui la necessità di avere un mapping che, in base alle operazioni di vendita/storno, vada a decretare lo status di quella vendita. Il KPI realizzato è **SALE IN PROGRESS FL**, ovvero un flag che determina se la vendita è ancora in corso o meno. È chiaro che una sale, dopo un determinato periodo, rimane tale decadendo i termini consentiti per lo storno.
- **INTERACTION**: è il mapping che si occupa di storicizzare le interazioni che sono state effettuate dal cliente. A questo mapping sono associati i KPI che, hanno come prefisso il termine **INT**.

- *NEXT APPOINTMENT*: questo mapping, a partire dalla tabella sorgente che contiene tutte le informazioni relative agli appuntamenti presi dai clienti dell'azienda, stabilisce la data nonché il luogo in corrispondenza del quale si terrà il prossimo appuntamento.
- *TEST*: mapping realizzato per individuare tutti i KPI che fanno riferimento ad informazioni legate ai test effettuati dai clienti. Si ricorda che sono identificabili mediante l'utilizzo del prefisso TST.
- *TRIAL*: si occupa di storicizzare le informazioni sui trial effettuati dai clienti (prove di apparecchi), come la data in cui è stato effettuato il trial, lo status di quest'ultimo, (se si è concluso con vendita o altro) e così via. Essi sono identificabili tramite il prefisso TRI.

Tutti questi mapping vengono ripetuti analogamente nella partizione di ciascuna nazione interessata. Questo perchè, a seconda dei requisiti forniti dal business locale, potrebbero cambiare le regole di validità, le informazioni nelle tabelle sorgenti e le caratteristiche dei KPI.

## 4.2 Progettazione dei principali package in ambiente di test ODI

. Se, da una parte, i mapping consentono di stabilire come il dato viene trasformato, i package fungono da “controllori” stabilendo il modo con il quale i mapping vengono eseguiti. In generale, i mapping non vengono mai lanciati singolarmente, se non per effettuare dei check immediati su parte dei dati nelle tabelle target. Ciò che viene effettivamente eseguito sono i package, che, a loro volta, possono contenere altri package fino ad arrivare al livello più basso in corrispondenza del quale si trovano i mapping. L'utilizzo di variabili permette di automatizzare l'esecuzione dei mapping all'interno dei package. Infatti, in base al valore di una variabile, il package può decretare l'esecuzione di un particolare flusso di mapping piuttosto che di un altro, e se questi vengono eseguiti in maniera sincrona o asincrona.

In ambito KPI-Pedia i package che devono essere eseguiti, qualora si volesse avere riscontro degli indicatori nelle tabelle target, sono due ovvero:

- *CUSTOMER BASE ALL EVENTS*;
- *ENGINE*.

### 4.2.1 Package CUSTOMER BASE ALL EVENTS

Nella Figura [4.4](#) è riportato il package *CUSTOMER BASE ALL EVENTS*. Il nome è abbastanza autoesplicativo e fa riferimento alla presenza di ulteriori package che permettono di collezionare tutti gli eventi associabili ad un determinato customer. Tutti i package, qualora aperti, presentano la stessa struttura, formata dal solo mapping omonimo, visti nella sezione precedente. In altre parole, questa componente permette di racchiudere in un'unica esecuzione tutti i mapping della KPI-Pedia, gestendone il run.

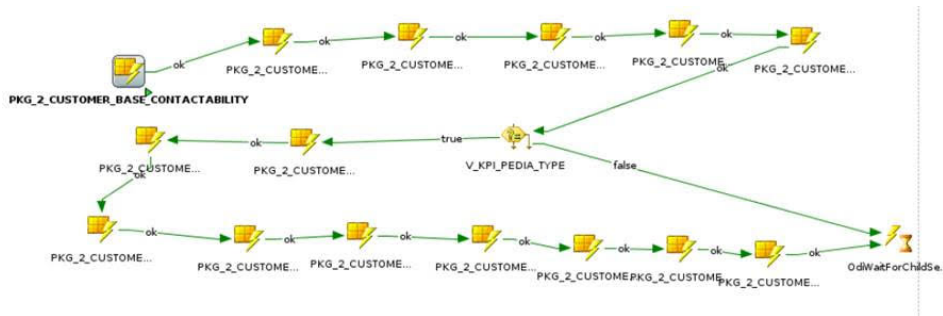


Figura 4.4. Package CB All Events

Com'è possibile notare, ciascun package è caratterizzato da una freccia uscente e da una entrante (ad eccezione del primo). Il verso di percorrenza della freccia non determina l'ordine di esecuzione del flusso. Infatti, i primi sei package, a partire da quello iniziale, vengono eseguiti contemporaneamente. Nel momento in cui questi primi sei finiscono l'esecuzione (non tutti hanno stessa durata, che dipende anche dal numero di record che vengono scritti a destinazione), supposto che terminino tutti senza errori, viene valutata una variabile chiamata *V\_KPI\_PEDIA\_TYPE*. Se il valore è posto a 0 allora l'esecuzione termina in quanto erano necessarie soltanto le informazioni dei primi sei package e, in questo caso, si parla di *KPI-Pedia Light*. Se, invece, il valore è 1 ("settabile" manualmente da SQL Developer) sono richieste delle informazioni aggiuntive all'interno della tabella target; per questo vengono eseguiti i mapping contenuti nei package successivi, sempre in contemporanea tra di loro. Se, al termine dell'esecuzione, non sono avvenuti errori (visibili nella finestra Operator di ODI) allora il package è terminato con successo; altrimenti, è necessario indagare tra i vari mapping per localizzare l'errore (molte volte complesso, a causa dell'elevata generalità dei messaggi di errore del software).

La tabella target interessata da questo package è la *CUSTOMER BASE 01 TT*, che, si ricorda essere una tabella di staging utilizzata successivamente dal package.

Dal momento che i mapping interessati in questo package sono country specific; anche questo deve essere ripetuto in maniera analoga per ciascuna nazione interessata.

### 4.2.2 Package ENGINE

Nella Figura 4.5 viene visualizzato il package *ENGINE*, che è caratterizzato da tre flussi paralleli che vengono eseguiti in sequenza.

I package del primo flusso sono accessori al popolamento di ulteriori tabelle di staging che presentano dati trasformati a partire da quelli contenuti nella *FT CUSTOMER BASE 01 TT*, popolata dal package visto nella sezione precedente.

Il secondo flusso (quello centrale), chiaramente, viene eseguito soltanto se il primo termina in maniera corretta, ed è il fulcro del package. Qui si trovano le logiche per popolare le due principali tabelle già ampiamente discusse; ovvero *FT CUSTOMER BASE* e *FT CUSTOMER BASE ACTUAL*.

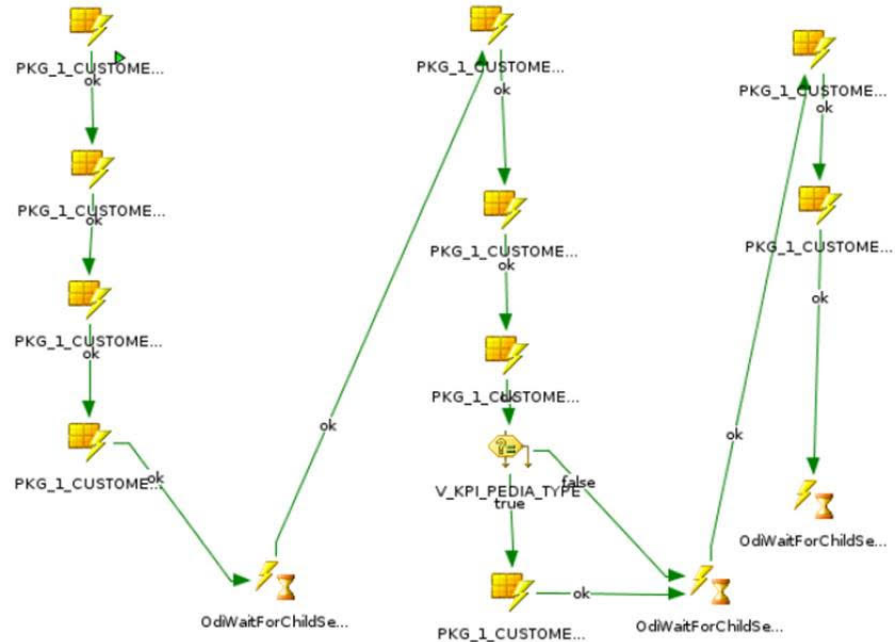


Figura 4.5. Package ENGINE

Nella Figura 4.6 è riportata l'espansione del primo package del flusso centrale, all'interno del quale sono presenti i mapping che permettono il popolamento di *FT CUSTOMER BASE*.



Figura 4.6. Espansione package CUSTOMER BASE COMPUTATION

Le sorgenti di questi mapping risultano essere sempre tabelle di staging che presentano dati via via trasformati. In questo caso, vengono attuate delle operazioni che permettono di storicizzare i record per ciascuna attività intrapresa dal cliente nei confronti dell'azienda di apparecchi medicali. I record sono, infatti, suddivisi, o meglio partizionati, per *Branch Code*, che determina la tipologia di attività

alla quale i dati fanno riferimento (Sale, Test, Trial, Appointment, etc.). Si crea, così, un'intera struttura dati che racchiude al proprio interno l'intera base clienti dell'azienda.

Anche questa tabella segue il consueto partizionamento per nazione.

Il package successivo nel flusso di mezzo è quello che consente il calcolo della tabella *FT CUSTOMER BASE ACTUAL*. La struttura interna è molto simile al package precedente, con la differenza che vengono utilizzati operatori come *Aggregate*, che permettono di schiacciare una serie di eventi in un unico record contenente le informazioni relative alle ultime (Last) attività svolte dal cliente (ultima vendita, ultimo appuntamento, ultima prova, ultimo test, etc.). La forza di questa struttura è legata al fatto che permette di visualizzare lo stato attuale del customer, in relazione alle sue ultime interazioni, all'interno di una tabella che presenta un solo record per customer e che, quindi, è molto più veloce da interrogare.

Se, anche, il flusso centrale viene eseguito correttamente, si hanno a disposizione le due strutture sopra citate che presentano al loro interno tutti i KPI.

È chiaro che la sola corretta esecuzione non è sinonimo di dati di qualità. Come anticipato, questi ultimi, così come i KPI, devono essere sottoposti a check rigorosi per garantirne la bontà e l'integrità.

Il package *ENGINE* risulta essere il medesimo per ciascuna nazione dell'area EMEA. Eventuali differenze tra un paese e l'altro possono essere gestite a livello di condizioni, e quindi di espressioni implementate all'interno dei mapping interessati, che leggono le variabili di contesto associate a ciascun paese.

In generale l'utilizzo di questo ambiente permette di testare ed effettuare check prima che i mapping e i package vengano rilasciati in produzione. Individuare e correggere un errore in fase di progettazione, e quindi di test, è meno dispendioso e a minor impatto rispetto alla correzione di quest'ultimo in fase di implementazione, soprattutto quando ogni singolo elemento dell'infrastruttura va ad incidere su tutti quelli che vi sono a valle.

Se tutti i test e i check del caso vanno a buon fine, mapping e package possono essere esportati in produzione.



## Implementazione della componente KPI-Pedia

*In questo capitolo viene affrontata l'implementazione in ambiente di produzione delle componenti sviluppate in fase di progettazione.*

### 5.1 Export da ambiente di test

Tutti i package e i mapping discussi nel capitolo precedente vengono sviluppati in ambiente di test all'interno del quale si ha visibilità solo su una piccola parte della totalità dei dati presenti nel database di origine. Ciò permette di testare la funzionalità delle componenti realizzate, non solo in merito alla corretta scrittura dei record nelle tabelle target, ma anche in merito di esecuzione di mapping e package. In ODI, l'utilizzo di particolari pattern permette di ridurre il tempo di run, aspetto fondamentale nel progetto in questione, dato l'elevato numero di package e mapping che concorrono.

L'ambiente di test è una sorta di banco di prova sul quale si possono testare tutte queste peculiarità, fino al raggiungimento degli obiettivi prefissati.

Una volta che un mapping o un package viene consolidato in test, esso è pronto per essere trasferito in un contesto che opera con la totalità dei dati a disposizione.

Tale contesto è denominato ambiente di *produzione* ed è quello in cui mapping e package girano per popolare in maniera completa le tabelle target, che saranno, poi, utilizzate in una fase successiva per la realizzazione dei report, anche attraverso operazioni di creazione delle misure.

Ogni volta che in Oracle Data Integrator viene apportata una modifica ad un package (e, in alcuni casi, anche mapping), deve essere rigenerato il cosiddetto *scenario*, che tiene traccia dello storico delle modifiche effettuate sulla componente. Uno scenario può essere considerato come una sorta di versione "light" del mapping o package, in quanto contiene il codice eseguibile, al netto delle modifiche effettuate.

Nel momento in cui la fase di test può essere considerata conclusa, è possibile prelevare l'ultima versione disponibile dello scenario di quel mapping/package e farne l'*export* in una cartella condivisa all'interno di ODI.

Il motivo per cui vengono esportati soltanto gli scenari deriva dalla necessità di avere un ambiente di produzione quanto più leggero e comprensibile, a diffe-

renza di quello di test, in cui la presenza di copie di prova e di backup rideuce la comprensibilità e l'ordine delle componenti.

Nel progetto oggetto della presente tesi, i mapping e i package, che vengono esportati, sono quelli visti nel capitolo precedente, ovvero tutti quelli accessori alla realizzazione della KPI-Pedia.

## 5.2 Rilasci

Esportare gli scenari dei package e dei mapping da ambiente di test in una cartella condivisa non è l'unico passaggio da effettuare per consolidare una componente precedentemente progettata.

Essi devono essere, infatti, *rilasciati*, e quindi importati dalla cartella condivisa, in ambiente di produzione. La fase di rilascio è una delle più importanti e delicate all'interno dell'intero progetto.

Un rilascio non può essere effettuato in maniera tempestiva quando necessario, ma segue sempre un percorso di pianificazione. Una volta che esso è stato pianificato, il team Deloitte deve comunicare all'azienda questa azione attraverso l'apertura di un ticket. In quest'ultimo vengono riportate le caratteristiche principali del rilascio, come, ad esempio, le componenti che vengono impattate dal rilascio e l'effort necessario per compiere lo stesso. Tali informazioni vengono comunicate all'azienda tramite aggiornamento di un file Excel condiviso dalle due parti. Successivamente viene fatta richiesta di un codice che, in maniera univoca, permette di identificare il rilascio soltanto se quest'ultimo soddisfa la disponibilità in termini di effort del personale dedicato. Infatti, i rilasci vengono condotti da un team specifico interno a Deloitte.

Per far sì che vengano meno problematiche di comprensione legate ai passaggi da realizzare per completare un rilascio, vengono fornite, assieme agli scenari che devono essere importati, due importanti documenti, che sono:

- *Documento di rilascio;*
- *Realse Notes.*

Il primo racchiude, in maniera generale, tutte le informazioni sul rilascio, come il suo identificativo, le componenti impattate, gli owner del rilascio (ciascun rilascio ha uno o più proprietari, che corrispondono ai componenti del team che si sono occupati di quel particolare stream di progetto) e gli step da eseguire per ultimare lo stesso. Nel secondo, invece, vengono riportate informazioni più dettagliate.

Durante la fase di rilascio, viene creata un'apposita chat tra gli owner e i componenti del team che prende in carico il rilascio, in modo tale da chiarire in corso d'opera eventuali dubbi circa gli step da eseguire.

È chiaro che durante il rilascio possono insorgere diverse problematiche; proprio per tale motivo, questa è un'attività che potrebbe occupare gran parte della giornata lavorativa di un componente del team.

Una delle problematiche maggiori che si possono presentare in un progetto di questo tipo è proprio la rottura dell'esecuzione di scenari rilasciati in produzione, in quanto l'interruzione di questo flusso potrebbe colpire in cascata tutte le componenti realizzate a valle, come misure su Tabular e dashboard.

## 5.3 Implementazione misure su Tabular Editor

Se, da una parte, tabelle target come *FT CUSTOMER BASE ACTUAL* contengono già KPI customer specific pronti per la visualizzazione su dashboard, invece altre, come *FT DATA ENRICHMENT*, hanno dati che devono essere manipolati per poter essere visualizzati su apposite dashboard.

In questo caso si passa su uno strumento interno di Microsoft Power BI, *Tabular Editor*, che, come già anticipato, permette di creare misure in linguaggio *DAX*, utilizzando i campi presenti nelle tabelle all'interno del *Data Model*.

Nella Figura 5.1 è riportata una lista di misure create su Tabular Editor, che hanno riscontro di utilizzo nella dashboard relativa alla *Contactability*.

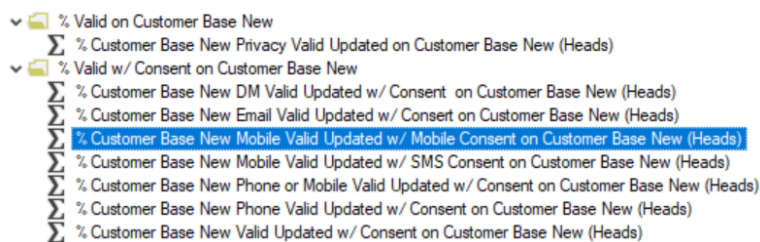


Figura 5.1. Lista di misure percentuali su *Tabular Editor*

La lista comprende tutte misure, in percentuale, relative ai nuovi clienti. In particolare, esse indicano il numero di update validi e con consenso del relativo canale (DM, PHONE, MOBILE, EMAIL, PRIVACY) rispetto alla totalità dei nuovi clienti (*Customer Base New*).

Nella Figura 5.2, invece, viene presentato un estratto di una delle misure sopra riportate in linguaggio *DAX*.

```
[% Customer Base New Mobile Valid Updated w/ Mobile Consent on Customer
1 DIVIDE (
2 [Customer Base New Mobile Valid Updated w/ Mobile Consent (Heads)],
3 [Customer Base New Customers (Heads)]
4 )
```

Figura 5.2. Misura update validi con consenso su canale MOBILE

La misura in esame prende il numero dei nuovi clienti che hanno effettuato un update valido sul canale MOBILE con consenso e lo divide (comando *DIVIDE*) per la totalità dei nuovi clienti (*Customer Base New Customers (Heads)*).

La struttura della misura è la medesima di tutte le altre della lista.

Come avviene per mapping e package, anche le misure su Tabular vengono prima sviluppate in ambiente di test. Dopo aver condotto i check del caso, anche Tabular è soggetto a rilascio quando necessario. In questo caso, ciò che viene importato in ambiente di produzione è proprio il file *.bim* che racchiude il modello da rilasciare.

Tale modello deve essere allineato rispetto a tutte le modifiche, che, vengono apportate in locale da tutti i membri del team che vi lavorano (operazioni di *fetch* e *pull* prima della modifica di una o più misure, e *commit* e *push* dopo le modifiche, per allineare il modello condiviso).

Il protocollo da seguire è analogo a quello visto nella sezione precedente; ovviamente, in questo caso, la componente impattata è proprio Tabular.

Terminati i rilasci, si hanno, dunque, all'interno del *Data Model*, campi pronti per essere inseriti all'interno degli oggetti visivi nei dashboard.

## 6

# Uno sguardo alle componenti realizzate

In questo capitolo verrà dato uno sguardo più approfondito ai principali dashboard realizzati.

## 6.1 Dashboard KPI-Pedia

La componente KPI-Pedia, si ricorda, è quella che permette di racchiudere tutta una serie di indicatori di natura customer specific, necessari all'azienda a capire, in qualsiasi istante, quale sia lo status del cliente interessato. La modalità con la quale si arriva ad ottenere questo insieme di KPI è stata già ampiamente discussa nei capitoli precedenti. Non rimane altro che vedere nello specifico come questi vengano visualizzati all'interno di un report PBI, il quale costituisce il deliverable da consegnare all'azienda.

Nella Figura 6.1 è riportata una delle pagine della dashboard in esame.

The screenshot shows a dashboard with three main sections. The top section is a table titled 'BDFL SALES' with columns: Company Code, ORDER\_NUMBER\_ID, Serial Number ID, ORDER\_TYPE\_CODE\_CD, Delivery Date, Market Type Descr, Product Band Descr, Product Family Descr, Product Type Descr, and Rechargeable Flag. It contains three rows of data and a 'Totale' row. The middle section is split into two panels: 'BDFL TEST' with columns 'ACTIVITY\_DATE\_DT' and 'M\_HEARING\_LOSS\_FL', and 'BDFL PRODUCT' with columns 'ORDER\_NUMBER\_ID' and 'Delivery Date'. The bottom section is titled 'KPI-Pedia Actual' and contains a grid of KPI values, many of which are '(Vuoto)'. A warning icon and text are visible above the KPI grid.

BDFL SALES									
Company Code	ORDER_NUMBER_ID	Serial Number ID	ORDER_TYPE_CODE_CD	Delivery Date	Market Type Descr	Product Band Descr	Product Family Descr	Product Type Descr	Rechargeable Flag
002	03XNPE		SA	25/01/2019	Libre	Excellence	Other		Rechargeable
002	03XNUB		SA	26/01/2019	Libre	Excellence	Other		Not Rechargeable
002	03XNUF		SA	26/01/2019	Libre	Excellence	Other		Rechargeable
Totale									

BDFL TEST		BDFL PRODUCT	
ACTIVITY_DATE_DT	M_HEARING_LOSS_FL	ORDER_NUMBER_ID	Delivery Date
02/01/2019	0.00	001001	08/10/2019
02/01/2019	1.00	001002	11/10/2019
03/01/2019	0.00	001003	24/10/2019
03/01/2019	1.00	001004	24/10/2019

KPI-Pedia Actual									
(Vuoto)	(Vuoto)	(Vuoto)	(Vuoto)	(Vuoto)	(Vuoto)	(Vuoto)	(Vuoto)	(Vuoto)	(Vuoto)
COM_AVG_HA...	COM_HA_SALE...	COM_HA_SALE...	COM_LAST_HA...	COM_LAST_HA...	COM_LAST_HA...	COM_LAST_HA...	COM_LAST_HA...	COM_LAST_HA...	COM_LAST_HA...
(Vuoto)	(Vuoto)	0	(Vuoto)	(Vuoto)	(Vuoto)	(Vuoto)	-6.172.72	06/08/2019	€ 3.254.55
COM_LAST_HA...	COM_LAST_HA...	COM_LAST_HA...	COM_LAST_HA...	COM_LAST_HA...	COM_LAST_HA...	COM_LAST_HA...	COM_LAST_QU...	COM_LAST_QU...	COM_LAST_QUOT...
€ 9.427.27	AUDEO M7...	-	(Vuoto)	(Vuoto)	(Vuoto)	(Vuoto)	(Vuoto)	(Vuoto)	0.00
COM_LAST_QU...	COM_LAST_QU...	COM_PROFILE...	COM_PROPENS...	COM_TOT_HA...	COM_LAST_HA...	COM_LAST_HA...	COM_LAST_HA...	COM_LAST_HA...	COM_HA_BOTH_SI...

Figura 6.1. Pagina con KPI di natura Commercial

Essa è denominata *KPI-Pedia Sales*, in quanto racchiude tutti i KPI *Commercial*, che fanno, appunto, riferimento agli acquisti sostenuti dal cliente.

A partire dal basso troviamo una prima tabella dati, denominata *KPI-Pedia Actual*. Quest'ultima contiene tutti gli indicatori con prefisso COM che si trovano all'interno della tabella target, precedentemente discussa, FT CUSTOMER BASE ACTUAL. In particolare, si hanno dati relativi all'ultima data di vendita, alla tipologia dell'ultimo prodotto venduto e a quanto è recente l'ultima vendita (cluster 0-3, 3-6, 6-9, 9-12 month, 1-2, 2-3 year etc.). Inoltre, vi sono informazioni di natura più economica, come il numero di apparecchi venduti, il prezzo di listino relativo ad un ordine di vendita ed il prezzo netto effettivamente pagato in virtù della particolare scontistica applicata. Infine, si trova un dato interessante relativo allo status della vendita, che permette di capire se quest'ultima è stata effettivamente portata a termine, oppure il prodotto o i prodotti oggetto della stessa sono stati stornati, in seguito a problemi rilevati dal cliente.

Nelle tre tabelle al di sopra, invece, vi sono informazioni relative al *BDFL* che permettono di ricondurre la correttezza del KPI in relazione ai dati che si hanno nella sorgente. I dati principali sono la nazione di appartenenza del cliente, l'identificativo e la data dell'ordine effettuato, la tipologia dell'ordine stesso, la fascia commerciale di appartenenza del prodotto, etc.

Tali informazioni non sono strettamente necessarie al team interno Deloitte, ma lo sono per l'azienda, la quale, non avendo accesso ai permessi per effettuare query sui database sorgente, riesce a testare la validità degli indicatori. Inoltre, non tutti i KPI possono essere testati seguendo questa procedura. Molti dati sorgente, necessari a compiere i check, non risalgono la catena di gestione del dato fino al campo all'interno del tabular. In questi casi va fornito all'azienda un estratto dei dati sorgente capace di completare il pacchetto informativo necessario alla validazione dei KPI sviluppati.

Tutta la parte di validazione degli indicatori avviene in sede di *UAT*, una fase del ciclo di vita del progetto che verrà discussa nel capitolo successivo.

Una precisazione è necessaria in merito all'ammontare dei campi *null* presenti nella dashboard. Ciò non costituisce un errore, in quanto non è detto che per un determinato cliente si abbia la totalità dei dati sulla sorgente. Possibile è anche la mancata risalita di un campo lungo tutta la catena di manipolazione del dato. Questi fattori, inevitabilmente, contribuiscono alla visualizzazione di valori *null* nella dashboard, i quali, però, non sono riconducibili ad errori.

## 6.2 Dashboard Contactability

Oltre ai KPI customer specific, devono essere visualizzati anche indicatori che permettano al business aziendale di monitorare la situazione generale della base clienti nazione per nazione. Per questo viene realizzata un'ulteriore dashboard correlata alla componente *Contactability* già discussa nei capitoli precedenti.

Per questo report si utilizzano visual più accattivanti dal punto di vista estetico, che permettono, a chi le guarda, di avere un riscontro quasi immediato dell'andamento dell'azienda in termini di gestione dei rapporti con i clienti.

Una delle principali pagine di questa dashboard è riportata nella Figura [6.2](#).



**Figura 6.2.** Trend Overview - Contactability

Si tratta di una pagina ampiamente consultata dal management; all'interno di essa vengono riportati i trend di aggiornamento dei vari canali di comunicazione, con un'aggregazione mensile.

Ciascuna barra dell'istogramma riporta il numero di update che sono stati fatti sul rispettivo canale di comunicazione oggetto del visual, in un mese di un anno selezionato tramite apposito filtro di tipo calendario. Si ricorda che i canali soggetti ad aggiornamento possono essere: PRIVACY, DM, EMAIL, PHONE, MOBILE. La particolarità che si nota nella pagina è una simile ripartizione degli update su PHONE e MOBILE. Ovviamente ciò non è un errore, in quanto, come è stato visto in sede di analisi dei requisiti, è possibile che, per la nazione interessata, la regola di update del primo canale è la medesima di quella del secondo canale. Questo significa che se un cliente ha aggiornato il numero di telefono mobile, è come se avesse anche aggiornato quello fisso, e viceversa. Questa peculiarità è meglio visibile quando le misure interessate non sono gli update, ma update validi. Attraverso l'utilizzo di un bookmark, in questa pagina, è possibile cambiare la visualizzazione da update ad update validi e update utili.

Un'altra importante pagina della stessa dashboard è quella che permette di avere un'ottica sull'intero storico della base dei clienti; tale dashboard è riportata nella Figura [6.3](#).

Selezionando tutti i filtri nella parte sinistra della pagina, nel visual sono riportati i numeri dei lead, prospect e customer con tutti gli ageing di test e con tutti i possibili risultati dell'ultimo test medicale effettuato. Ciascun dato nella tabella indica il numero di clienti all'interno della customer base, con relativo canale di comunicazione. I valori sono rappresentati sia in forma decimale che percentuale.

## 6.3 Altre dashboard realizzate

All'interno del progetto oggetto di questa tesi, non vi sono soltanto le dashboard riportate nel capitolo precedente. Per completezza verranno riportate alcune visual significative di questi report aggiuntivi.

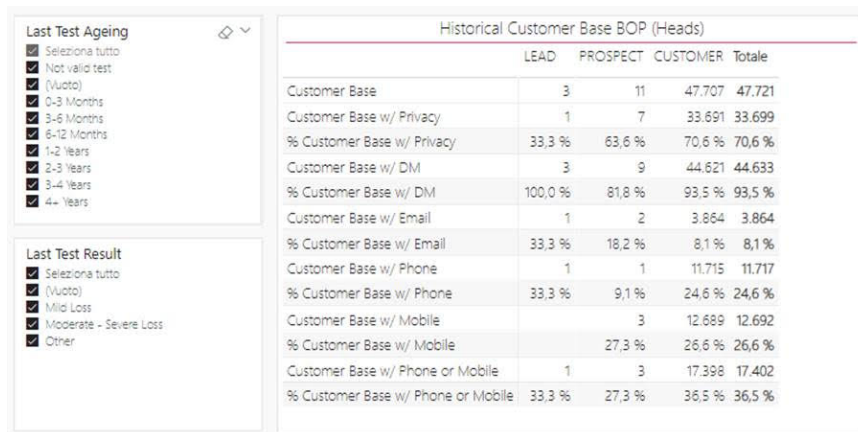


Figura 6.3. Historical Customer Base

Nella Figura 6.4 è riportata una pagina molto simile a quella presente nella Contactability, che, infatti, prende in esame le stesse misure, ma con una diversa visualizzazione, integrate in un contesto differente.

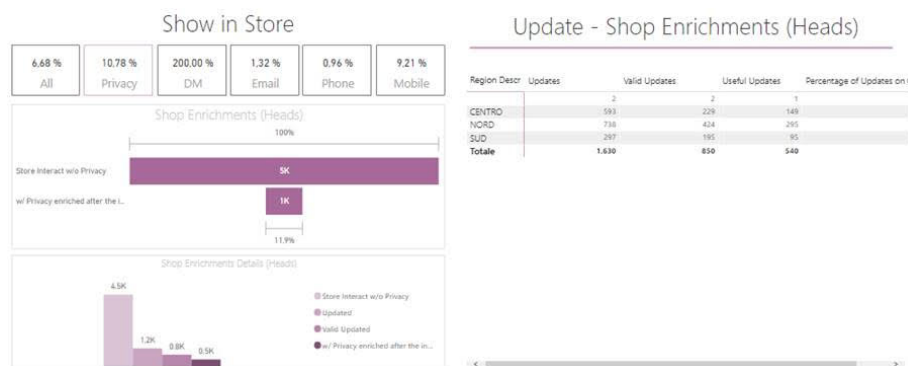


Figura 6.4. Historical Customer Base

Nella parte sinistra sono riportati tre diversi oggetti visivi. In alto vi sono le “card” che mostrano, in percentuale, il numero di appuntamenti presi senza il corrispettivo canale di comunicazione valido. I due gradini nel mezzo indicano, rispettivamente, il numero di interazioni effettuate con lo store senza almeno un canale di comunicazione valido e il numero di interazioni che hanno portato all’arricchimento di quel canale. Nella parte inferiore vi è un istogramma che riporta il numero di interazioni con lo store senza quel canale valido, il numero di update, il numero di valid update e il numero di interazioni che ha portato all’arricchimento di quel canale.

Nella parte destra, invece, vi è una semplice tabella pivot dove, per ogni colonna, vengono visualizzati gli update, gli update validi, gli update utili e la percentuale di



aggiornamenti sulla base dei clienti. Su questa tabella è implementato il drill-down e drill-up che permettono di scendere e salire a livello gerarchico tra area della nazione interessata e shop di riferimento.

L'ultima visual analizzata è quella riportata nella Figura [6.5](#).

Customer Type Group	Aging Cluster English Desc	HA Sales (Value)	HA Sales (Units)
LEAD		1.868.628 €	1.222
PROSPECT		437.094 €	304
PROSPECT	USER - Cold	2.047 €	1
CUSTOMER		8.051 €	7
CUSTOMER	USER - Cold	490.343 €	344
CUSTOMER	USER - Hot	2.026.556 €	1.285
<b>Totale</b>		<b>4.832.718 €</b>	<b>3.163</b>

Customer Type Group	Customer Type Descr	Aging Cluster English Desc	Trial Activated (Units)
LEAD	ENGAGED		1.541
LEAD	NEW LEAD		57
LEAD	RETURNING LEAD		145
PROSPECT	NEW PROSPECT		100
PROSPECT	RETURNING PROSPECT		583
CUSTOMER	CUSTOMER		1
CUSTOMER	CUSTOMER	USER - Cold	506
CUSTOMER	CUSTOMER	USER - Hot	1.913
<b>Totale</b>			<b>4.846</b>

**Figura 6.5.** Dashboard - CLC

Questa pagina, che fa parte del filone progettuale relativo al CLC (*Customer LifeCycle*), presenta informazioni relative alle unità di apparecchi medicali venduti, con il rispettivo valore economico e il numero di trial attivati. Il tutto viene suddiviso nei tre cluster possibili, ovvero lead, prospect e customer. Questo schema viene anche riproposto con altre misure, relative ad esempio al branch dei test.

Tutte le dashboard realizzate sono corredate al loro interno con filtri di tipo calendario che permettono, all'occorrenza, di visualizzare i dati di uno specifico anno, mese e giorno. Nel sistema sono stati implementati ulteriori filtri particolarmente complessi che non verranno descritti nella presente tesi.



## Discussione in merito al lavoro svolto

*In questo capitolo verrà discusso con ottica critica il lavoro svolto all'interno del progetto.*

### 7.1 Lezioni apprese

In questa sezione esamineremo alcune delle “lezioni apprese” durante le attività connesse con la presente tesi.

#### 7.1.1 L'importanza della Business Intelligence e dei KPI

L'enorme quantità di dati a disposizione di un'azienda non sempre è sinonimo di conoscenza. Essi necessitano di procedure di trasformazione affinché possano essere utilizzati dal management per migliorare le prestazioni complessive della società. È proprio da questa necessità che si è sviluppato il concetto di *Business Intelligence*, ovvero il processo di trasformazione di dati e informazioni in conoscenza.

Avere alle spalle un ottimo sistema di *Data Analytics*, associato a sofisticate analisi di business, fornisce, alle imprese, il potenziale di avere intuizioni sulle condizioni di mercato e sul comportamento dei clienti, rendendo l'attività decisionale più efficace e veloce rispetto alla concorrenza. Ed è proprio questo l'obiettivo a lungo termine dell'azienda per la quale Deloitte lavora, ovvero riuscire a mettere in campo soluzioni sempre più innovative in ambito di gestione di relazioni con i clienti, per guadagnarsi la posizione di leader di mercato.

Il prodotto finale, che traghetta la società verso questa posizione, fornito dal team Deloitte, non è altro che un insieme di dashboard dinamiche, pronte a mettere in evidenza eventuali lacune, nonché punti di forza, nella gestione dei clienti. Nel fare ciò si è sviluppato un vero e proprio strumento software, che, partendo dall'acquisizione e manipolazione di masse di dati presenti su database e archivi non strutturati, fornisce report, statistiche, indicatori, grafici costantemente aggiornati, facilmente adattabili e configurabili persino dall'utente.

All'interno del progetto oggetto di questa tesi, è stato posto il focus su alcuni degli elementi forse più importanti di un sistema di BI e data analytics, ovvero i KPI.

Avere dei KPI ben progettati e strutturati permette al management di consultare un indicatore che, in maniera decisa e puntuale, mostra qual è l'andamento dell'azienda in corrispondenza di una particolare tematica.

Realizzare un KPI secondo una logica errata potrebbe compromettere, in cascata, tutta una serie di decisioni strategiche aziendali e che inevitabilmente condurrebbe ad uno scenario negativo. Da qui la cura e il focus del team Deloitte nella progettazione di indicatori quanto più fedeli possibili alle richieste del cliente. Fondamentale per il team è anche capire in maniera esatta quali siano le esigenze del management al fine di rendere lo sviluppo dei KPI quanto meno problematico possibile.

Non centrare a pieno la richiesta dell'azienda vorrebbe dire progettazione e sviluppo errati, con la gravante di compromettere tutta la parte di sistema a valle. Per questo lavorare nel team in questione è particolarmente complesso, date anche le elevate responsabilità prese in carico.

### 7.1.2 L'importanza dell'analisi dei requisiti

Quanto detto nella sezione precedente, ovviamente, non viaggia all'interno di un compartimento stagno. Capire le richieste del cliente è una fase molto delicata che, in generale, viene attuata all'interno dell'analisi dei requisiti. Questa rappresenta la prima fase del processo di sviluppo, e condurla in maniera errata porterebbe danni in tutto ciò che si trova a valle.

Problematica comune è, infatti, giungere al rilascio di un prodotto che è stato sviluppato seguendo requisiti errati che, in parte, potrebbero essere attribuiti ad una mancata comprensione del team Deloitte, mentre altre volte sono riconducibili ad errate spiegazioni lato cliente. È sempre meglio risolvere eventuali incomprensioni sul nascere, piuttosto che a prodotto ultimato. Effettuare backtracking in quest'ultimo caso porta con sé sempre molti problemi. All'interno del progetto sono diversi i team che collaborano per giungere ad un risultato finale e l'errore da parte di uno di essi potrebbe compromettere anche il lavoro degli altri.

Dunque, più l'analisi dei requisiti viene fatta in maniera meticolosa, e meno sarà la probabilità di insorgenza di problematiche sull'intero sistema.

Diversa è la questione quando vengono richieste delle modifiche a requisiti precedentemente approvati. In questi casi la modifica richiesta non è attribuita ad errori lato team Deloitte, in quanto le modifiche possono essere semplicemente dovute a richieste esplicite del management, in seguito al cambiamento degli obiettivi aziendali. Le *Change Request*, comunemente dette, anche, *CR*, devono essere stimate in giorni-uomo e approvate dal management che andrà a pagare un surplus dovuto al lavoro aggiuntivo. In quest'ottica, le CR vengono sempre viste in maniera positiva dal team Deloitte.

### 7.1.3 L'importanza delle sessioni di UAT

Ogniquale volta un prodotto è pronto ad essere consegnato al cliente, si sostengono delle sessioni di UAT, in genere della durata di poche settimane. In questa fase l'azienda ha visione del prodotto che deve essere rilasciato per condurre analisi minuziose e puntuali sulla sua qualità. Nel sistema realizzato, oggetto della presente

tesi, gli UAT vengono realizzati sulla KPI-Pedia e Contactability, in cui il team locale della nazione interessata prende in carico il prodotto, fornendo dei feedback costanti e graduali che si concludono con la definitiva accettazione del deliverable.

Questa fase è di assoluta importanza, perchè fa in modo che non venga rilasciato un prodotto che presenti errori o bug, in modo tale che eventuali problematiche abbiano margine di correzione. Il tutto è accompagnato dall'utilizzo di documentazione e fogli Excel condivisi che permettono di tenere traccia dei risultati dei test. All'occorrenza vengono fissati meeting all'interno dei quali si discutono le modalità con cui determinate componenti sono state realizzate, qualora sorgessero dubbi e perplessità lato aziendale.

Queste sessioni sono, inevitabilmente, considerate uno strumento utile, nonchè un'opportunità, per individuare problemi ed errori prima che questi diventino irreversibili, o, comunque, difficili da gestire, una volta che il prodotto viene consegnato.

#### 7.1.4 L'importanza dei rilasci

Tutti i pacchetti realizzati in ambiente di test devono essere chiaramente rilasciati in ambiente di produzione. Sono proprio i flussi di quest'ultimo che alimentano l'intero sistema di reportistica. Uno o più errori potrebbero compromettere più dashboard che leggono i dati da strutture condivise.

È proprio da qui che nasce l'importanza di condurre dei rilasci quanto più meticolosi possibili per evitare che i mapping si rompano in produzione compromettendo il sistema.

Effettuare un check in più durante la fase di test è sempre buona norma per evitare problematiche a valle. Più volte accade che piccoli dettagli o sviste facciano fallire il run degli scenari in produzione. Si pensi, ad esempio, ad un campo mappato erroneamente o alla mancata attribuzione di una partizione all'interno di una tabella.

Inoltre, la fase dei rilasci segue una schedulazione particolarmente selettiva e rigida e, pertanto, eventuali modifiche per rispondere ad errori rischiano di non poter essere integrate all'interno di una stessa sessione, riportando alla necessità di aprire un nuovo ticket e richiedere un nuovo slot di rilascio.

Più volte è capitato al team Deloitte di dover apportare correzioni durante una fase di rilascio, con il rischio di compromettere il sistema, anche se, in generale, si è sempre riusciti a portare a conclusione il rilascio nel migliore dei modi.

## 7.2 Analisi SWOT

Nella seguente sezione viene effettuata un'analisi SWOT, conosciuta anche come matrice SWOT. Essa rappresenta uno strumento di pianificazione strategica usato per valutare i punti di forza, i punti di debolezza, le opportunità e le minacce di un progetto.

### 7.2.1 Punti di forza

Tra i punti di forza del progetto non si può che mettere subito in evidenza la molteplicità di dashboard implementate nel sistema di Business Intelligence. All'interno di questa tesi è stata, infatti, analizzata solo una minima parte del contenuto reportistico complessivo in dotazione all'interno del progetto. Questo permette di fornire all'azienda quanta più conoscenza potenzialmente estraibile dai dati e dalle informazioni contenute all'interno dei database. Non tutte le tematiche aziendali riportate nella reportistica sono state affrontate in questa tesi, a causa della loro complessità e diversità contestuale.

Un altro punto di forza, specifico di ogni singola dashboard, è la quantità e qualità dei filtri utilizzati. Essi permettono di navigare tra i dati con maggior velocità, precisione ed accuratezza, rendendo i report estremamente dinamici ed adattabili.

Seppure i requisiti richiesti dall'azienda possono cambiare da una nazione all'altra, lo scheletro della componente KPI-Pedia rimane fundamentalmente il medesimo. Condurre, dunque, la fase di sviluppo ed implementazione in parallelo, nazione per nazione, consente di risparmiare tempo di lavoro, in quanto la maggior parte dei mapping e pacchetti possono essere riutilizzati, una volta che abbiano superato i test del caso. Qualora, in futuro, sarà necessario implementare il sistema per una nuova nazione, si potranno riutilizzare le componenti già sviluppate in precedenza. Aver progettato il sistema in questo modo costituisce, inevitabilmente, un punto di forza dell'intero progetto.

Un altro punto di forza è, sicuramente, l'utilizzo di un modello di dati condiviso all'interno di Tabular. In questo modo il file `.bim`, e quindi il data model su Power BI, rimane il medesimo per ogni nazione; quello che cambia è solo il database al quale il report punta per attingere ai dati della nazione interessata.

Dal punto di vista organizzativo, il fatto che ciascun team è allocato su uno stream di progetto, e non su una nazione specifica, rende la cooperazione e la collaborazione migliore.

Infine, la schedulazione di meeting ricorrenti, anche più volte a settimana, tra il team Deloitte e il cliente, rende il raggiungimento degli obiettivi di progetto prefissati più semplici e alla portata.

### 7.2.2 Punti di debolezza

Se, da una parte, la quantità di dashboard realizzate all'interno del progetto costituisce un punto di forza, dall'altra, l'elevata complessità rappresenta un punto di debolezza. Caricare un report con un elevato numero di filtri e oggetti visivi per ciascuna pagina causa inevitabilmente un rallentamento non trascurabile nella reattività della dashboard stessa. Questo aspetto è ancora più persistente quando vengono utilizzati dei bookmark che richiedono la duplicazione delle pagine, appesantendo ulteriormente il sistema di BI.

A livello più basso, l'utilizzo di ODI pone un freno all'usabilità, rispetto ai più aggiornati concorrenti tra i software di gestione e integrazione dei dati. Molte volte, l'eccessiva lentezza del software, unita ad una user interface non particolarmente soddisfacente, pongono l'esperienza di utilizzo al di sotto delle aspettative, costituendo, così, un elemento di debolezza.

Infine, un altro aspetto importante è la quantità di memoria limitata dei database. Per facilitare l'interrogazione di tabelle e per metterle a confronto, molto spesso vengono realizzate delle copie di backup, che, nella maggior parte dei casi, vengono salvate senza essere eliminate. Ciò porta alla saturazione della memoria disponibile, con conseguente rallentamento del lavoro e dei processi in corso.

### 7.2.3 Opportunità

Tra le varie opportunità che potrebbero essere sfruttate al fine di portare un valore aggiunto al progetto c'è la possibilità di passare all'utilizzo di un nuovo software di integrazione dei dati. In questo modo si riuscirebbe a garantire un'esperienza di utilizzo migliore sotto diversi aspetti. Ovviamente una scelta del genere andrebbe fatta in maniera ponderata e avendo cura delle eventuali complicazioni che potrebbero nascere nel porting dei database da un software ad un altro.

Inoltre non si esclude la possibilità di estensione del progetto ad altre nazioni all'interno delle quali l'azienda opera, sia a livello europeo che mondiale. Questo porterebbe inevitabilmente lavoro aggiuntivo su diversi fronti, garantendo continuità e crescita del team Deloitte.

Per quanto l'intero progetto sia grande e complesso, attualmente, le analisi che vengono condotte, e, quindi, la conoscenza che si può apprendere dalla reportistica realizzata, sono di natura descrittiva. Una grande opportunità, potrebbe essere quella di utilizzare i vari tool di analisi predittiva messi a disposizione dello stesso Power BI. In questo modo si genererebbe uno stream che va ad occuparsi solo ed esclusivamente di predizioni, creando tutta una serie di reportistica a supporto. Nonostante l'affidabilità degli algoritmi di previsione, la quale non sarà mai assoluta, avere delle proiezioni future in ambito di base clienti potrebbe aiutare, e non poco, a raggiungere meglio e prima gli obiettivi aziendali.

### 7.2.4 Minacce

Portare avanti un progetto di questo calibro richiede inevitabilmente un elevato numero di risorse umane all'opera. Una potenziale minaccia potrebbe essere, dunque, la carenza di personale dedicato, che andrebbe a rallentare di molto lo sviluppo dell'intero sistema.

Un altro aspetto, da considerare in ottica minatoria del progetto, è quello degli attacchi cyber. Un'azione malevola nei confronti del sistema rischierebbe di "far cadere" l'intero database con conseguente collasso della reportistica a supporto. Gli strumenti posti a difesa di questa minaccia non sono trattati all'interno di questa tesi.

Infine, se, da una parte, l'evoluzione tecnologica costituisce una potenziale opportunità da sfruttare, non è escluso lo scenario in cui, al contrario, sarebbe una minaccia per l'intero sistema. Qualora tutti gli strumenti utilizzati diventassero tecnologicamente obsoleti non sarebbe semplice effettuare una migrazione dell'intero sistema verso altri tool, e questo costituisce, inevitabilmente, una minaccia per l'intero progetto.





## Conclusioni

In questo lavoro di tesi è stata progettata e sviluppata una componente in grado di racchiudere al proprio interno una variegata lista di indicatori di performance, relativamente ad un sistema di data analytics. Lo scopo ultimo del progetto in questione è quello di fornire all'azienda committente, che si ricorda operare nel settore retail di apparecchi medicali, uno strumento a supporto del raggiungimento degli obiettivi aziendali. Il prodotto finale non è altro che un ampio ventaglio di reportistica che tocca diversi aspetti in ambito di gestione delle relazioni con il cliente. Più dettagliata è la conoscenza che si ha in merito, maggiore è la consapevolezza dell'azienda nei confronti del percorso che viene intrapreso dagli individui, quali clienti della stessa.

L'esperienza presso l'azienda si è configurata come un'attività lavorativa a tutti gli effetti. L'inserimento all'interno di un team specifico, e il rispetto di un orario lavorativo full time, sono tutti aspetti che riflettono una reale integrazione all'interno del progetto e dell'azienda.

I risultati ottenuti, intesi come feedback dell'azienda committente, sui prodotti consegnati, sono molto buoni. Ovviamente, ciò non significa che l'intero sistema non sia migliorabile, anzi è prerogativa standard del team Deloitte lavorare sempre per raggiungere dei risultati migliori e che possano soddisfare al massimo le richieste del cliente.

Seppur il rapporto con l'azienda in questione prosegue ormai da diversi anni, il progetto è ancora in fase iniziale e ben lontano dalla sua conclusione. Questo è merito delle continue richieste di cambiamento che vengono inoltrate su deliverable già testati ed accettati, nonchè richieste di sviluppo di filoni progettuali completamente nuovi.

Tra gli sviluppi futuri vi è, sicuramente, la possibilità di estendere l'intero progetto ad altre nazioni, sia in Europa che nel resto del mondo, e di collocare un team in uno stream che si occupi esclusivamente di analisi predittive. Infine, si può pensare ad un'eventuale migrazione del sistema verso software di gestione ed integrazione dei dati aventi una tecnologia più avanzata ed al passo con i tempi moderni, anche in virtù della disponibilità di risorse in termini di occupazione della memoria.



---

## Riferimenti bibliografici

1. Antonio Badia. *SQL for Data Science: Data Cleaning, Wrangling and Analytics with Relational Databases*. Springer Nature, 2020.
2. Gary Cokins. *Performance management: Integrating strategy execution, methodologies, risk, and analytics*, volume 21. John Wiley & Sons, 2009.
3. Fabrizio Di Crosta. *Indicatori di performance aziendali. Come identificare gli indicatori più adatti per misurare le performance: dagli obiettivi ai risultati: Come identificare gli indicatori più adatti per misurare le performance: dagli obiettivi ai risultati*. FrancoAngeli, 2015.
4. Paul Goodey. *Salesforce CRM—The Definitive Admin Handbook*. Packt Publishing Ltd, 2015.
5. Heli Helskyaho. *Oracle SQL Developer Data Modeler for Database Design Mastery*. McGraw Hill Professional, 2015.
6. Ihab F Ilyas and Xu Chu. *Data cleaning*. Morgan & Claypool, 2019.
7. Mark Jeffery. *Data-driven marketing: the 15 metrics everyone in marketing should know*. John Wiley & Sons, 2010.
8. Harold Kerzner. *Project management metrics, KPIs, and dashboards: a guide to measuring and monitoring project performance*. John Wiley & Sons, 2017.
9. Ralph Kimball and Joe Caserta. The data warehouse etl toolkit practical techniques for extracting, cleaning, conforming and delivering data. *Canada: Wiley Publishing, Inc*, 2004.
10. Andy Kirk. *Data visualisation: A handbook for data driven design*. Sage, 2016.
11. Devin Knight, Brian Knight, Mitchell Pearson, Manuel Quintana, and Brett Powell. *Microsoft Power BI Complete Reference: Bring your data to life with the powerful features of Microsoft Power BI*. Packt Publishing Ltd, 2018.
12. Bernard Marr. *Key performance indicators for dummies*. John Wiley & Sons, 2015.
13. Errin O'Connor. *Microsoft Power BI Dashboards Step by Step*. Microsoft Press, 2018.
14. Scott E Page. *The model thinker: What you need to know to make data work for you*. Basic Books, 2018.
15. Thomas A Runkler. *Data analytics*. Springer, 2020.
16. Marco Russo and Alberto Ferrari. *The Definitive Guide to DAX: Business intelligence for Microsoft Power BI, SQL Server Analysis Services, and Excel*. Microsoft Press, 2019.
17. Bryan Syverson and Joel Murach. *Murach's SQL Server 2019 for Developers: Training and Reference*. Mike Murach & Associates, Incorporated, 2020.
18. Carlo Vercellis. *Business intelligence. Modelli matematici e sistemi per le decisioni*. McGraw-Hill, 2006.



---

## Ringraziamenti

I primi ringraziamenti sono per i miei genitori, che mi hanno sempre sostenuto durante questo percorso e senza i quali non avrei mai raggiunto questo obiettivo.

Ringrazio, poi, il Prof. Ursino, una grande persona prima che un grande Professore, per l'aiuto fornito nel cogliere al volo una grande opportunità per la mia crescita personale e professionale.

Non posso che ringraziare anche i miei amici e colleghi Laura, Luca e Sara; con voi ho condiviso tutte le vicissitudini universitarie, ma siamo sempre riusciti a concludere i progetti e portato a casa il risultato.

Grazie, poi, ai miei attuali colleghi Carmine e Matteo; avete saputo prelevarmi dal mondo universitario e trasportatomi in quello lavorativo contribuendo alla mia crescita professionale; da voi ho solo da imparare.

Un ringraziamento speciale ai miei amici Adriano, Alessio e Alessandro; le nostre partite online sono state di grande aiuto durante la pandemia per spezzare la monotonia dello studio; rimanete sempre i più ricchi di Chieti.

Potrebbe passare per autocelebrazione ma sento di fare l'ultimo ringraziamento a me stesso e alla mia forza di volontà, per aver superato qualsiasi ostacolo, anche quelli all'apparenza insormontabili.

*Questo non è un traguardo ma un punto di partenza, Ad Maiora!*