

Università degli Studi di Modena e Reggio Emilia

Dipartimento di Ingegneria “Enzo Ferrari”

Corso di Laurea in Ingegneria Informatica

Analisi dello strumento di BI Qlik Sense e  
applicazione presso l'azienda USCO S.p.A.

Relatore:

Prof.ssa Sonia Bergamaschi

Candidata:

Gaia Frabetti

---

Anno Accademico 2020-2021



# Ringraziamenti

Con questo elaborato si conclude un lungo periodo di studio, tra alti e bassi, nel quale ho avuto accanto persone fantastiche che mi hanno sempre sostenuto senza mai giudicare i miei fallimenti e festeggiando con me le mie vittorie.

In primo luogo, la mia famiglia, che mi ha supportato sempre, anche nelle sessioni e momenti peggiori;

il mio compagno di vita, Marco, che mi ha sempre spronato a non mollare, a fare del mio meglio senza mai lasciarmi andare alla deriva, ricordandomi ogni volta chi sono e che nulla è inaffrontabile;  
*a te devo tutto.*

i miei compagni di corso e amici, Debora, Giorgia, Davide, Naima, Rajaa e Amrit, che mi hanno sempre aiutata nel momento del bisogno e con i quali ho condiviso le gioie e i dolori di ogni singolo esame. In particolar modo ringrazio Dylan, col quale ho vissuto questa avventura fin dall'inizio e che non ha mai perso la pazienza;

Laura, Francesca e Gaia, la mia triade di migliori amiche, che fin da tempi lontani mi conoscono così bene da saper comprendere ogni lato del mio carattere e farmi ripartire più carica di prima ad ogni caduta.

Un ringraziamento speciale va a Cinzia, Alessia, Enrico, Mario, Filippo e Matteo: mi avete sempre sostenuta e mai giudicata. So di poter contare su di voi nel momento del bisogno, questo vale più di ogni cosa.

Ringrazio l'azienda USCO S.p.A. per avermi permesso di effettuare lo stage aziendale su cui si basa questo elaborato. Un ringraziamento a Nicola Landi che è stato il mio tutor aziendale, a Roberto Trenti e Alessandra Lisotti, con i quali ho lavorato e che mi hanno accompagnata nello sviluppo del progetto e a tutte le persone con cui ho collaborato, che mi hanno accolto fin dal primo momento insegnandomi tutto ciò che potevano.

Ringrazio la Prof. Sonia Bergamaschi per il tempo dedicatomi e per i suggerimenti che ha saputo darmi.

Infine ringrazio me stessa, perchè è doveroso farlo: senza la mia testardaggine e razionalità probabilmente avrei abbandonato al primo ostacolo.

*KISS - KEEP IT SIMPLE AND STUPID*



# Indice

<b>1. Introduzione .....</b>	<b>6</b>
1.1.    USCO S.p.A.....	6
1.2.    Attività svolte durante il tirocinio.....	7
<b>2. Business Intelligence.....</b>	<b>8</b>
2.1.    Definizione e campo di applicazione.....	8
2.2.    Architettura di una soluzione di BI.....	10
2.3.    Confronto tra competitor: Qlik Sense VS Power BI.....	17
<b>3. Qlik Sense.....</b>	<b>20</b>
3.1.    Descrizione software.....	20
3.2.    Gestione dati e sintassi .....	24
<b>4. Sviluppo Dashboard Helpdesk.....</b>	<b>30</b>
4.1.    Motivazioni.....	30
4.2.    Analisi delle richieste.....	31
4.3.    Fonte dati e caricamento.....	32
4.4.    Creazione: strumenti e metodi utilizzati .....	36
4.4.1. Primo foglio.....	37
4.4.2. Secondo foglio .....	42
4.4.3. Terzo foglio.....	43
<b>5. Conclusioni.....</b>	<b>45</b>
<b>6. Bibliografia.....</b>	<b>46</b>

# 1. Introduzione

La redazione di questo elaborato ha lo scopo di descrivere le attività svolte durante il tirocinio curricolare presso l'azienda USCO S.p.A..

In particolar modo verranno analizzate le attività svolte sulla piattaforma Qlik Sense, strumento di business intelligence per l'analisi dei dati e decision making.

## 1.1 USCO S.p.A.

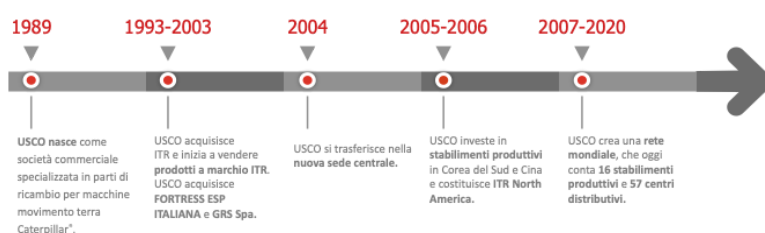


Usco S.p.A. nasce nel 1989 a Modena come società di trading specializzata nelle parti di ricambio per macchine movimento terra Caterpillar.

Nel 1993 acquisisce ITR ampliando notevolmente la gamma di prodotti e diventa produttrice di pompe, riduttori, pignoni, miscellanea ed altri ricambi per macchine movimento terra.

In pochi anni è stata protagonista di una grande espansione, ancora oggi in atto, grazie ad acquisizioni, investimenti e realizzazioni di nuovi centri produttivi e distributivi in tutto il

mondo oltre che in Italia: a partire dai primi anni 2000 acquisisce GRS S.p.A., leader mondiale nella produzione di pompe a ingranaggi per macchine movimento terra nel settore aftermarket [\*], acquisisce KUT, azienda con sede in Corea del Sud, costituisce il primo centro distributivo a Chicago, ITR North America, in seguito anche ITR Middle East a Dubai (UAE) e TrackOne a Modena dedicata alla progettazione e sviluppo di sottocarri completi per clienti



[\*]Letteralmente traducibile con 'post vendita', il mercato aftermarket si riferisce a tutto quel settore che racchiude i ricambi e gli accessori, originali e non, anche se col tempo l'utilizzo si è riferito maggiormente ai ricambi non originali.

OEM(Original Equipment Manufacturer). Negli anni successivi fino ad oggi sono state costituite altre sedi produttive e di distribuzione in Australia, Cina, America e Sud America, Africa e Sudafrica, Olanda, Spagna, Germania e Inghilterra, con esportazioni in oltre 120 Paesi nel mondo.

USCO ha sviluppato nel tempo una gamma di parti di ricambio fra le più complete disponibili sul mercato dei componenti per macchine movimento terra. Tale gamma, costantemente in evoluzione, include attualmente oltre 50.000 articoli, rendendo l'azienda leader nella produzione e distribuzione di componenti aftermarket per macchine movimento terra e fornitore di parti OEM.

## **1.2 Attività svolte durante il tirocinio**

Durante il tirocinio svolto presso USCO S.p.A. ho potuto svolgere diverse attività, tra cui:

- Sviluppo e modifica dashboards su Qlik Sense
- Redazione di manuali (Qlik Sense, CRM)
- Corsi ( Qlik Sense, CRM, Nicim MES aziendale)
- Analisi di nuovi software da acquistare e implementare (Nicim)
- Report su excel e sistemazione dati

Questo elaborato finale si concentra in particolar modo sull'attività che ha occupato la maggior parte del tempo di questa esperienza di tirocinio, ovvero lo sviluppo e la manutenzione di dashboards sulla piattaforma di business intelligence Qlik Sense.

## 2. Business Intelligence

### 2.1 Definizione e impiego

La locuzione Business Intelligence<sup>(1)</sup> è stata coniata nel 1958 da Hans Peter Luhn, ricercatore e inventore tedesco. Costituisce un ramo della cosiddetta informatica aziendale.

Con l'espressione business intelligence (BI) ci si può solitamente riferire a:

1. un insieme di processi aziendali per raccogliere dati ed analizzare informazioni strategiche;
2. la tecnologia utilizzata per realizzare questi processi;
3. le informazioni ottenute come risultato di questi processi.

Business intelligence quindi è un termine onnicomprensivo che riguarda i processi e i metodi per raccogliere, memorizzare e analizzare i dati tratti dalle operazioni o attività aziendali con l'obiettivo di migliorare le prestazioni.

Questo strumento, mostrando dati attuali e storici all'interno del contesto aziendale, consente alle aziende di prendere decisioni migliori e mirate al loro contesto. Infatti permette di:

- Identificare i modi per aumentare i profitti
- Analizzare il comportamento dei clienti
- Confrontare i dati con i concorrenti
- Monitorare le prestazioni
- Migliorare le operazioni
- Prevedere il successo
- Individuare i trend di mercato
- Scoprire complicazioni o problemi

Un'importante caratteristica delle BI è che consentono di raccogliere e uniformare i dati archiviati all'interno di basi di dati differenti, basate su diverse tecnologie, dove può capitare che i dati siano frammentari e non comunichino tra di loro, aspetto che rende difficile realizzare una reportistica completa.

Difatti, in passato, gli strumenti di business intelligence si basavano su un modello tradizionale. La business intelligence era gestita dall'organizzazione IT



e si era soliti usare report statici per rispondere alla maggior parte delle domande di analisi, se non a tutte. Perciò, se qualcuno aveva un ulteriore dubbio sul report ricevuto, la sua richiesta veniva messa in fondo alla coda di reporting e il processo doveva ripartire dal principio. Viceversa, la business intelligence moderna è interattiva e accessibile: sebbene i reparti IT siano ancora fondamentali per la gestione dell'accesso ai dati, gli utenti possono personalizzare le dashboard e creare report anche con poco preavviso e in autonomia, riuscendo così a visualizzare i dati e rispondere alle proprie domande.

Tanto è vero che uno dei vantaggi più interessanti delle piattaforme di Business Intelligence è che vengono progettate appositamente per essere utilizzate anche da persone senza conoscenze tecniche. Ciò avviene grazie all'utilizzo di cruscotti aziendali e indicatori di performance di facile comprensione, con diagrammi e grafici, per permettere anche a chi non ha alcuna conoscenza di analisi dei dati di avere un'immagine istantanea di ogni tipo di attività in poco tempo.

Negli ultimi anni, la business intelligence si è sviluppata includendo più processi e attività per consentire il miglioramento delle prestazioni. Tali processi includono:

- **Data mining:** uso di database, statistiche e apprendimento automatico per svelare i trend in ampi set di dati;
- **Elaborazione di report:** condivisione delle analisi dei dati con i soggetti interessati, affinché possano trarre conclusioni e prendere decisioni;
- **Metriche e benchmarking delle prestazioni:** confronto dei dati sulle prestazioni attuali con i dati storici;
- **Analisi descrittiva:** utilizzo di analisi dei dati preliminari per comprendere cosa è accaduto;
- **Esecuzione delle query:** interrogazione dei dati con specifiche domande, per cui la BI estrae le risposte dai set di dati;
- **Analisi statistica:** partendo dai risultati dell'analisi descrittiva, ulteriore esplorazione dei dati usando le statistiche, per esempio in relazione a come e perché si sia verificato un determinato trend;
- **Visualizzazione dei dati:** trasformazione dell'analisi dei dati in rappresentazioni visive, come grafici, diagrammi e istogrammi, per una fruizione dei dati più facile;

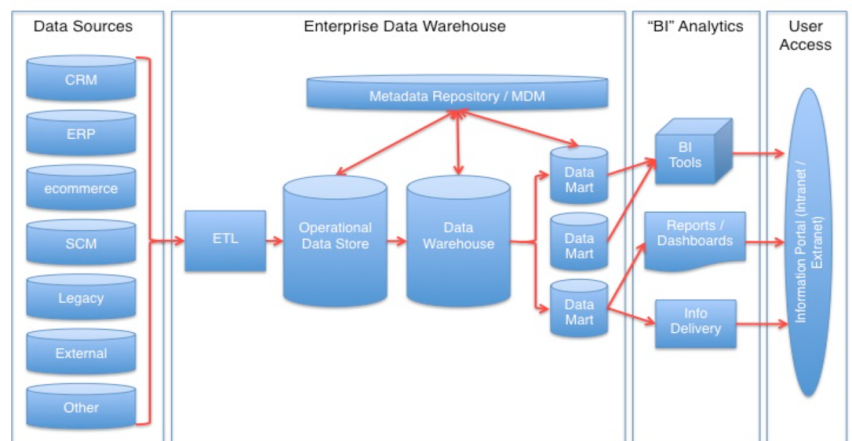
- **Analisi visiva:** esplorazione dei dati attraverso le rappresentazioni visive, per comunicare informazioni rapidamente e seguire il flusso dell'analisi;

Disporre di questa tecnologia all'interno di un'azienda comporta quindi diversi vantaggi tra i quali l'automatizzazione della raccolta e trasformazione dei dati, maggiore flessibilità rispetto a un normale ERP, indipendenza tra i reparti funzionali e maggiore qualità dei dati e puntualità, in quanto lo strumento di BI può condurre ogni giorno le proprie analisi preimpostate su dati aggiornati quotidianamente.

## 2.2 Architettura di una soluzione di BI tradizionale

L'architettura<sup>(2)</sup> generale dei sistemi di Business Intelligence è articolata su tre livelli:

1. I sistemi che producono e contengono dati elementari (sistemi alimentanti/fonti dati)
2. I sistemi per l'integrazione e l'archiviazione dei dati semilavorati (sistemi di Data Warehouse e Data mart)
3. I sistemi per l'accesso ai dati e la produzione di informazioni "finite" (Metodologie di business intelligence).



I sistemi di origine dei dati elementari, o sistemi alimentanti, sono costituiti dai sistemi gestionali di tipo ERP o tradizionali, dai sistemi che interfacciano il mercato, dai sistemi web (portali, e-commerce, siti informativi) e da tutti gli altri Sistemi Informativi formali di tipo operativo e/o transazionali.

I dati in seguito vengono trasformati e immagazzinati nei database adibiti alle analisi di business intelligence mediante strumenti detti *ETL* (*Extract, Transform, Load*) che svolgono le operazioni di estrazione, pulitura, trasformazione, caricamento dei dati nel *Data Warehouse (DW)*. Queste operazioni vengono spesso indicate con il termine “riconciliazione” che, durante il processo di alimentazione del DW avviene in due occasioni: quando il DW viene popolato per la prima volta e periodicamente quando viene aggiornato. Le operazioni svolte sono:

- **Estrazione:** durante questa fase i dati rilevanti vengono estratti dalle sorgenti e questa operazione può essere di tipo:
  - Statico: viene effettuata quando il DW deve essere popolato per la prima volta e consiste concettualmente in una fotografia dei dati operazionali;
  - Incrementale: viene usata per l’aggiornamento periodico del DW, e cattura solamente i cambiamenti avvenuti nelle sorgenti dall’ultima estrazione. L’idea alla base è quella di utilizzare i cambiamenti registrati a livello dei dati per aggiornare il DW. Il beneficio è il volume molto piccolo dei dati coinvolti di volta in volta nell’operazione rispetto all’estrazione statica, in quanto vengono analizzati solo i dati che hanno subito modifiche.
- **Pulitura:** è la fase che si occupa di migliorare la qualità dei dati andando ad eliminare quelli “sporchi” dovuti a duplicazioni, inconsistenze, dati mancanti, valori errati etc.
- **Trasformazione:** è la fase centrale del processo di riconciliazione e ha l’obiettivo di convertire i dati dal formato operativo sorgente a quello del DW.
- **Caricamento:** in questa fase avviene il caricamento dei dati sul DW attraverso due modalità alternative:
  - refresh: i dati vengono riscritti integralmente sostituendo completamente quelli precedenti. In generale questa tecnica viene utilizzata solo durante la fase iniziale di popolamento del DW;
  - update: vengono aggiunti al DW solo i cambiamenti avvenuti sui dati senza sovrascrivere ad ogni iterazione tutti i dati. Questa tecnica viene utilizzata, in abbinamento all’estrazione incrementale per l’aggiornamento periodico del DW.

Il secondo livello dall'architettura è costituito dai sistemi di Data Warehouse. Un *Data Warehouse* è una collezione di dati che rappresenta l'anello di collegamento tra i dati di origine e i sistemi informativi di supporto alle attività di controllo e decisione. Presenta le seguenti caratteristiche:

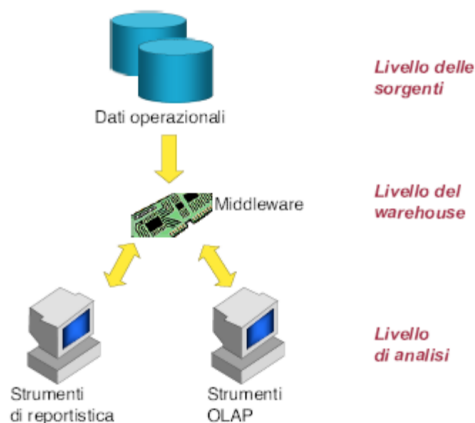
- *è orientata ai soggetti di interesse*, nel senso che include tutti i dati che possono essere utilizzati nel processo di controllo e di decisione, raggruppandoli per aree, fatti o temi di interesse e finalizzandoli a chi li utilizza e non a chi li genera
- *è integrata e consistente*: il DW si appoggia a più fonti di dati eterogenee come dati estratti dall'ambiente di produzione, e quindi originariamente archiviati in basi di dati aziendali, o addirittura provenienti da sistemi informativi esterni all'azienda. Di tutti questi dati il DW restituisce una visione unificata.
- *è rappresentativa dell'evoluzione temporale*, ossia i dati in esso sono costituiti da “fotografie” periodiche dei fatti correnti o storici con un orizzonte temporale storico anche di 2-5 anni.
- *non volatile*, nel senso che il dato viene caricato periodicamente fuori linea, cioè una volta memorizzato correttamente può essere acceduto, ma non modificato, dall'utente

I Data Warehouse possono essere articolati su uno o più livelli. La necessità, infatti, di avere a disposizione dati di dettaglio differente induce a progettare architetture su più livelli e con tecnologie differenti: mentre i dati di sintesi sono spesso precalcolati e memorizzati in database multidimensionali<sup>[\*\*]</sup>, i dati di dettaglio sono più spesso memorizzati in database relazionali a un livello inferiore dell'architettura dati.

---

[\*\*] Il modello multidimensionale è stato ideato da Edgar F. Codd, un informatico britannico, fondatore della teoria delle basi di dati relazionali. Negli anni sessanta e settanta, mentre lavorava per l'IBM, creò il modello relazionale per la gestione delle basi di dati. Con suo grande disappunto, l'IBM tardò a sfruttare i suoi suggerimenti fino a quando la concorrenza cominciò ad implementarli. Ad esempio, Oracle costruì il proprio database sulla base delle idee di Codd.

## Architetture a 1 livello



## Architetture a 2 livelli



L'architettura dati può essere, quindi, costruita in modo da non accedere direttamente al DW ma ai cosiddetti Data Mart. Un *Data Mart (DM)* è un raccogliatore di dati specializzato su un particolare segmento/area aziendale, o in termini più tecnici, è un sottoinsieme logico o fisico di un DW di maggiori dimensioni. La differenza fondamentale consiste nel fatto che la creazione del Data Warehouse avviene prima in maniera generalizzata per poi subire modifiche per potersi adattare a esigenze specifiche, mentre il data mart viene generalmente creato per venire incontro ad una esigenza già determinata.

Nell'architettura a 2 livelli, i Data Mart possono essere dipendenti o indipendenti dal DW. I DM dipendenti sono utili come blocchi costruttivi durante la realizzazione del DW in quanto, essendo di dimensioni inferiori, permettono di raggiungere prestazioni migliori. Nei DM indipendenti invece l'assenza di un DW primario snellisce le fasi progettuali ma può aumentare il rischio di inconsistenza tra i data mart

La progettazione di Data Warehouse e Data Mart si basa su un paradigma di rappresentazione multidimensionale dei dati, il cosiddetto *Dimensional Fact Model (DFM)*, nel quale i *fatti* di interesse sono rappresentati in *cubi* in cui ogni cella contiene *misure* numeriche che quantificano il fatto da diversi punti di vista; ogni asse rappresenta una *dimensione* di interesse per l'analisi; ogni

dimensione può essere la radice di una gerarchia di attributi usati per aggregare i dati memorizzati nel cubo.

Un fatto è un concetto di interesse per il processo decisionale, tipicamente modella un insieme di eventi che accadono nell'impresa (ad esempio: vendite, spedizioni, ...). È essenziale abbia aspetti dinamici, ovvero evolva nel tempo.

Le misure del cubo sono informazioni numeriche che rappresentano in modo quantitativo i fenomeni di interesse aziendale e racchiudono in sé dei valori espressi secondo unità di misure precise. Ogni misura o fatto può riferirsi a molteplici dimensioni (tempo, prodotto, cliente...).

Le dimensioni di analisi, invece, rappresentano le chiavi di lettura ritenute significative per l'analisi delle misure. Ogni dimensione di analisi può essere caratterizzata da una o più relazioni gerarchiche tra i propri attributi.

L'ultimo livello dell'architettura di BI è rappresentato dai sistemi di Business Intelligence che permettono di estrarre e utilizzare i dati per la costruzione di modelli matematici e la produzione e analisi di informazioni. Questi sistemi permettono quindi agli utenti di "crearsi" le informazioni di cui hanno bisogno secondo i propri tempi decisionali e quindi in tempo utile per analizzare un problema emerso o per la presa delle decisioni di azione.

I sistemi di analisi e accesso ai dati sono:

1. **Reporting:** l'insieme delle attività mediante le quali sono rilevati, misurati e comunicati i risultati raggiunti. È un approccio che permette agli utenti di accedere in modo tempestivo ai dati contenuti nei DW e nei DM. Il documento prodotto viene chiamato report e si presenta come una combinazione di tabelle e grafici che presentano le misure di rilievo per i vari fenomeni analizzati. Un rapporto può essere sia generato manualmente dall'utente che automaticamente per una distribuzione periodica agli utenti interessati, per esempio mediante posta elettronica.
2. **On-Line Analytical Processing (OLAP):** si intende la possibilità fornita all'utente di costruire, modificare e personalizzare report decidendo

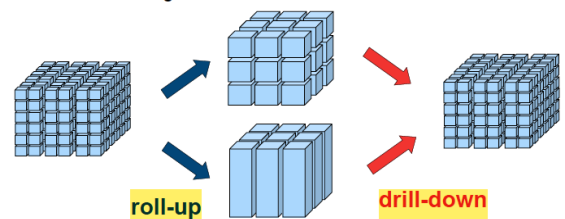
autonomamente dimensioni di analisi (righe, colonne, pagine), livello di dettaglio, quali dati numerici sottoporre a elaborazione (ricavi, costi, ecc.).

Il sistema OLAP consente la navigazione sull'informazione contenuta in un cubo e quindi con una logica multidimensionale. Si distingue da un sistema di reporting per il fatto che quest'ultimo offre un set predefinito di report, al limite modificabili e integrabili (colonne calcolate), mentre OLAP evidenzia la facilità per l'utente di cambiare totalmente strategia di analisi senza che, a priori, siano stati creati report dedicati.

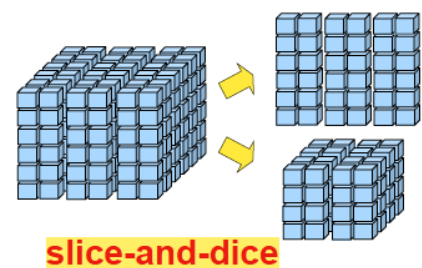
Ogni passo della sessione di analisi è scandito dall'applicazione di un operatore OLAP che trasforma l'ultima interrogazione formulata in una nuova interrogazione. Eseguendo successivamente delle interrogazioni sui dati è possibile ottenere risposte in tempi decisamente ridotti rispetto alle stesse operazioni effettuate su altre tipologie di database.

Gli operatori di base di uno strumento OLAP sono:

ROLL UP: conosciuto anche come "consolidamento", comporta l'accumulo di tutti i dati che possono essere raccolti e il calcolo di tutte le loro relazioni in una o più dimensioni.



DRILL DOWN: è l'operazione inversa del roll-up. Inizia con un ampio set di dati, quindi suddiviso in parti più piccole, consentendo agli utenti di esaminare i dettagli.

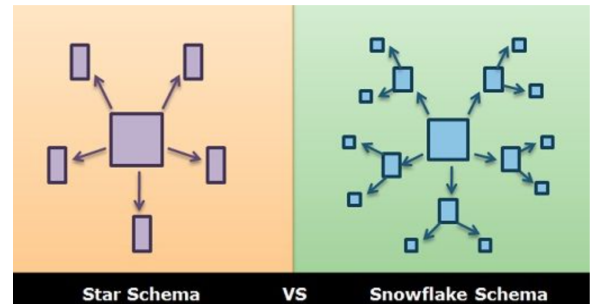


SLICE: L'operazione di slicing viene eseguita fissando un specifico valore per una delle dimensioni del "cubo", estraendo quindi una "fetta" e ottenendo un nuovo cubo con una dimensione in meno rispetto a quello di partenza;

DICE: esegue una selezione su una o più dimensioni. L'operazione di dicing viene eseguita quando l'analisi viene focalizzata su un sottoinsieme del "cubo".

Ma come possiamo riportare un cubo in un DBMS relazionale?

La modellazione multidimensionale su sistemi relazionali è basata sul cosiddetto schema a stella (star schema). L'altro schema disponibile per la modellazione viene detto schema a fiocco di neve (snowflake schema).



Uno schema a stella è composto da:

- Un insieme di tabelle, chiamate tabelle delle dimensioni (*dimension table*). Ciascuna di queste tabelle è caratterizzata da una chiave primaria e da un insieme di attributi che descrivono le dimensioni di analisi a diversi livelli di aggregazione;
- Una tabella chiamata tabella dei fatti (*fact table*) in cui sono presenti le chiavi di tutte le tabelle delle dimensioni. La chiave primaria di questa tabella sarà data dall'insieme delle chiavi esterne delle dimension table. La tabella dei fatti contiene inoltre un attributo per ogni misura.

Nel modello a fiocco di neve invece, pur confermandosi la presenza della tabella dei fatti centrale, una tabella dimensionale potrebbe essere collegata anche ad un'altra tabella dimensionale, nel caso in cui ne rappresenti un suo attributo.

3. **Simulazioni e previsioni (analisi what-if):** sono tecniche di analisi previsionale che permettono di valutare il comportamento di un sistema reale assumendo un particolare insieme di condizioni iniziali. Questo tipo di analisi supera uno dei limiti fondamentali di reportistica e OLAP, ossia quello di registrare solo il passato e di non consentire di analizzare scenari futuri. Per raggiungere l'obiettivo, bisogna essere in grado di disegnare un modello che riproduca con buona approssimazione il comportamento del sistema in esame; ovviamente maggiore sarà l'aderenza del modello alla realtà, maggiore sarà l'accuratezza delle stime ottenute.



4. **Data Mining:** si tratta di un processo per estrarre conoscenza da banche dati di grandi dimensioni tramite l'applicazione di algoritmi che individuano le associazioni nascoste tra le informazioni e le rendono visibili. Il data mining raccoglie quindi tecniche di intelligenza artificiale e pattern recognition per aiutare l'utente nella ricerca di pattern: è sufficiente indicare cosa e dove si vuole ricercare

## 2.3 Confronto tra competitor: Qlik Sense VS Power BI

Attualmente esistono molti fornitori di BI sul mercato. Tuttavia nel report del 2021, “Magic Quadrant for Analytics and Business Intelligence Platforms”, Gartner ha nominato solo tre fornitori come leaders: Power BI, Tableau e Qlik. In questo elaborato verranno analizzati Qlik Sense e Microsoft Power BI<sup>(3)</sup>, così come è stato fatto in USCO, per carpirne pro e contro.

Innanzitutto, il Quadrante Magico di Gartner è una mappa aggiornata dell'ecosistema delle soluzioni di BI disponibili, basata su una serie di rapporti di ricerca di mercato pubblicati dalla società di consulenza IT Gartner per dimostrare le tendenze del mercato, come direzione, maturità e partecipanti.

Questa società ha definito un set di criteri di valutazione che permettono di categorizzare i provider di tecnologia in una semplice matrice a quattro quadranti costruita su due variabili fondamentali:

- 1) *Completeness of vision*, ovvero quanto il provider è in grado di anticipare i trend di mercato, se non di influenzarli.
- 2) *Ability of execute*, ossia quanto il provider sia in grado di implementare le tecnologie nell'azienda cliente o partner.

Le quattro zone forniscono un posizionamento competitivo grafico di quattro tipi di fornitori di tecnologia: *leaders*, *visionaries*, *niche players* e *challengers*.



- Leaders : sulla carta, i migliori;
- Visionaries: capiscono dove sta andando il mercato o hanno una visione per cambiare le regole del mercato, ma non funzionano ancora bene;
- Niche Players: si concentrano con successo su un piccolo segmento;
- Challengers: possono dominare un ampio segmento, ma non dimostrano di comprendere la direzione del mercato;

Come si può notare dall'immagine, sia Qlik che Microsoft sono posizionati tra i Leaders: hanno rubato la scena a molti altri grazie alla loro caratteristica di riuscire a centralizzare in un unico software la connessione a sorgenti differenti per modellare i dati e la creazione di nuove formule e misure complesse.

Ma quali sono le effettive differenze tra le loro soluzioni di BI, Qlik Sense e Power BI? Mettiamoli a confronto:

	<b>Power BI</b>	<b>Qlik Sense</b>
<i>Versioni di base</i>	La versione di Power BI Desktop è gratuita	Qlik Sense non è un software gratuito, dispone però di una versione “desktop” gratuita, da usare tassativamente su un solo PC
<i>Installazione</i>	L'installazione di Power BI è molto semplice, avviene come l'installazione di un qualsiasi altro software sul pc.	L'installazione di Qlik Sense è piuttosto complessa e richiede tempo. E' necessario un esperto nel campo per iniziare.
<i>Capacità di memoria</i>	Con la versione gratuita di Power BI è disponibile fino a 1 GB di spazio di archiviazione, con la versione pro si può estendere il limite di archiviazione a 10 GB.	Qlik Sense non ha spazio di archiviazione gratuito e la versione cloud di archiviazione può fornire 500 GB di spazio di archiviazione.
<i>Funzionalità cloud</i>	Le origini dati cloud di Power BI supportano e sono compatibili	Qlik Sense può accedere all'origine dati cloud ma richiede una parte di

	principalmente con i server Microsoft Azure.	programmazione per recuperare i dati.
<i>Facilità di utilizzo</i>	Power BI risulta molto più semplice, grazie alla sua interfaccia utente molto intuitiva. Assomiglia infatti quasi ad una versione migliorata e integrata di Excel.	Qlik Sense non dispone di un'interfaccia intuitiva, risulta quindi più semplice da utilizzare per esperti di data science o programmatori.
<i>Prezzi</i>	Power BI richiede \$ 10 per utente al mese.	Il prezzo di Qlik Sense viene fornito a un prezzo quotato dopo aver contattato il team di marketing di Qlik: la cifra annuale per una decina di licenze con consulenza inclusa si aggira intorno ai 10.000/15.000€.
<i>Supporto per dispositivi mobili</i>	Power BI può supportare dispositivi mobili con buone caratteristiche e capacità, ma permette solo la visualizzazione e non la creazione di nuovi report.	Qlik Sense ha capacità e caratteristiche migliori in termini di supporto dei dispositivi mobili. L'app offre un'esplorazione completa dei dati online e offline.
<i>Intelligenza artificiale</i>	Alcune funzionalità di intelligenza artificiale sono state aggiunte a Power BI, ma per accedervi serve acquistare un pacchetto aggiuntivo. Inoltre queste funzionalità sono oggetti su una dashboard che bisogna aggiungere ogni volta.	L'intera gamma di funzionalità AI di Qlik è integrata nella piattaforma. Ciò consente all'AI di Qlik di suggerire approfondimenti e nuove connessioni mentre si esplorano i dati.

Per quanto riguarda l'aspetto dell'intelligenza artificiale, non sono state approfondite le differenze tra le due soluzioni nell'ambito del tirocinio.

Dopo un'accurata analisi, in USCO la scelta finale è ricaduta su Qlik Sense per vari motivi. In primo luogo è stato preso in considerazione il fatto che viene già utilizzato Qlik View come strumento per costruire report, di conseguenza passare a una BI di un provider completamente diverso come Microsoft avrebbe avuto costi maggiori per quanto riguarda il tempo e le risorse impiegate per

mettere in atto il passaggio tra due software differenti. In seguito, tenendo conto che l'azienda utilizza un gestionale non molto diffuso (DNA, ditta ITACME), si è notato che Qlik Sense dispone degli strumenti per importarne i dati, mentre Power BI ne è sprovvisto. Questo dettaglio ha notevolmente influito su quella che è stata la scelta finale, seppure Qlik Sense abbia richiesto il pagamento di un prezzo maggiore rispetto a quello che sarebbe stato per l'altro prodotto. Inoltre Qlik Sense offre un design liquido, con un'ottima adattabilità ai dispositivi mobile, e grazie all'intelligenza artificiale è in grado di proporre visualizzazioni grafiche in base agli input. Si tratta di un aspetto molto importante nell'ottica di rendere indipendente gli utenti finali, i quali possono personalizzare le proprie dashboards arrangiando in modi differenti i dati certificati forniti dagli sviluppatori, senza avere la necessità di richiedere aiuto tecnico.

### 3. Qlik Sense



#### 3.1 Descrizione software

Qlik<sup>(4)</sup> è una impresa di software statunitense con sede a King of Prussia, in Pennsylvania, ma fondata a Lund, in Svezia, nel 1993. Offre come prodotti QlikView, Qlik Sense ed NPrinting, software di visualizzazione e business intelligence che permettono il rapido sviluppo di dashboard completamente personalizzabili in grado di fornire rapidamente informazioni utili sui dati a disposizione.

Qlik Sense in particolare è un prodotto di visualizzazione e rilevazione dati che consente di creare visualizzazioni flessibili e interattive e prendere decisioni significative.

La maggior parte dei prodotti di BI permette di rispondere a domande che possono essere formulate in anticipo. Ma come è possibile gestire le domande di follow-up, ossia quelle che possono sorgere dopo che un utente ha letto un report o ha esaminato una visualizzazione? Con il modello associativo di Qlik Sense è possibile rispondere a una domanda dopo l'altra completando i propri percorsi personali per giungere alle proprie conclusioni ed è possibile esplorare liberamente i dati.

L'elemento centrale di Qlik Sense sono le cosiddette app. Un'app consiste in uno o più fogli contenenti visualizzazioni, ossia grafici, tabelle e rappresentazioni di dati in generale, combinate con altre informazioni, che sarà possibile riutilizzare, modificare e condividere con altri utenti.

Effettuando selezioni in queste visualizzazioni, è possibile analizzare le informazioni per ottenerne altre più approfondite sui dati in completa autonomia, senza la necessità di ricorrere a un esperto.

Questa possibilità di analisi autonoma da parte dell'utente viene coadiuvata dall'interfaccia utente user-friendly di Qlik Sense. Grazie al drag-and-drop e alla liquidità degli elementi a disposizione, si possono creare visualizzazioni non solo utili ma anche intuitive e graficamente piacevoli da consultare. Inoltre l'utente viene ulteriormente aiutato dalla presenza dell'intelligenza artificiale, in grado di suggerire visualizzazioni in base al tipo di dato che viene considerato.

Quando si creano visualizzazioni sono disponibili le risorse dati seguenti:

- *Campi*: contengono i dati caricati in Qlik Sense e corrispondono alle colonne delle tabelle di un database. I valori di un campo possono essere numerici o alfanumerici ossia possono essere qualitativi o quantitativi. Quando si creano visualizzazioni, si utilizzano per creare dimensioni e misure. Vediamo la differenza tra qualitativi e quantitativi.

I valori dei dati quantitativi sono misurati numericamente su una scala ascendente. Possono essere rapporti o intervalli:

- **Rapporto**: i rapporti sono dati quantitativi sui quali è possibile eseguire operazioni aritmetiche, come il costo o l'età.  
Ad esempio, è possibile sommare i valori delle vendite di un mese per ottenere i totali.
- **Intervallo**: gli intervalli sono dati quantitativi sui quali non è possibile eseguire operazioni aritmetiche.  
Ad esempio, non è possibile calcolare la somma delle temperature della settimana, anche se è possibile calcolare la temperatura media, massima e minima di una giornata.

I dati qualitativi non possono essere misurati numericamente, ma possono essere descritti attraverso il linguaggio. Possono essere nominali o ordinali:

- **Nominali:** i campi con dati nominali hanno valori qualitativi distinti, ma senza un ordine definito.  
Ad esempio, i nomi di prodotti o di clienti sono dati nominali, in quanto hanno valori distinti ma non hanno un ordine prescritto.
  - **Ordinali:** i campi con dati ordinali hanno valori qualitativi che possono essere ordinati o classificati. L'ordine dei dati ordinali non corrisponde all'ordine alfabetico.  
Ad esempio, basso, medio, alto sono valori ordinali. Anche piccolo, medio e grande sono valori ordinali.
- **Misure:** sono i dati che si desidera visualizzare, vengono create da un'espressione composta da funzioni di aggregazione, ad esempio Sum o Max, combinate con uno o più campi.
  - **Dimensioni:** determinano il metodo di raggruppamento dei dati all'interno della visualizzazione. Ad esempio: vendite totali per paese o numero di prodotti per fornitore. Le dimensioni mostrano i valori distinti del campo selezionato come dimensione. Possono anche essere calcolate utilizzando un'espressione.

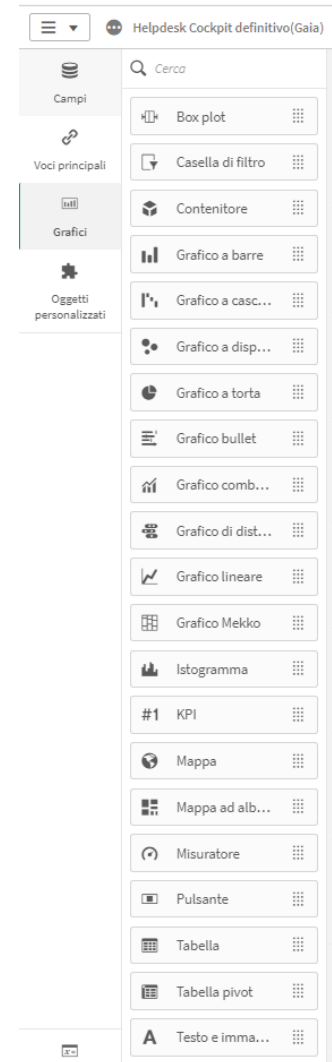
A nostra disposizione abbiamo poi le *espressioni*, una combinazione di funzioni, campi e operatori matematici (+ \* / =). Vengono utilizzate per elaborare i dati nell'app al fine di produrre un risultato visibile all'interno di una visualizzazione. Difatti le espressioni sono utilizzate principalmente per creare misure, dimensioni calcolate o per impostare proprietà all'interno di visualizzazioni diverse. Ad esempio, è possibile utilizzare espressioni per definire i limiti degli intervalli dei misuratori o le linee di riferimento dei grafici a barre.

Inoltre è possibile realizzare visualizzazioni più dinamiche e avanzate con espressioni per titoli, sottotitoli o note a piè di pagina. Questo significa, ad esempio, che al posto del titolo di una visualizzazione costituito da testo statico, è possibile utilizzare un'espressione il cui risultato cambia in base alle selezioni effettuate.

Grazie a questi elementi di base si possono creare varie visualizzazioni, partendo da quelle basilari: i grafici, che illustrano i dati con elementi visivi come barre, linee o punti e le visualizzazioni basate su dati testuali, come tabelle o filtri.

Qui a lato possiamo vedere l'elenco delle possibilità che l'utente ha per creare le visualizzazioni, tra cui ad esempio:

- Grafico a barre: visualizza una barra per ogni valore. La lunghezza della barra corrisponde al relativo valore numerico della misura.
- Grafico combinato: riunisce le barre e le linee nello stesso grafico. Le barre e le linee presentano assi diversi che consentono di confrontare le percentuali e le somme.
- Grafico di distribuzione: è adatto per il confronto degli intervalli e delle distribuzioni di gruppi di dati numerici. I dati sono tracciati come punti dei valori lungo un asse.
- Misuratore: consente di visualizzare il valore di una singola misura, senza dimensioni. Graficamente ricorda un tachimetro.
- Istogramma: adatto per visualizzare la distribuzione di dati numerici su un intervallo continuo o per un certo periodo di tempo.
- Grafico lineare: visualizza le linee dei dati tra i valori. Vengono spesso utilizzati per visualizzare una tendenza nei dati su un intervallo di tempo.
- Mappa: viene utilizzata per combinare dati geospaziali e valori di misura, come le vendite relative a una regione o a un negozio.
- Mappa ad albero: mostra i dati gerarchici e può mostrare un gran numero di valori simultaneamente all'interno di uno spazio limitato.
- Grafico a torta: visualizza la relazione tra una singola dimensione e una singola misura.
- Grafico a dispersione: presenta i valori da due misure. È utile per rappresentare i dati nei casi in cui a ciascuna istanza corrispondano due numeri, ad esempio una nazione (popolazione e relativo tasso di crescita). È possibile utilizzare una terza misura opzionale che viene riflessa nella dimensione delle bolle. Quando si visualizzano serie di dati di grandi



dimensioni, per rappresentare la dimensione della misura verranno utilizzati i colori anziché la dimensione della bolla.

- Casella di filtro: consente di controllare i dati mostrati nelle visualizzazioni su un foglio, agendo contemporaneamente sui dati di più dimensioni.
- KPI: viene utilizzato per presentare dati quantitativi fondamentali relativi alle prestazioni.
- Tabella pivot: mostra dimensioni e misure come righe e colonne di una tabella. Consente di analizzare i dati contemporaneamente in più dimensioni, i quali possono essere raggruppati in base a una combinazione di dimensioni e possono essere visualizzate anche somme parziali.
- Tabella: visualizza i valori sotto forma di record, in modo che ogni riga della tabella contenga i campi calcolati utilizzando le misure. In generale, una tabella contiene una dimensione e diverse misure.
- Testo e immagine: è possibile utilizzare la visualizzazione testo e immagine per aggiungere testo, immagini, misure e collegamenti a una pagina Web.
- Pulsante: è possibile usare i pulsanti per aggiungere collegamenti rapidi ai fini di una facile selezione e navigazione nella propria app.

Per utilizzare questi elementi è sufficiente trascinare la visualizzazione sul foglio su cui si sta lavorando oppure cliccare sull'icona corrispondente due volte, aggiungere le dimensioni e le misure che servono (il numero di dimensioni e misure da aggiungere varia in base al tipo di visualizzazione) e infine regolare la presentazione, ad esempio applicando un ordinamento, colori o etichette.

Se le visualizzazioni predefinite non soddisfano lo scopo, Qlik Sense dispone di una serie di estensioni scaricabili da installare.

### **3.2 Gestione dati e sintassi**

Qlik Sense consente di connettersi ai dati di una vasta gamma di sorgenti grazie a un'ampia scelta di connettori Qlik e di altri tipi di connessioni dati.

È possibile connettersi a database, dati di social media, file locali, file remoti e file Web di cui si è proprietari o per cui si possiedono diritti di accesso in lettura o aggiornamento



Ad esempio, tramite connettori di database ODBC preconfigurati è possibile connettersi a una sorgente dati ODBC come Amazon Redshift, Apache Drill, Azure SQL, Google BigQuery, MongoDB, Oracle, PostgreSQL, Microsoft SQL Server e molti altri.

Oppure, come già anticipato, è possibile connettersi anche a social media o a sorgenti dati basate sul Web, come ad esempio Github, Dropbox, Google Analytics/Calendar/Ad Manager/Drive, Facebook Manager, Twitter, Office 365 Sharepoint, Youtube Analytics, Outlook 365 e tanti altri ancora.

Per quanto riguarda i file locali come sorgente, Qlik Sense è in grado di leggere i dati dai fogli di calcolo di Microsoft Excel. I formati di file supportati sono *XLS*, *XLSX*, *XLW* e *XLSM*. Nella fase di caricamento dati è possibile selezionare aree denominate di un foglio, come le colonne, un singolo foglio o tutti i fogli del file di un foglio di calcolo. Ciascun foglio di Excel viene poi caricato in una tabella separata su Qlik Sense, le quali rimangono sempre consultabili per avere una visione generale della base dati che si ha a disposizione per creare app.

Dal punto di vista della sicurezza e la protezione da modifiche non autorizzate dei dati, Qlik Sense offre quello che viene chiamato “*Section Access*”: rappresenta una parte dello script di caricamento dei dati in cui si aggiunge una tabella di sicurezza per definire i diritti di visualizzazione e modifica dei dati e app. Qlik Sense quindi utilizza tali informazioni per ridurre i dati all'ambito appropriato quando l'utente apre l'applicazione, pertanto alcuni dei dati nell'app vengono nascosti dall'utente in base alla relativa identità.

Avendo nominato lo script di caricamento dei dati, vediamo di approfondire l'argomento.

In uno script di caricamento dei dati vengono definiti il nome della sorgente dati, i nomi delle tabelle e i nomi dei campi inclusi nella logica. La sintassi dello script e la sintassi della riga di comando di Qlik Sense vengono descritte in una notazione denominata metasintassi Backus-Naur Form o codice BNF.

La BNF (Backus-Naur Form o Backus Normal Form)<sup>(5)</sup> è una metasintassi, ovvero un formalismo attraverso cui è possibile descrivere la sintassi di linguaggi formali (il prefisso *meta* ha proprio a che vedere con la natura *circolare* di questa definizione). Si tratta di uno strumento molto usato per descrivere in modo preciso e non ambiguo la sintassi dei linguaggi di programmazione, dei protocolli di rete e così via,

Nella tabella indicata di seguito viene fornito un elenco di simboli utilizzati nel codice BNF <sup>(6)</sup>, con una descrizione di come vengono interpretati:

**Simbolo    Descrizione**

	Operatore OR logico: il simbolo può essere utilizzato da entrambi i lati.
()	Parentesi che stabiliscono la precedenza: vengono utilizzate per strutturare la sintassi BNF.
[ ]	Parentesi quadre: gli elementi racchiusi sono opzionali.
{ }	Parentesi graffe: gli elementi racchiusi possono essere ripetuti zero o più volte.
Simbolo	Categoria sintattica nonterminale: può essere suddivisa ulteriormente in altri simboli. Ad esempio, composizione dei simboli precedenti, altri simboli nonterminali, stringhe di testo e così via.
::=	Contrassegna l'inizio di un blocco che definisce un simbolo.
LOAD	Simbolo terminale che consiste in una stringa di testo. Deve essere scritto nello script così come è visualizzato.

Una specifica BNF è un insieme di regole di derivazione, ciascuna espressa nella forma:

<simbolo> ::= \_\_espressione\_\_

Nelle regole di derivazione <simbolo> (i caratteri < e > sono obbligatori) viene detto *non-terminale* e \_\_espressione\_\_ è costituita da una o più sequenze di

simboli terminali o non-terminali, identificabili perché racchiusi tra < >; se le sequenze sono più di una, sono separate dalla barra verticale “ | “. La regola esprime il fatto che il non-terminale a sinistra della regola può essere sostituito da una qualsiasi delle sequenze indicate sulla destra. Inoltre in una sequenza alcuni simboli o sottosequenze possono essere indicati come opzionali racchiudendoli fra parentesi quadre.

I simboli che non appaiono mai a sinistra di una regola di derivazione sono detti *terminali*. I terminali sono in un certo senso il punto di arrivo, perché rappresentano elementi che si troveranno effettivamente in un testo scritto secondo le regole sintattiche che la specifica BNF descrive.

Per fornire un esempio, prendiamo il caso di voler descrivere in modo formale le regole da seguire per scrivere l'indirizzo su una lettera:

```
<indirizzo postale> ::= <destinatario> <indirizzo> <localita>  
<destinatario> ::= [<titolo>] [<nome>|<iniziale>] <cognome> <a capo>  
<indirizzo> ::= <via> <numero civico> <a capo>  
<localita> ::= [<CAP>] <comune> <provincia>
```

Questo frammento di specifica può essere tradotto in italiano come segue:

- ▷ un indirizzo postale include un destinatario, seguito da un indirizzo, seguito da una indicazione di località;
- ▷ il destinatario comprende sicuramente un cognome, a cui si possono far precedere, nell'ordine, un titolo (come *Sig.* o *Dott.* ecc.) e un nome o una iniziale;
- ▷ l'indirizzo comprende necessariamente una indicazione di via e il numero civico;
- ▷ l'indicazione della località comprende un codice di avviamento postale opzionale, seguito dal nome del comune e dalla provincia.

La specifica della sintassi di un linguaggio di programmazione ha tipicamente un non-terminale principale <programma> e un insieme di regole di derivazione che descrivono come un programma è strutturato in moduli, i moduli in istruzioni, le istruzioni in espressioni, e via dicendo.

Più avanti nel testo avremo modo di vedere questa sintassi applicata alle espressioni create su Qlik Sense.

Quando si crea un nuovo file Qlik Sense le prime righe di codice vengono generate automaticamente. I valori predefiniti di queste variabili di interpretazione numerica derivano dalle impostazioni internazionali del sistema operativo.

Una variabile in Qlik Sense può essere definita come un raccoglitore che memorizza un valore statico o un calcolo, ad esempio un valore numerico o alfanumerico. Quando si utilizza la variabile nell'app, qualsiasi modifica apportata alla variabile viene applicata ovunque venga utilizzata. Per impostare il valore di una variabile si utilizzano le istruzioni Let o Set nello script di caricamento dei dati, utilizzando la sintassi:

```
set variablename = string
```

oppure

```
let variable = expression
```

Il comando Set inserisce il testo a destra del segno dell'uguale rispetto alla variabile, mentre il comando Let valuta l'espressione.

Le variabili di interpretazione numerica sono definite dal sistema, come già anticipato, ossia vengono generate automaticamente in base alle impostazioni regionali attuali del sistema operativo al momento della creazione di una nuova app. In Qlik Sense ciò avviene in base alle impostazioni del sistema operativo del computer oppure in base al sistema operativo del server su cui è installato.

Tra tutte abbiamo *DateFormat* e *TimeFormat*, che definiscono il formato col quale vogliamo visualizzare le date e gli orari; *LongDayNames* che sostituisce la convenzione usata per i nomi estesi dei giorni della settimana del sistema operativo e *FirstWeekDay*, numero intero che definisce il giorno da utilizzare come primo giorno della settimana.

```
1 SET ThousandSep=',';
2 SET DecimalSep='.';
3 SET MoneyThousandSep=',';
4 SET MoneyDecimalSep='.';
5 SET MoneyFormat='$$,##0.00;-$$,##0.00';
6 SET TimeFormat='h:mm:ss TT';
7 SET DateFormat='M-D-YYYY';
8 SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
9 SET FirstWeekDay=6;
10 SET BrokenWeeks=1;
11 SET ReferenceDay=0;
12 SET FirstMonthOfYear=1;
13 SET CollationLocale='en-US';
14 SET CreateSearchIndexOnReload=1;
15 SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
16 SET LongMonthNames='January;February;March;April;May;June;July;August;September;October;November;December';
17 SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';
18 SET LongDayNames='Monday;Tuesday;Wednesday;Thursday;Friday;Saturday;Sunday';
19 SET NumericalAbbreviation='3:k;6:M;9:G;12:T;15:P;18:E;21:Z;24:Y;-3:m;-6:µ;-9:n;-12:p;-15:f;-18:a;-21:z;-24:y';
20
```

In generale, lo script di Qlik Sense è costituito da una serie di istruzioni. Un'istruzione può essere un'istruzione di script regolare o un'istruzione di controllo dello script:

- Le istruzioni regolari vengono generalmente utilizzate per la manipolazione dei dati. Queste possono essere scritte su un qualsiasi numero di righe nello script e devono sempre terminare con un punto e virgola.
- Le istruzioni di controllo invece vengono utilizzate per controllare il flusso di esecuzione dello script. Ogni clausola di un'istruzione di controllo deve essere mantenuta in una singola riga dello script e può terminare con un punto e virgola oppure con un fine riga.

Possiamo avere:

**Call** → consente di chiamare una subroutine che deve essere definita da un'istruzione sub precedente.

```
Call name ( [ paramlist ] )
```

**Do..loop** → è un costrutto per la ripetizione di script che esegue una o più istruzioni finché non incontra una condizione logica.

```
do [ ( while | until ) condition ] [statements]  
[exit do [ ( when | unless ) condition ] [statements]  
loop [ ( while | until ) condition ]
```

**Exit script** → interrompe l'esecuzione dello script. Può essere inserita in un punto qualsiasi dello script.

```
exit script[ (when | unless) condition ]
```

**For each ..next** → costrutto per la ripetizione di script che esegue una o più istruzioni per ogni valore in un elenco le cui voci sono separate da virgole. Le istruzioni incluse nel ciclo fra for e next verranno eseguite per ogni valore nell'elenco.

```
For Each var in list [statements] [exit for [ ( when | unless )  
condition ] [statements] next [var]
```

**For..next** → costrutto per la ripetizione di script con un contatore. Le istruzioni all'interno del ciclo incluso tra for e next verranno eseguite per ogni valore del contatore in base ai limiti inferiore e superiore specificati.

```
For counter = expr1 to expr2 [ stepexpr3 ] [statements] [exit  
for [ ( when | unless ) condition ] [statements] Next [counter]
```

**If..then** → costruito per la selezione di script che forza l'esecuzione dello script su percorsi diversi in base a una o più condizioni logiche.

```
If condition then [ statements ] { elseif condition then [ statements ] } [ else [ statements ] ] end if
```

**Sub** → definisce una subroutine che può essere richiamata da un'istruzione call.

```
Sub name [ ( paramlist ) ] statements end sub
```

**Switch** → costruito per la selezione di script che forza l'esecuzione dello script su percorsi diversi, in base al valore di un'espressione.

```
Switch expression {case valuelist [ statements ]} [default statements] end switch
```

Dopo aver preso visione di come è strutturato Qlik Sense in generale, passiamo all'analisi del progetto che ho avuto modo di sviluppare su questo strumento.

## 4. Sviluppo dashboard Helpdesk

### 4.1 Motivazioni

All'interno dell'azienda viene utilizzato dal 2018 un servizio di Helpdesk su SharePoint dove tutti i dipendenti hanno la possibilità di aprire ticket. I ticket sono richieste di assistenza che possono riguardare la parte software (problemi col sistema gestionale interno, mail, e-commerce o in generale con programmi software di utilizzo comune nelle varie aree aziendali ) o la parte hardware (stampanti, schermi, telefonia ecc.).

Attraverso il pannello di controllo su SharePoint i dipendenti dell'ufficio IT possono gestire tutti i ticket che vengono creati: assegnano a ciascuno di loro le richieste in base alle rispettive competenze, in modo da dividersi il carico di lavoro; stabiliscono la tipologia della richiesta (problema, incidente,

progetto, domanda...); marcano come “solved” o “in progress” un ticket risolto o che richiede un riscontro o collaborazione dal dipendente che lo ha aperto; assegnano a ogni ticket il tempo che è stato impiegato per la risoluzione del problema. Stabilire il tempo di risoluzione impiegato è essenziale per creare statistiche sull’efficienza dell’ufficio IT, il quale non dovendo svolgere solo questa attività di assistenza necessita di un feedback costante a riguardo del tempo impiegato in totale a risolvere le problematiche tecniche dei dipendenti.

L’Helpdesk su Sharepoint non dispone di questa funzionalità, difatti fino a poco tempo fa le statistiche venivano realizzate su Excel dopo aver esportato i dati da SharePoint.

Con l’acquisto di Qlik Sense si è pensato di creare una dashboard apposita per l’Helpdesk che fosse il più completa possibile, per esaminare le prestazioni a livello di ufficio e di singolo addetto.

## **4.2 Analisi delle richieste**

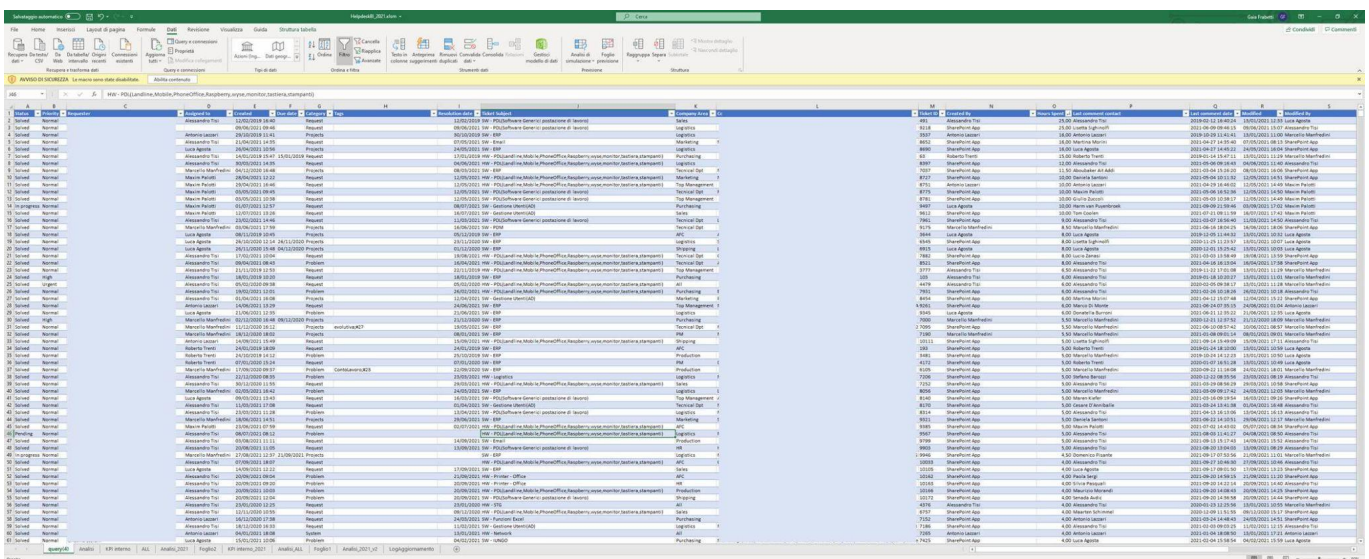
Dopo aver analizzato attentamente la situazione corrente e le esigenze di ciascun addetto IT, i dati che sono risultati fondamentali da graficare in questa app composta da 3 fogli sono:

- numero di ticket assegnati a ciascuna persona dell'ufficio
- tempo speso totale/da ognuno per ogni categoria di ticket
- tempo complessivo/di ciascuna persona speso risolvendo ticket
- numero di ticket per categoria
- numero totale di ticket
- numero di ticket per ogni requester
- percentuale di tempo speso per risolvere ticket sul tempo totale di lavoro
- percentuale di ticket risolti sul totale di ticket creati
- tempo medio di chiusura ticket

In seguito è stata avanzata un'ulteriore richiesta: fare in modo che i grafici fossero di facile consultazione, intuitivi e disposti in modo logico all'interno della dashboard.

### 4.3 Fonte dati e caricamento

Le fondamenta di tutto il processo di creazione di questo progetto è la base dati derivante dall'aggiornamento automatico e giornaliero di un file Excel, il quale riceve dati dall'Helpdesk su Sharepoint..



Nel file Excel ogni colonna contiene valori differenti (data di creazione dei ticket, nome dell'utente, categoria, area aziendale, tempi di risoluzione ecc) e ciascuna di esse viene importata in Sense con la sintassi LOAD, diventando un campo ed eventualmente parametro per espressioni nella creazione di misure.

Su Sense è disponibile un'area apposita dentro la dashboard su cui si lavora per importare tramite uno script la base dati necessaria. Nel caso di questo progetto, lo script creato è il seguente:

**Variabili:**

```
Let vCurrentYearNum = year(today());
Let vCurrentMonthNum = num(month(today()));
Let vCurrentDayNum = day(today());
```



```
Let v7DaysAgo = today()-7;
Let v7DaysAgoTrunc = MakeDate(year(v7DaysAgo), Month(v7DaysAgo), Day(v7DaysAgo));
Let v30DaysAgo = today()-30;
Let v30DaysAgoTrunc = MakeDate(year(v30DaysAgo), Month(v30DaysAgo),
Day(v30DaysAgo));
Let vToday = today();
```

**main:**

```
SET ThousandSep='.';
SET DecimalSep='';
SET MoneyThousandSep='.';
SET MoneyDecimalSep='';
SET MoneyFormat='#.##0,00 €;-#.##0,00 €';
SET TimeFormat='hh:mm:ss';
SET DateFormat='DD/MM/YYYY';
SET TimestampFormat='DD/MM/YYYY hh:mm:ss[.fff]';
SET FirstWeekDay=0;
SET BrokenWeeks=0;
SET ReferenceDay=4;
SET FirstMonthOfYear=1;
SET CollationLocale='it-IT';
SET CreateSearchIndexOnReload=1;
SET MonthNames='gen;feb;mar;apr;mag;giu;lug;ago;set;ott;nov;dic';
SET
LongMonthNames='gennaio;febbraio;marzo;aprile;maggio;giugno;luglio;agosto;settembre;ott
obre;novembre;dicembre';
SET DayNames='lun;mar;mer;gio;ven;sab;dom';
SET LongDayNames='lunedì;martedì;mercoledì;giovedì;venerdì;sabato;domenica';
SET NumericalAbbreviation=
'3:k;6:M;9:G;12:T;15:P;18:E;21:Z;24:Y;-3:m;-6:µ;-9:n;-12:p;-15:f;-18:a;-21:z;-24:y';
```

**LOAD**

```
Status,
Priority,
Requester,
"Assigned to",
Created,
"Due date",
Category,
"Tags",
"Resolution date",
"Ticket Subject",
"Company Area",
Cc,
```

"Ticket ID",  
"Created By",  
"Hours Spent",  
"Last comment contact",  
"Last comment date",  
Modified,  
"Modified By",  
"Service rate",  
"Support channel",  
Title,  
Cancelled,  
"Item Type",  
Path,  
CreatedOK,  
ResolutionOK,  
SAL as SLA,  
Fascia,  
"Mese Created",  
Company,  
"Anno Created",  
Period,  
"week",  
"week-anno"

```
FROM [lib://HELPDESK_BI (usco_luca.agosta)/HelpdeskBI_2021.xlsm]  
(ooxml, embedded labels, table is [query(4)]);
```

Nella parte di LOAD si può notare che alcuni nomi di colonne vengono inseriti tra virgolette (""): questo è dovuto al fatto che durante questa fase viene fatta l'associazione tra un nome senza spazi e una colonna. Se viene inserito un nome con uno spazio senza virgolette, il software lo riconosce come due nomi separati, andando a creare due colonne separate, sbagliando.

Helpdesk Cockpit defaultivo(Sala) | Prepara | Editor caricamento dati | Analisi | Foglio | Nome | Racconto | Carica dati

VARIABLE | Main

```

1 Let vCurrentYearMon = year(today());
2 Let vCurrentMonthMon = num(month(today()));
3 Let vCurrentDayMon = day(today());
4 Let vDaysAgo = today()-7;
5 Let vDaysAgoTrunc = MakeDate(year(vDaysAgo), Month(vDaysAgo), Day(vDaysAgo));
6 Let v30DaysAgo = today()-30;
7 Let v30DaysAgoTrunc = MakeDate(year(v30DaysAgo), Month(v30DaysAgo), Day(v30DaysAgo));
8 Let vToday = today();
9

```

Output

Tutte le modifiche sono state salvate

Conessioni dati

- ArchivedLogFolder
- Cartella
- File allegati
- COMMON
- DIREZIONALI
- downloads
- HELPOESK\_BI (socio\_luca.agosta)
- IMPORT
- IMPORT\_SENSE (socio\_service.sense)
- IMPORTQLIK
- INCLUDE
- INCLUDELOCAL
- monitor\_apps\_glikview\_logs

Helpdesk Cockpit defaultivo(Sala) | Prepara | Editor caricamento dati | Analisi | Foglio | Nome | Racconto | Carica dati

VARIABLE | Main

```

1 SET ThousandSep=".";
2 SET Decimals="";
3 SET MoneyThousandSep=".";
4 SET MoneyDecimals="";
5 SET MoneyFormat="#,###,00 €;#,###,00 €";
6 SET TimeFormat="hh:mm:ss";
7 SET DateFormat="DD/MM/YYYY";
8 SET FirstMonday="DD/MM/YYYY hh:mm:ss[.fff]";
9 SET BrokenWeek="0";
10 SET ReferenceDay="4";
11 SET FirstMonthOfYear="1";
12 SET CollationLocal="it-IT";
13 SET CreateSearchIndexOnLoad="1";
14 SET MonthNames="gen;feb;mar;apr;mag;giu;lug;ago;set;ott;nov;dic";
15 SET LongMonthNames="gennaio;febbraio;marzo;aprile;maggio;giugno;luglio;agosto;settembre;ottobre;novembre;diciembre";
16 SET DayNames="Lunedì;Martedì;Mercoledì;Giovedì;Venerdì;Sabato;Domenica";
17 SET LongDayNames="Lunedì;Martedì;Mercoledì;Giovedì;Venerdì;Sabato;Domenica";
18 SET NumericAbbreviation="3;12;6;9;6;12;7;5;5;9;18;6;21;22;24;7;3;6;9;9;9;15;15;18;18;21;24;7";
19
20
21
22
23
24
25 LOAD //tra virgolette ci vanno i nomi delle colonne che hanno uno spazio in mezzo altrimenti si confonde e ne crea 2
26 Status,
27 Priority,
28 Requester,
29 "Assigned to", //data e ora di creazione formato gg/mm/aaaa xx:xx
30 Created,
31 "Due date",
32 Category,
33 "Resolution date", //data e ora di risoluzione
34 "Ticket Subject",
35 "Company Area", //colonna che indica a chi è stato inoltrato la mail per conoscenza
36 Cr,
37 "Ticket ID",
38 "Created By",
39 "Hours Spent",
40 "Last comment contact", //ultima persona che ha risposto in un ticket
41 "Last comment date", //data di ultimo risposta nel ticket
42 Modified,
43 "Modified By",
44 "Service rate",
45 "Support channel",
46 Title, //titolo del ticket
47 Canceled, //contiene vero o falso in base a se il ticket è stato cancellato o meno
48 "Ticket type",
49 Path,
50 CreatedAt, //solo la data della colonna "created" (che ha anche l'orario)
51 ResolvedAt, //solo la data della colonna "resolution date"
52 SLA, as SLA, //tempo che intercorre tra risoluzione e creazione del ticket
53 Faccia, //contiene le fasce di tempi di risoluzione (1 day, 3-3 day, 3-7 day, 7 day...)
54 "Mese Created", //numero del mese della data di creazione ticket
55 Company, //nome della compagnia di appartenenza della persona che ha aperto il ticket
56 "Anno Created",
57 Period, //mese e anno di creazione
58 "week", //numero di settimana di creazione
59 "week-anno", //numero di settimana e anno di creazione
60 FROM [lib://HELPOESK_BI (socio_luca.agosta)/HelpdeskBI_2021.xlsx]
61 (column, embedded labels, table in [query4]);
62
63
64
65
66
67
68
69

```

Output

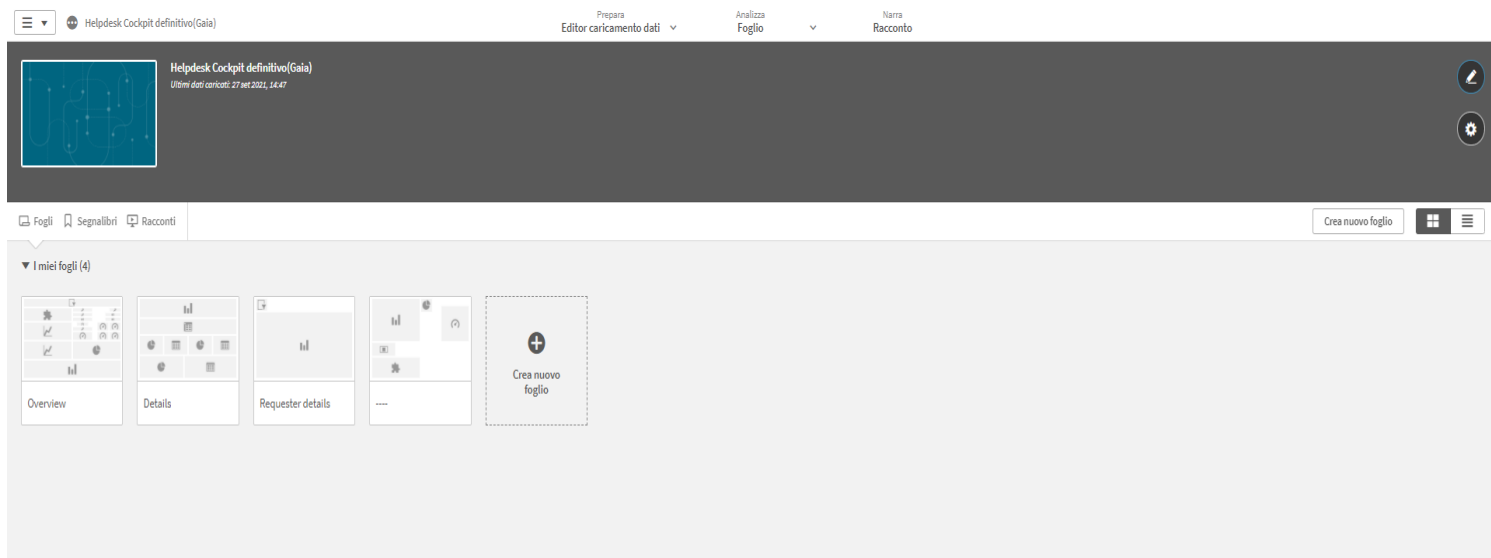
Tutte le modifiche sono state salvate

Conessioni dati

- ArchivedLogFolder
- Cartella
- File allegati
- COMMON
- DIREZIONALI
- downloads
- HELPOESK\_BI (socio\_luca.agosta)
- IMPORT
- IMPORT\_SENSE (socio\_service.sense)
- IMPORTQLIK
- INCLUDE
- INCLUDELOCAL
- monitor\_apps\_glikview\_logs

## 4.4 Creazione: strumenti e metodi utilizzati

Preso atto delle richieste sopra elencate, come prima cosa ho deciso di dividerle nel modo più logico possibile. Da ciò ne deriva la creazione di tre fogli, vale a dire tre pagine contenenti vari elementi grafici al loro interno: “Overview”, “Details”, “Requester Details”.

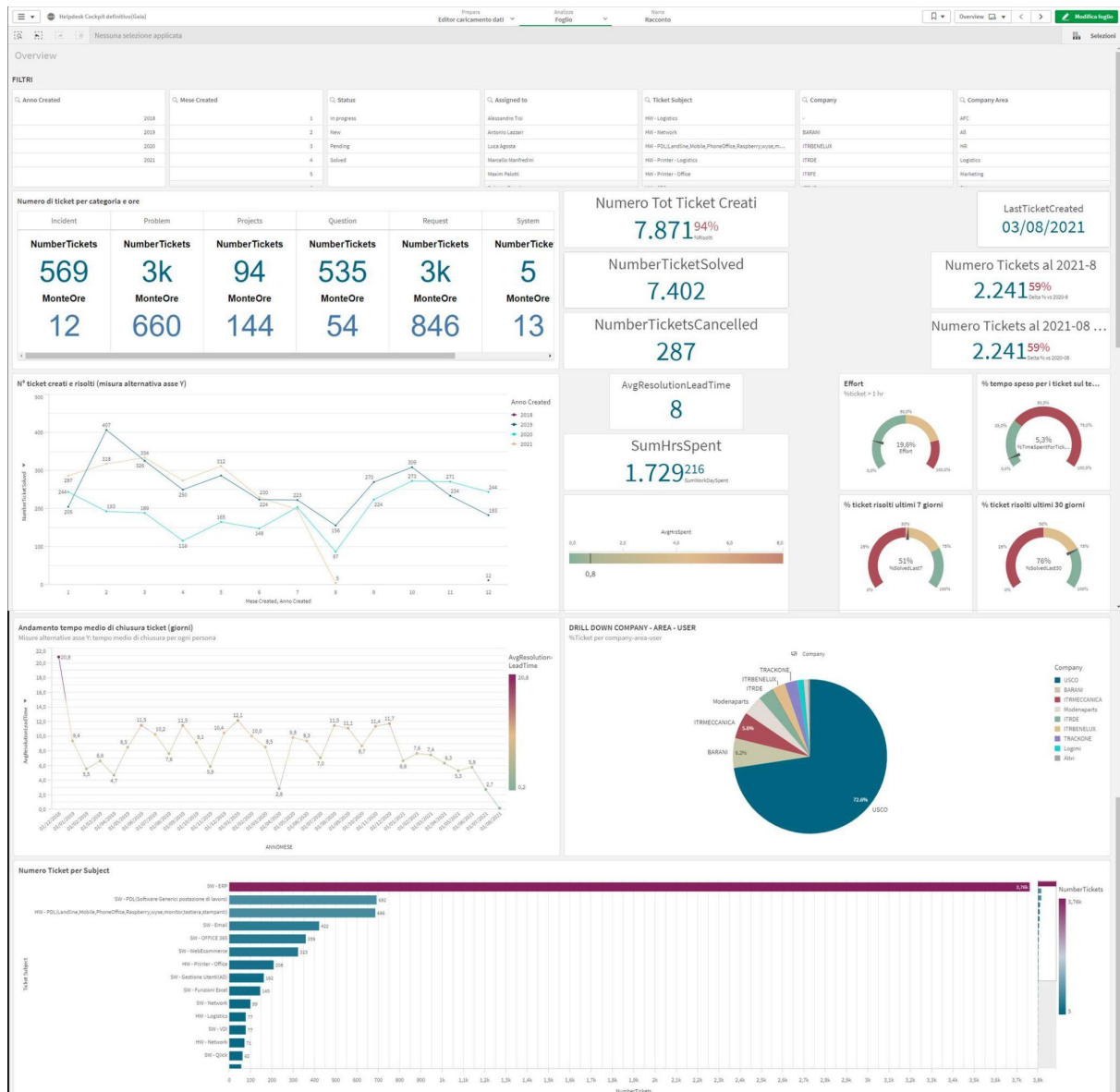


All’interno di ciascun foglio, grazie al design liquido di Sense, ho avuto la possibilità di scegliere e adattare grafici, tabelle, kpi ecc. come meglio pensavo per poter rappresentare le statistiche richieste, lasciando allo stesso tempo che rimanessero intuitive e facilmente consultabili.

Esaminando tutte le alternative a disposizione, ho optato per l’utilizzo di solo alcuni strumenti, in quanto non tutti avrebbero soddisfatto la richiesta di rappresentare i dati in modo efficace.

Tra i selezionati abbiamo i grafici a barre, lineari e a torta, misuratori, kpi, tabelle lineari e tabelle pivot. Infine è stato aggiunta una casella di filtro in cima al primo foglio, contenente le dimensioni e i campi principali, per poter selezionare solo i dati necessari alle analisi.

## 4.4.1. Primo foglio: Overview



Il foglio “Overview” ha lo scopo di fornire una panoramica sui dati consultati più di frequente, per questo motivo ho inserito :

- un box composto da più kpi che per ogni categoria mostra il numero di tickets e il numero di ore che sono state impiegate per risolverli;
- un'altra serie di kpi, in questo caso disposti singolarmente, raffiguranti il numero di tickets totali creati, risolti o cancellati , il numero di ore spese e il tempo medio (espresso in giorni) per risolverli. A lato invece sono stati aggiunti un indicatore con la data dell'ultimo ticket creato e un ulteriore paio che mostrano il numero di ticket creati fino a una determinata data. Questi ultimi possono sembrare identici ma hanno una caratteristica che li

differenza: la sensibilità alla selezione dei filtri. In questo modo, quando si va a selezionare una data diversa da quella attuale, uno dei due riporterà sempre i dati aggiornati ad oggi;

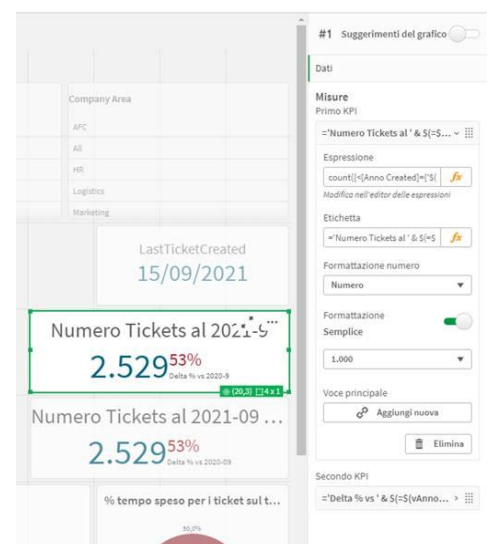
- quattro misuratori (tachimetri) per stimare l'effort (percentuale dei tickets che hanno richiesto più di un'ora per essere risolti), la percentuale di ticket risolti negli ultimi 7 e 30 giorni e la percentuale di tempo speso sul totale delle ore lavorative;
- due grafici lineari per mostrare in base ai mesi e agli anni l'andamento del numero di ticket creati e risolti e del tempo medio di chiusura dei ticket (valutata in giorni). Quest'ultimo ha una misura alternativa sull'asse delle ordinate per valutare il tempo medio per ogni dipendente dell'ufficio IT.
- un grafico a torta basato su una dimensione drill-down che permette di vedere a prima vista la percentuale di tickets per ogni azienda facente parte del gruppo USCO; cliccando su una parte del grafico permette di vedere poi l'area aziendale da cui provengono i ticket e per ultimo gli utenti che li creano, seguendo lo stesso procedimento ;
- un grafico a barre per mostrare qual è la materia principale dei ticket creati (hardware o software).

Procediamo con l'analisi di alcuni elementi grafici:

## - Numero di tickets creati alla data attuale

I KPI, acronimo di *Key Performance Indicators*, sono gli indicatori di performance più importanti, utili a misurare le performance di un'azienda per determinare quanto sia vicina a raggiungere un obiettivo strategico per la propria attività. Proprio per questo motivo sono il tipo di visualizzazione grafica migliore per lo scopo di questa dashboard.

Sense offre la possibilità di inserire due kpi in uno, come si può vedere in questo caso. La finestra di modifica permette di definire il primo e il secondo kpi, assegnando a ciascuno una misura o una formula. Per ciascuno di essi è poi possibile aggiungere un'etichetta con testo e/o variabili,



scegliere la formattazione del numero (automatica o a nostra scelta tra valuta, percentuale, numero con decimali...) e il colore.

Il kpi principale (di colore blu) rappresenta il numero di ticket alla data attuale e come anticipato in precedenza non è sensibile alla scelta dei filtri. Nella formula della misura infatti si può osservare la sintassi impiegata per ottenere questo risultato: ad esempio “*Category=,*” implica che la misura è indipendente dalle scelte nel filtro sulla categoria che troviamo in alto nel foglio.

Espressione:

```
count({<[Anno Created]={'$(= $(vCurrentYear))'},[Mese Created]='<=$(= $(vCurrentMonthNum))"},Category=,Status=,[Assigned to]=,[Ticket Subject]=,Requester=,[Company Area]=,Company=>}distinct [Ticket ID])
```



Lo scopo di questa espressione è contare (count()) il numero di ticket in base al ticket ID (distinct [Ticket ID]) tenendo conto solo dell'anno corrente (vCurrentYear) fino al mese corrente (vCurrentMonthNum) senza considerare eventuali filtri applicati sulla categoria, lo stato del ticket, a chi è assegnato, la materia, l'utente che l'ha richiesto, l'area dell'azienda o l'azienda del gruppo da cui proviene.

Il kpi secondario (di colore rosso) rappresenta in percentuale la differenza di ticket creati alla data attuale rispetto alla stessa data dell'anno precedente.

Espressione:

*(count({<[Anno Created]='{'\$(\$vCurrentYear)'}',[Mese Created]='{"<=\$(\$vCurrentMonthNum)"}',Category=,Status=,[Assigned to]=,[Ticket Subject]=,Requester=,[Company Area]=,Company=>} distinct [Ticket ID])*

-

*count({<[Anno Created]='{'\$(\$vAnnoPrecedente)'}',[Mese Created]='{"<=\$(\$vCurrentMonthNum)"}',Category=,Status=,[Assigned to]=,[Ticket Subject]=,Requester=,[Company Area]=,Company=>} distinct [Ticket ID]))*

/

*count({<[Anno Created]='{'\$(\$vAnnoPrecedente)'}',[Mese Created]='{"<=\$(\$vCurrentMonthNum)"}',Category=,Status=,[Assigned to]=,[Ticket Subject]=,Requester=,[Company Area]=,Company=>} distinct [Ticket ID])*



```
1 (
2 count({<[Anno Created]='{'$($vCurrentYear)'}',[Mese Created]='{"<=$($vCurrentMonthNum)"}',Category=,Status=,[Assigned to]=,[Ticket Subject]=,Requester=,[Company Area]=,Company=>} distinct [Ticket ID])
3 -
4 count({<[Anno Created]='{'$($vAnnoPrecedente)'}',[Mese Created]='{"<=$($vCurrentMonthNum)"}',Category=,Status=,[Assigned to]=,[Ticket Subject]=,Requester=,[Company Area]=,Company=>} distinct [Ticket ID])
5 )
6 /
7 count({<[Anno Created]='{'$($vAnnoPrecedente)'}',[Mese Created]='{"<=$($vCurrentMonthNum)"}',Category=,Status=,[Assigned to]=,[Ticket Subject]=,Requester=,[Company Area]=,Company=>} distinct [Ticket ID])
```

La struttura è simile all'espressione vista per il primo kpi (cambia solamente la variabile `vCurrentMonthNum` in `vAnnoPrecedente`), ma in questo caso, trattandosi di una percentuale, il calcolo che viene effettuato è :

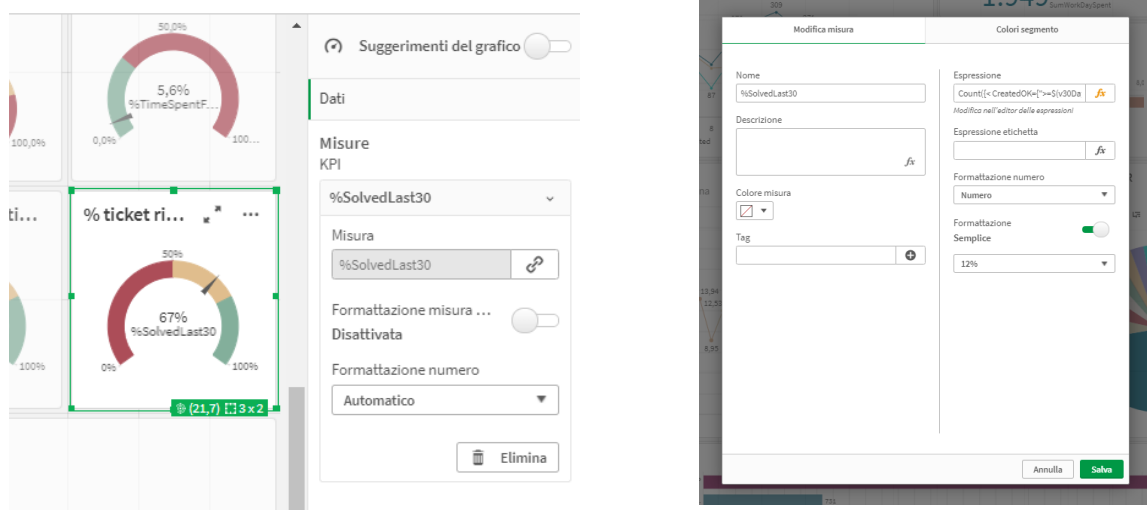
(numero di ticket alla data attuale - numero di ticket alla stessa data dell'anno precedente) / numero di ticket alla stessa data dell'anno precedente.



## - Misuratore %SolvedLast30

Questo elemento grafico ha lo scopo di mostrare la percentuale di tickets risolti negli ultimi 30 giorni, un dato molto importante per realizzare un bilancio mensile sulle effettive performance.

Per questo tipo di dato utilizzare un misuratore è ottimale: a livello visivo, risulta più impattante rispetto a una semplice tabella. Infatti, grazie alla possibilità di assegnare un colore ad ogni range di valori, è facile comprendere a colpo d'occhio se i risultati sono soddisfacenti o meno.



Anche in questo caso si ha a disposizione una finestra di modifica nella quale possiamo stabilire quale misura preesistente assegnare al misuratore, il colore dei range di valori che vogliamo stabilire e la formattazione del numero.

La misura inserita è “%SolvedLast30”, la cui espressione è:

```
Modifica espressione
1 Count({< CreatedOK=">=${v30DaysAgoTrunc}], Status = {'Solved'}, Cancelled="<'1'>}] [Ticket ID])
2 /
3 Count({< CreatedOK=">=${v30DaysAgoTrunc}], Cancelled="<'1'>}] [Ticket ID])
4
5
6
```

Viene calcolato il rapporto tra il totale dei ticket con data entro gli ultimi 30 giorni dalla data attuale che abbiano lo stato settato a “Solved” che non abbiano il campo “Cancelled” uguale a 1, e il totale dei ticket creati ma non cancellati.

## 4.4.2. Secondo foglio : Details



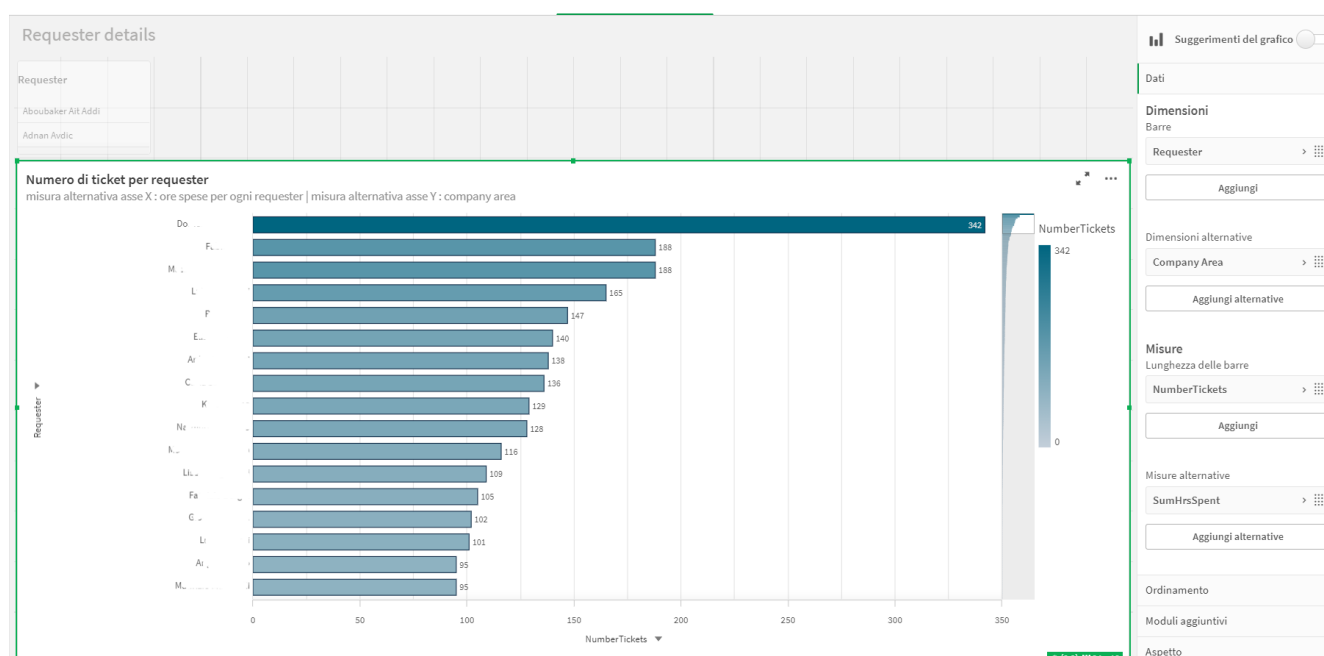
Il secondo foglio è stato creato per fornire dei grafici con dati più dettagliati rispetto al primo. Si può vedere:

- un grafico a barre che fornisce il numero di ticket risolti per ogni persona dell'ufficio e per ogni categoria;
- una tabella pivot che riassume i dati del grafico a barre e che di conseguenza contiene le stesse misure e dimensioni;
- una serie di grafici a torta con relative tabelle riassuntive che riportano rispettivamente il numero di ticket per categoria, il carico di lavoro in percentuale per ogni addetto IT e il numero in percentuale di ticket per materia.

L'utilizzo di grafici a torta è strategico in quanto permettono immediatamente di capire quale dato occupa la maggior parte del tempo o delle risorse sul totale.

Per quanto riguarda invece la preferenza di utilizzo di una tabella pivot rispetto a una semplice tabella, i vantaggi derivano dal fatto che in una tabella pivot le misure e le dimensioni sulle righe e le colonne possono essere interscambiate dall'utente per modificare la visualizzazione dei dati.

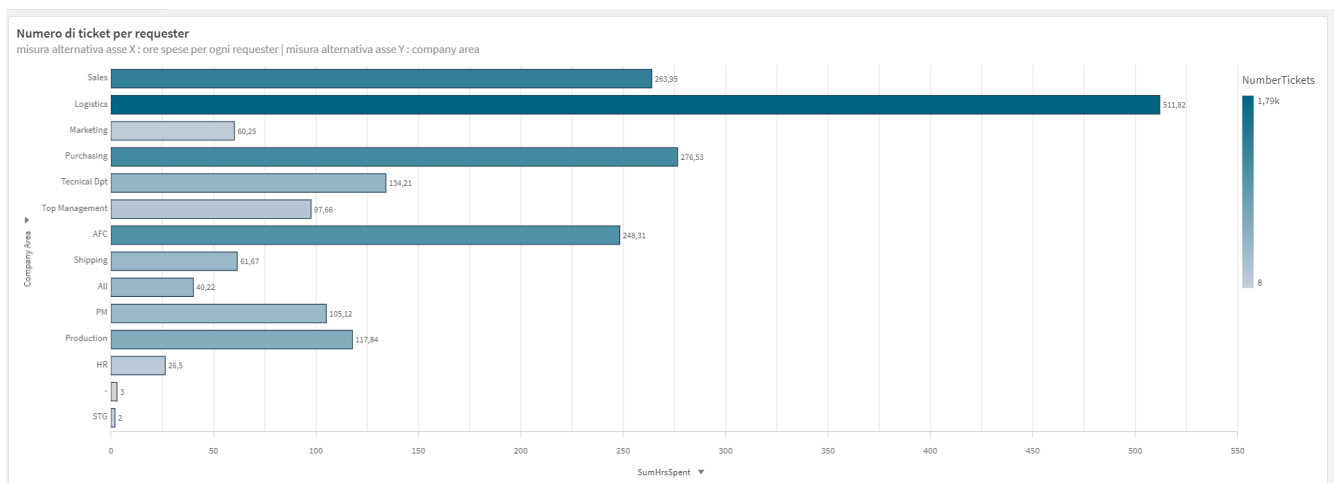
### 4.4.3. Terzo foglio: Requester details



Nel terzo foglio si può notare la presenza di un solo grafico a barre con lo scopo specifico di mostrare il numero di ticket per ogni “requester”, vale a dire gli utenti che li creano sul portale Helpdesk. In questo modo è possibile notare

immediatamente chi crea un numero maggiore di ticket e intervenire nel caso la situazione diventasse insostenibile a livello di ore spese per questa persona.

Questo grafico offre un'ulteriore visualizzazione grazie alla presenza di misure alternative su entrambi gli assi: è possibile consultare anche la somma delle ore spese per ogni area aziendale. Tutte queste misure si possono combinare sui due assi, vale a dire che le possibili visualizzazioni sono varie e a scelta dell'utente.



Struttura del grafico:

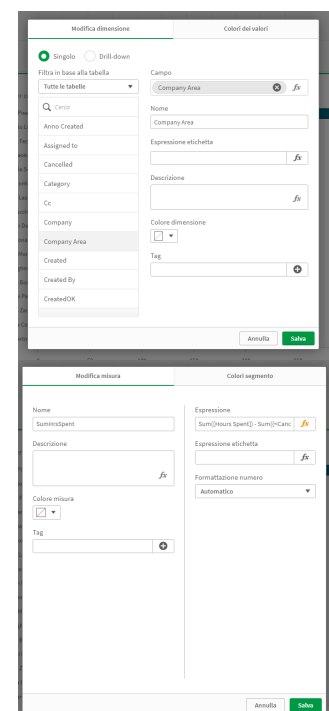
Dimensione principale: *Requester*, derivante dall'omonima colonna del file excel di caricamento dei dat.

Dimensione alternativa: *CompanyArea*, anche questa derivante direttamente dal file excel.

Misura: *NumberTickets* (totale numero di ticket)

Misura alternativa: *SumHrsSpent* (somma delle ore spese per risolvere tickets)

Espressione *NumberTickets*:



```

Modifica espressione
1 Count([Ticket ID]) - Count({<Cancelled={ '1' }>[Ticket ID])
2

```

Espressione *SumHrsSpent*:

```
Modifica espressione
1 Sum([Hours Spent]) - Sum({<Cancelled={ '1' }>}[Hours Spent])
2
3 |
```

## 5. Conclusioni

Nell'ultimo decennio l'utilizzo di strumenti come le business intelligence all'interno delle aziende ha subito una crescita esponenziale, di pari passo alla necessità di analizzare una mole sempre più notevole di dati.

Nel contesto dell'azienda USCO S.p.A. il vantaggio che consegue dall'utilizzo di una business intelligence come Qlik Sense è che ad oggi viene effettuato un monitoraggio costante dei dati, con possibilità di farlo in modo interattivo. Nel caso specifico della mia personale attività svolta, l'ufficio IT ad oggi dispone di grafici e tabelle sempre aggiornati per poter migliorare le proprie performance.

In futuro si può prendere in considerazione di apportare dei miglioramenti, tra cui:

- introduzione di un collegamento diretto tra il portale dell'Helpdesk e Qlik Sense senza passare da un file excel come avviene ora, con lo scopo di creare un canale diretto per il caricamento dei dati;
- introduzione della sola lingua inglese per tutta la dashboard creata (nomi di variabili, campi, misure ecc).

## 6. Bibliografia e Sitografia

- (1) [https://it.wikipedia.org/wiki/Business\\_intelligence](https://it.wikipedia.org/wiki/Business_intelligence)
- (2) pdf “Introduzione alla business Intelligence e al Data Warehousing” tratto dal libro “*Data Warehouse - teoria e pratica della Progettazione*”- Matteo Golfarelli, Stefano Rizzi , Editore: McGraw-Hill (fornito dal professore D. Beneventano)
- (3) <https://it.know-base.net/7580448-power-bi-vs-qlik-sense>
- (4) <https://help.qlik.com/it-IT/sense>
- (5) [https://it.wikipedia.org/wiki/Backus-Naur\\_Form](https://it.wikipedia.org/wiki/Backus-Naur_Form)
- (6) Beneventano D., Bergamaschi S., Guerra F., Vincini M., “*Progetto di basi di dati relazionali. Lezioni ed esercizi.*”, Bologna, Pitagora Editrice, 2007