



**INTESA SANPAOLO
GROUP SERVICES**

DIREZIONE SISTEMI INFORMATIVI

-- Servizio --

-- Ufficio --

AFU - Analisi Funzionale

-- Nome Progetto --

"Modifica Proprietà Documento"

Fare Doppio Clic sul bottone per Modificare le proprietà del Documento.

SSA Acronimo	-- SSA Acronimo --		
Codice Progetto	-- Codice Progetto --		
Classificazione	-- Classificazione --		
Autorizzati	-- Autorizzati --		
Autore	-- Autore --		
Nome file	AFU - Analisi Funzionale.doc		
Versione	-- Versione --	Stato	-- Stato --
Approvato da	-- Approvato da --		
Data creazione	gg/mm/aaaa	Data ultimo aggiornamento	gg/mm/aaaa

Si rammentano le seguenti Note Operative:

1. I documenti classificati "ad uso interno" non possono essere divulgati all'esterno di Intesa - San Paolo Sanpaolo S.p.A.
2. La persona che venisse in possesso di un documento classificato "riservato" e non trovi il suo nominativo nella lista del personale autorizzato all'utilizzo è invitato a consegnare il documento ad una delle persone citate nella lista di cui sopra.

Cronologia

Nella seguente tabella è riportata la cronologia delle modifiche al presente documento

[illegible]

Approvazione Documento

Nella seguente tabella è riportata la lista degli stakeholder con cui il documento è stato condiviso e da cui è stato approvato.

[illegible]

Indice Generale:

CRONOLOGIA	2
APPROVAZIONE DOCUMENTO	3
INDICE GENERALE:	4
INDICE TABELLE:	5
NOTE GENERALI PER LA COMPILAZIONE	6
1 SCOPO E AMBITO	7
2 ARCHITETTURE APPLICATIVE.....	8
2.1 ARCHITETTURE APPLICATIVE PROGETTUALI	8
2.1.1 Architettura ad un solo livello (Diretta)	8
2.1.2 Architettura a 2 livelli con QVD	9
2.1.3 Architettura a 2 livelli con QVF (QlikMart)	10
2.1.4 Architettura a 3 livelli con QVF (Base App e QlikMart)	11
2.1.5 Architettura a 3 livelli mista QVD e QVF	12
2.2 ARCHITETTURE DI PROGETTO DI RIFERIMENTO	13
2.2.1 Architettura ad un solo livello (Diretta)	13
2.2.2 Architettura a 3 livelli mista QVD e QVF	14
3 ANALISI DEI PROCESSI.....	17
3.1 CARTELLA APPLICATIVA DI RIFERIMENTO.....	17
3.2 GESTIONE CARTELLA QVD COMUNI (CROSS PROGETTO)	21
3.3 DEVELOPMENT FRAMEWORK.....	21
3.3.1 Applicazioni Qlik naming convention	21
3.3.2 Back-end.....	23
3.3.3 Front-end	24
3.4 ARCHITETTURE	25
3.4.1 livello "0" - data source	26
3.4.2 Extraction level.....	28
3.4.3 Specifiche di setup e uso	29
3.4.4 Impostazione delle variabili di base	29
3.4.5 Transformation level	39
3.4.6 Specifications of setup and use Specifiche di setup e uso.....	41
3.4.6.1 SetUp Base Variables.....	42
3.4.7 Livello presentation	43
3.5 ARCHITETTURA AD UN SOLO LIVELLO (DIRETTA)	43
4 GESTIONE DELL'INFRASTRUTTURA	45
4.1 GESTIONE EXTENSION	45
4.2 CREAZIONE CONNESSIONI ODBC	47
4.3 GESTIONE CARTELLE GENERICHE PER UTENTE	47

Indice Tabelle:

Tabella 1 - Elenco Funzionalità	Errore. Il segnalibro non è definito.
Tabella 2 - Elenco Flussi Applicazione.....	Errore. Il segnalibro non è definito.
Tabella 3 - <Funzione 1> - Elenco Flussi Operativi	Errore. Il segnalibro non è definito.
Tabella 4 - <Funzione 1> - Dettaglio < Flusso 1>.....	Errore. Il segnalibro non è definito.
Tabella 5 - <Funzione 1> - Elenco Pagine / Schermate	Errore. Il segnalibro non è definito.
Tabella 6 - <Funzione 1> - Dettaglio < Pagina/Schermata 1> ...	Errore. Il segnalibro non è definito.
Tabella 7 - <Funzione 1> - Elenco Report.....	Errore. Il segnalibro non è definito.
Tabella 8 - <Funzione 1> - Dettaglio < Report 1>	Errore. Il segnalibro non è definito.
Tabella 9 - <Funzione 1> - Elenco Business Service utilizzabili	Errore. Il segnalibro non è definito.
Tabella 10: Rappresentazione di regole di business da importare sulla componente architetturale Rule Engine	Errore. Il segnalibro non è definito.
Tabella 10 - Elenco Entità Informative	Errore. Il segnalibro non è definito.
Tabella 11 - Dettaglio <Entità 1>.....	Errore. Il segnalibro non è definito.
Tabella 12 - Oggetti di Sizing	Errore. Il segnalibro non è definito.

Note generali per la compilazione

- 1) *Vanno compilati i capitoli effettivamente necessari alla descrizione dell'intervento. Laddove vi siano capitoli nei quali non è possibile descrivere le attività pertinenti l'iniziativa, è comunque opportuno lasciare traccia nel capitolo stesso, inserendo una breve motivazione (per esempio: Non Applicabile in quanto non sviluppata dal progetto in esame).*
- 2) *Le frasi in blu corsivo presenti all'inizio di ogni capitolo descrivono con quali argomenti va compilato. Questo per rendere confrontabili ed omogenei tra loro i diversi documenti ed offrire per il redattore una guida ed uno spunto. Ovviamente non fanno parte del documento finale e debbono pertanto essere rimossi*

1 Scopo e Ambito

Intesa - San Paolo ha definito e adottato una serie di linee guida per i progetti di Business Intelligence basati sulla piattaforma Qlik. La conformità a queste linee guida servirà a:

- Fornire una serie di principi e regole generali per gli sviluppatori.
- Facilitare le operazioni di manutenzione delle applicazioni.
- Migliorare le capacità della piattaforma tecnologica.

Queste linee guida non sono obbligatorie; ogni sviluppatore può adottarle in modo flessibile durante lo sviluppo del progetto Qlik.

Tuttavia, la ragione per non conformarsi alle linee guida deve essere valutata e giustificata.

Queste linee guida saranno valide per tutti i nuovi progetti Qlik sviluppati in Intesa - San Paolo dal rilascio di queste linee guida.

Per tutte le applicazioni Qlik già esistenti, in caso di necessità di ripristino o di manutenzione delle stesse, dovrebbe essere presa in considerazione l'adozione di alcuni o di tutti i principi definiti nelle linee guida, alla luce dell'utilità e dei vantaggi che questo porterà alla qualità dell'applicazione.

Al fine di facilitare l'adozione di queste linee guida, Intesa - San Paolo fornisce un 'kit' di applicazioni standard parametriche che:

- Aiuteranno gli sviluppatori.
- Ridurranno il tempo necessario per lo sviluppo del progetto durante la sua implementazione.
- Facilitaranno l'attività di manutenzione dei progetti.

Questo documento è costituito dalla documentazione che definisce lo standard metodologico per lo sviluppo di applicazioni BI sulla piattaforma Qlik e fornisce una descrizione dettagliata di come utilizzare le applicazioni parametriche introdotte.

Gli standard metodologici per lo sviluppo sono:

- Definizione dell'architettura di riferimento:
 - livelli, struttura delle cartelle dei progetti e le diverse componenti dell'applicazione.
- Definizione dello standard per lo sviluppo:
 - strutturazione e organizzazione dello script, linee guida per l'utilizzo delle variabili in Presentation, gestione dei metadati, convenzioni di denominazione per variabili e parametri e procedure di controllo qualità.
- Linee guida per la progettazione del layout: principi di visual design e delle best practice

Questo documento si riferisce alla definizione dell'architettura, alla descrizione dettagliata dei diversi livelli, dell'organizzazione di directory dei progetti e dei componenti dell'applicazione.

2 Architetture applicative

Le seguenti pagine contengono, in primo luogo, diverse architetture di dati che possono essere implementate come parte delle soluzioni Qlik.

Queste pagine non rappresentano tutte le possibili architetture. Infatti, rappresentano solo una piccola parte di quelle possibili.

Infatti rappresentano i concetti più utilizzati (ad eccezione dell'architettura ad un solo livello) per promuovere la scalabilità, il riutilizzo e la coerenza nello sviluppo di applicazioni Qlik.

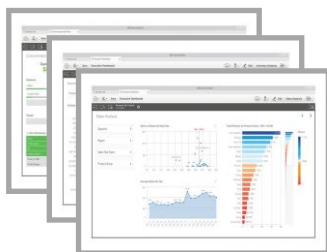
Di seguito viene indicata l'architettura applicativa di riferimento che Intesa - San Paolo adotta come standard per lo sviluppo delle soluzioni Qlik.

Viene, infine, delineata l'architettura applicativa a cui Intesa - San Paolo intende evolvere.

2.1 Architetture applicative progettuali

2.1.1 Architettura ad un solo livello (Diretta)

Presentation Layer



Utilizzato per l'accesso diretto alle fonti alimentanti.
Essendoci diverse applicazioni che accedono agli stessi dati esiste una ridondanza di codice e un overlap di scarichi di dati.

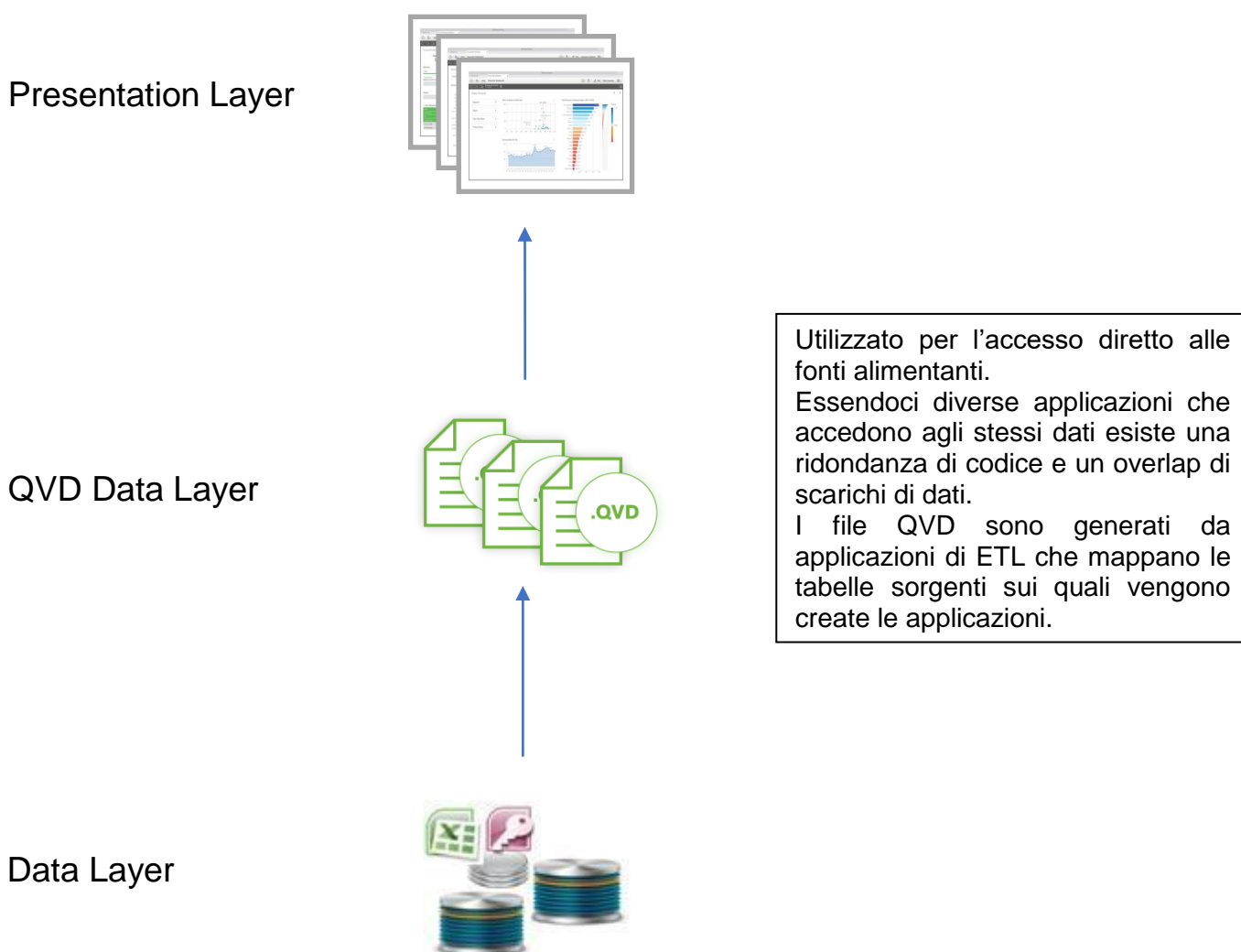
Data Layer



Questa architettura coinvolge solo query dirette verso i database di origine. Questa è la soluzione più costosa e complessa in termini di manutenzione perché tutte le applicazioni a livello di presentation hanno connessioni ai database e contengono le proprie query SQL e gli script Qlik per le trasformazioni e l'aggregazione di tutti i dati necessari.

Questo approccio è però adatto per progetti "Big Data" con funzionalità «Accesso diretto» o con generazione di «App on Demand».

2.1.2 Architettura a 2 livelli con QVD



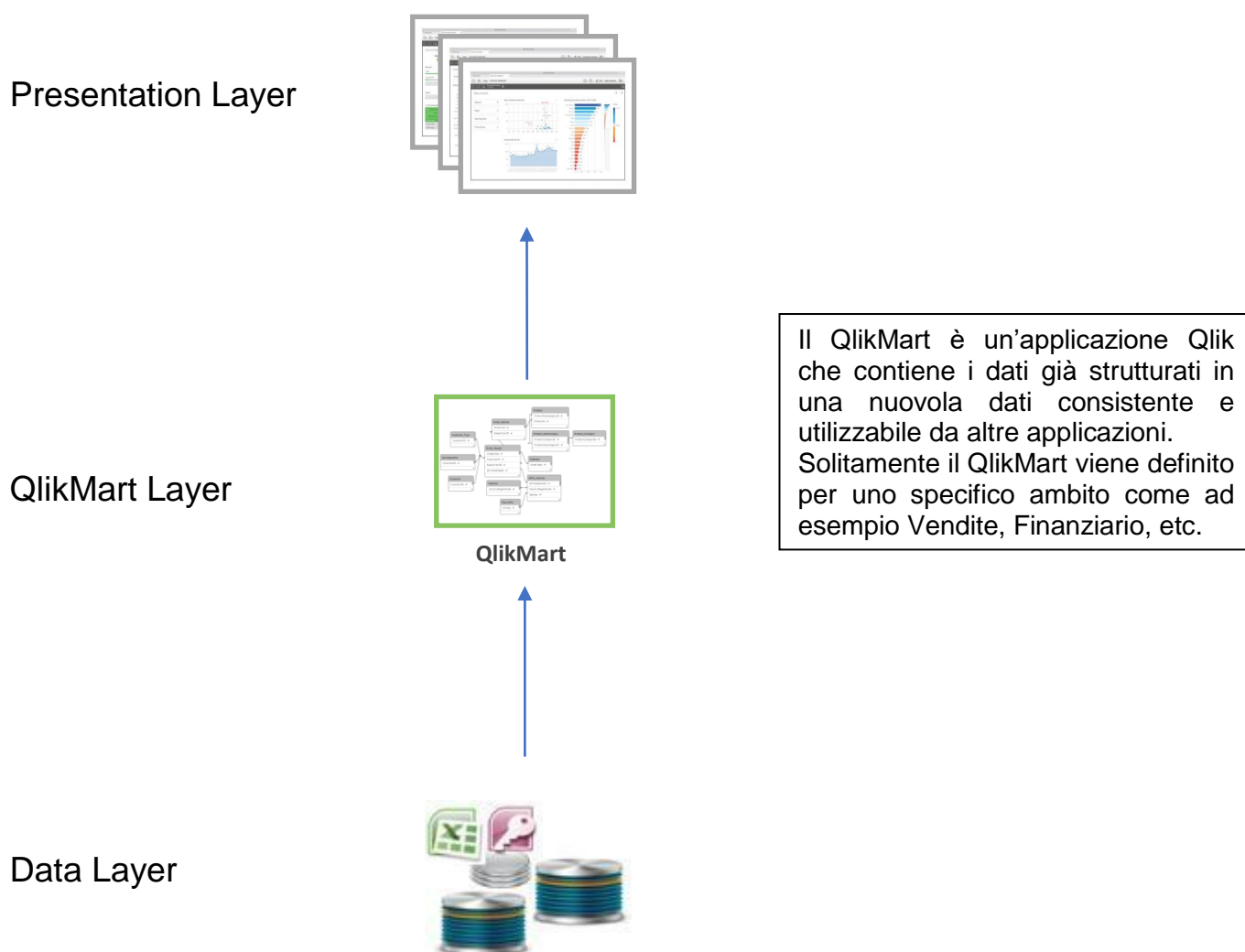
Questa architettura include l'uso di file QVD per un secondo livello di dati. A livello di presentation le applicazioni genereranno modelli di dati da diversi file QVD, ma non avranno bisogno di estrarre direttamente dai database di origine. Ciò aiuta a promuovere il riutilizzo e la coerenza tra le diverse app.

Questo tipo di architettura consente di separare lo sviluppo dell'applicazione di presentazione dai dati dei database di origine.

Uno svantaggio di questa architettura è la definizione del cloud associativo nelle applicazioni di presentazione: questo è fondamentale per la manutenzione applicativa, in particolare per il modello di dati ad alta complessità.

Questa architettura potrebbe essere utilizzata per lo sviluppo di progetti con requisiti di ricondizionamento intraday (quasi in tempo reale). Si consiglia quando le origini dati sono normalizzate e ottimizzate per il layer di presentation.

2.1.3 Architettura a 2 livelli con QVF (QlikMart)



Questa architettura include l'uso di file di applicazioni "QlikMart" come secondo livello di dati. Questi QlikMarts sono applicazioni Qlik senza il layer di presentation.

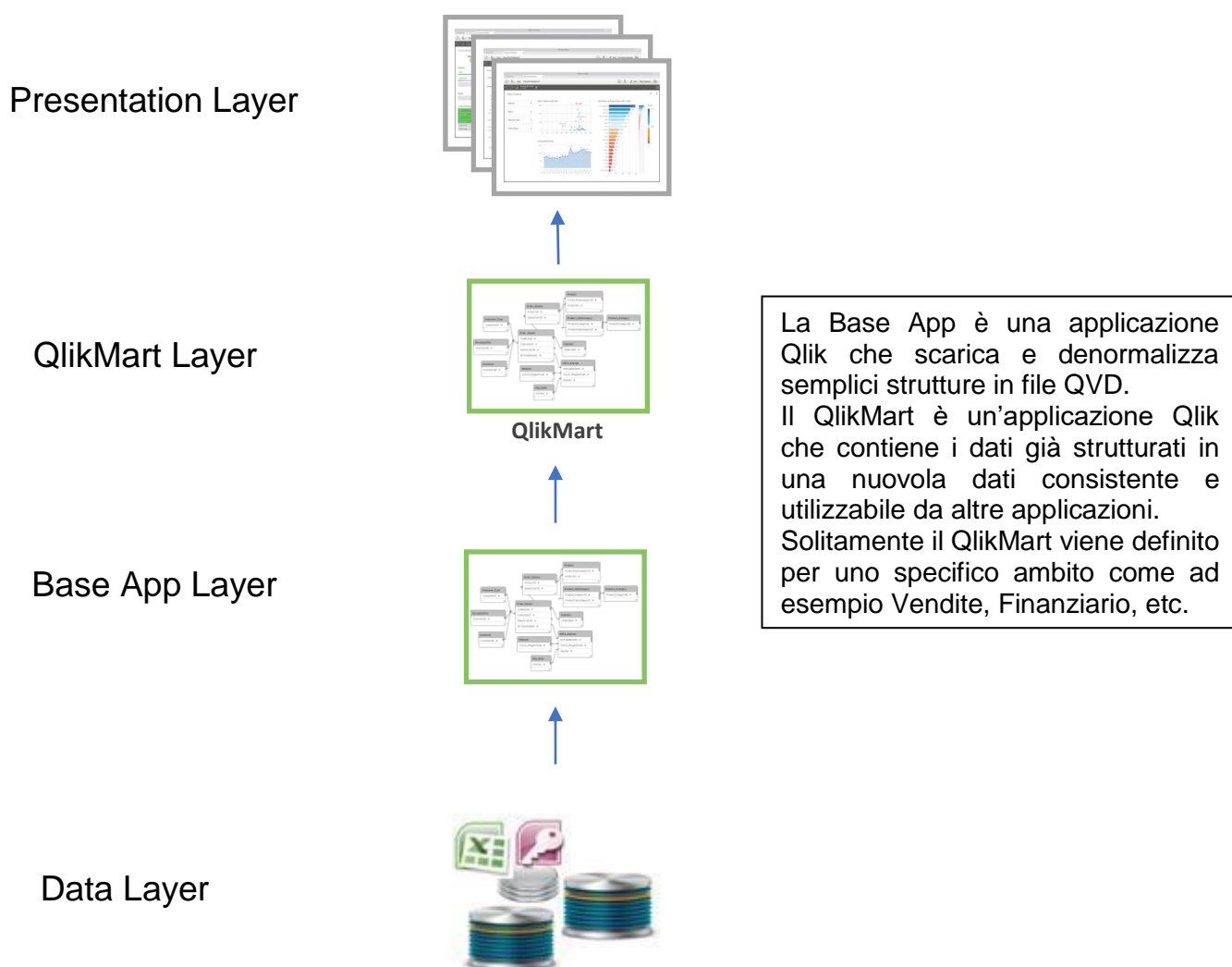
Funzionano come modelli di dati che possono essere caricati in un'applicazione di presentazione con una singola riga di codice (chiamata carico binario).

Il principale handicap di questa architettura è rappresentato dalla "rigidità" della funzionalità "Carico binario": questa funzionalità carica l'intero modello di dati; se è necessario integrare il modello con altre tabelle, potrebbe essere difficile e non flessibile. Questa architettura non gestisce il

caricamento incrementale da origini dati e la conservazione della cronologia o riutilizzo in molte applicazioni.

Questa architettura può essere utilizzata per lo sviluppo di progetti che comportano un volume elevato di dati con frequenti aggiornamenti. E' consigliata quando più di un'applicazione di presentazione è basata sullo stesso modello di dati.

2.1.4 Architettura a 3 livelli con QVF (Base App e QlikMart)

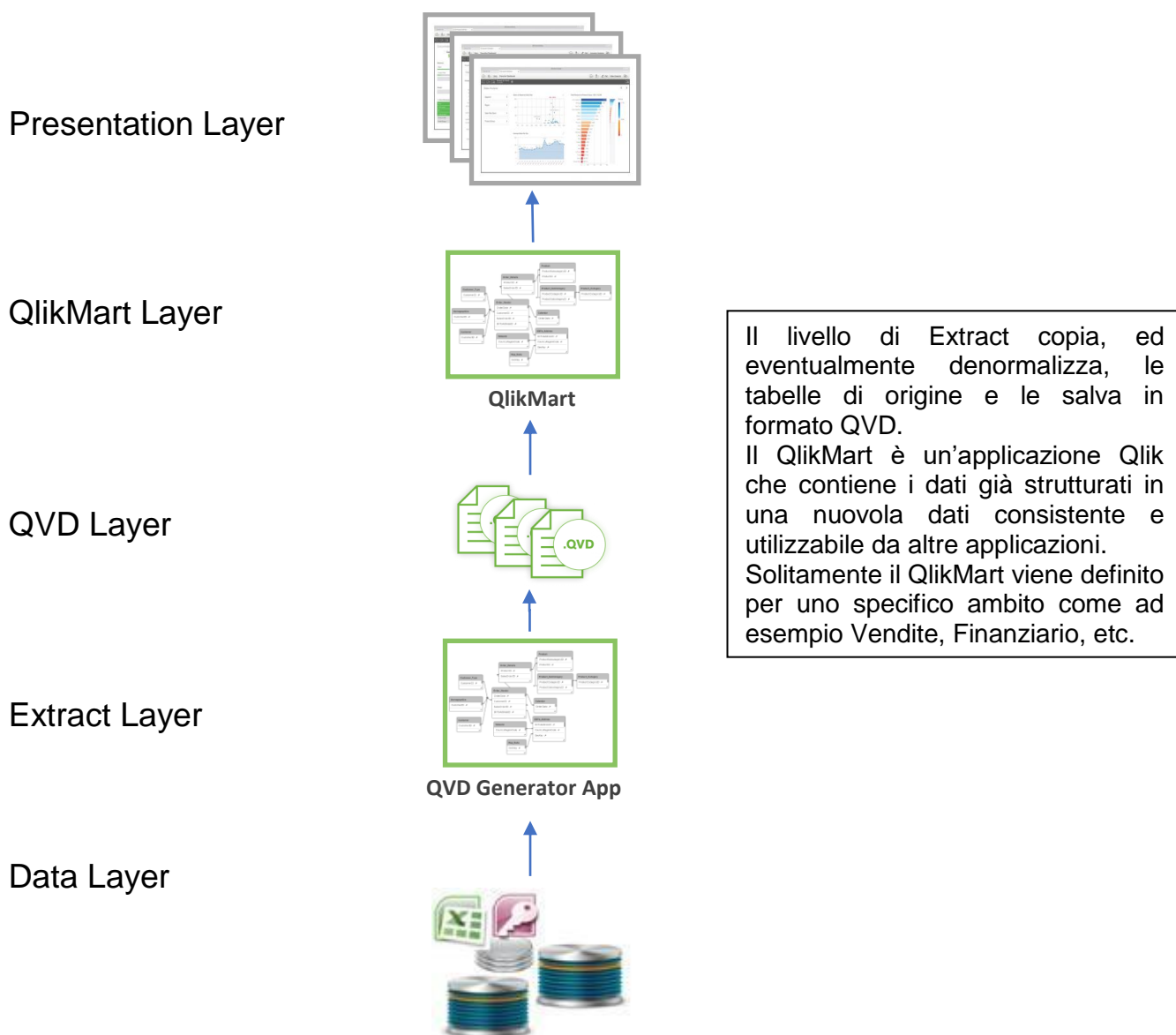


Questa architettura include l'uso di file QVF come secondo e terzo livello di dati. Ciò consente al primo strato QVF di contenere modelli di dati relativamente semplici con un livello più normalizzato del database di origine, con poche attività di aggregazione o trasformazione.

Nel livello QlikMart avvengono le trasformazioni e le aggregazioni. Questa architettura avrà lo scripting più complesso a livello di QlikMart poiché un intero modello di dati viene caricato dal livello precedente prima di attivare le trasformazioni sui dati.

Questa architettura può essere utilizzata per lo sviluppo di progetti che comportano un volume elevato di dati con frequenti aggiornamenti. Si consiglia quando più di un'applicazione di presentazione è basata sullo stesso modello di dati. Rispetto al singolo modello QlikMart, in alcuni casi, il primo livello di QVF consente di gestire l'accesso alle fonti di dati utilizzando la segregazione.

2.1.5 Architettura a 3 livelli mista QVD e QVF



Questa architettura include l'utilizzo dei file QVD come secondo livello dati, basato sul quale viene generato il livello QlikMart. Ciò consente alle applicazioni del livello di presentation di essere dei «Binary Load» del livello precedente. Rispetto al precedente modello QlikMart, il primo strato di

QVD consente la gestione dei caricamenti incrementali e la conservazione della cronologia delle informazioni. Rimane la rigidità della funzionalità «Binary Load» nelle applicazioni di presentazione; Questa architettura allunga la catena del processo limitando il vantaggio del QlikMart in termini di tempo di ricarica. Resta però valida, ad esempio, per lo sviluppo di progetti con più di una presentazione con lo stesso modello di dati.

2.2 Architetture di progetto di riferimento

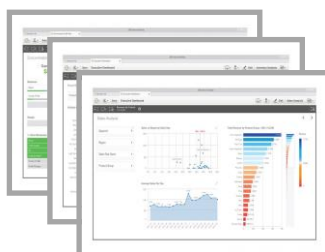
L'architettura applicativa di riferimento per i progetti che Intesa adotta per lo sviluppo standard dei progetti Qlik sono due:

- Architettura ad un solo livello (Diretta)
- Architettura a due livelli (QVD e QVF)

Vediamo in dettaglio entrambe le architetture.

2.2.1 Architettura ad un solo livello (Diretta)

Presentation Layer



Data Layer



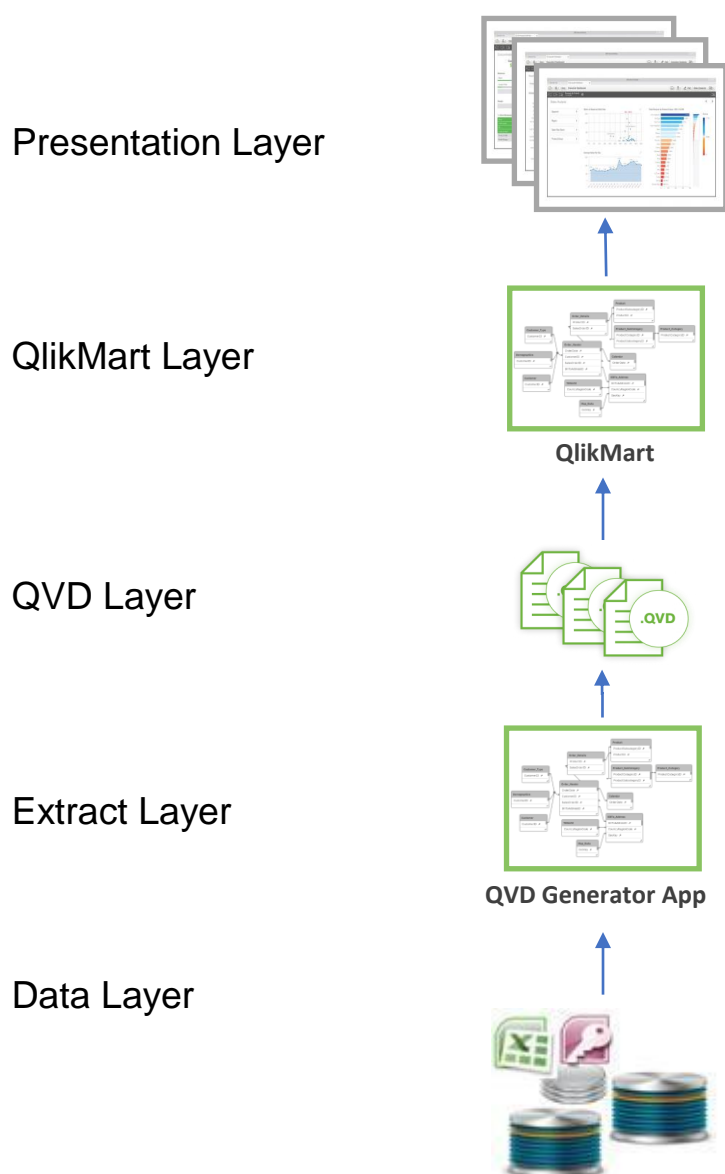
Questa struttura risponde alle esigenze di:

- Accesso al repository Big Data
- Flessibilità per l'analisi degli utenti

- BI self-service
- Analisi ad hoc e mashup di dati

A partire dalle origini dati, il livello Presentazione legge direttamente i dati dalla sorgente nell'applicazione Qlik. L'origine dati può essere uno o più database o file QVD o una combinazione di entrambi.

2.2.2 Architettura a 3 livelli mista QVD e QVF



A partire dalle origini dati il livello **Estrazione** genera un repository di file QVD che sono corrispondenza one-to-one delle tabelle DB di origine.

Il livello di **QlikMart** opera con la normalizzazione e l'aggregazione dei dati per definire il modello di dati associativo (cloud associativo), questo consente il riutilizzo (a volte parziale) in

applicazioni di presentazione ottimizzando le prestazioni di reload.

Il livello di **Presentation** include solo lo sviluppo della visualizzazione dei dati e le funzionalità di data discovery.

Questa architettura offre questi vantaggi:

- Livello di **Estrazione**:
 - Permettere di separare lo sviluppo del progetto (modello di dati e presentazione) dalle fonti dati
 - Consentire la gestione del caricamento incrementale e il congelamento delle informazioni
 -
- Livello **QlikMart**:
 - le trasformazioni e le aggregazioni vengono definite a questo livello. Tutta la logica e le regole necessarie sono presenti in un unico punto.
 - Riutilizzo dello stesso modello per diverse applicazioni
 - Gestione di un volume elevato di dati con reload frequenti
- Livello di **Presentation**:
 - Applicazioni di front-end con il solo sviluppo della visualizzazione se ci si basa sull'architettura a 3 livelli
 - Applicazioni di front-end con trasformazione e aggregazione di dati e sviluppo della visualizzazione se ci si basa su un'architettura a livello singolo
 - Consente di separare lo sviluppo delle applicazioni di Presentation dal back end che possono essere gestiti da sviluppatori diversi.

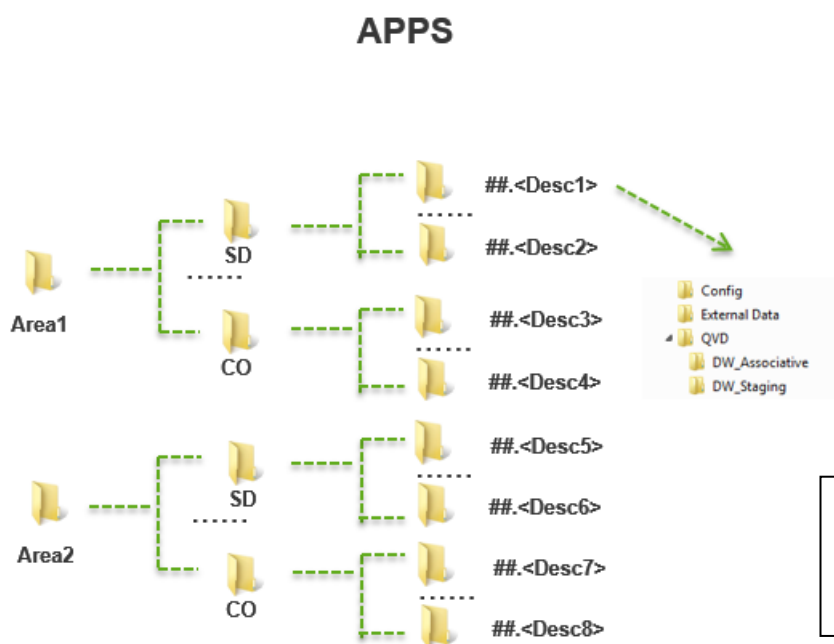
3 Analisi dei Processi

Le pagine seguenti descrivono come implementare l'architettura di riferimento definita precedentemente in termini di:

- Scopo dei componenti dell'applicazione
- Struttura della cartella di riferimento e descrizione dei contenuti
- Naming convention di cartelle e file
- Specifiche e utilizzo development framework per i progetti

3.1 Cartella applicativa di riferimento

La struttura della cartella di riferimento è la seguente:



La struttura della directory delle applicazioni si riferisce agli sviluppi dei documenti di Qlik ed è strutturata per area funzionale. La convenzione di naming per questo livello di directory è un acronimo a 3 caratteri.

Esempi di aree funzionali sono:

Area Funzionale (3 caratteri)	Descrizione
COP	Customer operations
MKT	Marketing
SAL3	Sales

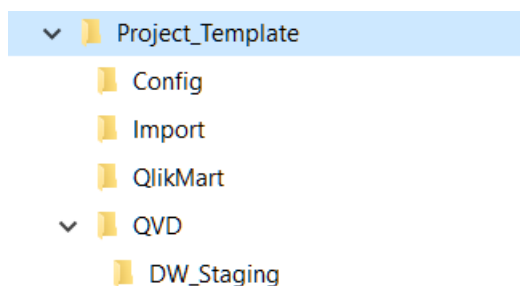
All'interno di ogni cartella definite dall'area funzionale sono presenti le cartelle di progetto.
La naming convention delle cartelle di progetto è: **##.[Breve descrizione del progetto]** dove:

= Numero incrementale di 2 cifre

[Breve descrizione del progetto] = Descrizione del progetto in massimo 15 caratteri

Esempio: **01.MyTestPage**

Ogni cartella di progetto contiene la seguente struttura di cartelle:



Le diverse cartelle conterranno:

- **QVD:** File QVD generati dai processi di ETL
 - **DW_Staging:** cartella contenente i file QVD generate dalle applicazioni di ETL01; questa cartella potrebbe contenere delle sottocartelle ad esempio per le Anagrafiche o degli anni per dati segmentati
- **Config:** cartella che contiene i file di settings ETL01_Setting.xls ed eventuali altri file di configurazione
 - **ETL Audit Log:** contiene file QVD generate dal processo di ETL con dati di monitoring delle esecuzioni;
 - **Logos:** cartella utilizzata come backup contenente loghi, immagini, icone, etc. utilizzati nelle app
- **Import:** cartella contenente file di appoggio provenienti da altri sistemi o manuali
- **Qlik Mart:** cartella contenente i file QlikMart. Queste sono applicazioni che contengono una nuova dati completa e che possono essere utilizzate come fonte dati per i documenti di presentation

Ogni progetto può avere diverse fonti:

- Database sorgenti
- Database locali (QlikMart e QVD)
- Flat file

Nel caso in cui il progetto richieda dati da un database locale il processo di ETL01 salverà i file nella cartella DW_Staging con la seguente naming convention:

- In caso di reload incrementale o completo:
 - o [sorgente]_[E]_[nome tabella originale].QVD
- In caso di storicizzazione: [sorgente]_[E]_[nome tabella originale]_[tempo (a seconda della modalità storica scelta: YYYY, YYYYMM, YYYYMMDD)].QVD

La naming convention per l'applicazione è la seguente:

[area funzionale]_[breve descrizione del progetto]_[nome del file].qv

Dove:

- **Area funzionale:** area funzionale (3 digits)
- **Breve descrizione del progetto:** una breve descrizione (max 15 digits)
- **Nome file:** ETL01 (Mandatory)

Esempio: **COP_MyTestPage_ETL01.qvf;**

Le linee guida per le estrazioni sono:

- Corrispondenza uno a uno delle tabelle sorgenti – QVD
- Estrazione consigliata in incrementale e/o segmentato per volumi importanti
- Estrazione in * (tutti i campi della tabella)

I QlikMart devono essere salvati nella cartella QlikMart con la seguente naming convention:

[area funzionale]_[breve descrizione del progetto]_[file name].qv

Dove:

- **Area funzionale:** area funzionale (3 digits)
- **Breve descrizione del progetto:** una breve descrizione (max 15 digits)
- **Nome file:** MART (Mandatory)

Esempio: **COP_MyTestPage_MART.qvf;**

Il documento di presentazione dovrà caricare:

- QVF dalla cartella QlikMart dalla area di *progetto*

La naming convention dell'applicazione sarà:

[area funzionale]_[breve descrizione del progetto]_[file name].qvf

Dove

- **Area funzionale:** area funzionale (3 digits)
- **Breve descrizione del progetto:** una breve descrizione (max 15 digits)
- **Nome file:** nome file (max 15 digits)
 - o **Il nome del file sarà modificato da Qlik Sense quando l'applicazione sarà caricata nel server.** In ogni caso l'adozione della naming convention è importante per mantenere una corretta overview delle app nella management console e negli streams

Esempio: **COP_MyTestPage_Dashboard.qvf;**

Se viene utilizzata una architettura single tier (diretta) il file qvf conterrà sia la data transformation che la parte di visualizzazione e la naming convention dovrà essere:

[area funzionale]_[breve descrizione del progetto]_[file name].qvf

Dove:

- **Area funzionale:** area funzionale (3 digits)
- **Breve descrizione del progetto:** una breve descrizione (max 15 digits)
- **Nome file:** DIRECT (Mandatory)

Example: **COP_MyTestPage_DIRECT.qvf;**

L'applicazione di presentation caricherà:

- Dati direttamente dalle fonti dati
- File flat dalla cartella di Import

La naming convention dell'applicazione sarà:

[area funzionale]_[breve descrizione del progetto]_[file name].qvf

Dove:

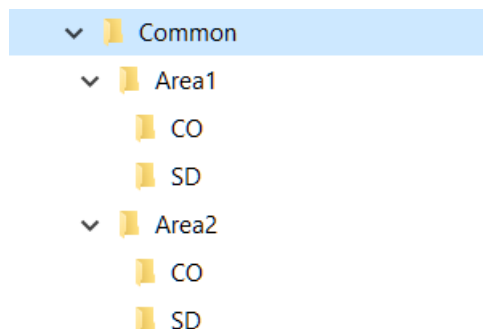
- **Area funzionale:** area funzionale (3 digits)
- **Breve descrizione del progetto:** una breve descrizione (max 15 digits)
- **Nome file:** nome file (max 15 digits)
 - o **Il nome del file sarà modificato da Qlik Sense quando l'applicazione sarà caricata nel server.** In ogni caso l'adozione della naming convention è importante per mantenere una corretta overview delle app nella management console e negli streams

Esempio: **COP_MyTestPage_Dashboard.qvf**;

3.2 Gestione cartella QVD comuni (cross progetto)

Per poter gestire dati che sono utilizzati da diversi progetti contemporaneamente si definisce una cartella chiamata Common che, suddivisa per Area, conterrà al suo interno i file QVD condivisi.

Le gestione degli script di creazione di questi QVD sarà gestita centralmente da DSI che ne garantirà l'aggiornamento secondo quanto pianificato in fase di implementazione dello script di generazione.



3.3 Development framework

Per facilitare l'adozione dell'architettura applicativa di progetto di cui sopra per lo sviluppo di applicazioni Qlik e massimizzare il vantaggio delle implementazioni, Intesa adotta un framework di sviluppo che contiene applicazioni Qlik parametriche e una struttura di directory conforme all'architettura di riferimento

Le pagine seguenti descrivono la funzionalità e l'uso di queste applicazioni.

Vengono identificate due macro aree, a seconda degli scopi dei componenti che ne fanno parte:

- Back - End (Extraction Layer, Transformation Layer)
- Front - End (Presentation Layer)

3.3.1 Applicazioni Qlik naming convention

I file QVF devono seguire la seguente naming convention:

[area funzionale]_[breve descrizione del progetto]_[file name].qvf

Dove

- **Area funzionale:** area funzionale (3 digits)
- **Breve descrizione del progetto:** una breve descrizione (max 15 digits)
- **Nome file:** ETL01 per i processi di ETL e nome file per le presentation (max 15 digits)

- **Il nome del file sarà modificato da Qlik Sense quando l'applicazione sarà caricata nel server.** In ogni caso l'adozione della naming convention è importante per mantenere una corretta overview delle app nella management console e negli streams

Esempi:

COP_MyTestPage_ETL01.qvf,

COP_MyTestPage_MART.qvf

COP_MyTestPage_DIRECT.qvf

COP_MyTestPage_DashBoard.qvf;

3.3.2 Back-end

L'area macro Back-End è composta da tre applicazioni parametriche utilizzate per creare il cloud associativo dalle origini dati:

- ETL01.qvf
- QlikMart.qvf

La prima applicazione, ETL01.qvf, gestisce l'estrazione delle tabelle di origine e la loro archiviazione in formato file binario QVD. I QVD di output dell'applicazione ETL01 devono essere denominati con lo stesso nome della tabella di origine, con la seguente convenzione:

- In caso di reload incrementale o completo:
 - *[sorgente]_[E]_[nome tabella originale].QVD*
- In caso di storicizzazione: *[sorgente]_[E]_[nome tabella originale]_[tempo (a seconda della modalità storica scelta: YYYY, YYYYMM, YYYYMMDD)].QVD*

La seconda applicazione, QlikMart.qvf, utilizza i file QVD, l'output dell'applicazione precedente, per creare il cloud associativo che verrà inserito nelle applicazioni front-end tramite il carico in Binary.

Queste applicazioni, che si trovano nella cartella di riferimento rispettano le seguenti linee guida:

- rendere parametrico il maggior numero possibile di funzionalità di scripting
- utilizzare le variabili per rendere lo script dinamico e riutilizzabile
- estrarre dallo script il maggior numero possibile di funzioni: elenco di tabelle da estrarre, dizionario dei dati dell'applicazione, condizioni per l'estrazione delle tabelle
- gestione dei metadati
- congelamento del controllo dei processi dei file
- script di supporto con commenti appropriati per consentire una facile lettura

Le impostazioni e l'utilizzo specifici relativi a ciascuna applicazione sono esposti in dettaglio di seguito.

3.3.3 Front-end

L'area front-end comprende le applicazioni "Presentation", orientate all'analisi e alla navigazione dei dati.

A seconda dell'architettura utilizzata per il singolo progetto, possiamo avere due scenari:

1. L'applicazione di presentazione legge direttamente il Database di origine. Viene utilizzato con le applicazioni App-On Demand o Direct Discovery
2. L'applicazione di presentazione legge il file QVF di QlikMart. Solo un file può essere letto in questo modo

Quando viene utilizzato l'approccio 1), l'applicazione può avere uno script complesso all'interno per caricare i dati.

Quando viene utilizzato l'approccio 2), viene eseguito un carico binario da un'applicazione QlikMart da cui viene caricato tutto il cloud associativo.

Le applicazioni di "presentazione" si concentrano su metodi e tecniche di visualizzazione dei dati, nonché sulla massimizzazione delle prestazioni di navigazione.

Le prestazioni sono influenzate sia dalla struttura del cloud associativo sia dal numero e dalla struttura delle espressioni utilizzate negli oggetti Qlik che compongono le applicazioni "Presentazione".

Pertanto, le applicazioni front-end dovrebbero seguire le linee guida seguenti:

- Semplicità di script composta principalmente da istruzioni come Carica *, per importare il formato file QVD creato nell'area Back-End
- Uso di variabili per espressioni di calcolo, utilizzate in più oggetti
- Visualizzare oggetti sul foglio di visualizzazione in modo efficace
- Basso numero di oggetti sullo stesso foglio

Nel capitolo "Ergonomia e progettazione visiva" saranno presentati i principi di ergonomia e design visivo, questi principi sono utili per l'implementazione di applicazioni "Presentazione" che migliorano le capacità grafiche della piattaforma Qlik. Un esempio di applicazione basata su questi principi è allegato, per supportare sviluppatori e progettisti.

3.4 Architetture

Questo capitolo descrive la struttura delle architetture, in relazione al flusso di informazioni.

A seconda del tipo di architettura selezionata Diretta o a 3 livelli è necessario adottare un approccio diverso.

Architettura a 3 livelli

Identifichiamo due processi ETL sequenziali basati su applicazioni QVF (ETL01.qvf, QlikMart.qvf) che memorizzano i dati nel file binario nativo di Qlik QVD, uno o più file qvf di QlikMart e una o più applicazioni per analizzare e navigare i dati importati dai file QVD creati nei processi ETL e / o da QlikMart per visualizzare le informazioni in modo chiaro ed efficace.

Questa architettura si basa quindi sul concetto di "triplo strato misto di QVD e QVF" ed è considerata una delle soluzioni adeguate alle esigenze di Intesa.

Il primo livello di ETL legge le origini dati e crea una prima serie di file in formato QVD, gestendo il carico incrementale, quando necessario. Il secondo livello utilizza un'applicazione Mart che carica i file QVD creati nel primo livello di ETL, normalizza e aggrega i dati per generare un file QlikMart.qvf che contiene la nuvola associativa. Una o più applicazioni di "Presentation" costituiscono il quarto e ultimo livello di questa architettura.

I livelli architetturali sono definiti in maniera sintetica come:

- Livello 0 - Sorgente
- Livello 1 – Estrazione
- Livello 2 - Trasformazione
- Livello 3 - Presentazione

Architettura ad un solo livello (Diretta)

Quando viene utilizzata questa architettura, esiste una sola applicazione che contiene sia ETL che elementi di presentazione.

Questo tipo di approccio è per progetti in cui è necessario leggere enormi quantità di dati o creare app "on demand".

All'interno dell'app sono presenti l'intero processo di lettura dei dati dal database di origine, la trasformazione dei dati, la creazione del cloud associativo e gli elementi di presentazione. Per un confronto con l'Architettura a 3 livelli tutti i livelli da 0 a 3 sono all'interno della stessa app.

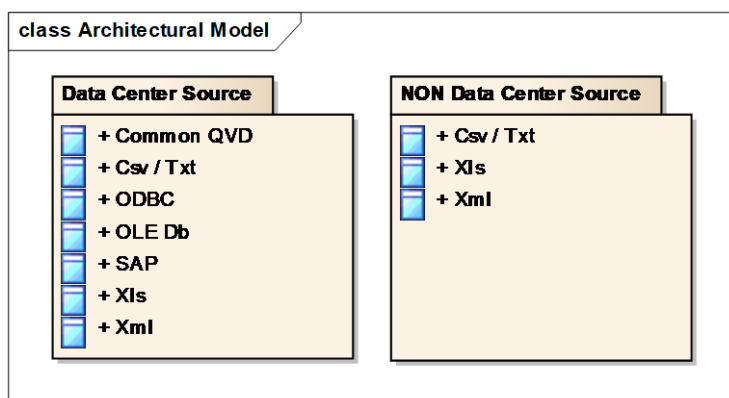
Il database di origine può essere anche file QVD.

3.4.1 livello "0" - data source

Il "Livello 0" è costituito da fonti esterne contenenti i dati che alimentano la piattaforma Qlik per ciascun progetto

- sources ODBC / OLE Db
- QVD files
- SAP Connector
- sources Xml
- sources Csv, Txt
- sources Xls

Le fonti sono divise tra quelle memorizzate nel Data Center di Intesa e quelle archiviate esternamente al Data Center, come mostrato nello schema seguente:



Le fonti nel Data Center sono accessibili tramite indirizzo IP (ODBC, OLE Db, SAP) con USER-ID e Password.

Le fonti situate al di fuori del Data Center sono disponibili solo tramite collegamento al Filesystem.

ODBC / OLE Db

Questa è la fonte più comune in cui vengono raccolte le informazioni. Si compone di database relazionali esterni come Oracle, SQL Server, MySql, Teradata, DB2, ecc.

Quando possibile, la connessione OLE DB è migliore di ODBC perché ha prestazioni migliori.

In alcuni casi (ad esempio per elevate complessità delle tabelle originali) può essere valutata attentamente la possibilità di creare una vista logica per separare i database.

QVD

QVD è una fonte gestita dal team centrale come descritto sopra

Gli scopi di questa fonte sono:

- generare un set di file QVD disponibili per gli sviluppatori senza accesso diretto al database di origine aziendale
- gestire centralmente le informazioni richieste trasversalmente da diversi progetti Qlik
- minimizzare l'accesso al database di origine aziendale
- minimizzare la ridondanza delle informazioni nel livello di estrazione

Sources Xml

Le fonti Xml richiedono particolare attenzione a causa della struttura di queste fonti. Pertanto, è necessario esaminare attentamente ciascun campo e identificare esattamente le informazioni da importare.

Sources Csv / Txt / Microsoft Excel

Le fonti consistevano nel file Csv o Txt caricati in Qlik molto semplicemente e non mostrano particolari punti di attenzione

3.4.2 Extraction level

Le attività sono eseguite dall'applicazione ETL01.qvf allegata al presente documento. Di solito, c'è solo un ETL01.qvf per ogni progetto. A volte è possibile configurare più di una applicazione ETL01.qvf per un progetto specifico se vi è una diversa tempistica di ricarica dei dati per una tabella dei fatti diversa.

L'utilizzo parametrico di questa applicazione è definito dalle impostazioni contenute nel file di Microsoft Excel "ETL01_Settings.xls".

In questi file vengono definiti i parametri e i metodi di estrazione dei dati dalle origini dati e i nomi dei file corrispondenti in formato QVD, in cui sono memorizzate le tabelle. Pertanto, l'applicazione ETL01.qvf è un motore parametrico per l'estrazione dei dati in cui i parametri per l'estrazione vengono gestiti esternamente dallo script.

Ogni applicazione ETL01.qvf interfaccia il proprio "ETL01_Settings.xls": la connessione tra i due file è 1: 1. Il collegamento è fatto dal nome: ad esempio:

COP_MyTestPage_ETL01.qvf  **COP_MyTestPage_ETL01_Settings.xls**

Non è necessario modificare lo script dell'applicazione ETL01.qvf poiché l'elenco delle origini dati e i metodi di estrazione sono definiti tramite file esterni Microsoft Excel. Tuttavia, esiste una sezione (Foglio "Estrazioni personalizzate") nella struttura del componente di scripting in cui gli sviluppatori possono creare il proprio script personalizzato per l'estrazione (es: estrazioni attraverso query che richiedono operazioni di unione tra le tabelle di fonti che non possono essere gestite con impostazioni parametriche sul file Microsoft Excel "ETL01_Settings.xls"). La funzionalità di cui sopra dovrebbe essere usata con parsimonia per non vanificare lo scopo dell'applicazione. Lo strato di QVD elementare è il risultato delle operazioni eseguite dall'applicazione ETL01.qvf. Il Qvd elementare di questo livello replica le fonti di dati sostanzialmente "as-is", con l'applicazione di alcune trasformazioni come la condizione Where o il partizionamento annuale o mensile delle tabelle dei fatti molto pesanti. Ciò significa che in questo livello le fonti dei dati di alimentazione vengono replicate in un rapporto 1-> 1 (1 tabella sorgente-> 1 file QVD) o, se necessario, diviso per periodi in un rapporto 1-> n (1 tabella sorgente -> n file QVD per ogni anno, ad esempio). A questo livello, le operazioni tra le tabelle di origine come Join, concatenate, mapping, non vengono eseguite. Lo scopo principale di questo livello è creare un'area estratta costituita da file in formato QVD pronti per essere utilizzati dall'applicazione QlikMart.qvf. La manutenzione e l'aggiornamento di questo livello è molto semplice e consente di disporre di un'area di staging dei dati separata da fonti originali, più facile accesso e accesso di lettura molto veloce. In questa area estratta è possibile effettuare una serie di controlli di qualità.

3.4.3 Specifiche di setup e uso

Questo paragrafo descrive le specifiche di installazione e uso l'applicazione ETL01.qvf e del relativo file Excel "ETL01_Settings.xls", che compongono il livello di architettura 1.

3.4.4 Impostazione delle variabili di base

L'impostazione delle variabili di base (principalmente variabili di percorso) e l'istruzione include sono definite in Tab Main dello script ETL01.qvf come mostrato di seguito:

```
////---Development Kit

//Basic Config
set vL.Prefix_QVD_Extracted = 'St_'; //Prefix used for file format QVD created by ETL01.qvw in Extracted area.
set vL.NumOfScriptLogStored = 10; //Number of processes observed through Log file created by default by ETL01.qvw application,
//displayed in presentation of ETL01.qvw and stored on file format QVD ScriptLogETL01.qvd
set vL.NumOfMonitoringLogStored = 200; //Number of processes observed in synthetic mode displayed in presentation of the ETL01.qvw
//and stored on file format QVD MonitoringLogETL01.qvd

//Paths Config
let vL.Path_ExternalData = '..\..\8.Import\';
let vL.Path_QVD_Extracted = '..\..\2.QVD\1.Extraction_QVD\';
let vL.Path_Settings = '..\..\5.Config\';
let vL.Path_Include = '..\..\3.Include\';

let vL.Path_QVD_Log_ETL = '$(vL.Path_Settings)' & 'ETL Audit Log\';
let vL.Path_Sub_ETL = '$(vL.Path_Include)' & '4.Sub\1.ETL01_Subs\';
let vL.Path_Sub_Base = '$(vL.Path_Include)' & '1.BaseVariable\';

//Routines Init
$(Include=$(vL.Path_Sub_ETL)\ETL01_InitConfigurations.qvs);

//Base Root
$(Include=..\Base_Root_Settings.qvs);
$(Include=..\Base_Root_Settings.qvs);
$(Include=..\Base_Root_Settings.qvs);
$(Include=..\Base_Root_Settings.qvs);
$(Include=..\Base_Root_Settings.qvs);
$(Include=..\Base_Root_Settings.qvs);

//Load variables path for Common QVD area
$(Include=$(vL.Path_Sub_Base)\Load_DWH_QVD_Path_Variables.qvs);
```

Variabili di base sono:

- **vL.Prefix_QVD_Extracted**: Prefisso per il file QVD generato da ETL01.qvf (seguendo la convenzione di denominazione definita come prefisso) [Source]_E_)
- **vL.NumOfScriptLogStored**: Numero di processi osservati tramite il file di registro creato per impostazione predefinita dall'applicazione ETL01.qvf e visualizzato nella presentazione di ETL01.qvf e archiviato in formato file QVD ScriptLogETL01.qvd
- **vL.NumOfMonitoringLogStored**: Numero di processi osservati in modalità sintetica displayed nella presentazione di ETL01.qvf e memorizzati in formato file QVD MonitoringLogETL01.qvd
- **vL.Path_ExternalData** Path relativa per I file esterni
- **vL.Path_QVD_Extracted** Path relativa per I file QVD generate da ETL01.qvf
- **vL.Path_Settings** Path relativa per I file di configurazione
- **vL.Path_QVD_Log_ETL** Path relativa per I file di log dei QVD

Microsoft Excel file "ETL01_Settings.xls"

Il file Microsoft Excel "ETL01_Settings.xls" consente la standardizzazione parametrica della fase di estrazione, quindi è possibile evitare di modificare direttamente lo script dell'applicazione ETL01.qvf. Il file contiene 3 fogli di lavoro:

Sheet	Description
Connections SetUp	Impostare le connessioni alle origini dati: includere le istruzioni di un file QVS per database relazionale o percorso per origini esterne
Tables SetUp	Tabelle da estrarre e specifiche di estrazione.
Fields SetUp	Campi di installazione (alias, campi calcolati)

Worksheet [Connections SetUp]

Il foglio di lavoro [Connections SetUp] consente di impostare le connessioni ai database relazionali, tramite la corrispondente istruzione di inclusione del file QVS memorizzato in 3.Include \ 3.ConnString della struttura di directory e / o di origini dati esterne con la sottodirectory specifica. *Nelle seguenti tabelle l'asterisco (*) indica che il campo è richiesto per il corretto funzionamento dello script ETL. L'asterisco tra virgolette (**) indica che il campo è richiesto in condizioni specifiche required on specific conditions*

Field	Description
Source ID (*)	Formato: <ul style="list-style-type: none">• Stringa di connessione; <p>Dirige la lettura sul connettore destro. Questi valori devono essere utilizzati come elenco di selezione dal campo [ID sorgente] del foglio [Impostazione tabella].</p> <p>IMPORTANT: È necessario creare un valore univoco di StringSource per ogni combinazione [SubPathQVD] / [SubPathExternalSources]</p>
Enable (*)	Picklist: <ul style="list-style-type: none">• Y• N (o vuoto) <p>Y = connessione da eseguire N / empty = script non esegue la connessione</p>
Source Type (*)	Picklist: <ul style="list-style-type: none">• DB• External• ODS (con DSO SAP Connector) <p>Identifica il tipo di Datasources</p>
DB Type	Solo per Source Type = 'DB' e il reload incrementale richiesto Picklist: ORA = Oracle

		SS = MS SQL Server ACC = MS Access (serial date)
Connection String (*)		Include del file QVS contained connection string (ODBC/OLE DB) <i>Campo obbligatorio se Source Type="DB"</i>
SubPath - External Source		Formato: <ul style="list-style-type: none"> \Path Sub-directory dove salvare I files relative a sorgenti dati esterne (xls, txt) ..\8.Import\SubPathExternalSources Se vuoto il default è ..\8.Import\
SubPath - Store Staging		Formato: <ul style="list-style-type: none"> \Path Sub-directory dove salvare I file in formato QVD elementare (Extracted) ..\QVD\DW_Staging\SubPathQVD Se vuota è la root ..\QVD\DW_Staging_QVD\ In caso di segmentazione, è utile impostare una sottocartella a seconda dell'anno di segmentazione; per fare ciò la subdirectory può essere riempita con la variabile \$(vL.Year) \
Description		Campo descrittivo

Esempio di compilazione:

Source ID	Enable	Source Type	DB Type	Connection String	SubPath - External Source	SubPath - Store Extracted	Description
RCO_VIEW	Y	DB	SS	\$(Include=..\3.include\3.connstring\RCO_VIEW.txt);		\$(vL.Year)\	
SQL	Y	DB	ORA	\$(Include=..\3.include\3.connstring\Connessione_ORACLE.qvs);			
Txt	Y	External			\Folder_DS_1		
Xls		External			\Folder_DS_2	\Folder to store	
Xls2	N	External				\Folder to store	
QVD	Y	External					

Esempio di Include \$(Include=..\3.include\3.connstring\Connessione_ORACLE.qvs);

Worksheet [Tables SetUp]

Il foglio di lavoro contiene l'elenco delle tabelle da estrarre e le specifiche dell'estrazione per ogni tabella

.

Campo	Descrizione
Source ID (*)	Picklist: <ul style="list-style-type: none"> Foglio: [Connections Setup] Campo: [Source ID] Identifica la source da leggere. Indirizza la lettura al connettore corretto.

	ATTENTION: i valori inseriti nel campo [ID sorgente] devono corrispondere ai valori nel foglio di lavoro [Impostazioni connessioni] Campo: [ID sorgente]
Table Name (*)	Formaot: <ul style="list-style-type: none"> Relational DB: [table name] External Sources: [File name].[File extension]
QVD Name (*)	E' il nome del file QVD in cui sarà salvata la tabella.
Description	Descrizione
Extraction Mode (*)	Picklist: <ul style="list-style-type: none"> FULL INCR empty Tipo di estrazione. Se vuoto la tabella non viene estratta
Store Partial	Picklist: <ul style="list-style-type: none"> Y N (o empty) In combinazione con Extraction Mode = 'INCR' il requisito è quello di memorizzare la parte incrementale in una QVD separata con suffisso il campo di incrementale
Where ('*')	Picklist: <ul style="list-style-type: none"> Y N (o empty) Y = where condition applicata N / empty = where condition non applicata
Where on SELECT / LOAD	Picklist: <ul style="list-style-type: none"> SELECT LOAD Mostra se la clausola deve essere utilizzata nell'istruzione Select o Load (carico precedente). ATTENZIONE: Campo obbligatorio se viene usata la clausola Where
Where Clause ('*')	Codice relativo alla clausola Where. Il codice utilizzato deve essere strutturato con la sintassi Qlik se la clausola Where è definita in "LOAD" o con una sintassi specifica della sorgente da cui viene estratta la tabella. ATTENZIONE: Campo obbligatorio se viene usata la clausola Where
Note	Field Note

Import Configuration - External Sources (**)	<p>Format:</p> <ul style="list-style-type: none"> (specific import statement) <p>Dichiarazioni specifiche di importazione in lingua Qlik per le fonti esterne (txt, xls, qvd). Esempio: LOAD * FROM Path \ File.ext specificImp;</p> <p>ATTENZIONE: campo obbligatorio se TypeDB = External</p>
To Segment	<p>Picklist:</p> <ul style="list-style-type: none"> Y N <p>'Y' mostra che la tabella deve essere congelata, ovvero la tabella originale viene estratta e archiviata in diversi formati di file QVD suddivisi per anno o anno o ogni giorno in base a una data di campo definita.</p> <p>In questo modo le tabelle vengono memorizzate in tanti file in formato QVD come i periodi (anno o anno). I file sono memorizzati con il nome [vs_Prefix_Name_Extracted] _ [nametable] _ [YEAR o YEARMONTH o YEARMONTHDAY] .qvd.</p>
Segment Type (**)	<p>Picklist:</p> <ul style="list-style-type: none"> YEARLY MONTHLY DAILY <p>Mostra se il congelamento deve essere eseguito per periodo anno o mese o giorno</p> <p>Yearly: qvd stored [vs_Prefix_Name_Extracted]_[table name]_[YYYY].qvd</p> <p>Monthly: qvd stored [vs_Prefix_Name_Extracted]_[table name]_[YYYYMM].qvd</p> <p>Daily: qvd stored [vs_Prefix_Name_Extracted]_[table name]_[YYYYMMDD].qvd</p> <p>ATTENZIONE : Campo obbligatorio se To Segment = 'Y'</p>
Segment Field (**)	<p>Identifica il campo contenente la data (memorizzazione mensile e giornaliera) o l'anno (archiviazione annuale) Per il congelamento annuale se il campo è una data è necessario utilizzare la formula: year (fieldData)</p> <p>eg monthly, daily: fieldData eg annual: year(fieldData)</p> <p>ATTENZIONE : Campo obbligatorio se To Segment = 'Y'</p>
Segment From (**)	<p>Identifica la data (mensile, congelamento giornaliero) o l'anno</p>

	<p>(congelamento annuale) da cui inizia il congelamento della tabella</p> <p>es monthly: AddMonths(today(),-3) or 20160101 es annual: (year(today()))-2) or 2010</p> <p>ATTENZIONE : Campo obbligatorio se To Segment = 'Y'</p>
Segment To ('*')	<p>Identifica la data (mensile, congelamento giornaliero) o l'anno (congelamento annuale) in cui deve finire il congelamento della tabella</p> <p>es monthly: today() es annual: year(today())</p> <p>ATTENZIONE : Campo obbligatorio se To Segment = 'Y'</p>
Incremental Load - Date Field ('*')	<p>Mostra il campo che nella tabella specifica la data di immissione del nuovo record. Il campo viene anche utilizzato per mostrare la data del tipo da utilizzare per il caricamento incrementale. Di seguito, questo campo sarà discusso in profondità.</p>
Incremental Load - Counter Filed	<p>Viene utilizzato per specificare il campo che in una tabella contiene il contatore di tipo numerico da utilizzare per il caricamento incrementale. Di seguito, questo campo sarà discusso in profondità</p>
Incremental Load - Update - Table Primary Key	<p>Mostra il campo che nella tabella specifica il primario per il carico di aggiornamento incrementale.</p>
Incremental Load - Delete - Table Primary Key	<p>Mostra il campo che nella tabella specifica il primario per il carico di cancellazione incrementale.</p>

Esempio di compilazione:

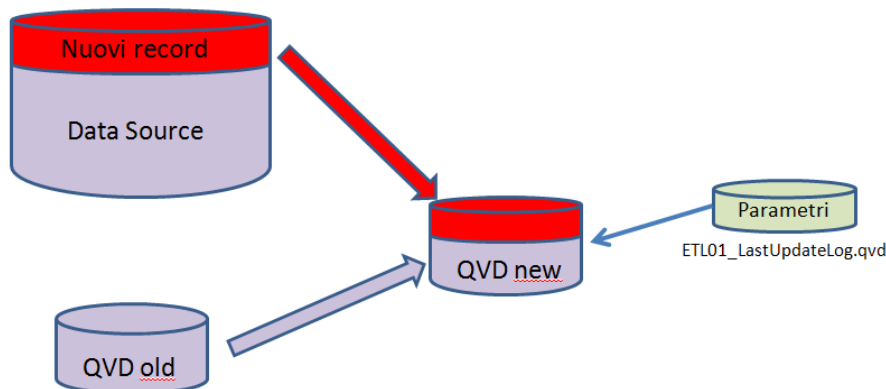
Source ID	Table Name	QVD Name	Description	Extraction Mode	Store Partial	Where	Where on SELECT / LOAD	Where Clause	Note	Import Configuration - External Sources
SQL	dbo.DemoTable	TableSQL		FULL		Y	SELECT	[Product]='A003'		
SQL	[dbo.Demo Table with space]	TableSQL		INCR	Y	N				
Txt	Text_1.txt	TableTXT		FULL						(ansi, fix, no labels, header is 0, record is line)
Xls1	MS_Excel_1.xls	XLS1		FULL						(biff, embedded labels, table is DatiBilancio\$)
Xls1	MS_Excel_1.xls	XLS1		FULL						(biff, embedded labels, table is Respo_Vendite\$)
Xls2	MS_Excel_2.xls	XLS2		FULL						(biff, embedded labels, table is Respo_Saldi\$)
qvd	QVD_1.qvd	XLS3		INCR		Y	LOAD	Customer<>'ABC'		(qvd)

Segue:

L	M	N	O	P	Q	R	S	T
To Segment	Segment Type	Segment Field	Segment From	Segment To	Incremental Load - Date Field	Incremental Load - Counter Filed	Incremental Load - Update - Table Primary Key	Incremental Load - Delete - Table Primary Key
					DATA_INOLTRO		[Primary ID]	[Primary ID]
Y	YEARLY	year(DATA_REG)	2008	year(today())				
Y	MONTHLY	DATA_REG	AddMonths(today(),-3)	Today()				
						COUNTER	[Primary ID]	[Primary ID]

La funzionalità di caricamento incrementale consente di specificare se una tabella di origine deve essere caricata interamente o applicare la logica incrementale, al fine di ottimizzare le prestazioni del processo.

Questa funzionalità può essere utilizzata se nella tabella per caricare c'è un campo che consente di identificare nuovi record dal carico precedente.



Il campo che identifica i nuovi record può essere una data o un contatore numerico. Per ogni caricamento, il valore massimo di questo campo viene memorizzato in una tabella di sistema e utilizzato come filtro durante i successivi caricamenti incrementali.

A livello di script, la modalità di caricamento è determinata dal valore (FULL o INCR) del campo Modalità di estrazione

Il carico incrementale potrebbe essere:

- INSERT: caricamento di nuovi record
- UPDATE: caricamento di nuovi record e record aggiornati
- DELETE: caricamento di nuovi record, record aggiornati e record cancellati

INSERT

Per abilitare il caricamento incrementale su una tabella in modalità INSERT è necessario impostare nel foglio di lavoro [Impostazione tabelle] del campo ETL01_Settings.xls [Campo data caricamento incrementale] (se il campo che identifica nuovi record è il tipo di dati) o il campo [Contatore carico incrementale Campo] (se il campo che identifica i nuovi record è di tipo numerico)

UPDATE and DELETE

Per abilitare il caricamento incrementale su una tabella in modalità UPDATE e DELETE è necessario impostare nel foglio di lavoro [Table Setup] del file ETL01_Settings.xls il campo [Incremental Load - Update - Table Primary Key] e [Incremental Load - Delete - Table Primary Key] con il valore o la concatenazione di valori che identificano univocamente i record

P	Q	R	S
Incremental Load - Date Field	Incremental Load - Counter Filed	Incremental Load - Update - Table Primary Key	Incremental Load - Delete - Table Primary Key
Table_Date_1		[Primary ID]	[Primary ID]
	ID_Table_1	[Primary ID]	[Primary ID]

WorkSheet [Fields SetUp]

The worksheet [Fields SetUp] permette di gestire i campi estratti da ogni tabella. In particolare consente di:

- Decidi quali campi di una tabella / file estrarre
- Assegna un alias ai campi estratti
- Aggiungi nuovi campi calcolati
- Applicare una funzione nella fase di carico precedente

L'impostazione del foglio di lavoro [Fields SetUp] è facoltativa: se una tabella è menzionata nel foglio [Table SetUp] e non è dettagliata nel foglio di lavoro [Fields SetUp], l'applicazione ETL01.qvf esegue un LOAD * importando tutti i campi.

<i>Field</i>	<i>Description</i>
QVD Name (*)	Picklist: <ul style="list-style-type: none"> • [table name] worksheet [Tables Setup]
To Read (*)	Picklist: <ul style="list-style-type: none"> • Y • N (o vuoto) Y = il campo viene estratto N / vuoto il campo non viene estratto
Field (*)	Nome campo (è possibile usare *)
Alias	Se compilato l'alias viene applicato
To Load Preceding	Picklist: <ul style="list-style-type: none"> • Y • N (o vuoto) Y = viene applicata la preceding load

Field / Function - Preceding Load	Nome campo / Preceding Load.
Alias - Preceding Load	Se compilato l'alias viene applicato / Preceding Load
Note	Note

Esempio di compilazione:

QVD Name	To Read	Field	Alias	To Preceding Load	Field / Function - Preceding Load	Alias - Preceding Load	Note
TableTXT	n	[@1:2T]					
TableTXT	y	[@3:32T]	CONTRAENTE				
TableTXT		[@33:67T]					
TableTXT	y	[@157:159T]	COD_PROD				
TableTXT	y	[@160:167T]	tmp_mov	Y	Year(tmp_mov)	Y_Movimento	
XLS1	y	AREA					
XLS1	y	[RESP ZONA]		Y	num([RESP ZONA],000)	[RESP 000]	
TableSQL	y	*		y			
TableSQL	y	date(campo_data)	Data_tmp	y	Data_tmp	Data	

ETL01.qvf Application

Il file ETL01.qvf ha le seguenti caratteristiche:

- Memorizza in formato QVD i dati caricati da diverse origini dati: database relazionali (testati con connettore OLE DB e ODBC) e / o fonti esterne
- Caricare fonti esterne dalla sottodirectory / memorizzare il QVD creato nella sottodirectory
- Impostare le singole tabelle da estrarre con l'istruzione import corrispondente
- Applicare una condizione di Where sulla tabella singola (in Select o in Load (Preceding Load))
- Congela le tabelle annuali, mensili o giornaliere
- Esegue un'estrazione incrementale (in insert per data o per contatore incrementale)
- Permette di specificare i campi da estrarre specificando un alias con la possibilità di eseguire una trasformazione nella Preceding Load
- Aggiungere campi calcolati (cn alcune limitazioni)
- Calcolare la data dell'ultimo aggiornamento di una tabella estratta
- Gestire delle estrazioni personalizzate
- Mostra tabelle riepilogative con informazioni relative ai processi di estrazione
- Monitorare le estrazioni (cardinalità, prestazioni, tabelle estratte, scripting)

Per la gestione di estrazioni personalizzate, nello scripting dell'applicazione ETL01.qvf c'è il foglio "Custom Extractions".

The screenshot shows the ETL01_Extractor application. On the left is a dark sidebar with a menu containing options like 'Incremental', 'Segment, Init & Where', 'Fz_Add_Where', 'Aliasing', 'Code_Generator', 'Coe_Execution', 'Fz_Store_Table', 'LastUpdateLog', 'Load_LastUpdateLog', 'Reset_Variables', and '**** END SUBS ****'. The 'Custom Extraction' option is selected at the bottom. The main area on the right displays a script with line numbers 1 through 27. The script is a VBA macro for custom extraction, including comments in Italian and code for connecting to a data source, executing a SQL query, and storing the results in a table with auditing.

```

1  /*-----
2  Custom extraction (not configured on Settings.xls)
3  Insert the custom code inside Sub CustomCode
4
5  Syntax:
6  - 1 - Insert the custom code (es: SAP OLAP connector, query SQL...)
7  - 2 - To execute the store and drop of the table with auditing features use: Call Fz_StoreTable('Table Name')
8  -----
9  Sub Sub_CustomCode
10     Trace ('CUSTOM EXTRACTIONS')
11
12
13
14     /*1 - Extraction script code
15     CustomTable:
16     SQL SELECT
17
18     FROM 'Test_Table_2'
19
20     /*2 - Store and drop of the table with auditing
21     Call Fz_StoreTable('CustomTable')  /*'CustomTable' it's the table name
22
23
24 End Sub
25
26
27
**** END SUBS ****

```

In questo foglio è possibile inserire il codice che consente ad esempio di estrarre una tabella risultante da una query SQL, che collega in join tabelle di fonti diverse. Altre estensioni personalizzate sono per istruzioni che non includono "LOAD ... FROM" o "SQL SELECT ... FROM" (ad esempio SAP OLAP Connector, importazione massiva di file flat). Questo tipo di estrazione non è gestito in ETL01_Settings.xls.

Nello script c'è un esempio di 3 blocchi che mostra come usare la [Sub CustomCode] con connessione parametrica al foglio [Connessioni] del file Microsoft Excel "ETL01_Settings.xls", per poi inserire del codice personalizzato e memorizzare / scaricare dalla memoria la tabella, preservare le funzioni di monitoraggio dell'applicazione ETL01.qvf (Extractor):

1. **Chiamata alla connessione:** call Fz_ConnectionCustom ('Name Connection') il nome della connessione nel campo [StringSource] di ETL01_Settings.xls
2. **Codice script della estrazione:** (eg: query SQL...)
3. **Store and Drop** con funzioni di Auditing: Call Fz_StoreTable('Name Table')

System Tables

Durante l'esecuzione di ETL01.qvf I seguenti file QVD possono essere creati se le relative chiamate di monitoring sono attive nello script (..\Config\ETL Audit Log\):

- ETL01_MonitoringLog:
 - *ETL_ExtractionCounter*: contatori
 - *ETL_Source*: origine dati
 - *ETL_ExtractionDuration*: durata esecuzione
 - *ETL_ExtractionMode*: modalità di estrazione
 - *ETL_StoredTableName*: nome QVD
 - *ETL_StorePath*: QVD path
 - *ETL_StoreQvdName*: nome file QVD

- *ETL_DropMode*: drop mode (system field)
- *ETL_RowNum*: numero di records nel file QVD
- *ETL_FieldNum*: number of fields nel file QVD
- *ETL_StoreDate*: store date nel file QVD
- *ETL_StoreTime*: store time nel file QVD
- ETL01_ScriptLog:
 - *ETL_ScriptCounter*: contatore
 - *ETL_Key_ETL01Timestamp*: timestamp inizio processo
 - *ETL_ScriptLog_TimeStamp*: timestamp inizio logging
 - *ETL_ScriptLog_ScriptRows*: righe di script
 - *ETL_ScriptLog_RowNo*: riga di script
- ETL01_TraceLog:
 - *ETL_Key_ETL01Timestamp*: timestamp inizio processo
 - *ETL_TraceDate*: data di trace
 - *ETL_TraceMsg*: messaggio di trace definite nello script
- ETL01_MonitoringLogSintesi:
 - *ETL_Key_ETL01Timestamp*: timestamp inizio processo
 - *ETL_StartTimeStamp*: timestamp inizio estrazione
 - *ETL_EndTimeStamp*: timestamp fine estrazione
 - *ETL_Duration*: durata processo
 - *ETL_TotalFieldNum*: numero di campi estratti
 - *ETL_TotalNoOfTable*: numero di tabelle estratte
 - *ETL_TotalRowNo*: numero of records estratti

3.4.5 Transformation level

Le operazioni vengono eseguite dall'applicazione QlikMart.qvf, se si utilizza l'architettura a 3 livelli. Tutti i passaggi necessari come trasformare, normalizzare e aggregare i dati memorizzati nel file QVD creato dall'applicazione ETL01, in modo funzionale per creare il cloud associativo utilizzato dalle applicazioni di "Presentation" viene fatto sull'app QlikMart.

La stessa base dati, quindi, può creare nuvole associative strutturate in modo diverso a seconda delle finalità e dei metodi di navigazione delle informazioni richieste nelle applicazioni di "Presentation".

Tuttavia, nell'implementazione di uno script specifico, è richiesto agli sviluppatori di rispettare gli standard di sviluppo.

Un principio molto importante è il seguente: l'applicazione QlikMart.qvf deve considerare tutte le possibili trasformazioni, aggregazioni, formattazioni necessarie per l'applicazione "Presentazione", che devono semplicemente ottenere le informazioni da questo secondo livello senza eseguire altre elaborazioni.

Pertanto, la parte principale della logica che guida il cloud associativo del documento finale Qvf deve essere realizzata a questo livello.

Per prestare particolare attenzione alle prestazioni nella navigazione nelle applicazioni di "Presentation", vengono identificate le seguenti linee guida per strutturare la nuvola associativa in particolare durante lo sviluppo di progetti che coinvolgono grandi volumi di dati:

- Mancanza di chiavi sintetiche native di Qlik: l'associazione tra le tabelle di un data cloud deve essere gestita attraverso i campi chiave definiti all'interno dei file QVD di livello 2.
- Adottare il più possibile strutture di cloud associativo come Star Schema: questo modello consente, a differenza di schemi Snow Flake, di "accorciare" i percorsi associativi tra campi di diverse tabelle e tra campi utilizzati all'interno degli oggetti delle applicazioni "Presentation" (sia come dimensioni e come espressioni).
- Gestione dei campi non utilizzati nell'applicazione "Presentation": non importare le informazioni che sicuramente non sono oggetto di analisi e navigazione da parte degli utenti.
- Precalcolare le espressioni complicate replicate negli oggetti di "Presentation" all'interno dello script dell'applicazione QlikMart.qvf, in modo da renderle disponibili come campi semplici a questo livello.

3.4.6 Specifications of setup and use Specifiche di setup e uso

Questo paragrafo descrive le specifiche di installazione e uso dell'applicazione QlikMart.qvf che compongono il livello di architettura 2.

3.4.6.1 SetUp Base Variables

L'impostazione di Variabili Base (principalmente variabili di percorso) e l'istruzione include sono definite nella Scheda Principale dello script QLIK MART.qvf come mostrato di seguito.

```
////---Development Kit
//Basic Config
set vL.Prefix_QVD_Staging      = 'St_';          //Prefix used for file format QVD created by ETL01 - Extractor.qvw in Staging area.
set vL.NumOfScriptLogStored    = 10;             //Number of processes observed through Log file created by default by ETL01 - Extractor.qvw application,
set vL.NumOfMonitoringLogStored = 200;          //Number of processes observed in synthetic mode displayed in presentation of the ETL01 - Extractor.qvw

Let vL.ETLName                 = 'ETL01_Sales_EMEA';
// Let vL.QvwName               = 'ETL02.qvf';

Directory lib://Dev@QlikCentralNode/Sales\EMEA\;

//Paths Config
let vL.Path_ExternalData      = 'Import\';
let vL.Path_QVD_DwStaging     = 'QVD\DW_Staging\';
let vL.Path_Settings         = 'Config\';

let vL.Path_QVD_Log_ETL       = '$(vL.Path_Settings)' & '\ETL Audit Log\';
let vL.Path_Subs_ETL          = '$(vL.Path_Include)' & '\4.Sub\1.ETL01_Subs\';
```

Le Variabili di Base sono:

- **vL.Prefix_QVD_Extracted**: Prefisso dei file QVD generati da ETL01.qvf
- **vL.NumOfScriptLogStored**: Numero di processi osservati tramite il file di registro creato per impostazione predefinita dall'applicazione QLIK MART.qvf e visualizzato nella presentazione di QLIK MART.qvf e archiviato in formato file QVD ScriptLogQLIK MART.qvd
- **vL.Path_ExternalData** Path relativa alla cartella dei file esterni
- **vL.Path_QVD_Extracted** Path relativa ai file QVD generati in ETL01
- **vL.Path_Settings** Path relativa ai file di configurazione
- **vL.Path_QVD_Log_ETL** Path relativa per I log QVD

Le principali subroutines disponibili sono:

- **Fz_StoreTable**: subroutine per lo store e drop delle tabelle
 - Syntax: Fz_StoreTable ('TableName' [,DropMode,StorePath,StoreQvdName])
- **Fz_Calendar**: subroutine per la generazione del calendario
 - Call Fz_Calendar('Static Calendar' , ' St' , '01/08/2010' , '04/08/2012')

Le subroutines con prefisso Fz_Check e Fz_Trace sono disponibili per il trace log e dei controlli base sui dati.

3.4.7 Livello presentation

Il terzo livello consiste in applicazioni di "Presentation" orientate all'analisi e alla navigazione dei dati da parte degli utenti

Le elaborazioni integrate a questo livello devono essere considerate come eccezioni: lo scripting delle applicazioni "Presentation" dovrebbe contenere principalmente istruzioni per l'importazione dei dati dalle applicazioni di QlikMart. Le eccezioni di calcolo (lato script) devono essere valutate attentamente, perché se da un lato può essere utile per evitare di lavorare sul livello sottostante, dall'altro lato peggiorano le prestazioni di ricarica delle applicazioni e possono creare una perdita di flessibilità del cloud associativo.

Per prestare particolare attenzione alle prestazioni di navigazione delle applicazioni a questo livello, vengono identificate le seguenti linee guida per progettare gli oggetti di "Presentation", in particolare durante lo sviluppo di progetti che coinvolgono grandi volumi di dati è opportuno tenere conto di:

- Limitare il numero di oggetti sullo stesso foglio
- Limitare l'uso di Set Analysis nelle espressioni, in special modo se nidificate
- Limitare le dimensioni calcolate negli oggetti grafico
- Utilizzare le variabili per le espressioni replicate in più oggetti

Inoltre, al fine di massimizzare l'efficacia delle applicazioni di "Presentation", devono essere utilizzati i principi di ergonomia e design visivo.

3.5 Architettura ad un solo livello (Diretta)

Questo livello consiste in una singola app che contiene tutti i livelli necessari per navigare e analizzare i dati.

Questo tipo di approccio viene utilizzato nel caso di un approccio specializzato all'analisi di big data, applicazioni altamente personalizzate, creazione di app on demand e altre applicazioni guidate da esigenze aziendali.

Lo scripting delle Architetture ad un livello dovrebbe contenere principalmente istruzioni per l'importazione dei dati dal database di origine, sia esso di origine database o QVD, direttamente o tramite connettori. Le eccezioni sui calcoli effettuati in script devono essere valutate attentamente, perché se da un lato può essere utile per evitare di lavorare sul livello sottostante, dall'altro lato peggiorano le prestazioni di ricarica delle applicazioni e possono creare una perdita di flessibilità del modello associativo.

Come per le applicazioni a 3 livelli è opportuno tenere conto dei seguenti suggerimenti:

- Limitare il numero di oggetti sullo stesso foglio
- Limitare l'uso di Set Analysis nelle espressioni, in special modo se nidificate
- Limitare le dimensioni calcolate negli oggetti grafico
- Utilizzare le variabili per le espressioni replicate in più oggetti

Inoltre, al fine di massimizzare l'efficacia delle applicazioni devono essere utilizzati i principi di ergonomia e design visivo.

Direzione Sistemi Informativi

-- Servizio --

-- Ufficio --

4 Gestione dell'infrastruttura

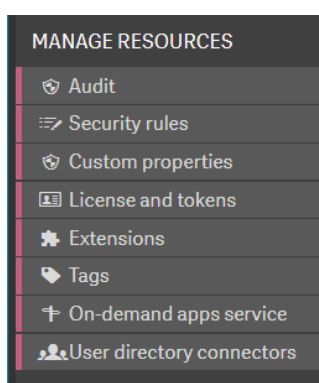
In questo capitolo andremo a definire quelle area di governance che sono collegate alla parte applicativa e che gestiscono oggetti e elementi necessari per la corretta utilizzazione della piattaforma da parte dell'utente.

4.1 Gestione extension

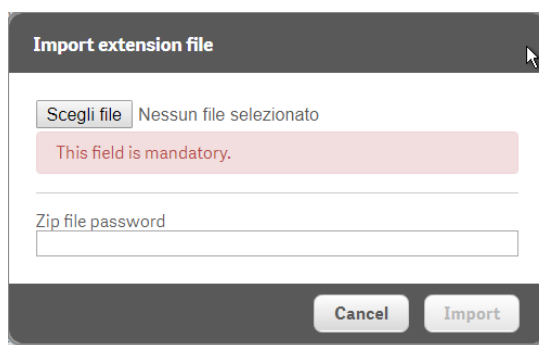
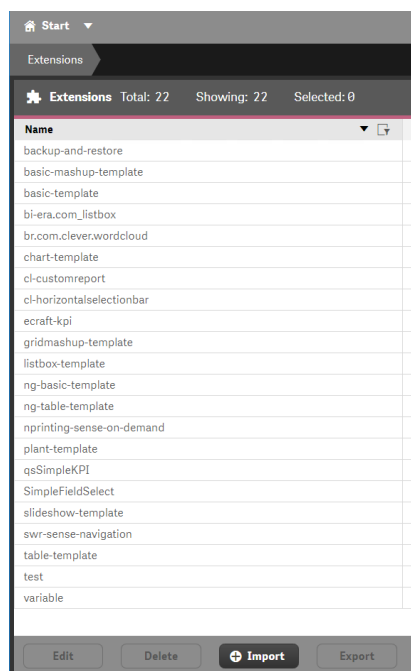
Per estendere le funzionalità di front end di Qlik Sense possono essere utilizzate delle Extension che ampliano le capacità di visualizzazione, calcolo e interfacciamento delle applicazioni.

Per accedere alla QMC utilizzare il link [XXXXXXXXXXXXXXXXXXXXXXXXXXXX](#)

Le extension vengono gestite attraverso la QMC dal relativo menù all'interno del gruppo MANAGE RESOURCES:



Una extension per poter essere utilizzata deve essere importata sul server. Per fare questo è necessario che, dal relativo menù, si utilizzi la funzione di Import:



Il file atteso in ingresso è uno zip che contiene tutti gli elementi, i file, propri dell'estensione stessa. Il server si occuperà di scompattare i file nella opportuna cartella.

Una volta che l'extension è stata caricata sarà possibile gestirne le sue caratteristiche di visibilità e accesso attraverso il seguente menù:

The screenshot shows the 'Edit extension' interface. On the left, a sidebar lists 'Extensions' and 'ecraft-kpi'. The main area is titled 'Edit extension' and contains several sections: 'IDENTIFICATION' with fields for 'Name' (ecraft-kpi) and 'Owner' (Qlik Sense2 (DAYCO\qlik.sense2)); 'TAGS' with a tag 'EMEA'; and 'CUSTOM PROPERTIES'. On the right, a sidebar shows 'Properties' (Identification, Tags, Custom properties) and 'Associated items' (User access, Security rules). At the bottom, there are 'Apply' and 'Cancel' buttons.

La visibilità dell'extension è controllabile attraverso l'uso delle Custom Properties e delle Security Rules definite nell'ambiente.

The screenshot shows the 'Extension: associated items' interface. It features a table of security rules with columns: Name, Description, Resource filter, Actions, and Disabled. The table lists two rules: 'Extension' and 'ContentAdmin'. On the right, a sidebar shows 'Properties' (Identification, Tags, Custom properties) and 'Associated items' (User access, Security rules).

Name	Description	Resource filter	Actions	Disabled
Extension	Everyone can view extensions	Extension_*	Read	No
ContentAdmin	Content admin should have access rights to conte...	Stream_*,App*,ReloadTask_*,UserSyncTask...	Create, Read, ...	No

Le Extensions sono proprie dell'ambiente in cui sono caricate e non vengono inglobate nelle app che le utilizzano.

E' necessario quindi che ogni extension sia trasportata in ambiente di produzione per poter essere utilizzata. Il processo resta lo stesso descritto sopra.

4.2 Creazione connessioni ODBC

Per poter avere accesso ai dati da una applicazione è necessario utilizzare una Data Connection. Saranno disponibili tre data connection, tutte di tipo ODBC, per accedere ai diversi sistemi.

Le connection saranno di tre tipi:

- Teradata
- Hadoop
- Oracle

Le connessioni saranno già configurate per accedere ai sistemi sorgenti ma, per poter attivare la connessione, saranno richieste le credenziali dello sviluppatore che accede al dato. Questa scelta è legata a regole di sicurezza necessarie per la tracciatura degli accessi.

IMPORTANTE: *questa impostazione richiederà l'inserimento di nuove credenziali nel momento in cui alla persona che ha sviluppato l'applicazione dovessero essere, per qualsiasi motivo, tolti o cambiati i diritti di accesso*

4.3 Gestione cartelle generiche per utente

Per gli utenti che utilizzano la piattaforma sarà necessario predisporre delle cartelle personali all'interno delle quali saranno salvati i file di dati che l'utente stesso utilizza in modalità self service. Tali cartelle saranno generate sulla NAS di riferimento da DSI e saranno ad accesso esclusivo di ogni singolo utente.

La cartella sarà denominata con lo userid dell'utente e avrà uno spazio di storage limitato, uguale per ogni utente.

Le necessità di incremento di spazio dovranno essere richieste dall'utente a DSI.