ISTITUTO NAZIONALE DI FISICA NUCLEARE
Direzione Sistemi Informativi

INFN-23-08-DSI
6 marzo 2023

Business Intelligence INFN : Modello Dati Multidimensionale per Ordini e Acquisti

Thomas Angelini[1], Barbara Demin[1], Claudio Galli[1]

[1)] *INFN, Direzione Sistemi Informativi, 00044 Frascati (Roma) Italy*

Abstract

L'obiettivo di questo elaborato è documentare le difficoltà incontrate, e le soluzioni tecniche adottate, nella creazione di un modello multidimensionale che consenta l'esplorazione dei dati di Gare e Ordini dell'INFN. Da principio ci si è concentrati sull'analisi delle fonti, tra cui il tool *RDA* che dal 2020 è fonte autoritativa per le gare sotto la soglia dei 40keuro. In un secondo momento, sono stati presi in esame i dati storici e sopra soglia provenienti direttamente dal sistema contabile. I due modelli sono stati uniformati per ottenere una visione coerente e consistente di tutti i dati in possesso dell'ente. Nel documento vengono utilizzati due diversi formalismi descrittivi: i diagrammi *E/R (Entity Relationships)* [2] e *DFM (Dimensional Fact Model)*[3] utilizzati, rispettivamente, nella fase di indagine dell'*as is* e nella progettazione multidimensionale. A seguire viene presentata l'implementazione del modello sul *data warehouse* della Direzione Sistemi Informativi attraverso i processi di *ETL (Extract Transform Load)*. Infine sono riportate alcune note conclusive e i vantaggi ottenuti.

DOI: 10.15161/oar.it/76972

Published by
Laboratori Nazionali di Frascati

1 Analisi delle risorse

Con la costruzione del modello dati per gli Ordini di acquisto INFN si è scelto di intraprendere un percorso di progettazione diverso da quello utilizzato finora. Con il tempo, infatti, ci si è resi conto che i modelli sviluppati in passato non riuscivano ad essere sufficientemente versatili da rispondere adeguatamente alle sempre più mutevoli esigenze degli utenti. Piuttosto che concentrarsi sui requisiti necessari per implementare i report finali, in questa occasione si è scelto di ribaltare l'approccio, partendo da un'analisi dello stato dell'arte dei dati coinvolti e dalla raccolta della documentazione disponibile. Il processo scelto è, dunque, quello proposto nel libro di *Golfarelli e Rizzi*[3].

Una della prime difficoltà incontrate è stata sicuramente la scarsa documentazione di progetto prodotta nel passato per cui è stato necessario ricostruire il dettaglio dei dati e degli applicativi coinvolti nelle operazioni di lettura e scrittura. Inoltre, andando avanti nell'analisi, ci si è presto resi conto che le logiche di *business* dei vari applicativi avevano subito diverse modifiche nel corso degli anni: alcune delle relazioni emerse erano deprecate o valide solo in riferimento a precisi archi temporali.

Per risolvere queste criticità, si è lavorato a stretto contatto con gli esperti di area del servizio *Sviluppo e Gestione Applicativi*, revisionando e verificando in modo continuo i dati e le loro relazioni.

Questo continuo *trade-off*, oltre ad assicurare una corretta contestualizzazione di dati e applicativi, ha contribuito ad individuare con maggiore precisione le informazioni realmente utili da integrare nel modello.

1.1 Query sui dati come metodologia d'indagine

Uno dei problemi osservati era che le principali basi dati esaminate non presentavano, ed è tuttora così, chiavi primarie (*primary key*) o chiavi esterne (*foreign key*). Per cui, era necessario fare emergere tali informazioni dalle tabelle in essere e dalle logiche di business degli applicativi. A tale scopo, è stata condotta un'analisi sulle cardinalità dei dati con ausilio di alcune *query SQL* riportate nel seguito.

La prima *query* utilizzata mostra come ricercare possibili chiavi primarie di una tabella denominata "tab". Se il risultato della *query* non ritorna alcun record, i campi scelti nell'indagine fanno tutti parte di una chiave composta per la tabella.

```
select tab.campo, tab.campo2... tab.campoN, count(
   distinct(ROWNUM))
from tabella tab
where tab.campo1 is not null
and tab.campo2 is not null --ripetuto per ogni campo
group by tab.campo, tab.campo2... tab.campoN
having count(distinct(ROWNUM)) > 1;
```

Listing 1: Query per valutare se $campo1, campo2...campoN$ sono chiave

La seconda *query* utilizzata identifica possibili dipendenze funzionali tra i due campi *Campo1* e *Campo2* delle tabelle *tabella1* e *tabella2*.

```
select tab2.campo2, count(distinct(tab1.campo1)
from tabella2 tab2
join tabella1 tab1 on (tab2.id = tab1.id)
where tab2.campo2 is not null
and tab1.campo1 is not null
group by tab2.campo2
having count(distinct(ROWNUM)) > 1;
```

Listing 2: Query per valutare relazione 1-n tra tab1.Campo1 e tab2.Campo2.

I risultati ottenuti con le precedenti *query* sono stati condivisi ed analizzati con gli esperti di area del servizio *Sviluppo e Gestione Applicativi* come indicato nel paragrafo precedente.

2 Progettazione E/R

Una volta investigate le cardinalità e individuate le chiavi primarie delle entità in gioco, è stato necessario sviluppare il diagramma di tipo *E/R (Entity-Relationship)*)[4]. Questo passaggio ha rappresentato la base su cui poggiare la successiva modellazione multidimensionale in versione ROLAP.

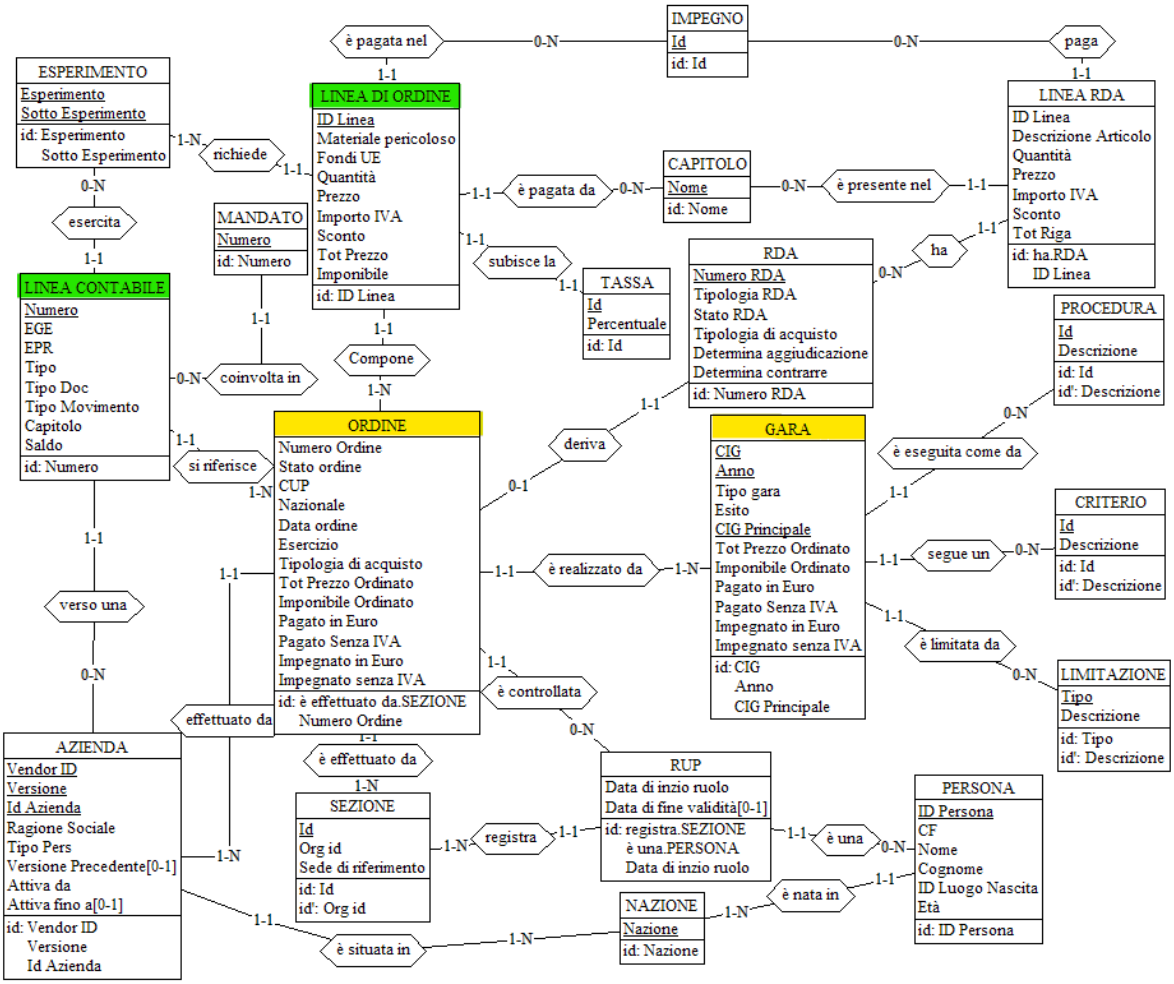

Figura 1: Schema E/R del modello Ordini.

Nella modellazione dello schema E/R sono stati considerati sia i contributi derivanti dal tool *RDA*[1] sia quelli del mondo contabile.

Nonostante il focus originale fosse la modellazione dei dati di *RDA* dal 2020 in poi, ben presto ci si è resi conto che risultava necessario includere nel modello anche il mondo contabile proprietario, invece, dei dati storici pregressi ante 2020. Inoltre, si è osservato che, spesso, le informazioni derivanti dal mondo contabile risultavano necessarie per poter estrapolare il costo reale degli ordini, poiché alcune scritture avvenivano/avvengono direttamente sul sistema contabile a rettifica dei costi preventivati nel tool *RDA*.

[1]RDA è il tool INFN per il trattamento delle richieste di acquisto fino a 40K€ dal 2020

Tutto ciò, ha sicuramente complicato la modellazione dello schema *E/R*, ma, di contro, ha permesso di conservare il valore dei dati storici, anche se talvolta non completi, affiancati e relazionati ai dati più recenti. Il modello così riformulato si mantiene anche aperto a possibili integrazioni future preservando entrambi i contributi.

Fatta questa premessa in [fig.1] si riporta il modello *E/R* completo, su cui sono state evidenziate le quattro entità di principale interesse.

L'entità **Gara** è caratterizzata da diversi elementi come la *Limitazione* di gara, il tipo di *Procedura* per l'espletamento ed il *Criterio* di aggiudicazione. In generale, la gara si concretizza in una serie di *Ordini* ed è contraddistinta da un codice univoco, il *CIG* (Codice Identificatore di Gara). Nel caso in cui da una gara sia necessario derivarne altre, ad esempio per l'acquisto di servizi ancillari rispetto alla fornitura principale, si ricorre a gare secondarie. In questo caso, le secondarie riportano come *CIG Primario* quello della gara padre a cui afferiscono mutuando essa le caratteristiche di Limitazione, Procedura e Criterio.

L'entità **Ordine** è un tipo di pratica effettuata presso una determinata *Sezione o Laboratorio* e fa riferimento ad una *Gara* INFN. Per ogni ordine, è prevista la figura di un *Responsabile Unico di Procedimento (RUP)*. L'entità è relazionata con cardinalità 1-N ad altre due molto rilevanti, *Linea d'Ordine* e *Linea Contabile*.

L'entità **Linea d'Ordine** è il modo in cui viene tenuta traccia dell'acquisto di uno o più *prodotti*[2] all'interno di un ordine. La Linea d'Ordine contiene informazioni di dettaglio per ciascun prodotto: *il prezzo unitario, la quantità, lo sconto* .... La **Linea RDA** è l'entità equivalente derivante dal tool *RDA* per le sole linee provenienti da *RDA*.

L'entità **Linea Contabile** traccia la singola spesa sostenuta dall'INFN attraverso il *saldo*. Gli attributi mostrano l'esercizio di provenienza *EPR*, l'esercizio attuale *EGE*, il *tipo di movimento* (competente o residuo) ed il *Tipo* di pratica trattata: *Impegno (IMP)*, *Pagamento (TFP)* o una *Liquidazione (LIQ)*.

Tipicamente l'importo pagato coincide con l'importo ordinato al momento della designazione dell'ordine di acquisto. Talvolta, però, può accadere che il prezzo effettivamente pagato sia differente rispetto a quanto ipotizzato in fase di inserimento dell'ordine. Tale variazione viene gestita sommando un saldo (positivo in caso di maggiorazione, negativo in caso di sconto) direttamente all'ordine, ed il dato viene tracciato in *Linea Contabile*. Ne consegue che per conoscere informazioni, quali *Pagato Senza IVA* e *Impegnato Senza IVA*, si debba attingere alle *Linea Contabile* per valutare il costo pagato o impegnato, a differenza della percentuale di tassazione che rimane relazionata alla *Linea d'Ordine*. Per ottenere questa "convergenza" di informazioni, si è deciso di sfruttare l'entità *Ordine*, essendo comune tra le due attraverso la seguente modalità:

[2] si utilizza il termine "prodotto" con la generica accezione di *bene o servizio*; ai fini dell'elaborato quindi una licenza software, un servizio di telefonia e un componente meccanico si equivalgono.

- da *Linea Contabile* si ottiene il saldo di un ordine per ogni *Tipo* di pratica (impegni o pagamenti),
- dalle *Linea d'Ordine* si ottiene la misura dell'impatto dell'IVA sul prezzo tramite il rapporto dell'*Imponibile* sul *Tot Prezzo*.

A questo punto siamo perfettamente in grado di calcolare l'imponibile sulle misure aggregate per ordine provenienti da *Linea Contabile*.

3 Progettazione DFM

Per la creazione del modello dati per gli Ordini e le Gare ai fini del Data Warehouse, come accennato nell'introduzione [Sez.1 pag.2], sono state apportate significative modifiche al processo di progettazione ed implementazione seguendo le linee guida tracciate nel libro di *Golfarelli e Rizzi*[3]. Queste vedono al centro del modello il concetto di tabella dei *fatti* di interesse per l'ente, le quali contengono le *misure* significative ai fini di indagini statistiche, mentre le tabelle *dimensione* rappresentano le coordinate di analisi del fatto stesso.

La modellazione risultante coinvolge la creazione di una ventina di tabelle, alcune delle quali prenderanno il nome di *dimensioni conformi*, per indicare la loro funzione di *"ponte"* tra i diversi fatti. Tramite queste dimensioni ponte sarà successivamente possibile navigare e correlare le informazioni provenienti anche da diversi modelli dati. Alcuni esempi di dimensioni conformi sono la tabella che mappa le sedi INFN, quella inerente alle anagrafiche del personale e la dimensione temporale la quale contiene informazioni sui giorni, mesi e anni oltre che bimestri, quadrimestri e semestri.

3.1 Descrizione del modello Ordini

A livello di progettazione del Data Warehouse si è optato per realizzare due tabelle dei *fatti*, la *Linea di Ordine* e la *Linea Contabile* [fig.2], e due *"viste"* **Ordini** e **Gare** che aggregano i dati provenienti dai fatti mettendo in relazione le informazioni del mondo contabile con quello di *RDA*.

I diagrammi a seguire sono conformi al linguaggio *DFM* [3] e riportano i fatti con corrispondenti misure e dimensioni (gli archi che partono dai fatti). Il prodotto delle cardinalità delle radici[3] delle gerarchie dimensionali[4] determinano il numero di eventi primari (record memorizzati dal fatto o della vista).

Al fine di avere una lettura efficace del diagramma *DFM* si deve considerare che all'interno della gerarchia dimensionale sussiste la relazione 1-N tra gli attributi, dove la maggior cardinalità di elementi è presente nella radice, mentre si otterranno aggregazioni via via sempre maggiori con gli attributi più distanti dal fatto. Una particolare eccezione sono gli attributi in rosso, essi sono in relazione 1-1.

[3] Attributo della dimensione collegata direttamente al fatto.
[4] Insieme degli attributi che fanno parte di una dimensione.

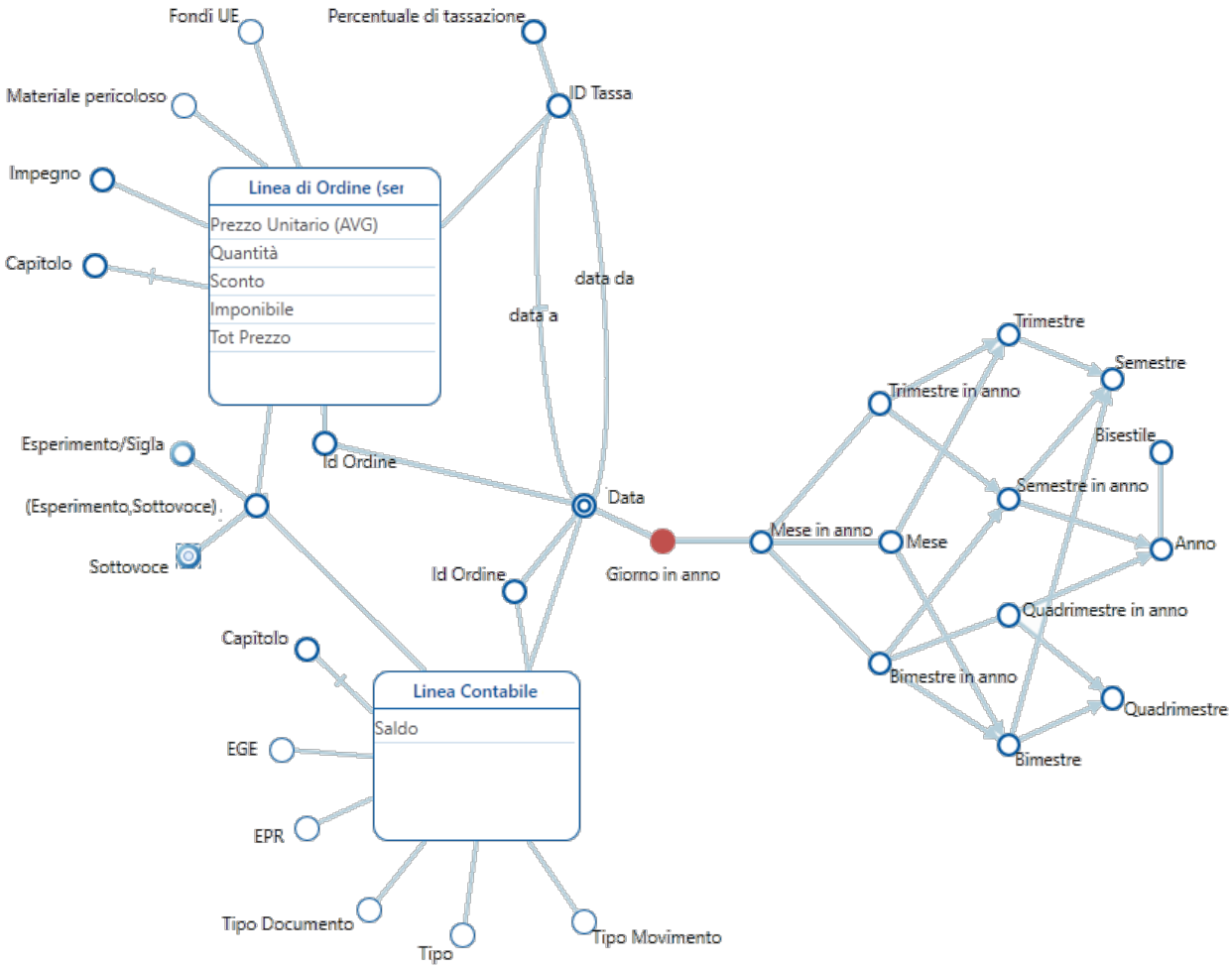

Figura 2: DFM che mostra Linee Contabili e Linee Ordini. I due Id Ordine indicano lo stesso valore

La tabella *Linea di Ordine* si relaziona alla vista *Ordini* (tramite l'identificatore dell'ordine stesso), alle dimensioni conformi inerenti a *Esperimenti*, *Tasse*, *Impegni* e *Capitoli*. Inoltre, sono presenti altre due dimensioni deformi (ovvero con un solo attributo): *Fondi UE* e *Materiale pericoloso*.

Linea contabili condivide la dimensione temporale, gli esperimenti, il capitolo e l'id ordine con *Linea di Ordine*. Anche in questo caso sono presenti anche altri attributi descritti a capitolo 2 e pagina 5.

Ordine a [fig.3] è una vista. Le viste sono tabelle che aggregano informazioni a partire da un fatto, quindi contengono le stesse misure del fatto ma la granularità analizzata differisce, perché le radici di dimensioni collegate sono un'aggregazione rispetto a quelle dei fatti. Ordine è un caso peculiare in quanto è ottenuta dall'aggregazione di due diversi fatti, i quali combinati permettono di ottenere nuovi e fondamentali informazioni. Le misure presenti sono: *Prezzo Totale, Imponibile, Sconto, Impegnato, Impegnato al netto IVA, Pagato e Pagato al netto IVA*. Tale caratteristica è un esigenza derivata da un proble-

ma di integrazione tra Linee di Ordine con Linee Contabili come descritto a capitolo [2] e pagina [5].

Aggregando da Ordine si ottiene la vista *Gara* per avere una tabella più vicina alle esigenze di alcune query. Dato che la cardinalità di un fatto/vista deriva dalle dimensioni con cui è correlata, si può notare a [fig.3] che gara non ha come unica dimensione CIG, bensì anche Anno e CIG Principale. Perciò per individuare una misura che si relazioni al solo CIG è sufficiente effettuare un raggruppamento per CIG, senza considerare perciò le restanti dimensioni.

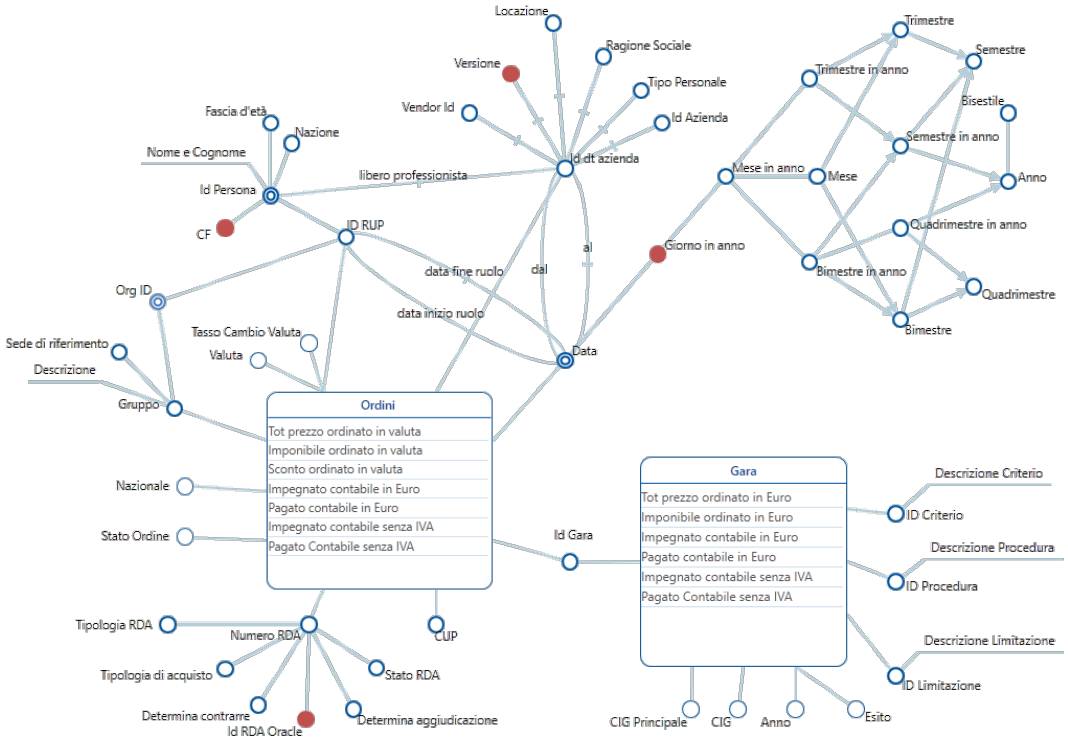

Figura 3: DFM Le viste Ordini e Gare

4 Implementazione del Modello Ordini in ETL

L'*ETL (Extract Transform Load)* è un processo che si occupa di estrarre i dati dalle sorgenti, trasformarli e ripulirli secondo determinate logiche, ed infine riversarli nel *data warehouse*, creando di fatto la struttura fisica del modello dati. La progettazione degli *step*, che compongono la *pipeline* di elaborazione, è una delle fasi più critiche nella creazione di un nuovo modello. Spesso infatti è un'operazione che richiede tempo ed è caratterizzata da diverse attività di *debug* e *tuning*. Per complessità risulta seconda solo alla fase di analisi qualora non si disponga di documentazione e si renda necessario ripartire da zero e ricostruire tutto il *know how* necessario.

Lo strumento di progettazione utilizzato è Tibco Jaspersoft ETL [2] che, nella versione "Professional v7.3.1", risente di diversi *bug* legati alla mappatura e conversione dei tipi di dato da NUMBER (Oracle) a BigDecimal (Java), tipi di dato ampiamente utilizzati nelle tabelle legate al bilancio INFN.

La struttura di una tipica pipeline di ETL comprende le seguenti fasi:

$$\textbf{\textit{Import}} \Rightarrow \textbf{\textit{Preparatory}} \Rightarrow \textbf{\textit{Staging}} \Rightarrow \textbf{\textit{Datamart}}$$

La fase di **Import** è la fase di copia dei dati necessari dalle sorgenti . Nella fase di **Preparatory** i dati vengono ripuliti ed elaborati. Durante le operazioni di **Staging** vengono predisposte le risorse nella loro forma definitiva e poste in un'area dedicata. Infine, durante la fase di **Datamart** le tabelle pronte vengono copiate dall'area di staging alla posizione finale, per essere visualizzate dagli applicativi di *front-end*.

Uno degli aspetti su cui fare maggiore attenzione è l'analisi delle dipendenze tra le varie tabelle e le fasi della pipeline.

In figura [fig.4] viene mostrato l'albero delle dipendenze che porta alla costruzione del modello Ordini. Il diagramma si legge a partire dal nodo nero al centro in alto, proseguendo verso le foglie in basso seguendo le frecce. Un nodo intermedio dipende dai nodi che lo precedono nel path, così le tabelle *DT_RUP* e *DT_AZIENDE* sono tra loro indipendenti e possono essere create in parallelo. D'altra parte nel ramo di destra la tabella *DT_GARE* dipende dal corretto esito di tutte le operazioni che la precedono.

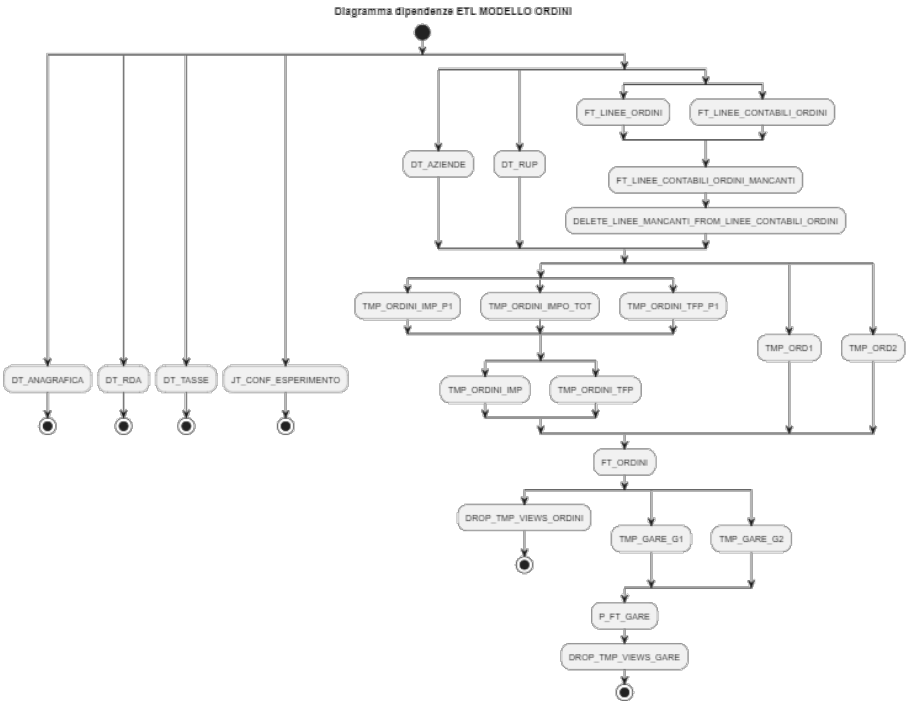

Figura 4: Albero delle dipendenze realizzato con PlantUML (2.0)[1]

L'intero diagramma del processo di ETL risulterebbe inutilmente complesso, perciò la figura [fig.5] si limita a mostrarne la parte che implementa il sotto-albero di destra della precedente [fig.4].

In particolare, vengono messi in evidenza in rosso i processi di creazione delle tabelle ed in blu i componenti che consentono di gestire l'esecuzione parallela di parti della pipeline. In questo secondo caso, i rami vengono contrassegnati con *Parallelize* per i processi che possono essere eseguiti parallelamente e con *Synchronize (Wait for All)*, il cui inizio di elaborazione deve attendere la terminazione dei flussi paralleli (similmente ad una barriera[5]).

[5]Elemento di sincronizzazione che attende la fine di tutti i processi entranti per eseguire i passi successivi.

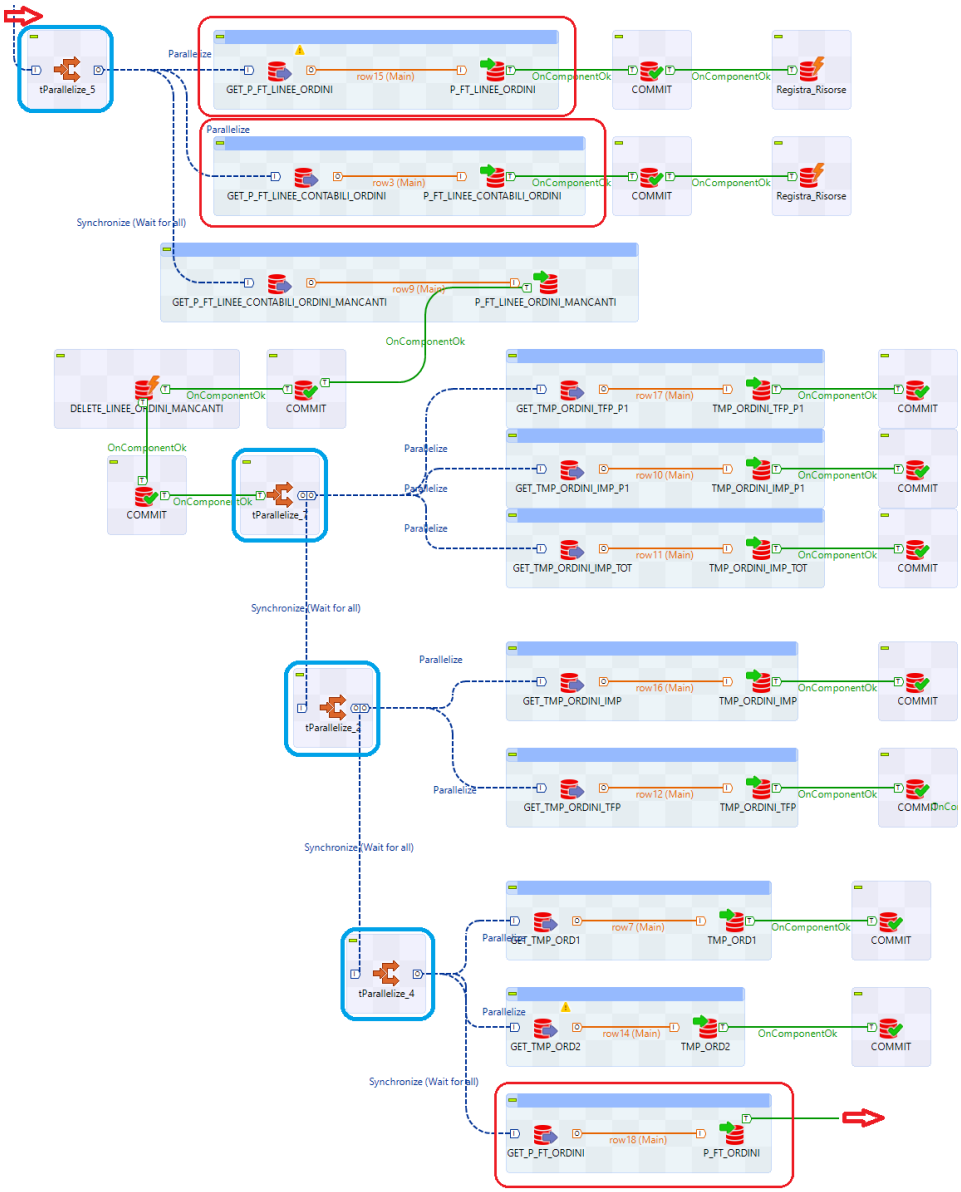

Figura 5: Frammento del diagramma di pipeline dell'ETL

5 Conclusioni

Il lavoro svolto presenta diversi benefici rispetto a quanto fatto in passato su altri modelli dati.

Innanzi tutto il processo di analisi, finalizzato alla creazione del modello E/R, ci ha consentito di recuperare il *know-how* sia sui dati sia sull'evoluzione nel tempo degli applicativi coinvolti.

Lo schema E/R si è reso anche molto utile nell'identificare tutte quelle informazioni statiche che potevano essere gestite con tabelle di *lookup*[6]. L'effetto positivo di questa razionalizzazione è un uso più efficiente dello spazio evitando inutili duplicazioni. Allo stesso modo questa operazione ha consentito di alleggerire anche la fase di implementazione dei job di ETL, concentrandosi sulle informazioni che realmente necessitavano di attenzione.

Anche il diagramma *DFM* presenta diverse caratteristiche particolarmente vantaggiose. Il primo punto di forza di questo formalismo è sicuramente la facilità di interpretazione che lo rende fruibile anche per chi non dovesse avere particolari competenze in ambito informatico. Un secondo elemento è dato dalla compattezza e ricchezza espressiva, che lo rende particolarmente adatto a documentare e condividere il *know-how* sui dati. Infine, pone l'attenzione sulle misure realmente di interesse per chi deve analizzare i dati, fornendo anche una chiave di lettura semplice delle possibili aggregazioni[4].

Tutte queste caratteristiche determinano una maggiore efficienza del modello dati prodotto e una maggiore leggibilità della documentazione a corredo.

[6] un esempio tipico è l'elenco delle codifiche che identificano le sedi INFN

Riferimenti bibliografici

[1] ®©2023 PlantUML. Plant uml v2.0.

[2] ®©2023 TIBCO. Tibco jaspersoft etl.

[3] S. R. Matteo Golfarelli. *Data™Warehouse: Teoria e pratica della progettazione*. In M.-H. Education, editor, *book*, pages 1–448. McGraw-Hill Education (Italy) S.r.l, Jan. 2006. Un manuale completo e aggiornato per la progettazione di un Data Warehouse.

[4] I.-Y. Song, M. Evans, and E. K. Park. A comparative analysis of entity-relationship diagrams. *Journal of Computer and Software Engineering*, 3, 01 1995.