

# **Tesina per il corso di “Seminari di Ingegneria del Software”**

Luigi Tullio

## Integrazione dei dati e Mapping

### ***La problematica dell'integrazione***

Il problema della *integrazione di dati* consiste nel voler combinare dati provenienti da sorgenti differenti fornendo all'utente una visione unificata.

Nella prima fase c'è il problema di individuare le corrispondenze tra concetti simili provenienti dalle diverse fonti e la risoluzione dei conflitti.

Un'ipotesi di lavoro è quella di adottare uno stesso modello dei dati e quindi avere uno schema globale. Lo schema globale deve fornire una visione iconciliata, integrata e virtuale delle sorgenti da integrare.

Quando si affronta la problematica dell'integrazione di più fonti di dati in un unico schema globale sorge la necessità di avere un meccanismo generale per rappresentare le correlazioni tra lo schema globale e gli schemi locali attraverso una sintassi che consenta di tradurre in modo automatico le query sullo schema globale in query sugli schemi locali, specificando anche come realizzare l'unione dei risultati parziali ottenuti dalle varie sorgenti.

In questo contesto, è necessario definire il sistema di integrazione attraverso *Mapping* formali tra le sorgenti.

E' possibile specificare tre tipi di mapping:

- *Mapping GAV (Global as View)*: lo schema globale è definito in termini di viste sugli schemi sorgente (locali).
- *Mapping LAV (Local as View)*: gli schemi locali sono definiti in termini di viste sullo schema globale.
- *Mapping GLAV (Global and Local as View)*: La relazione tra sorgenti e schema globale è stabilita attraverso un insieme di view, alcune sullo schema globale e alcune sulle sorgenti.

Indipendentemente dal tipo di mapping il sistema di integrazione deve fornire i seguenti servizi:

- Risponde alle query, che vengono poste in termini di schema globale

- Richiede uno step di riformulazione delle query (comunque formulate dall'utente) in termini di query sulle sorgenti

### **Formalizzazione del mapping**

Un sistema di integrazione di dati è una tripla  $\langle G; S; M \rangle$ , dove:

- **G** è lo schema globale.
- **S** è lo schema sorgente.
- **M** è il mapping tra S e G.

Un mapping  $M$  tra sorgenti e schema globale è costituito da un insieme di asserzioni:

$$q_S \rightarrow q_G$$

$$q_G \rightarrow q_S$$

In pratica, la prima asserzione specifica che il concetto rappresentato da una query  $q_S$  sullo schema di una sorgente  $S$  corrisponde al concetto specificato da  $q_G$  sullo schema globale. Viceversa per la seconda.

Le query al sistema integrato vengono poste in termini di  $G$  e specificano quali dati vogliamo estrarre dalla base di dati virtuale rappresentata dallo schema  $G$

### **Mapping GAV**

Un mapping GAV è un insieme di asserzioni, una per ogni elemento  $g$  di  $G$

$$g \rightarrow q_S$$

Si osserva che il mapping ci dice esplicitamente come valutare la query.

Questo tipo di mapping è adeguato se le sorgenti sono stabili, infatti diventa difficile estendere con una nuova sorgente, che può avere un notevole impatto sulla definizione dei vari elementi dello schema globale.

Con il mapping GAV, una query, sul globale, è elaborata effettuando un unfolding cioè espandendo gli atomi della query secondo le loro definizioni iterativamente fino ad avere una query in cui compaiono solo relazioni della sorgente.

## **Mapping LAV**

Un mapping LAV è un insieme di asserzioni, una per ogni elemento  $s$  di  $S$

$$s \rightarrow q_G$$

In questo caso il contenuto di ogni sorgente è caratterizzato in termini di una vista  $q_G$  sullo schema globale.

Questo mapping è adatto se lo schema globale è stabile (ad es. basato su un modello d'impresa o su un'ontologia), inoltre favorisce l'estensibilità del sistema infatti estendere con una nuova sorgente vuol semplicemente dire aggiungere una nuova asserzione (vista sul globale). Il problema del LAV è la complessità nel processare le query dovuto al fatto che la riscrittura della query è più complicata. E' necessario utilizzare un meccanismo inferenziale capace di riesprimere gli atomi dello schema globale in termini di atomi delle sorgenti. Nell'utilizzo dei mapping LAV si suppone di avere già uno schema globale che contiene le informazioni di interesse, e l'obiettivo diventa quello di far corrispondere i dati nelle sorgenti coinvolte ai dati che ci si aspetta dallo schema globale. Questo è il procedimento inverso del mapping GAV.

## **Mapping GLAV**

Il mapping GLAV può essere visto come un approccio misto GAV e LAV, è definito da un'insieme di asserzioni del tipo

$$q_S \rightarrow q_G$$

Praticamente il mapping avviene tra viste ( $q_G$ ) definite sullo schema globale e viste ( $q_S$ ) definite su quello locale.

Similmente al LAV questo tipo di mapping non dà delle informazioni dirette su come i dati soddisfano lo schema globale, ovvero per rispondere ad una query sullo schema globale bisogna inferire dal mapping le query che accedono allo schema sorgente.

## **GAV vs LAV**

Caratteristiche di GAV:

- La qualità dipende da quanto bene abbiamo mappato le sorgenti nello schema globale.

- Ognivolta che una sorgente cambia o viene aggiunta bisogna riconsiderare lo schema globale e il mapping.
- Il processamento delle query può essere basato sull'unfolding, quindi la riformulazione delle query è semplice.

Caratteristiche di LAV:

- La qualità dipende da quanto bene abbiamo caratterizzato le sorgenti.
- Alta modularità ed estendibilità, infatti se lo schema globale è stato progettato bene, quando una sorgente cambia si deve modificare solo il mapping di quella sorgente.
- Il processamento delle query non è semplice.

## Lavoro svolto

Il nostro compito consisteva nel realizzare l'integrazione di dati mediante mapping GAV e LAV su sei casi diversi e produrre della documentazione che analizzava questo lavoro.

Nella pratica abbiamo preso quattro specifiche di quattro compiti di esami di basi di dati e le abbiamo sviluppate separatamente per realizzare altrettanti schemi concettuali, l'intenzione era di poter ottenere rispettivamente due sorgenti di dati sullo stesso dominio di interesse ma realizzate, almeno in parte, in modo differente. Il passo successivo è stato di realizzare un terzo schema in modo congiunto, che rappresentava lo schema globale. Su queste prime quattro specifiche gli schemi realizzati separatamente sono risultati pressappoco identici, questo è dovuto al fatto che le specifiche che avevamo erano molto dettagliate e lasciavano poco ad interpretazioni differenti. Questo ha fatto sì che il mapping di questi casi è stato molto semplice. Visto tutto ciò, successivamente abbiamo deciso di realizzare le ultime due rimanenti integrazioni su database realizzati a partire da nostre specifiche volutamente rese molto generiche. Questo effettivamente ha portato a realizzare schemi logici differenti tra loro e quindi ad avere delle sorgenti adeguatamente differenti. In'oltre nella realizzazione dello schema globale di quest'ultimi due casi abbiamo seguito due approcci diversi. Precisamente in un caso abbiamo realizzato uno schema globale molto dettagliato, nell'altro caso abbiamo seguito la filosofia inversa. In questo modo è stato possibile vedere sia per il GAV che per il LAV tutte le problematiche che si possono riscontrare utilizzando uno di questi mapping. Infatti quando lo schema globale è molto dettagliato, rispetto le sorgenti, il mapping GAV presenta molte difficoltà nell'essere applicato, al contrario il LAV diventa molto più semplice. Rispettivamente succede l'opposto se lo schema globale è meno dettagliato delle sorgenti.

La documentazione di tutto il lavoro e delle conclusioni è stato realizzato separatamente ognuno su tre dei sei casi, due di quelli "semplici" e 1 uno di quelli complessi.

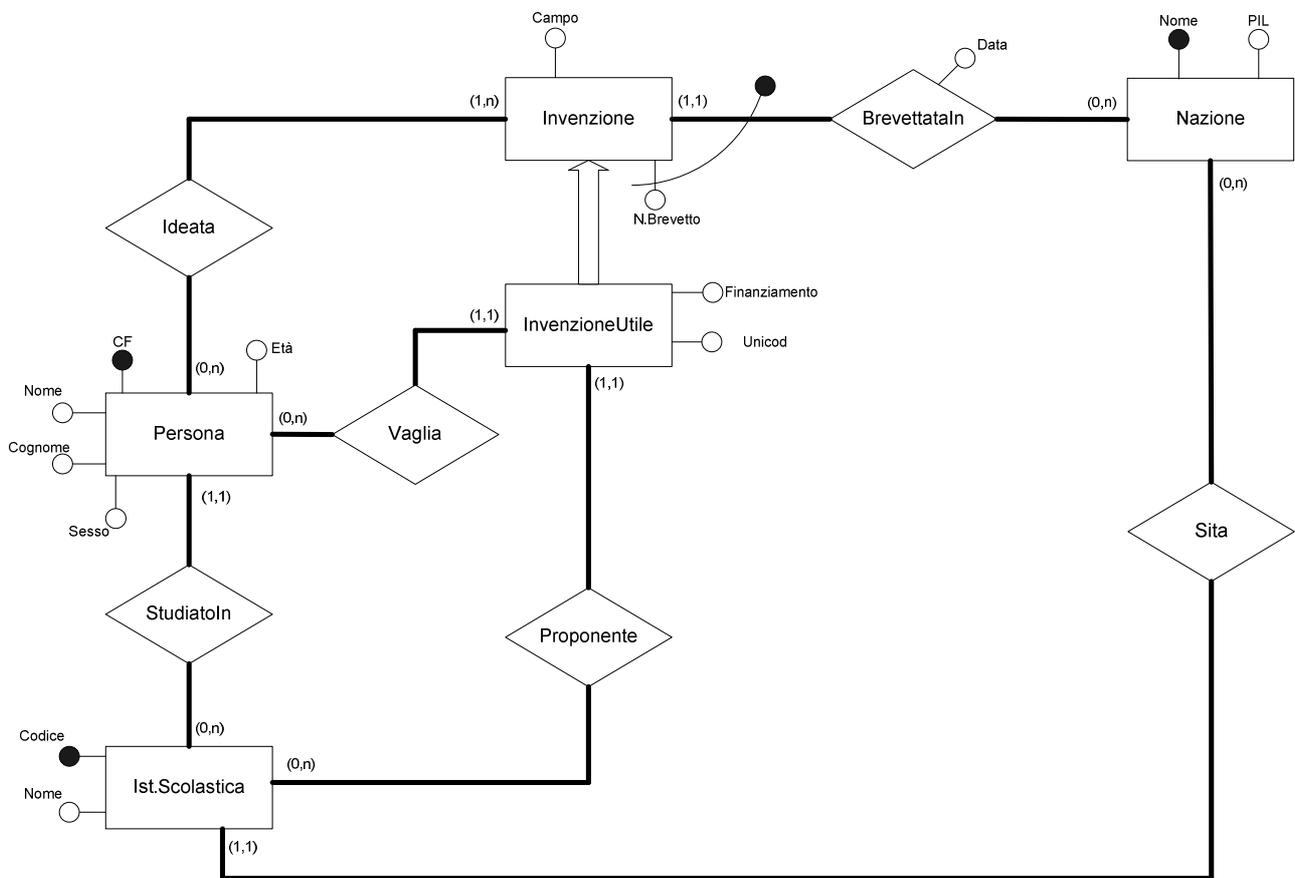
Nel mio caso ho realizzato e approfondito, tra l'altro, il caso in cui lo schema globale era meno dettagliato e quindi ho dovuto soffermarmi di più sul mapping LAV.

## Mapping DB 3 (Invenzioni brevettate)

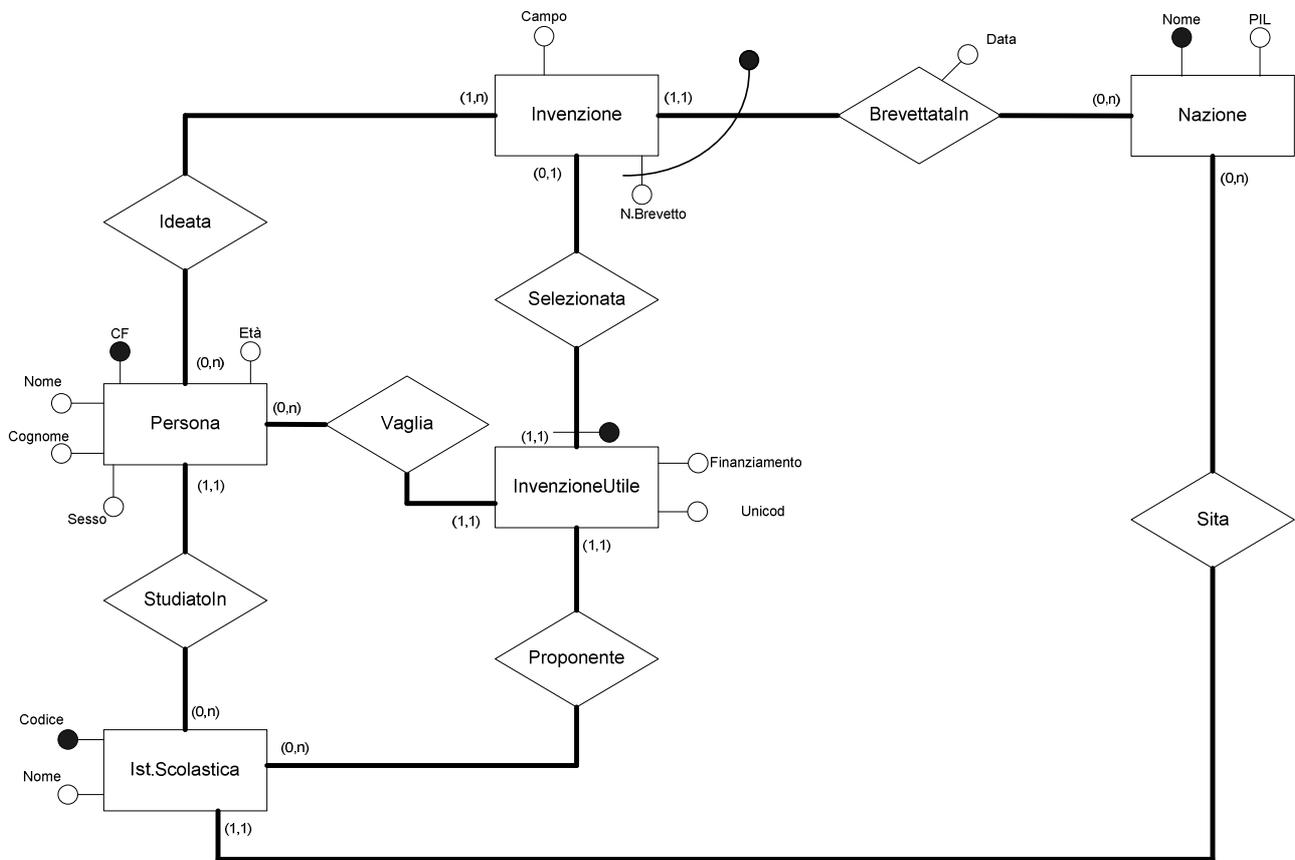
### **Specifiche:**

Si richiede di progettare un DB di un'applicazione relativa alle invenzioni brevettate. Ogni invenzione viene presentata da una o più persone ad un ufficio brevetti di una nazione, e di ogni invenzione interessa sapere: le persone che l'hanno presentata, la nazione (non l'ufficio) alla quale l'hanno presentata, la data di presentazione, ed il campo di interesse dell'invenzione stessa (medicina, ingegneria, svago, ecc.), ed il numero di brevetto assegnato (unico nell'ambito della nazione). Di ogni nazione interessa il nome (identificativo) ed il PIL (prodotto interno lordo). Di ogni persona interessa il codice fiscale (identificativo), il nome, il cognome, l'età, il sesso e l'istituzione scolastica nella quale ha ottenuto il diploma di istruzione (ogni persona ha esattamente un diploma di istruzione). Di ogni istituzione scolastica interessa il codice (identificativo), il nome e la nazione in cui si trova. Alcuni invenzioni, una volta presentate, vengono classificate come "utili", a seguito della proposta di una istituzione scolastica, proposta vagliata da una persona. Di ogni invenzione utile interessa: il codice (unico nell'ambito delle invenzioni utili), il finanziamento ottenuto (ogni invenzione utile ne riceve uno), l'istituzione scolastica (una ed una sola) proponente, e la persona che ha vagliato la proposta.

### Diagramma ER – DB 3 – Globale



### Diagramma ER Ristrutturato – DB 3 – Globale



### Schema logico – DB 3 – Globale

Invenzione(**NBrevetto, NomeNazione**, Campo, Data)  
*foreign key: Invenzione[NomeNazione]  $\subseteq$  Nazione[Nome]*

Nazione(**Nome**, PIL)

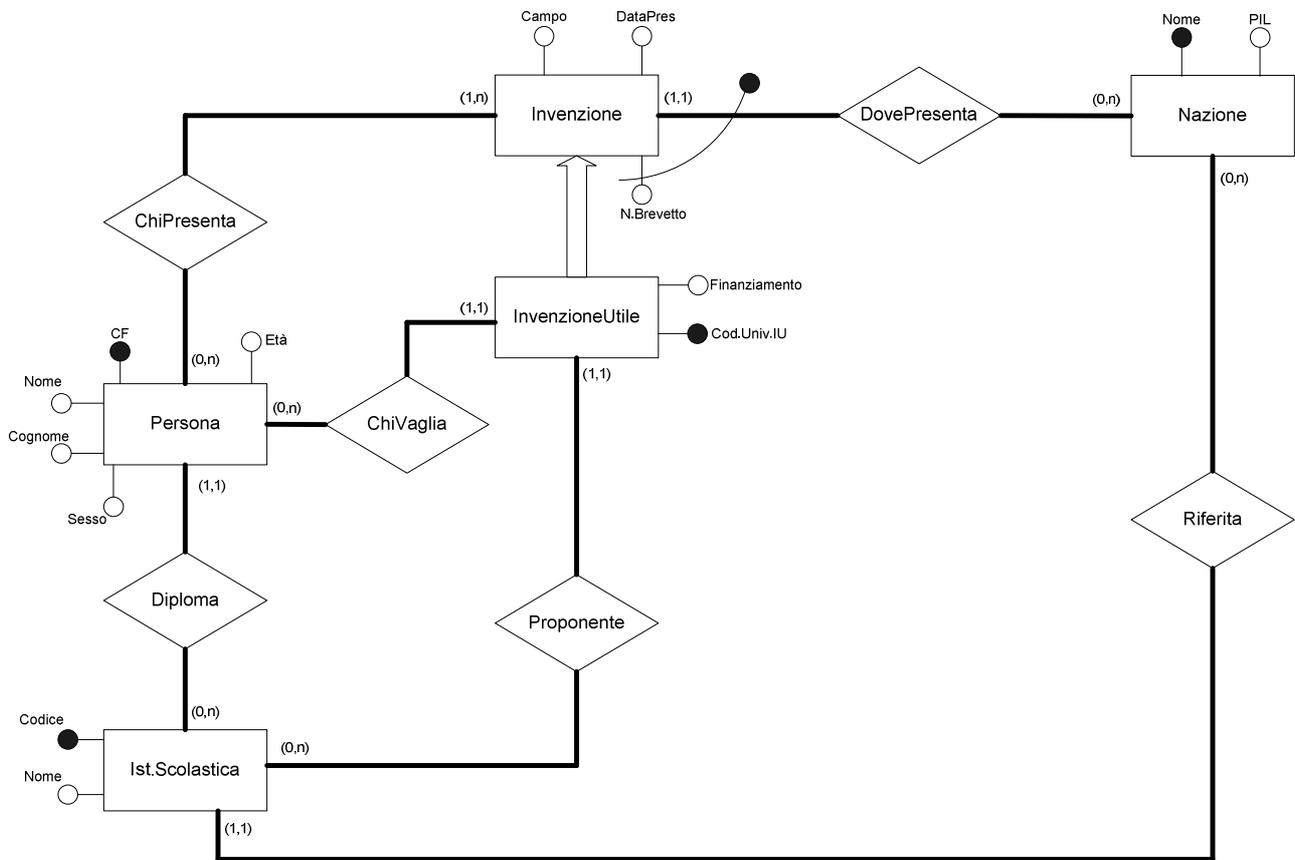
Persona(**CF**, Nome, Cognome, Sesso, Età, CodScuola)  
*foreign key: Persona[CodScuola]  $\subseteq$  IstScolastico[Codice]*

InvenzioneUtile(Unicod, **NBrevetto, NomeNazione**, Finanziamento, CFVaglia, IstScolastico)  
*foreign key: InvenzioneUtile[NBrevetto]  $\subseteq$  Invenzione[NBrevetto]*  
*foreign key: InvenzioneUtile[CFVaglia]  $\subseteq$  Persona[CF]*  
*foreign key: InvenzioneUtile[IstScolastico]  $\subseteq$  IstScolastico[Codice]*

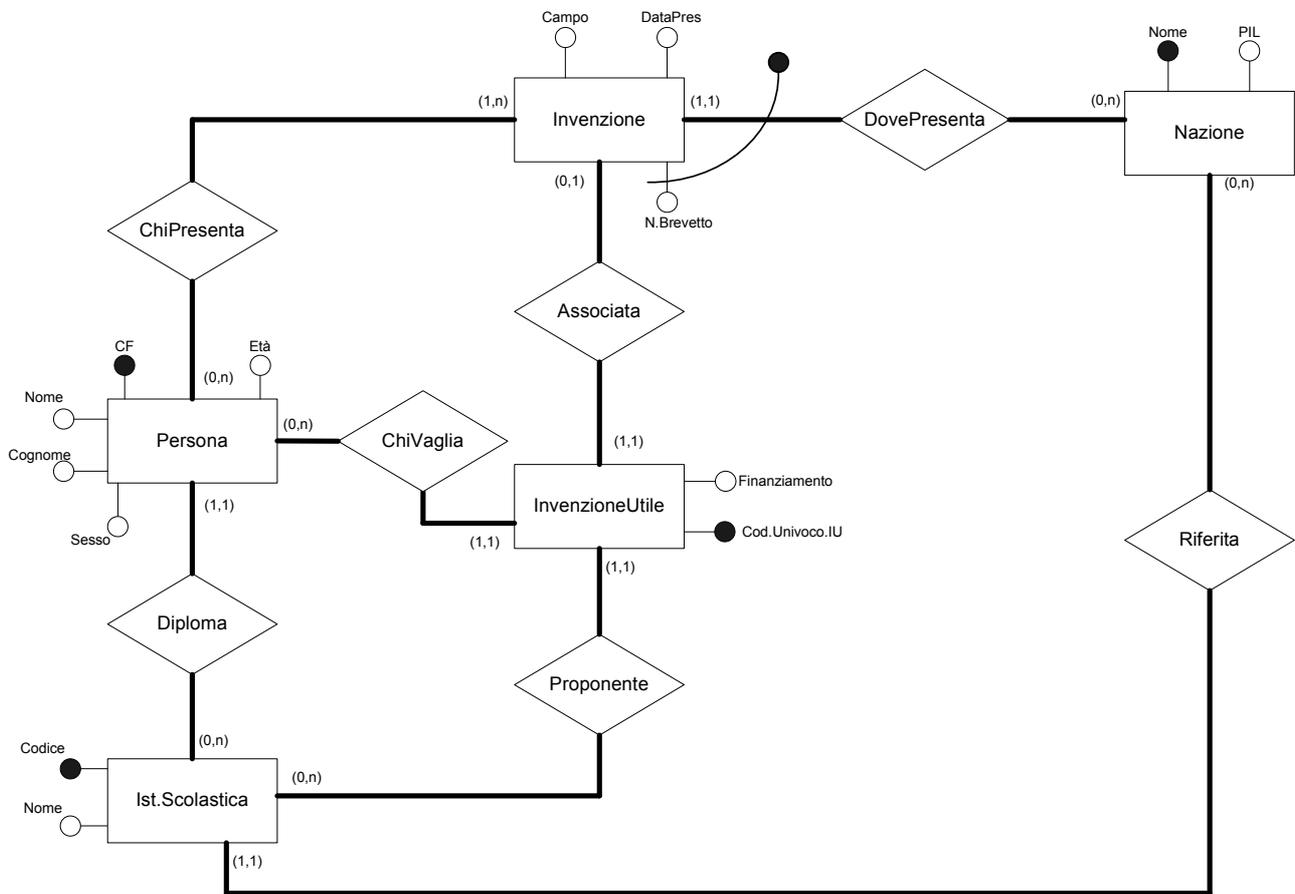
IstScolastico(**Codice**, Nome, NomeNazione)  
*foreign key: IstScolastico[NomeNazione]  $\subseteq$  Nazione[Nome]*

Ideata(**CF, NBrevetto, NomeNazione**)  
*foreign key: Ideata[CF]  $\subseteq$  Persona[CF]*  
*foreign key: Ideata[NBrevetto, NomeNazione]  $\subseteq$  Invenzione[NBrevetto, NomeNazione]*

### Diagramma ER – DB 3 – Sorgente A



### Diagramma ER Ristrutturato – DB 3 – Sorgente A



### Schema logico – DB 3 – Sorgente A

Invenzione(**NBrevetto, NomeNazione**, Campo, DataPres)  
*foreign key: Invenzione[NomeNazione]  $\subseteq$  Nazione[Nome]*

Nazione(**Nome**, PIL)

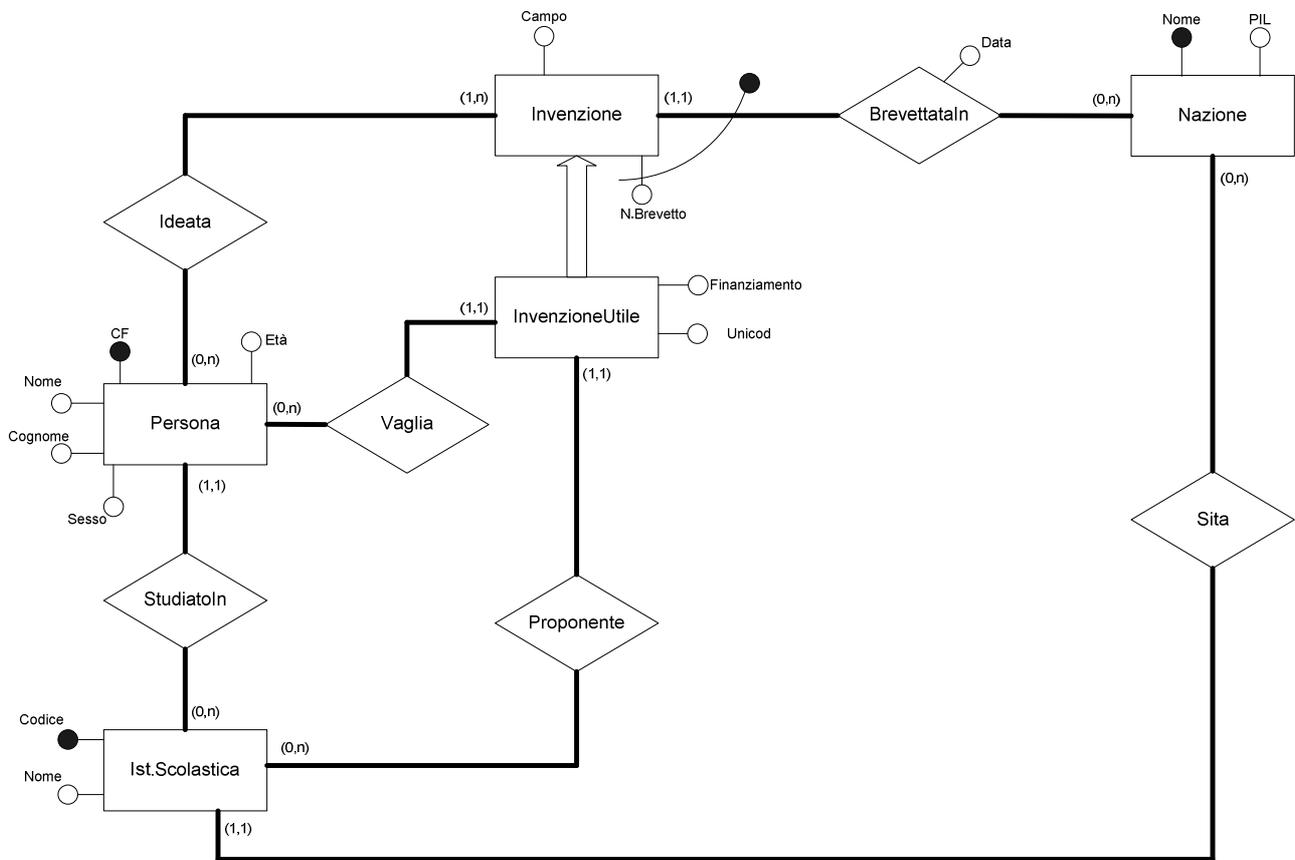
Persona(**CF**, Nome, Cognome, Sesso, Età, CodIstScol)  
*foreign key: Persona[CodIstScol]  $\subseteq$  IstScolastico[Codice]*

InvenzioneUtile(**Cod.Univoco.IU, NBrevetto, NomeNazione**, Finanziamento, CF, CodicelstScolastico)  
*foreign key: InvenzioneUtile[NBrevetto]  $\subseteq$  Invenzione[NBrevetto]*  
*foreign key: InvenzioneUtile[CF]  $\subseteq$  Persona[CF]*  
*foreign key: InvenzioneUtile[CodicelstScolastico]  $\subseteq$  IstScolastico[Codice]*

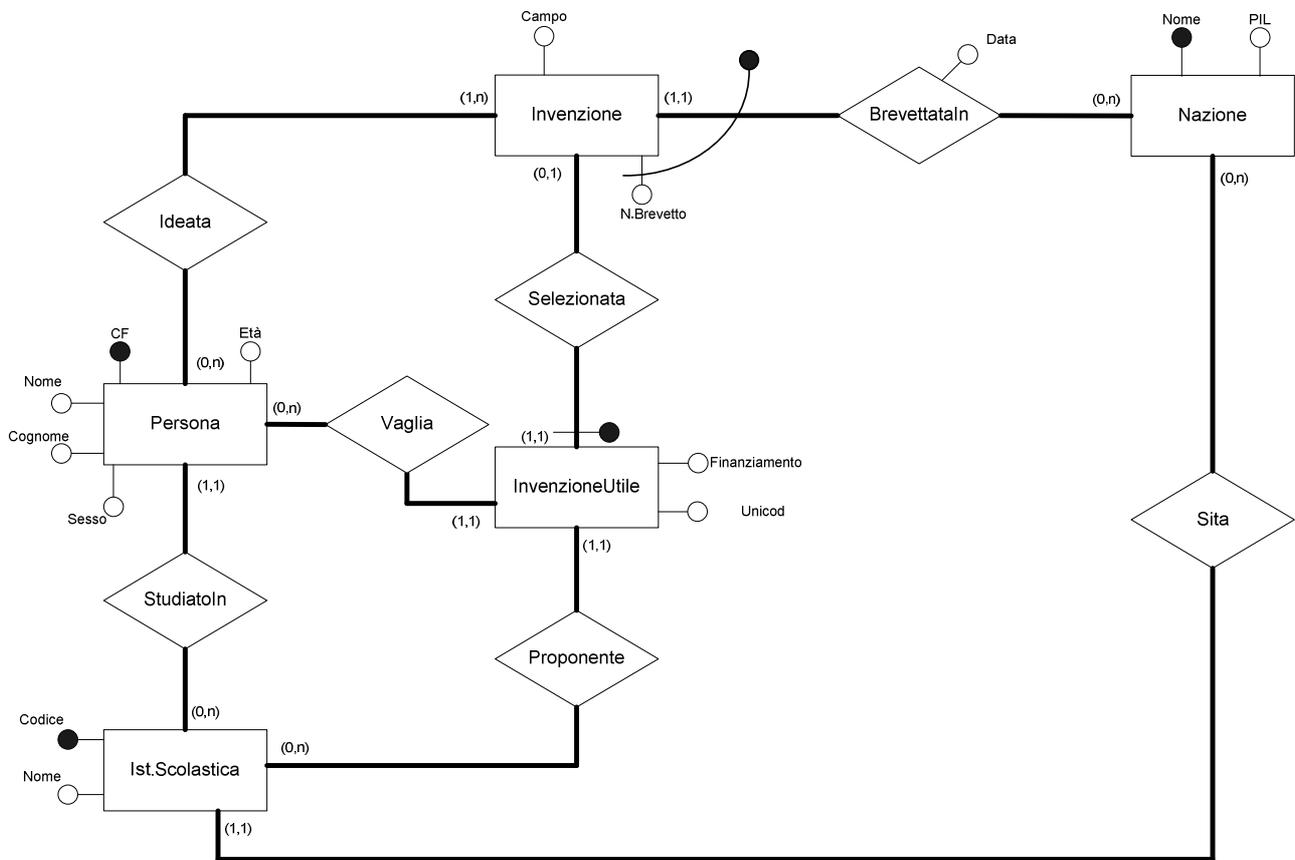
IstScolastico(**Codice**, Nome, NomeNazione)  
*foreign key: IstScolastico[NomeNazione]  $\subseteq$  Nazione[Nome]*

ChiPresenta(**CF, NBrevetto, NomeNazione**)  
*foreign key: ChiPresenta[CF]  $\subseteq$  Persona[CF]*  
*foreign key: ChiPresenta[NBrevetto, NomeNazione]  $\subseteq$  Invenzione[NBrevetto, NomeNazione]*

### Diagramma ER – DB 3 – Sorgente B



**Diagramma ER Ristrutturato – DB 3 – Sorgente B**



### **Schema logico – DB 3 – Sorgente B**

Invenzione(**NBrevetto, NomeNazione**, Campo, DataPresentazione)  
*foreign key: Invenzione[NomeNazione]  $\subseteq$  Nazione[Nome]*

Nazione(**Nome**, PIL)

Persona(**CF**, Nome, Cognome, Sesso, Età, CodScuola)  
*foreign key: Persona[CodScuola]  $\subseteq$  IstScolastica[Codice]*

InvenzioneUtile(Unicod, **NBrevetto, NomeNazione**, Finanziamento, CFVaglia, IstScolastico)  
*foreign key: InvenzioneUtile[NBrevetto]  $\subseteq$  Invenzione[NBrevetto]*  
*foreign key: InvenzioneUtile[CFVaglia]  $\subseteq$  Persona[CF]*  
*foreign key: InvenzioneUtile[IstScolastico]  $\subseteq$  IstScolastico[Codice]*

IstScolastico(**Codice**, Nome, NomeNazione)  
*foreign key: IstScolastico[NomeNazione]  $\subseteq$  Nazione[Nome]*

Ideata(**CF, NBrevetto, NomeNazione**)  
*foreign key: Ideata[CF]  $\subseteq$  Persona[CF]*  
*foreign key: Ideata[NBrevetto, NomeNazione]  $\subseteq$  Invenzione[NBrevetto, NomeNazione]*

### Mapping GAV – DB 3

Dato che le sorgenti e lo shema globale sono molto simili il mapping è stato diretto.

---

Globale:

Invenzione(**NBrevetto**, **NomeNazione**, Campo, Data)

Sorgente A:

Invenzione(**NBrevetto**, **NomeNazione**, Campo, DataPres)

Sorgente B:

Invenzione(**NBrevetto**, **NomeNazione**, Campo, DataPresentazione)

---

**[Invenzione,        Invenzione(X, Y, Z, K) :- InvenzioneA(X, Y, Z, K)**  
**Invenzione(X, Y, Z, K) :- InvenzioneB(X, Y, Z, K) ]**

---

Globale:

Nazione(**Nome**, PIL)

Sorgente A:

Nazione(**Nome**, PIL)

Sorgente B:

Nazione(**Nome**, PIL)

---

**[Nazione,        Nazione(X, Y) :- NazioneA(X, Y)**  
**Nazione(X, Y) :- NazioneB(X, Y) ]**

---

Globale:

Persona(**CF**, Nome, Cognome, Sesso, Età, CodScuola)

Sorgente A:

Persona(**CF**, Nome, Cognome, Sesso, Età, CodIstScol)

Sorgente B:

Persona(**CF**, Nome, Cognome, Sesso, Età, CodScuola)

---

**[Persona,        Persona(X, Y, Z, K, A, B) :- PersonaA(X, Y, Z, K, A, B)**  
**Persona(X, Y, Z, K, A, B) :- PersonaB(X, Y, Z, K, A, B) ]**

---

Globale:

InvenzioneUtile(Unicod, **NBrevetto**, **NomeNazione**, Finanziamento, CFVaglia, IstScolastico)

Sorgente A:

InvenzioneUtile(**Cod.Univoco.IU**, **NBrevetto**, **NomeNazione**, Finanziamento, CF,  
CodiceIstScolastico)

Sorgente B:

InvenzioneUtile(Unicod, **NBrevetto**, **NomeNazione**, Finanziamento, CFVaglia,  
IstScolastico)

**[InvenzioneUtile,**

**InvenzioneUtile(X, Y, Z, K, A, B) :- InvenzioneUtileA(X, Y, Z, K, A, B)**

**InvenzioneUtile(X, Y, Z, K, A, B) :- InvenzioneUtileB(X, Y, Z, K, A, B) ]**

---

Globale:

IstScolastico(**Codice**, Nome, NomeNazione)

Sorgente A:

IstScolastico(**Codice**, Nome, NomeNazione)

Sorgente B:

IstScolastico(**Codice**, Nome, NomeNazione)

**[IstScolastico,**

**IstScolastico(X, Y, Z) :- IstScolasticoA(X, Y, Z)**

**IstScolastico(X, Y, Z) :- IstScolasticoB(X, Y, Z) ]**

---

Globale:

Ideata(**CF**, **NBrevetto**, **NomeNazione**)

Sorgente A:

ChiPresenta(**CF**, **NBrevetto**, **NomeNazione**)

Sorgente B:

Ideata(**CF**, **NBrevetto**, **NomeNazione**)

**[Ideata,**

**Ideata(X, Y, Z) :- ChiPresentaA(X, Y, Z)**

**Ideata(X, Y, Z) :- IdeataB(X, Y, Z) ]**

### **Mapping LAV – DB 3 – Sorgente A**

Dato che le sorgenti e lo schema globale sono molto simili il mapping è stato diretto.

---

Globale:

Invenzione(**NBrevetto**, **NomeNazione**, Campo, Data)

Sorgente A:

Invenzione(**NBrevetto**, **NomeNazione**, Campo, DataPres)

**[Invenzione, Invenzione(X, Y, Z, K) :- Invenzione(X, Y, Z, K) ]**

---

Globale:

Nazione(**Nome**, PIL)

Sorgente A:

Nazione(**Nome**, PIL)

**[Nazione, Nazione(X, Y) :- Nazione(X, Y) ]**

---

Globale:

Persona(**CF**, Nome, Cognome, Sesso, Età, CodScuola)

Sorgente A:

Persona(**CF**, Nome, Cognome, Sesso, Età, CodIstScol)

**[Persona, Persona(X, Y, Z, K, A, B) :- Persona(X, Y, Z, K, A, B) ]**

---

Globale:

InvenzioneUtile(Unicod, **NBrevetto**, **NomeNazione**, Finanziamento, CFVaglia, IstScolastico)

Sorgente A:

InvenzioneUtile(**Cod.Univoco.IU**, **NBrevetto**, **NomeNazione**, Finanziamento, CF, CodicelstScolastico)

**[InvenzioneUtile, InvenzioneUtile(X, Y, Z, K, A, B) :- InvenzioneUtile(X, Y, Z, K, A, B) ]**

---

Globale:

IstScolastico(**Codice**, Nome, NomeNazione)

Sorgente A:

IstScolastico(Codice, Nome, NomeNazione)

**[IstScolastico, IstScolastico(X, Y, Z) :- IstScolastico(X, Y, Z) ]**

---

Globale:

Ideata(CF, NBrevetto, NomeNazione)

Sorgente A:

ChiPresenta(CF, NBrevetto, NomeNazione)

**[Invenzione, Invenzione(X, Y, Z) :- Invenzione(X, Y, Z) ]**

### **Mapping LAV – DB 3 – Sorgente B**

Dato che le sorgenti e lo schema globale sono molto simili il mapping è stato diretto.

---

Globale:

Invenzione(**NBrevetto**, **NomeNazione**, Campo, Data)

Sorgente B:

Invenzione(**NBrevetto**, **NomeNazione**, Campo, DataPresentazione)

**[Invenzione,            Invenzione(X, Y, Z, K) :- Invenzione(X, Y, Z, K) ]**

---

Globale:

Nazione(**Nome**, PIL)

SorgenteB:

Nazione(**Nome**, PIL)

**[Nazione,            Nazione(X, Y) :- Nazione(X, Y) ]**

---

Globale:

Persona(**CF**, Nome, Cognome, Sesso, Età, CodScuola)

Sorgente B:

Persona(**CF**, Nome, Cognome, Sesso, Età, CodScuola)

**[Persona,            Persona(X, Y, Z, K, A, B) :- Persona(X, Y, Z, K, A, B) ]**

---

Globale:

InvenzioneUtile(Unicod, **NBrevetto**, **NomeNazione**, Finanziamento, CFVaglia, IstScolastico)

Sorgente B:

InvenzioneUtile(Unicod, **NBrevetto**, **NomeNazione**, Finanziamento, CFVaglia, IstScolastico)

**[InvenzioneUtile,            InvenzioneUtile(X, Y, Z, K, A, B) :- InvenzioneUtile(X, Y, Z, K, A, B) ]**

---

Globale:

IstScolastico(**Codice**, Nome, NomeNazione)

Sorgente B:  
IstScolastico(Codice, Nome, NomeNazione)

**[IstScolastico, IstScolastico(X, Y, Z) :- IstScolastico(X, Y, Z) ]**

---

Globale:  
Ideata(CF, NBrevetto, NomeNazione)

Sorgente A:  
Ideata(CF, NBrevetto, NomeNazione)

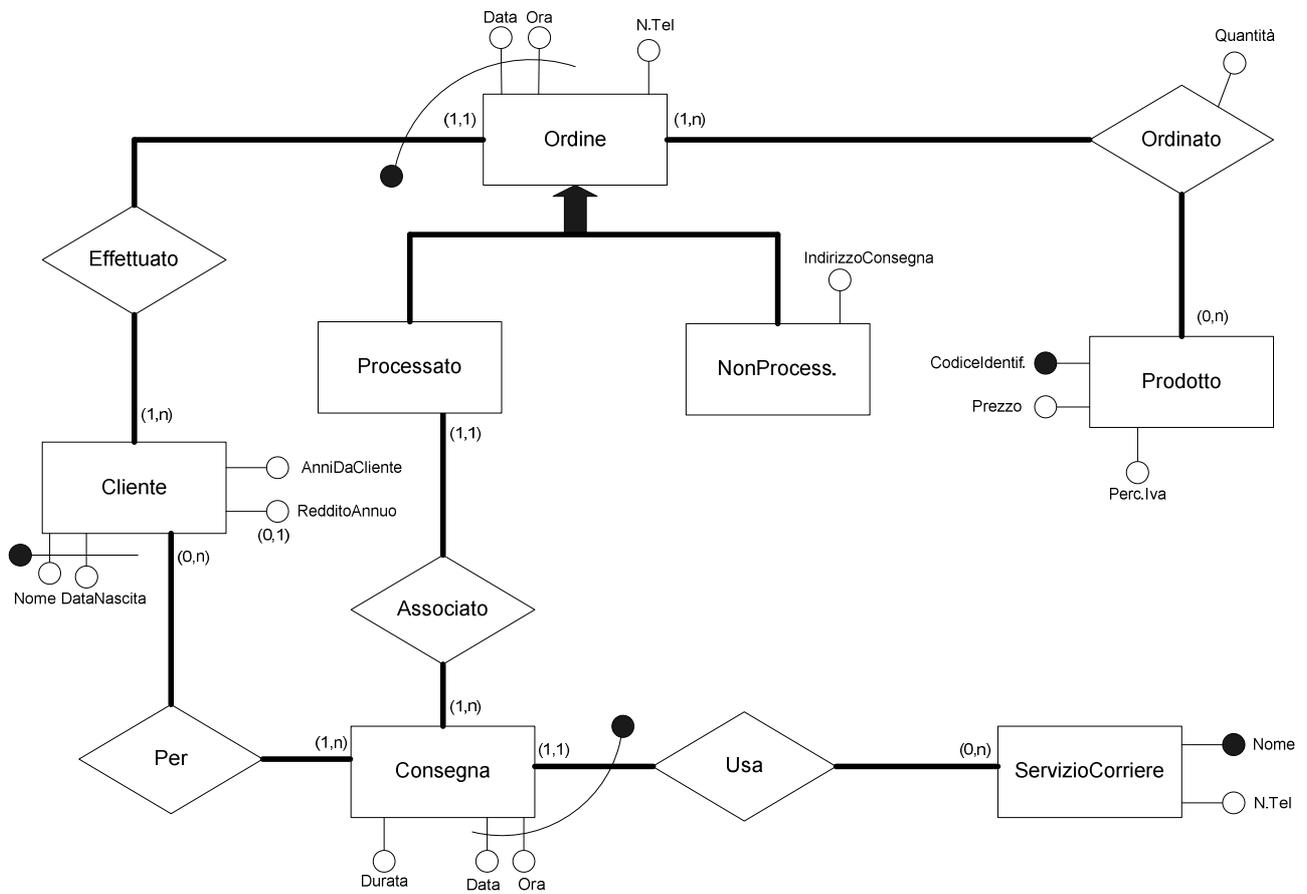
**[Invenzione, Invenzione(X, Y, Z) :- Invenzione(X, Y, Z) ]**

## Mapping DB 4 (Ordini di un'azienda)

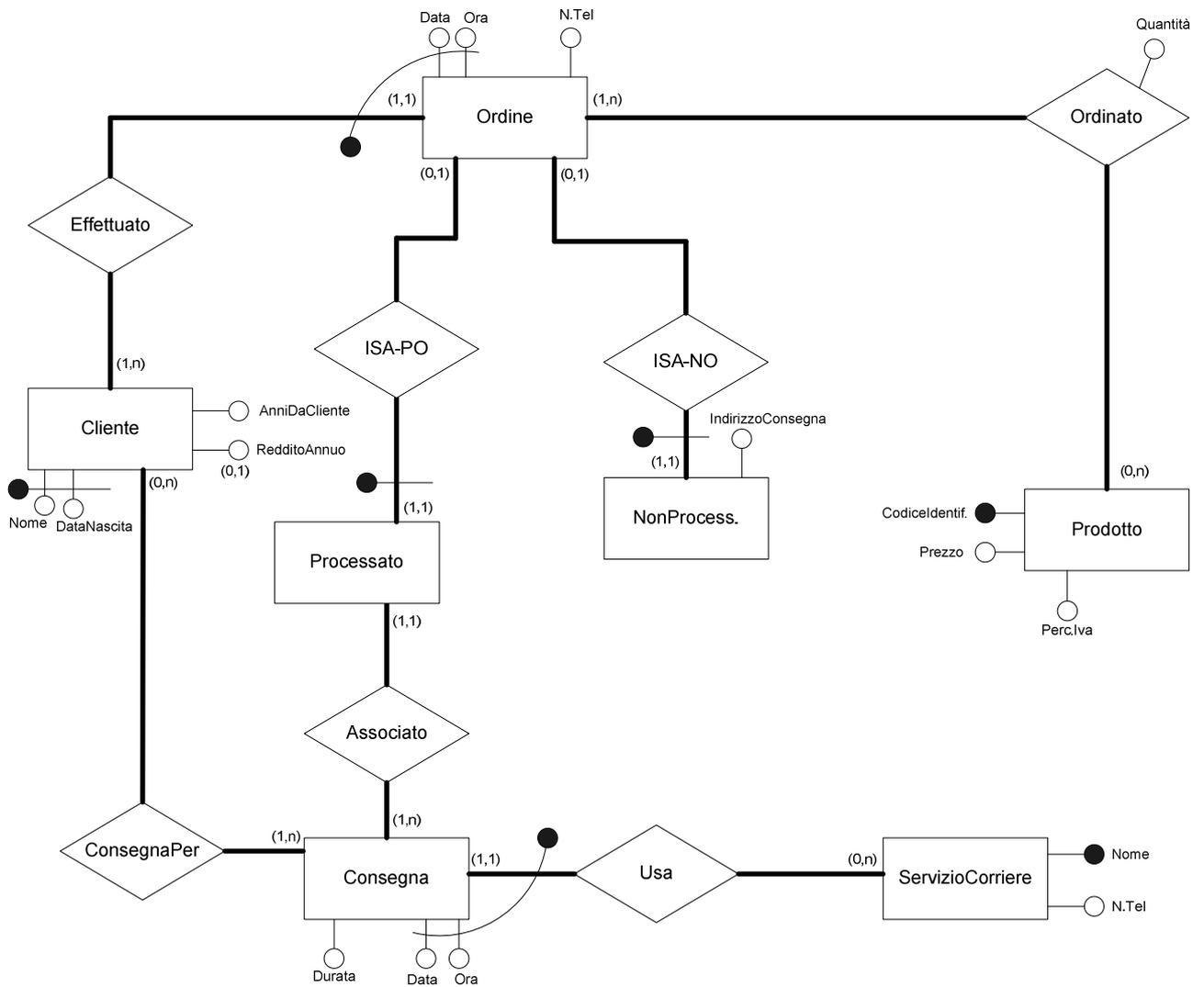
### **Specifiche:**

Si richiede di progettare DB di un'applicazione per la gestione degli ordini di un'azienda. Di ogni ordine interessa la data, l'ora, il numero di telefono di riferimento, il cliente (uno ed uno solo) che lo ha effettuato, e i prodotti ordinati, ciascuno in una data quantità. Si noti che ciascun cliente può effettuare al più un ordine in una certa ora di una certa data. Gli ordini si suddividono in processati e non processati. Di ogni ordine non processato interessa l'indirizzo di consegna. Di ogni ordine processato interessa la consegna (una ed una sola) attraverso cui è stato processato. Di ogni prodotto interessa il codice (identificativo), il prezzo, e la percentuale I.V.A. Ogni cliente è identificato dal suo nome e dalla sua data di nascita. Di ogni cliente interessa inoltre il numero di anni da cui è cliente e il reddito annuo, se noto. Si noti che ciascun cliente effettua almeno un ordine. Di ogni consegna, che è relativa ad un insieme di clienti (almeno uno), interessa la data, l'ora, la durata, ed il servizio di corriere usato (uno ed uno solo). Di ogni servizio di corriere interessa il nome (identificativo) e il numero di telefono. Si noti che uno stesso servizio di corriere non può essere usato per più di una consegna in una certa ora di una certa data.

### Diagramma ER – DB 4 – Globale



### Diagramma ER Ristrutturato – DB 4 – Globale



**Schema logico – DB 4 – Globale**

Ordine(Data, Ora, NomeCliente, DataNascitaCliente, Tel)

Ordinato(CodProdotto, DataOrdine, OraOrdine, NomeCliente, DataNascitaCliente,  
Quantità)

Prodotto(Codice, Prezzo, Iva)

Cliente(Nome, DataNascita, Reddito, AnniDaCliente)

ConsegnaPer(NomeC, DataNC, Data, Ora, NomeCorriere)

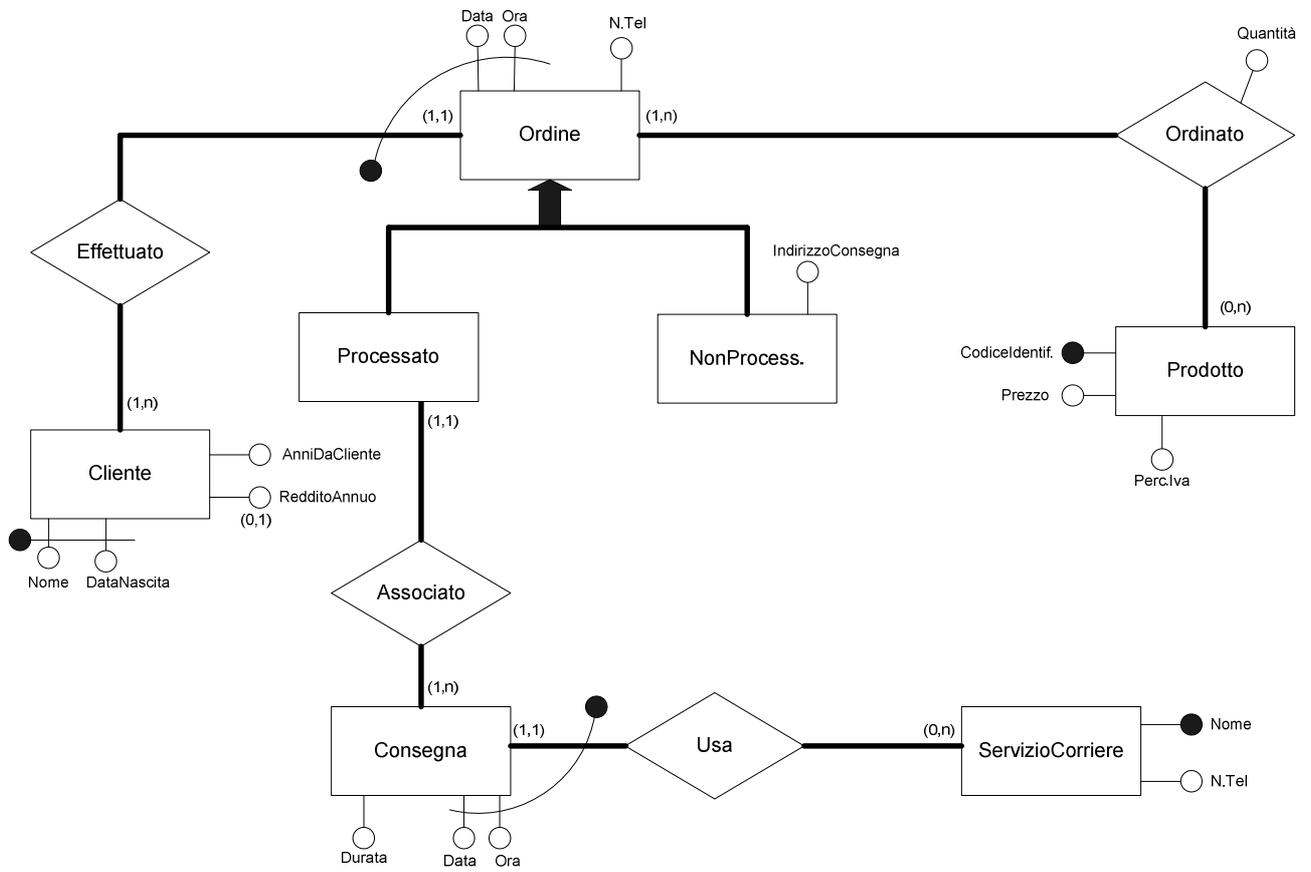
Consegna(Data, Ora, Corriere, Durata)

Corriere(Telefono, Nome)

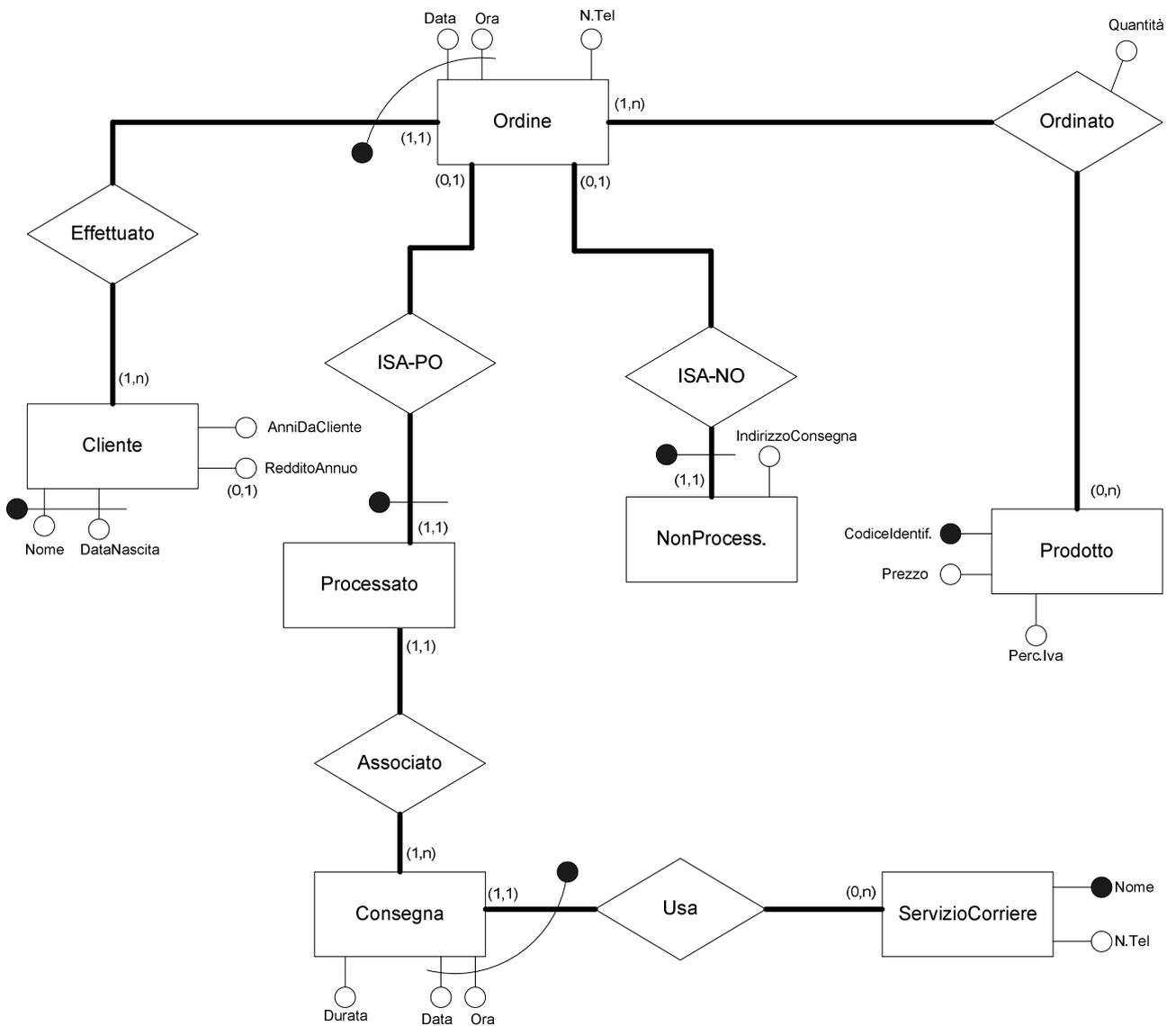
OrdineProc(DataOrd, OraOrd, NomeCli, DataNasCli, DataCon, OraCon, NomeCorr)

OrdineNonProc(DataOrd, OraOrd, NomeCli, DataNasCli, Indirizzo)

### Diagramma ER – DB 4 – Sorgente A



### Diagramma ER Ristrutturato – DB 4 – Sorgente A



**Schema logico – DB 4 – Sorgente A**

Ordine(Data, Ora, NomeCliente, DataNascitaCliente, Tel)

Ordinato(CodProdotto, DataOrdine, OraOrdine, NomeCliente, DataNascitaCliente,  
Quantità)

Prodotto(Codice, Prezzo, Iva)

Cliente(Nome, DataNascita, Reddito, AnniDaCliente)

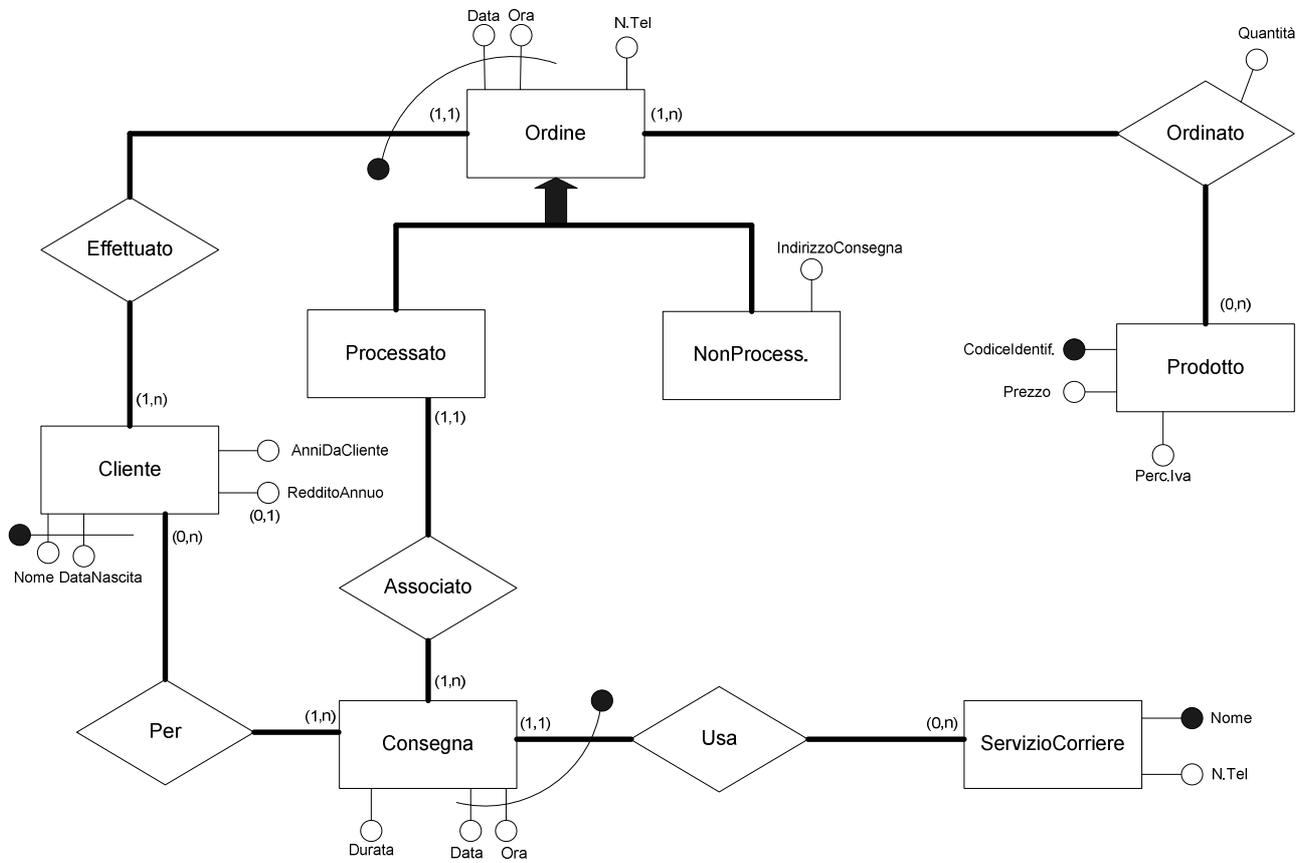
Consegna(Data, Ora, Corriere, Durata)

Corriere(Telefono, Nome)

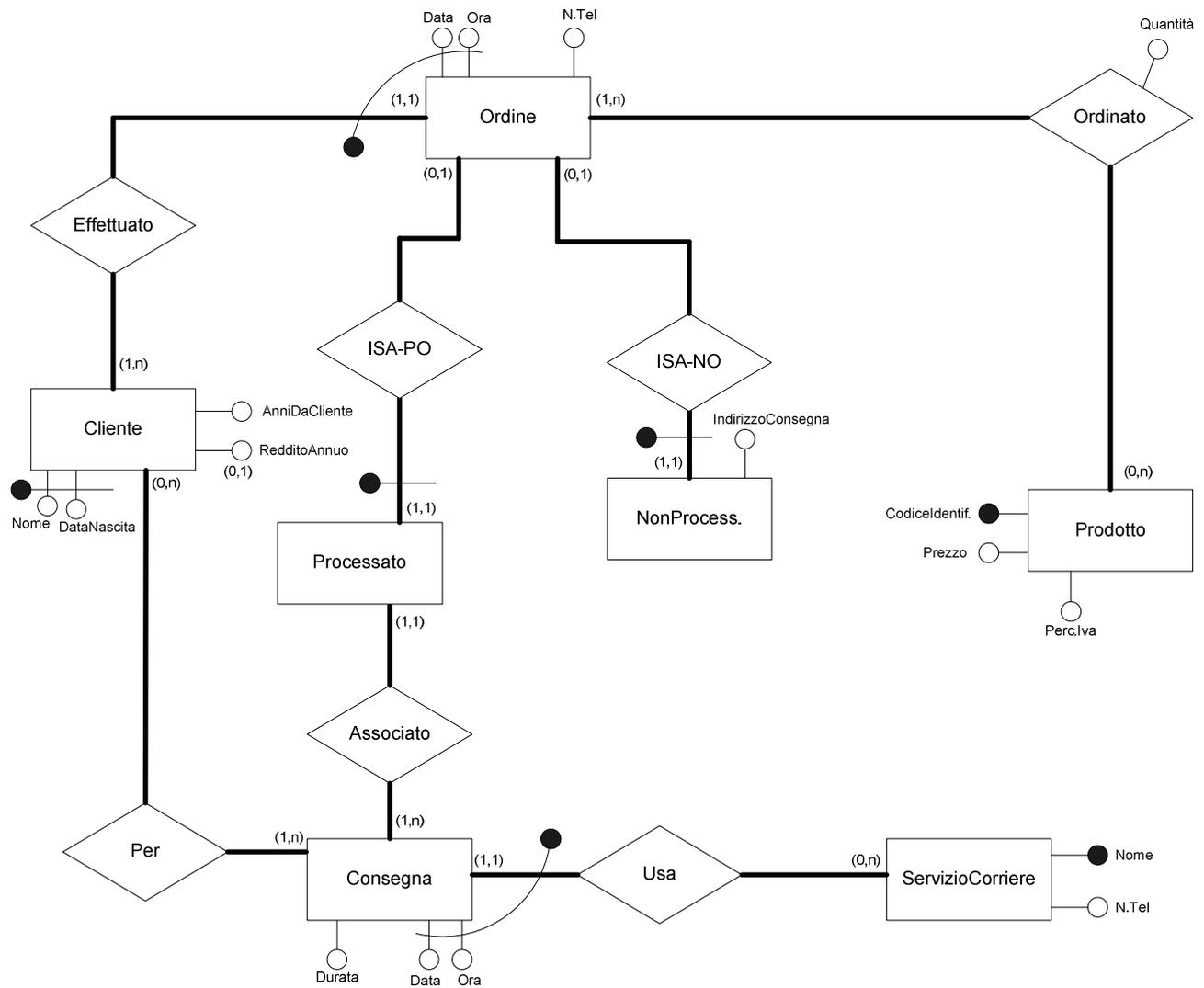
OrdineProc(DataOrd, OraOrd, NomeCli, DataNasCli, DataCon, OraCon, NomeCorr)

OrdineNonProc(DataOrd, OraOrd, NomeCli, DataNasCli, IndirizzoConsegna)

### Diagramma ER – DB 4 – Sorgente B



### Diagramma ER Ristrutturato – DB 4 – Sorgente B



**Schema logico – DB 4 – Sorgente B**

Ordine(Data, Ora, NomeCliente, DataNascitaCliente, Tel)

Ordinato(CodProdotto, DataOrdine, OraOrdine, NomeCliente, DataNascitaCliente,  
Quantità)

Prodotto(Codice, Prezzo, Iva)

Cliente(Nome, DataNascita, Reddito, AnniDaCliente)

Per(NomeC, DataNC, Data, Ora, NomeCorriere)

Consegna(Data, Ora, Corriere, Durata)

ServizioCorriere(Telefono, Nome)

OrdineProc(DataOrd, OraOrd, NomeCli, DataNasCli, DataCon, OraCon, NomeCorr)

OrdineNonProc(DataOrd, OraOrd, NomeCli, DataNasCli, IndirizzoConsegna)

### **Mapping GAV – DB 4**

Globale:

Ordine(Data, Ora, NomeCliente, DataNascitaCliente, Tel)

Sorgente A:

Ordine(Data, Ora, NomeCliente, DataNascitaCliente, Tel)

Sorgente B:

Ordine(Data, Ora, NomeCliente, DataNascitaCliente, Tel)

**[Ordine, Ordine(X, Y, Z, K, A) :- OrdineA(X, Y, Z, K, A)  
Ordine(X, Y, Z, K, A) :- OrdineB(X, Y, Z, K, A) ]**

---

Globale:

Ordinato(CodProdotto, DataOrdine, OraOrdine, NomeCliente, DataNascitaCliente,  
Quantità)

Sorgente A:

Odinato(CodProdotto, DataOrdine, OraOrdine, NomeCliente, DataNascitaCliente,  
Quantità)

Sorgente B:

Ordinato(CodProdotto, DataOrdine, OraOrdine, NomeCliente, DataNascitaCliente,  
Quantità)

**[Ordinato, Ordinato(X, Y, Z, K, A, B) :- OrdinatoA(X, Y, Z, K, A, B)  
Ordinato(X, Y, Z, K, A, B) :- OrdinatoB(X, Y, Z, K, A, B) ]**

---

Globale:

Prodotto(Codice, Prezzo, Iva)

Sorgente A:

Prodotto(Codice, Prezzo, Iva)

Sorgente B:

Prodotto(Codice, Prezzo, Iva)

**[Prodotto, Prodotto(X, Y, Z) :- ProdottoA(X, Y, Z)  
Prodotto(X, Y, Z) :- ProdottoB(X, Y, Z) ]**

---

Globale:

Cliente(Nome, DataNascita, Reddito, AnniDaCliente)

Sorgente A:

Cliente(**Nome, DataNascita**, Reddito, AnniDaCliente)

Sorgente B:

Cliente(**Nome, DataNascita**, Reddito, AnniDaCliente)

**[Cliente,      Cliente(X, Y, Z, K) :- ClienteA(X, Y, Z, K)  
                 Cliente(X, Y, Z, K) :- ClienteB(X, Y, Z, K) ]**

Globale:

ConsegnaPer(**NomeC, DataNC, Data, Ora, NomeCorriere**)

Sorgente A:

Ordine(**Data, Ora, NomeCliente, DataNascitaCliente**, Tel)

OrdineProc(**DataOrd, OraOrd, NomeCli, DataNasCli**, DataCon, OraCon, NomeCorr)

Sorgente B:

Per(**NomeC, DataNC, Data, Ora, NomeCorriere**)

**[ConsegnaPer,**

**ConsegnaPer(Z, K, A, B, C) :-**

**OrdineProcA(X, Y, Z, K, A, B, C), OrdineA(X, Y, Z, K, D)**

**ConsegnaPer(X, Y, Z, K, A) :- PerB(X, Y, Z, K, A) ]**

Globale:

Consegna(**Data, Ora, Corriere**, Durata)

Sorgente A:

Consegna(**Data, Ora, Corriere**, Durata)

Sorgente B:

Consegna(**Data, Ora, Corriere**, Durata)

**[Consegna,      Consegna(X, Y, Z, K) :- ConsegnaA(X, Y, Z, K)  
                 Consegna(X, Y, Z, K) :- ConsegnaB(X, Y, Z, K)]**

Globale:

Corriere(Telefono, **Nome**)

Sorgente A:

Corriere(Telefono, **Nome**)

Sorgente B:

ServizioCorriere(Telefono, **Nome**)

**[Corriere,      Corriere(X, Y) :- CorriereA(X, Y)**

---

**Corriere(X, Y) :- ServizioCorriereB(X, Y) ]**

---

Globale:

OrdineProc(DataOrd, OraOrd, NomeCli, DataNasCli, DataCon, OraCon, NomeCorr)

Sorgente A:

OrdineProc(DataOrd, OraOrd, NomeCli, DataNasCli, DataCon, OraCon, NomeCorr)

Sorgente B:

OrdineProc(DataOrd, OraOrd, NomeCli, DataNasCli, DataCon, OraCon, NomeCorr)

**[OrdineProc,        OrdineProc(X, Y, Z, K, A, B, C) :- OrdineProcA(X, Y, Z, K, A, B, C)  
                         OrdineProc(X, Y, Z, K, A, B, C) :- OrdineProcB(X, Y, Z, K, A, B, C) ]**

---

Globale:

OrdineNonProc(DataOrd, OraOrd, NomeCli, DataNasCli, Indirizzo)

Sorgente A:

OrdineNonProc(DataOrd, OraOrd, NomeCli, DataNasCli, IndirizzoConsegna)

Sorgente B:

OrdineNonProc(DataOrd, OraOrd, NomeCli, DataNasCli, IndirizzoConsegna)

**[OrdineNonProc,    OrdineNonProc(X, Y, Z, K, A) :- OrdineNonProcA(X, Y, Z, K, A)  
                         OrdineNonProc(X, Y, Z, K, A) :- OrdineNonProcB(X, Y, Z, K, A) ]**

### **Mapping LAV – DB 4 – Sorgente A**

Dato che le sorgenti e lo schema globale sono molto simili il mapping è stato diretto.

Globale:

Ordine(Data, Ora, NomeCliente, DataNascitaCliente, Tel)

Sorgente A:

Ordine(Data, Ora, NomeCliente, DataNascitaCliente, Tel)

**[Ordine, Ordine(X, Y, Z, K, A) :- Ordine(X, Y, Z, K, A) ]**

Globale:

Odinato(CodProdotto, DataOrdine, OraOrdine, NomeCliente, DataNascitaCliente,  
Quantità)

Sorgente A:

Odinato(CodProdotto, DataOrdine, OraOrdine, NomeCliente, DataNascitaCliente,  
Quantità)

**[Ordinato, Ordinato(X, Y, Z, K, A, B) :- Ordinato(X, Y, Z, K, A, B) ]**

Globale:

Prodotto(Codice, Prezzo, Iva)

Sorgente A:

Prodotto(Codice, Prezzo, Iva)

**[Prodotto, Prodotto(X, Y, Z) :- Prodotto(X, Y, Z) ]**

Globale:

Cliente(Nome, DataNascita, Reddito, AnniDaCliente)

Sorgente A:

Cliente(Nome, DataNascita, Reddito, AnniDaCliente)

**[Cliente, Cliente(X, Y, Z, K) :- Cliente(X, Y, Z, K) ]**

Globale:

Consegna(Data, Ora, Corriere, Durata)

Sorgente A:

Consegna(Data, Ora, Corriere, Durata)

**[Consegna, Consegna(X, Y, Z, K) :- Consegna(X, Y, Z, K) ]**

---

Globale:

Corriere(Telefono, Nome)

Sorgente A:

Corriere(Telefono, Nome)

**[Corriere, Corriere(X, Y) :- Corriere(X, Y) ]**

---

Globale:

OrdineProc(DataOrd, OraOrd, NomeCli, DataNasCli, DataCon, OraCon, NomeCorr)

Sorgente A:

OrdineProc(DataOrd, OraOrd, NomeCli, DataNasCli, DataCon, OraCon, NomeCorr)

**[OrdineProc, OrdineProc(X, Y, Z, K, A, B, C) :- OrdineProc(X, Y, Z, K, A, B, C) ]**

---

Globale:

OrdineNonProc(DataOrd, OraOrd, NomeCli, DataNasCli, Indirizzo)

Sorgente A:

OrdineNonProc(DataOrd, OraOrd, NomeCli, DataNasCli, Indirizzo)

**[OrdineNonProc, OrdineNonProc(X, Y, Z, K, A) :- OrdineNonProc(X, Y, Z, K, A) ]**

### **Mapping LAV – DB 4 – Sorgente B**

Dato che le sorgenti e lo schema globale sono molto simili il mapping è stato diretto.

Globale:

Ordine(Data, Ora, NomeCliente, DataNascitaCliente, Tel)

Sorgente B:

Ordine(Data, Ora, NomeCliente, DataNascitaCliente, Tel)

**[Ordine,                    Ordine (X, Y, Z, K, A) :- Ordine (X, Y, Z, K, A) ]**

Globale:

Ordinato(CodProdotto, DataOrdine, OraOrdine, NomeCliente, DataNascitaCliente,  
Quantità)

Sorgente B:

Ordinato(CodProdotto, DataOrdine, OraOrdine, NomeCliente, DataNascitaCliente,  
Quantità)

**[Ordinato,                    Ordinato(X, Y, Z, K, A, B) :- Ordinato(X, Y, Z, K, A, B) ]**

Globale:

Prodotto(Codice, Prezzo, Iva)

Sorgente B:

Prodotto(Codice, Prezzo, Iva)

**[Prodotto,                    Prodotto(X, Y, Z) :- Prodotto(X, Y, Z) ]**

Globale:

Cliente(Nome, DataNascita, Reddito, AnniDaCliente)

Sorgente B:

Cliente(Nome, DataNascita, Reddito, AnniDaCliente)

**[Cliente,                    Cliente(X, Y, Z, K) :- Cliente(X, Y, Z, K) ]**

Globale:

ConsegnaPer(NomeC, DataNC, Data, Ora, NomeCorriere)

Sorgente B:

Per(NomeC, DataNC, Data, Ora, NomeCorriere)

**[Per, Per(X, Y, Z, K, A) :- ConsegnaPer(X, Y, Z, K, A) ]**

---

Globale:

Consegna(Data, Ora, Corriere, Durata)

Sorgente B:

Consegna(Data, Ora, Corriere, Durata)

**[Consegna, Consegna(X, Y, Z, K) :- Consegna(X, Y, Z, K) ]**

---

Globale:

Corriere(Telefono, Nome)

Sorgente B:

ServizioCorriere(Telefono, Nome)

**[ServizioCorriere, ServizioCorriere (X, Y) :- Corriere(X, Y) ]**

---

Globale:

OrdineProc(DataOrd, OraOrd, NomeCli, DataNasCli, DataCon, OraCon, NomeCorr)

Sorgente B:

OrdineProc(DataOrd, OraOrd, NomeCli, DataNasCli, DataCon, OraCon, NomeCorr)

**[OrdineProc, OrdineProc(X, Y, Z, K, A, B, C) :- OrdineProc(X, Y, Z, K, A, B, C) ]**

---

Globale:

OrdineNonProc(DataOrd, OraOrd, NomeCli, DataNasCli, Indirizzo)

Sorgente B:

OrdineNonProc(DataOrd, OraOrd, NomeCli, DataNasCli, IndirizzoConsegna)

**[OrdineNonProc, OrdineNonProc(X, Y, Z, K, A) :- OrdineProc(X, Y, Z, K, A) ]**

## Mapping DB 5 (Agenzia di viaggio)

### ***Specifiche:***

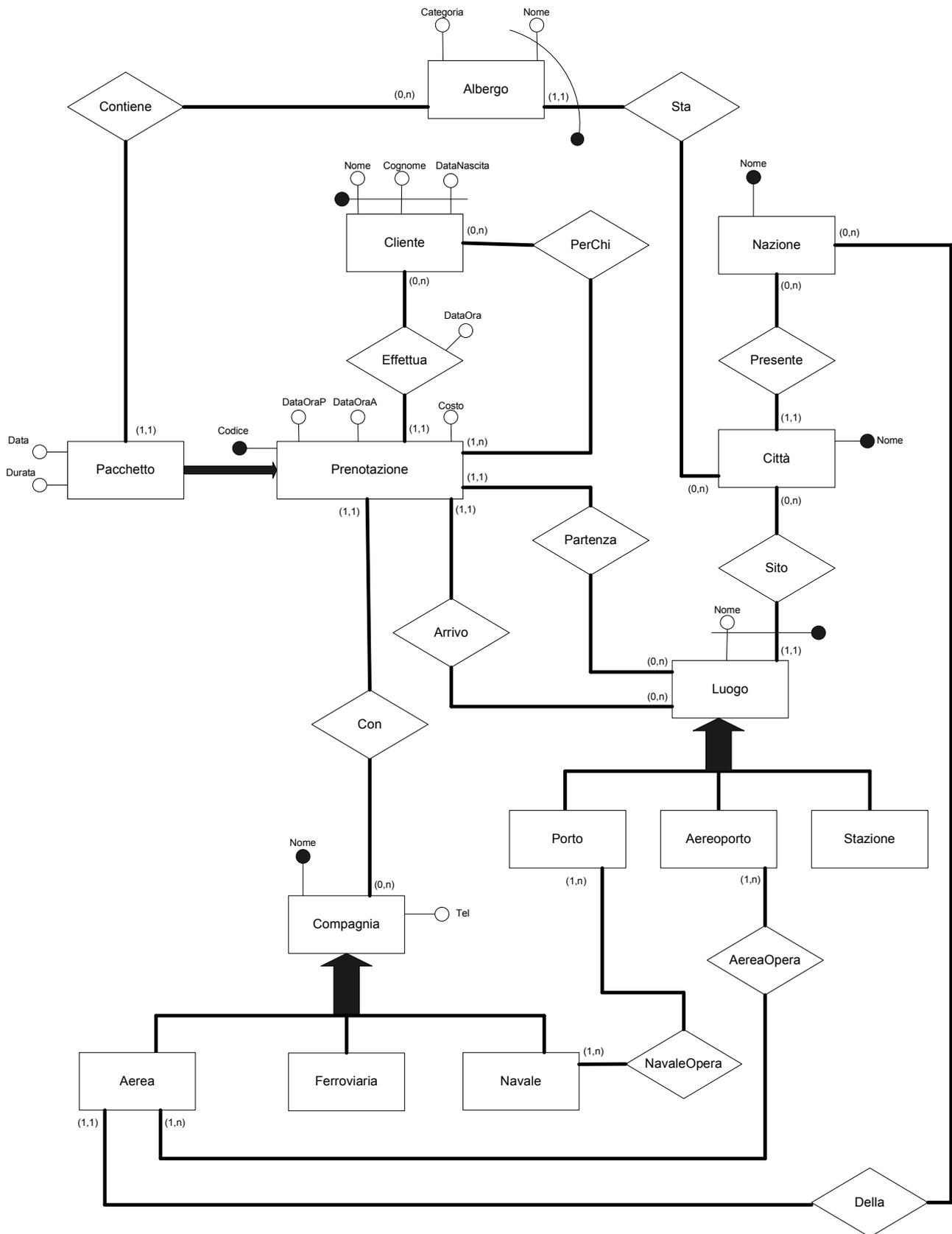
Si richiede di progettare DB di un'applicazione per la gestione di un'agenzia di viaggi. Bisogna poter gestire le informazioni delle prenotazioni, dei clienti e delle compagnie. Si deve tener traccia in un qualche modo tre tipi di viaggi: con aereo, nave e treno.

### ***Considerazioni:***

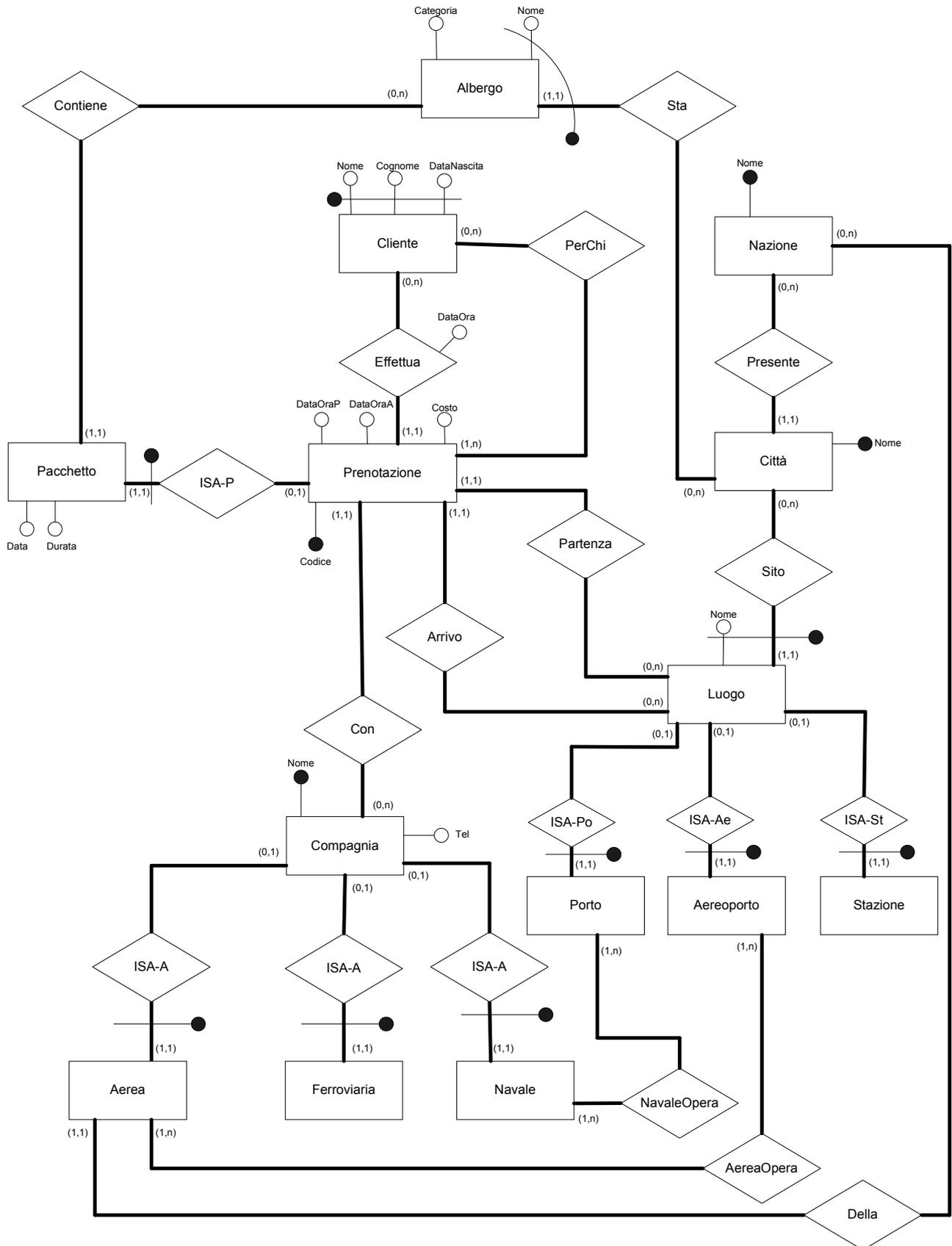
La specifica è stata data in modo molto vago per ottenere delle sorgenti diverse tra loro e con lo schema globale.

Lo schema globale è stato realizzato in modo tale che differisse dalle sorgenti sia per la presenza o meno di alcune informazioni considerate nelle sorgenti sia nella struttura. Questa strategia ha portato in evidenza vari problemi quando abbiamo applicato il mapping GAV e LAV.

### Diagramma ER – DB 5 – Globale



**Diagramma ER Ristrutturato – DB 5 – Globale**



## Schema logico – DB 5 – Globale

Cliente(**Nome, Cognome, DataNascita**, Tel)

Prenotazione(**Codice**, DataOraPrenotazione, DataOraPartenza, DataOraArrivo, Costo, NomeLuogoPartenza, CittàPartenza, NomeLuogoArrivo, CittàArrivo, NomeCompagnia, NomeChiCompra, CognomeChiCompra, DataNascitaChiCompra)

*foreign key: Prenotazione[NomeChiCompra, CognomeChiCompra, DataNascitaChiCompra]  $\subseteq$  Cliente[Nome, Cognome, DataNascita]*

*foreign key: Prenotazione[NomeCompagnia]  $\subseteq$  Compagnia[Nome]*

*foreign key: Prenotazione[NomeLuogoPartenza, CittàPartenza]  $\subseteq$  Luogo[NomeLuogo, NomeCittà]*

*foreign key: Prenotazione[NomeLuogoArrivo, CittàArrivo]  $\subseteq$  Luogo[NomeLuogo, NomeCittà]*

PerChi(**CodicePrenotazione, Nome, Cognome, DataNascita**)

*foreign key: PerChi[CodicePrenotazione]  $\subseteq$  Prenotazione[Codice]*

*foreign key: PerChi[Nome, Cognome, DataNascita]  $\subseteq$  Cliente[Nome, Cognome, DataNascita]*

Compagnia(**Nome**, Telefono)

Navale(**Nome**)

*foreign key: Navale[Nome]  $\subseteq$  Compagnia[Nome]*

Aerea(**Nome**, Nazione)

*foreign key: Aerea[Nome]  $\subseteq$  Compagnia[Nome]*

*foreign key: Aerea[Nazione]  $\subseteq$  Nazione[Nome]*

Ferroviana(**Nome**)

*foreign key: Ferroviana[Nome]  $\subseteq$  Compagnia[Nome]*

Luogo(**NomeLuogo, NomeCittà**, Tel)

*foreign key: Luogo[NomeCittà]  $\subseteq$  Città[Nome]*

Porto(**NomeLuogo, NomeCittà**)

*foreign key: Porto[NomeLuogo, NomeCittà]  $\subseteq$  Luogo[NomeLuogo, NomeCittà]*

Aereoporto(**NomeLuogo, NomeCittà**)

*foreign key: Aereoporto[NomeLuogo, NomeCittà]  $\subseteq$  Luogo[NomeLuogo, NomeCittà]*

Stazione(**NomeLuogo, NomeCittà**)

*foreign key: Stazione[NomeLuogo, NomeCittà]  $\subseteq$  Luogo[NomeLuogo, NomeCittà]*

NavaleOpera(**NomeCompagnia, NomeLuogo, NomeCittà**)

*foreign key: NavaleOpera[NomeCompagnia]  $\subseteq$  Navale[Nome]*

*foreign key: NavaleOpera[NomeLuogo, NomeCittà]  $\subseteq$  Porto[NomeLuogo, NomeCittà]*

AereaOpera(**NomeCompagnia, NomeLuogo, NomeCittà**)

*foreign key: AereaOpera[NomeCompagnia]  $\subseteq$  Aerea[Nome]*

*foreign key: AereaOpera[NomeLuogo, NomeCittà]  $\subseteq$  Aereoporto[NomeLuogo, NomeCittà]*

Città(**Nome**, Nazione)

*foreign key: Città[Nazione]  $\subseteq$  Nazione[Nome]*

Nazione(**Nome**)

Pacchetto(**CodicePrenotazione**, Data, Durata, Albergo, CittàAlbergo)

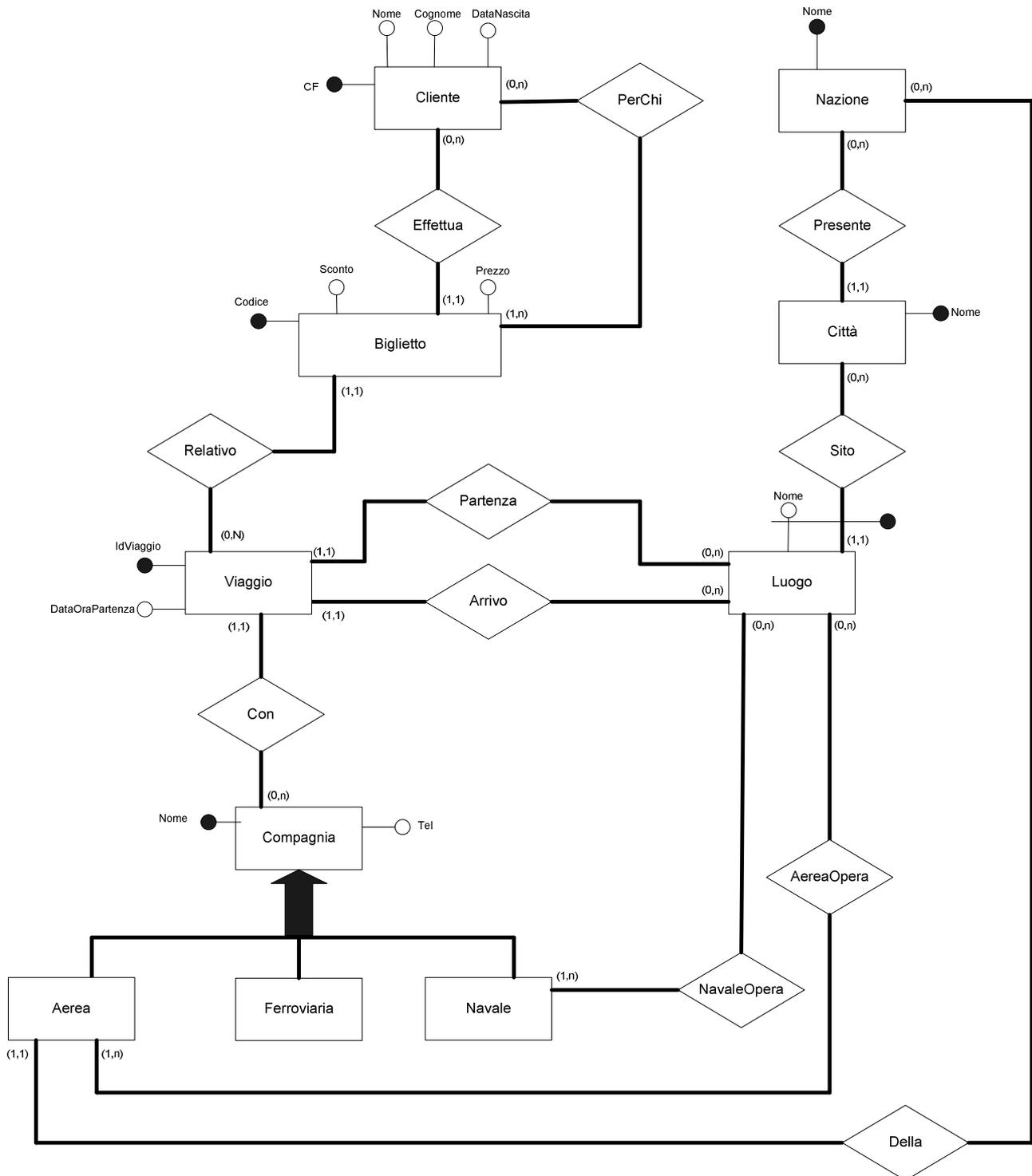
*foreign key: Pacchetto[CodicePrenotazione]  $\subseteq$  Prenotazione[Codice]*

*foreign key: Pacchetto[Albergo, CittàAlbergo]  $\subseteq$  Albergo[NomeAlbergo, NomeCittà]*

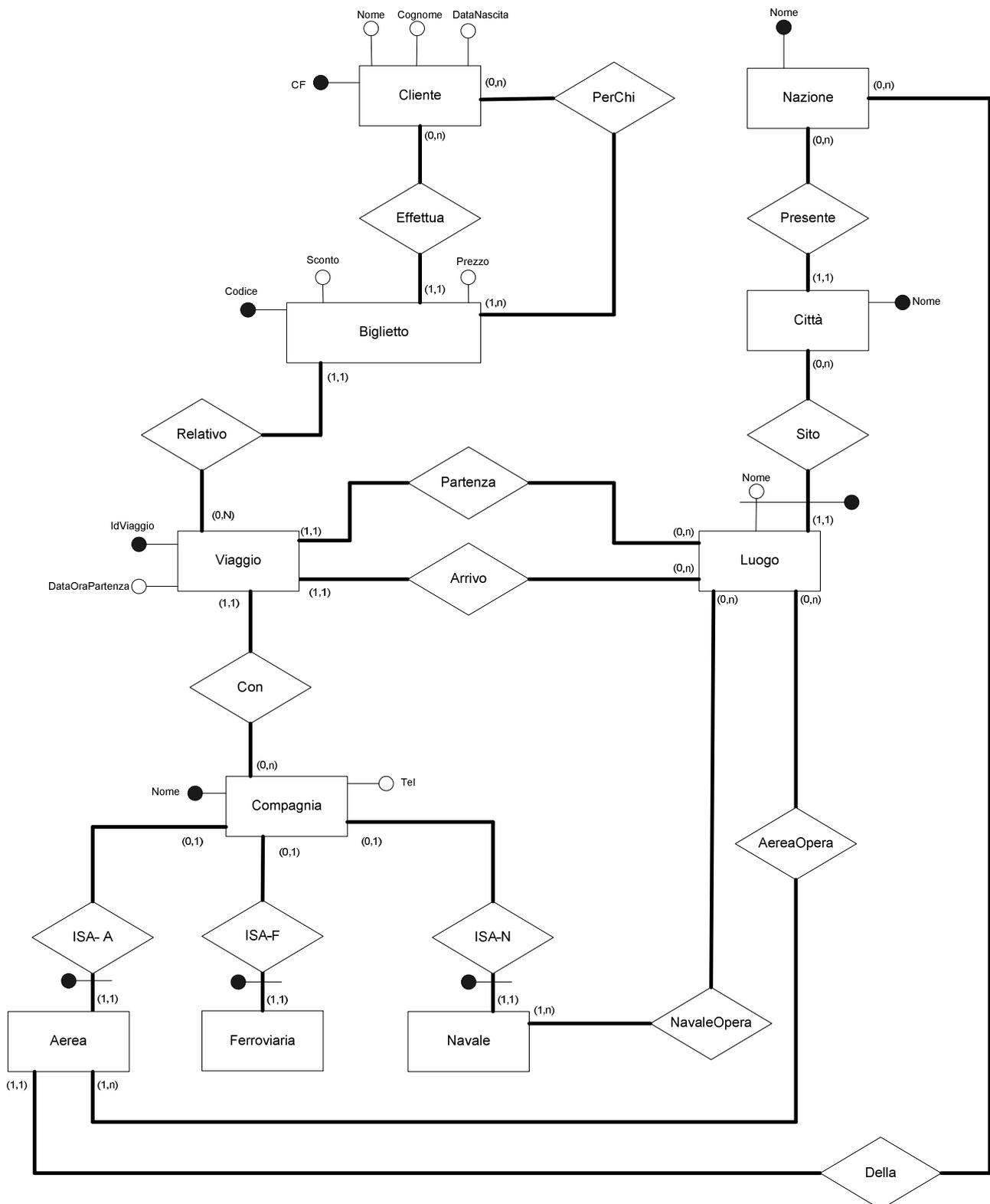
Albergo(**NomeAlbergo, NomeCittà**, Categoria)

*foreign key: Albergo[NomeCittà]  $\subseteq$  Città[Nome]*

### Diagramma ER – DB 5 – Sorgente A



### Diagramma ER Ristrutturato – DB 5 – Sorgente A



## Schema logico – DB 5 – Sorgente A

Cliente(Nome, Cognome, DataNascita, **CF**)

Biglietto(**Codice**, Prezzo, Sconto, CFChiCompra, IdViaggio)

*foreign key: Biglietto[CFChiCompra]  $\subseteq$  Cliente[CF]*

*foreign key: Biglietto[IdViaggio]  $\subseteq$  Viaggio[IdViaggio]*

PerChi(**CodiceBiglietto**, **CF**)

*foreign key: PerChi[CodiceBiglietto]  $\subseteq$  Biglietto[Codice]*

*foreign key: PerChi[CF]  $\subseteq$  Cliente[CF]*

Compagnia(**Nome**, Telefono)

Navale(**Nome**)

*foreign key: Navale[Nome]  $\subseteq$  Compagnia[Nome]*

Aerea(**Nome**, Nazione)

*foreign key: Aerea[Nome]  $\subseteq$  Compagnia[Nome]*

*foreign key: Aerea[Nazione]  $\subseteq$  Nazione[Nome]*

Ferroviaria(**Nome**)

*foreign key: Ferroviaria[Nome]  $\subseteq$  Compagnia[Nome]*

Luogo(**NomeLuogo**, **NomeCittà**, Tel)

*foreign key: Luogo[NomeCittà]  $\subseteq$  Città[Nome]*

NavaleOpera(**NomeCompagnia**, **NomeLuogo**, **NomeCittà**)

*foreign key: NavaleOpera[NomeCompagnia]  $\subseteq$  Navale[Nome]*

*foreign key: NavaleOpera[NomeLuogo, NomeCittà]  $\subseteq$  Luogo[NomeLuogo, NomeCittà]*

AereaOpera(**NomeCompagnia**, **NomeLuogo**, **NomeCittà**)

*foreign key: AereaOpera[NomeCompagnia]  $\subseteq$  Aerea[Nome]*

*foreign key: AereaOpera[NomeLuogo, NomeCittà]  $\subseteq$  Luogo[NomeLuogo, NomeCittà]*

Città(**Nome**, Nazione)

*foreign key: Città[Nazione]  $\subseteq$  Nazione[Nome]*

Nazione(**Nome**)

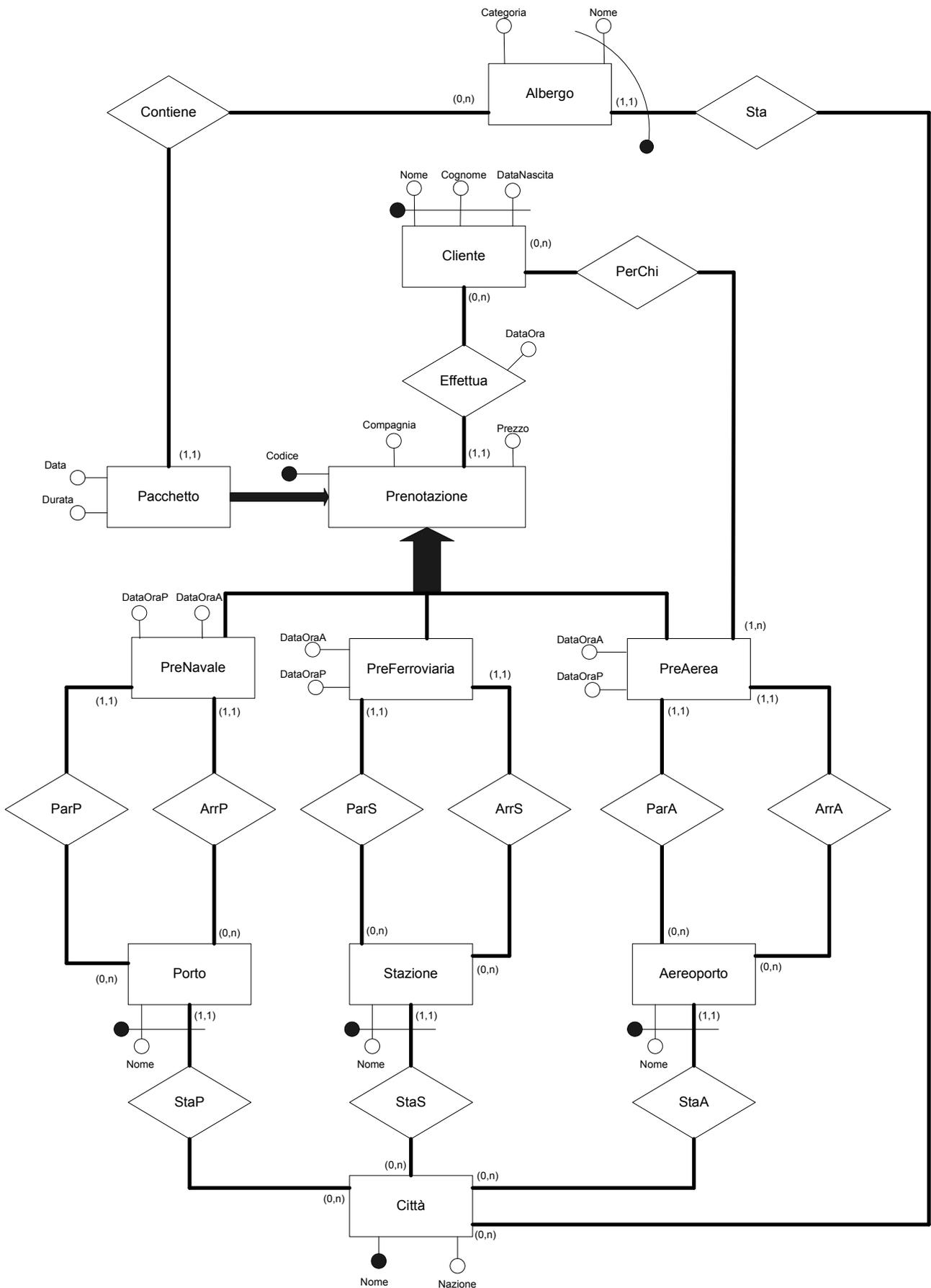
Viaggio(**IdViaggio**, NomeLuogoPartenza, NomeCittàPartenza, NomeLuogoDestinazione,  
NomeCittàDestinazione, NomeCompagnia, DataOraPartenza)

*foreign key: Viaggio[NomeCompagnia]  $\subseteq$  Compagnia[Nome]*

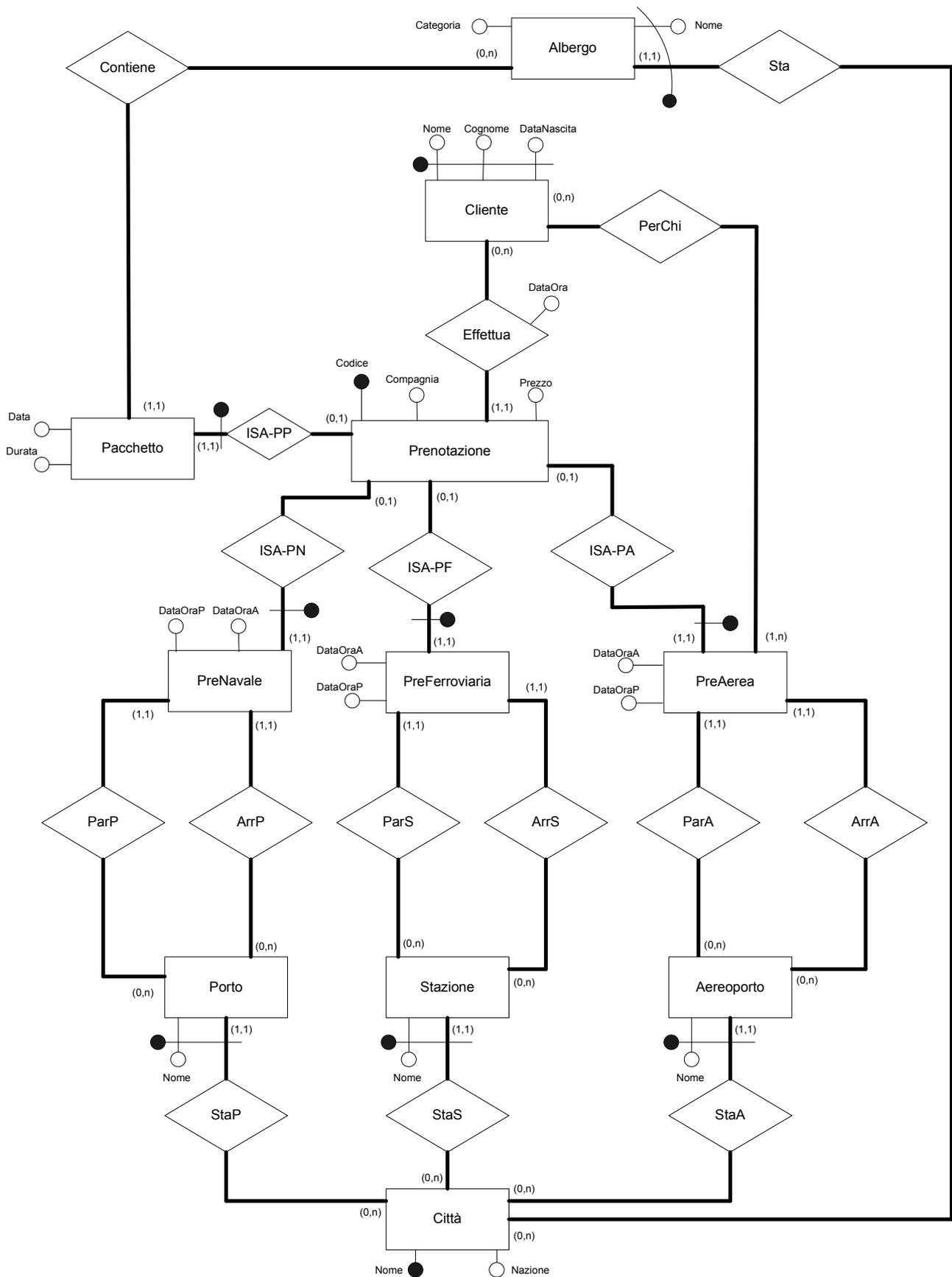
*foreign key: Viaggio[NomeLuogoPartenza, NomeCittàPartenza]  $\subseteq$  Luogo[NomeLuogo,  
NomeCittà]*

*foreign key: Viaggio[NomeLuogoDestinazione, NomeCittàDestinazione]  $\subseteq$   
Luogo[NomeLuogo, NomeCittà]*

### Diagramma ER – DB 5 – Sorgente B



**Diagramma ER Ristrutturato – DB 5 – Sorgente B**



## Schema logico – DB 5 – Sorgente B

Cliente(**Nome, Cognome, DataNascita**, Tel)

Prenotazione(**Codice**, DataOraPrenotazione, NomeCompagnia, Prezzo,  
NomeChiCompra, CognomeChiCompra, DataNascitaChiCompra)  
*foreign key: Prenotazione[NomeChiCompra, CognomeChiCompra,  
DataNascitaChiCompra]  $\subseteq$  Cliente[Nome, Cognome, DataNascita]*

PreAerea(**CodPrenotazione**, DataOraPartenza, NomeAereoportoPartenza,  
NomeCittàPartenza, DataOraArrivo, NomeAereoportoArrivo,  
NomeCittàArrivo)  
*foreign key: PreAerea[CodPrenotazione]  $\subseteq$  Prenotazione[Codice]*  
*foreign key: PreAerea[NomeAereoportoPartenza, NomeCittàPartenza]  $\subseteq$   
Aereoporto[Nome, NomeCittà]*  
*foreign key: PreAerea[NomeAereoportoArrivo, NomeCittàArrivo]  $\subseteq$  Aereoporto[Nome,  
NomeCittà]*

PreNavale(**CodPrenotazione**, DataOraPartenza, NomePortoPartenza,  
NomeCittàPartenza, DataOraArrivo, NomePortoArrivo, NomeCittàArrivo)  
*foreign key: PreNavale[CodPrenotazione]  $\subseteq$  Prenotazione[Codice]*  
*foreign key: PreNavale[NomePortoPartenza, NomeCittàPartenza]  $\subseteq$  Porto[Nome,  
NomeCittà]*  
*foreign key: PreNavale[NomePortoArrivo, NomeCittàArrivo]  $\subseteq$  Porto[Nome, NomeCittà]*

PreFerroviaria(**CodPrenotazione**, DataOraPartenza, NomeStazionePartenza,  
NomeCittàPartenza, DataOraArrivo, NomeStazioneArrivo,  
NomeCittàArrivo)  
*foreign key: PreFerroviaria[CodPrenotazione]  $\subseteq$  Prenotazione[Codice]*  
*foreign key: PreFerroviaria[NomeStazionePartenza, NomeCittàPartenza]  $\subseteq$   
Stazione[Nome, NomeCittà]*  
*foreign key: PreFerroviaria[NomeStazioneArrivo, NomeCittàArrivo]  $\subseteq$  Stazione[Nome,  
NomeCittà]*

Aereoporto(**Nome, NomeCittà**)  
*foreign key: Aereoporto[NomeCittà]  $\subseteq$  Città[Nome]*

Porto(**Nome, NomeCittà**)  
*foreign key: Porto[NomeCittà]  $\subseteq$  Città[Nome]*

Stazione(**Nome, NomeCittà**)

*foreign key: Stazione[NomeCittà] ⊆ Città[Nome]*

Città(**Nome**, Nazione)

PerChi(**CodicePrenotazioneAereo, Nome, Cognome, DataNascita**)

*foreign key: PerChi[CodicePrenotazioneAereo] ⊆ PreAerea[CodPrenotazione]*

*foreign key: PerChi[Nome, Cognome, DataNascita] ⊆ Cliente[Nome, Cognome, DataNascita]*

Pacchetto(**CodicePrenotazione**, Data, Durata, NomeAlbergo, CittàAlbergo)

*foreign key: Pacchetto[CodicePrenotazione] ⊆ Prenotazione[Codice]*

*foreign key: Pacchetto[NomeAlbergo, CittàAlbergo] ⊆ Albergo[NomeAlbergo, NomeCittà]*

Albergo(**NomeAlbergo, NomeCittà**, Categoria)

*foreign key: Albergo[NomeCittà] ⊆ Città[Nome]*

## **Mapping GAV – DB 5**

Per la filosofia seguita alla costruzione dello schema Globale (medo dettagliato), la realizzazione del Mapping GAV è stata meno problematica, infatti buona parte delle informazioni necessarie erano in gran parte presenti nelle sorgenti.

Globale:

Cliente(**Nome, Cognome, DataNascita**, Tel)

Sorgente A:

Cliente(Nome, Cognome, DataNascita, **CF**)

Sorgente B:

Cliente(**Nome, Cognome, DataNascita**, Tel)

**[Cliente,      Cliente(X, Y, Z, NULL) :- ClienteA(X, Y, Z, A),  
                 Cliente(X, Y, Z, K) :- ClienteB(X, Y, Z, K) ]**

Il NULL sul campo "Tel" è stato necessario perchè nella sorgente A questa informazione non è presente.

I vincoli di chiave sul globale vengono rispettati dai dati della sorgente B dato che fa uso della stessa chiave, per quelli provenienti dalla sorgente A teoricamente non è garantiti che gli elementi abbiano i campi nome, cognome e data di nascita univoci.

Globale:

Prenotazione(**Codice**, DataOraPrenotazione, DataOraPartenza, DataOraArrivo, Costo,  
                 NomeLuogoPartenza, CittàPartenza, NomeLuogoArrivo, CittàArrivo,  
                 NomeCompagnia, NomeChiCompra, CognomeChiCompra,  
                 DataNascitaChiCompra)

Sorgente A:

Cliente(Nome, Cognome, DataNascita, **CF**)

Biglietto(**Codice**, Prezzo, Sconto, CFChiCompra, IdViaggio)

Viaggio(**IdViaggio**, NomeLuogoPartenza, NomeCittàPartenza, NomeLuogoDestinazione,  
                 NomeCittàDestinazione, NomeCompagnia, DataOraPartenza)

Sorgente B:

Prenotazione(**Codice**, DataOraPrenotazione, NomeCompagnia, Prezzo, NomeChiCompra,  
                 CognomeChiCompra, DataNascitaChiCompra)

PreAerea(**CodPrenotazione**, DataOraPartenza, NomeAereoportoPartenza,  
                 NomeCittàPartenza, DataOraArrivo, NomeAereoportoArrivo, NomeCittàArrivo)

PreNavale(**CodPrenotazione**, DataOraPartenza, NomePortoPartenza,  
                 NomeCittàPartenza, DataOraArrivo, NomePortoArrivo, NomeCittàArrivo)

PreFerroviaria(**CodPrenotazione**, DataOraPartenza, NomeStazionePartenza,  
                 NomeCittàPartenza, DataOraArrivo, NomeStazioneArrivo, NomeCittàArrivo)

**[Prenotazione,**

**Prenotazione(A, NULL, B, NULL, E, F, G, H, I, L, M, N, O) :-**

**BigliettoA(A, E, Y, CF, D), ClienteA(M, N, O, CF), ViaggioA(D, F, G, H, I, L, B)**

**Prenotazione(A, B, C, D, E, F, G, H, I, L, M, N, O) :-**  
**PrenotazioneB(A, B, L, E, M, N, O), PreAereaB(A, C, F, G, D, H, I)**

**Prenotazione(A, B, C, D, E, F, G, H, I, L, M, N, O) :-**  
**PrenotazioneB(A, B, L, E, M, N, O), PreNavaleB(A, C, F, G, D, H, I)**

**Prenotazione(A, B, C, D, E, F, G, H, I, L, M, N, O) :-**  
**PrenotazioneB(A, B, L, E, M, N, O), PreFerroviariaB(A, C, F, G, D, H, I) ]**

Nella sorgente A le informazioni per la relazione Prenotazioni del globale sono distribuite su tre relazioni Biglietto, Cliente e Viaggio quindi l'asserzione deve essere composta dalle tre relazioni. Inoltre abbiamo assunto che il codice del biglietto della sorgente A viene usato come codice della prenotazione dello schema globale.

Per quanto riguarda la sorgente B sono in parte presenti nella relazione Prenotazione e in parte presenti nelle relazioni PreAerea, PreNavale e PreFerroviaria.

Globale:

PerChi(**CodicePrenotazione, Nome, Cognome, DataNascita**)

Sorgente A:

Cliente(Nome, Cognome, DataNascita, **CF**)

PerChi(**CodiceBiglietto, CF**)

Sorgente B:

PerChi(**CodicePrenotazioneAereo, Nome, Cognome, DataNascita**)

**[PerChi, PerChi(X, Y, Z, K) :- PerChiA(X, B), ClienteA(Y, Z, K, B)**  
**PerChi(X, Y, Z, K) :- PerChiB(X, Y, Z, K) ]**

La relazione *PerChi* rappresenta il legame tra una prenotazione e i vari clienti che sono interessati. Questa relazione è presente anche nelle sorgenti ma nella sorgente A per quanto riguarda il cliente contiene solo il CF (usato come chiave) e non il nome e cognome e data di nascita che invece è presente nella relazione *Cliente*.

Globale:

Compagnia(**Nome**, Telefono)

Sorgente A:

Compagnia(**Nome**, Telefono)

Sorgente B:

Prenotazione(**Codice**, DataOraPrenotazione, NomeCompagnia, Prezzo, NomeChiCompra, CognomeChiCompra, DataNascitaChiCompra)

**[Compagnia, Compagnia(X, Y) :- CompagniaA(X, Y)**  
**Compagnia(X, NULL) :- PrenotazioneB(A, B, X, C, D, E, F) ]**

Nella sorgente B l'informazione su una compagnia non è mantenuta in una relazione ma in un campo della relazione Prenotazione, quindi l'informazione sul nome della compagnia può essere ricavato da questo campo. Per le compagnie provenienti dalla sorgente B non è possibile ottenere l'informazione sul numero di telefono.

Per preservare il vincolo di chiave sulla relazione del globale bisogna prevedere di non considerare una stessa compagnia che è presente in entrambe le sorgenti. In'oltre nel caso avvenga questo si può pensare di prendere l'informazione dalla sorgente A che contiene anche il numero di telefono.

Globale:

Navale(**Nome**)

Sorgente A:

Navale(**Nome**)

Sorgente B:

Prenotazione(**Codice**, DataOraPrenotazione, NomeCompagnia, Prezzo, NomeChiCompra, CognomeChiCompra, DataNascitaChiCompra)

PreNavale(**CodPrenotazione**, DataOraPartenza, NomePortoPartenza, NomeCittàPartenza, DataOraArrivo, NomePortoArrivo, NomeCittàArrivo)

[Navale,

**Navale(X) :- NavaleA(X)**

**Navale(X) :- PrenotazioneB(A, B, X, C, D, E, F), PreNavaleB(A, G, H, I, L, M, N) ]**

La relazione *Navale* rappresenta una compagnia navale, nella sorgente A questo concetto è mappato direttamente. Per estrarre l'informazione dalla sorgente B è necessario ricavarla da due relazioni *PrenotazioneB* e *PreNavaleB*, infatti noi dobbiamo ricavare i nomi delle compagnie che sono navali. Nella sorgente B il nome della compagnia è presente nella relazione *Prenotazione*, invece la selezione delle sole compagnie navali la si può ottenere attraverso la relazione *PreNavale* che rappresenta una prenotazione riferita ad una compagnia navale.

Globale:

Aerea(**Nome**, Nazione)

Sorgente A:

Aerea(**Nome**, Nazione)

Sorgente B:

Prenotazione(**Codice**, DataOraPrenotazione, NomeCompagnia, Prezzo, NomeChiCompra, CognomeChiCompra, DataNascitaChiCompra)

PreAerea(**CodPrenotazione**, DataOraPartenza, NomeAereoportoPartenza, NomeCittàPartenza, DataOraArrivo, NomeAereoportoArrivo, NomeCittàArrivo)

[Aerea,

**Aerea(X, Y) :- AereaA(X, Y)**

**Aerea(X, NULL) :-**

**PrenotazioneB(A, B, X, C, D, E, F), PreAereaB(A, G, H, I, L, M, N) ]**

Lo stesso ragionamento fatto per il mapping della relazione *Navale* si applica su questa parte del mapping, usando ora *Prenotazione* e *PreAerea*.

---

Globale:

Ferroviana(**Nome**)

Sorgente A:

Ferroviana(**Nome**)

Sorgente B:

Prenotazione(**Codice**, DataOraPrenotazione, NomeCompagnia, Prezzo, NomeChiCompra, CognomeChiCompra, DataNascitaChiCompra)

PreFerroviana(**CodPrenotazione**, DataOraPartenza, NomeStazionePartenza, NomeCittàPartenza, DataOraArrivo, NomeStazioneArrivo, NomeCittàArrivo)

**[Ferroviana,**

**Ferroviana(X) :- FerrovianaA(X)**

**Ferroviana(X) :-**

**PrenotazioneB(A, B, X, C, D, E, F), PreFerrovianaB(A, G, H, I, L, M, N) ]**

Lo stesso ragionamento fatto per il mapping della relazione *Navale* si applica su questa parte del mapping, usando ora *Prenotazione* e *PreFerroviana*.

---

Globale:

Luogo(**NomeLuogo, NomeCittà**, Tel)

Sorgente A:

Luogo(**NomeLuogo, NomeCittà**, Tel)

Sorgente B:

Aereoporto(**Nome, NomeCittà**)

Porto(**Nome, NomeCittà**)

Stazione(**Nome, NomeCittà**)

**[Luogo,      Luogo(X, Y, Z) :- LuogoA(X, Y, Z)**

**Luogo(X, Y, NULL) :- AereoportoB(X, Y)**

**Luogo(X, Y, NULL) :- PortoB(X, Y)**

**Luogo(X, Y, NULL) :- StazioneB(X, Y) ]**

A differenza della sorgente A nella B l'informazione sui luoghi di interesse sono suddivise su tre relazioni *Aereoporto*, *Porto* e *Stazione*, queste tre relazioni contengono il nome e la città in cui sono situati. L'informazione sul numero di telefono di riferimento non è presente quindi non è possibile ricostruirla.

---

Globale:

Porto(**NomeLuogo, NomeCittà**)

Sorgente A:

NavaleOpera(**NomeCompagnia, NomeLuogo, NomeCittà**)

Sorgente B:

Porto(**Nome, NomeCittà**)

**[Porto,                    Porto(X, Y) :- NavaleOperaA(A, X, Y)  
                              Porto(X, Y) :- PortoB(X, Y) ]**

Al contrario del mapping della relazione *Luogo*, in questo caso abbiamo che la sorgente B contiene l'informazione direttamente mentre dalla sorgente A la si deve ricavare dalla relazione *NavaleOpera* che rappresenta la relazione tra una compagnia navale e il luogo in cui opera il suo servizio.

Globale:

Aereoporto(**NomeLuogo, NomeCittà**)

Sorgente A:

AereaOpera(**NomeCompagnia, NomeLuogo, NomeCittà**)

Sorgente B:

Aereoporto(**Nome, NomeCittà**)

**[Aereoporto,            Aereoporto(X, Y) :- AereaOperaA(A, X, Y)  
                              Aereoporto(X, Y) :- AereoportoB(X, Y) ]**

Discorso equivalente fatto per il mapping su *Porto*.

Globale:

Stazione(**NomeLuogo, NomeCittà**)

Sorgente A:

Sorgente B:

Stazione(**Nome, NomeCittà**)

**[Stazione,    Stazione(X, Y) :- StazioneB(X, Y) ]**

Discorso equivalente fatto per il mapping su *Porto*.

Globale:

NavaleOpera(**NomeCompagnia, NomeLuogo, NomeCittà**)

Sorgente A:

NavaleOpera(**NomeCompagnia, NomeLuogo, NomeCittà**)

Sorgente B:

Prenotazione(**Codice**, DataOraPrenotazione, NomeCompagnia, Prezzo, NomeChiCompra,  
CognomeChiCompra, DataNascitaChiCompra)

PreNavale(**CodPrenotazione**, DataOraPartenza, NomePortoPartenza,  
NomeCittàPartenza, DataOraArrivo, NomePortoArrivo, NomeCittàArrivo)

[NavaleOpera,

**NavaleOpera(X, Y, Z) :- NavaleOperaA(X, Y, Z)**

**NavaleOpera(X, Y, Z) :-**

**PrenotazioneB(A, B, X, C, D, E, F), PreNavaleB(A, G, I, L, H, Y, Z) ]**

Per la sorgente A il mapping è diretto. Nella sorgente B l'informazione necessaria per NavaleOpera deve essere estratta da *Prenotazione* e *PreNavale*.

Globale:

AereaOpera(**NomeCompagnia, NomeLuogo, NomeCittà**)

Sorgente A:

AereaOpera(**NomeCompagnia, NomeLuogo, NomeCittà**)

Sorgente B:

Prenotazione(**Codice**, DataOraPrenotazione, NomeCompagnia, Prezzo, NomeChiCompra,  
CognomeChiCompra, DataNascitaChiCompra)

PreAerea(**CodPrenotazione**, DataOraPartenza, NomeAereoportoPartenza,  
NomeCittàPartenza, DataOraArrivo, NomeAereoportoArrivo, NomeCittàArrivo)

[AereaOpera,

**AereaOpera(X, Y, Z) :- AereaOperaA(X, Y, Z)**

**AereaOpera(X, Y, Z) :-**

**PrenotazioneB(A, B, X, C, D, E, F), PreAereaB(A, G, Y, Z, H, I, L)**

**AereaOpera(X, Y, Z) :-**

**PrenotazioneB(A, B, X, C, D, E, F), PreAereaB(A, G, I, L, H, Y, Z) ]**

Ragionamento equivalente fatto per *NavaleOpera*.

Globale:

Pacchetto(**CodicePrenotazione**, Data, Durata, Albergo, CittàAlbergo)

Sorgente A: - - -

Sorgente B:

Pacchetto(**CodicePrenotazione**, Data, Durata, NomeAlbergo, CittàAlbergo)

[Pacchetto,

**Pacchetto(X, Y, Z, K, A) :- PacchettoB(X, Y, Z, K, A) ]**

Qui ci troviamo nel caso in cui una sorgente (la A) non contiene per niente un concetto, infatti nel globale è presente la relazione *Pacchetto* che rappresenta un viaggio con incluso l'alloggio in un'albergo. Quindi per questa relazione il mapping estrae le informazione sola dalla sorgente B che contiene questi dati.

Globale:

Albergo(NomeAlbergo, NomeCittà, Categoria)

Sorgente A: - - -

Sorgente B:

Albergo(NomeAlbergo, NomeCittà, Categoria)

**[Albergo, Albergo(X, Y, Z) :- AlbergoB(X, Y, Z) ]**

Per il mapping sulla relazione *Albergo* si ragiona come in precedenza per *Pacchetto*.

Globale:

Città(Nome, Nazione)

Sorgente A:

Città(Nome, Nazione)

Sorgente B:

Città(Nome, Nazione)

**[Città, Città(X, Y) :- CittàA(X, Y)  
Città(X, Y) :- CittàB(X, Y) ]**

In questo caso il mapping è diretto.

Globale:

Nazione(Nome)

Sorgente A:

Nazione(Nome)

Sorgente B:

Città(Nome, Nazione)

**[Nazione, Nazione(X) :- NazioneA(X)  
Nazione(X) :- CittàB(Y, X) ]**

Nella sorgente B l'informazione sulle nazioni non è presente in una relazione ma in un campo della relazione *Città*, quindi l'informazione sul nome della nazione può essere ricavato da questo campo.

## **Mapping LAV – DB 5 – Sorgente A**

Per la filosofia seguita alla costruzione dello schema Globale (medo dettagliata), la realizzazione del Mapping LAV è stata più problematica, infatti spesso le informazioni necessarie spesso non erano presenti nel Globale.

Globale:

Cliente(**Nome, Cognome, DataNascita**, Tel)

Sorgente A:

Cliente(Nome, Cognome, DataNascita, **CF**)

**[Cliente, Cliente(X, Y, Z, K) :- Cliente(X, Y, Z, A) ]**

Mapping diretto.

Globale:

Prenotazione(**Codice**, DataOraPrenotazione, DataOraPartenza, DataOraArrivo, Costo, NomeLuogoPartenza, CittàPartenza, NomeLuogoArrivo, CittàArrivo, NomeCompagnia, NomeChiCompra, CognomeChiCompra, DataNascitaChiCompra)

Sorgente A:

Biglietto(**Codice**, Prezzo, Sconto, CFChiCompra, IdViaggio)

**[Biglietto,**

**Biglietto(A, B, C, D, E, F, G) :-**

**Prenotazione(A, F, J, H, B, I, L, M, N, O, D, E, F) ]**

Per fare il mapping della relazione Biglietto ho bisogno di impostare una asserzione che rappresenta una query sullo schema globale in modo tale da avere le informazioni necessarie per la relazione. Dato che il codice della relazione Prenotazione è stato associato al codice della relazione Biglietto, questo fa sì che l'asserzione sia semplice.

Globale:

PerChi(**CodicePrenotazione, Nome, Cognome, DataNascita**)

Sorgente A:

PerChi(**CodiceBiglietto, CF**)

Cliente(Nome, Cognome, DataNascita, **CF**)

La relazione PerChi permette di associare dei clienti alla prenotazione o al biglietto. In entrambi gli schemi abbiamo questo concetto rappresentato ma con una differenza, nello schema globale la chiave di un cliente è il nome, cognome e data di nascita invece nella sorgente è dato dal suo codice fiscale. Inoltre nel globale l'informazione sul codice fiscale

non esiste. A complicare le cose c'è il fatto che questi campi appartengono alle chiavi delle rispettive relazioni.

Usando LAV l'unico modo per mappare queste relazioni è:

**[PerChi, PerChi(A, B) :- PerChi(A, C, D, E) ]**

Per avere un mapping più adeguato abbiamo bisogno di un mapping misto cioè GLAV. Con questa metodologia si possono realizzare asserzioni su viste di entrambe gli schemi. In questo modo dalla sorgente si può estrarre dalla relazione Cliente anche il nome, cognome e data di nascita che possono essere associati ai relativi campi dello schema del globale. La possibile asserzione è la seguente:

**[PerChi, PerChi(A, B), Cliente(N, C, D, B) :- PerChi(A, N, C, D) ]**

Da questo caso è evidente come il mapping GLAV permetta una più facile costruzione del mapping a discapito di una più difficile fase di riformulazione delle query.

Globale:

Compagnia(**Nome**, Telefono)

Sorgente A:

Compagnia(**Nome**, Telefono)

**[CompagniaA, CompagniaA(X, Y) :- Compagnia(X, Y)]**

Mapping diretto.

Globale:

Navale(**Nome**)

Sorgente A:

Navale(**Nome**)

**[NavaleA, NavaleA(X) :- Navale(X)]**

Mapping diretto.

Globale:

Aerea(**Nome**, Nazione)

Sorgente A:

Aerea(**Nome**, Nazione)

**[AereaA, AereaA(X, Y) :- Aerea(X, Y)]**

Mapping diretto.

---

Globale:

Ferroviaria(Nome)

Sorgente A:

Ferroviaria(Nome)

**[FerroviariaA, FerroviariaA(X) :- Ferroviaria(X) ]**

Mapping diretto.

---

Globale:

Luogo(NomeLuogo, NomeCittà, Tel)

Sorgente A:

Luogo(NomeLuogo, NomeCittà, Tel)

**[LuogoA, LuogoA(X, Y, Z) :- Luogo(X, Y, Z) ]**

Mapping diretto.

---

Globale:

NavaleOpera(NomeCompagnia, NomeLuogo, NomeCittà)

Sorgente A:

NavaleOpera(NomeCompagnia, NomeLuogo, NomeCittà)

**[NavaleOperaA, NavaleOperaA(X, Y, Z) :- NavaleOpera(X, Y, Z) ]**

Mapping diretto.

---

Globale:

AereaOpera(NomeCompagnia, NomeLuogo, NomeCittà)

Sorgente A:

AereaOpera(NomeCompagnia, NomeLuogo, NomeCittà)

**[AereaOperaA, AereaOperaA(X, Y, Z) :- AereaOpera(X, Y, Z) ]**

Mapping diretto.

---

Globale:

Città(**Nome**, Nazione)

Sorgente A:

Città(**Nome**, Nazione)

**[CittàA, CittàA(X, Y) :- Città(X, Y) ]**

Mapping diretto.

---

Globale:

Nazione(**Nome**)

Sorgente A:

Nazione(**Nome**)

**[NazioneA, NazioneA(X) :- Nazione(X) ]**

Mapping diretto.

---

Globale:

Prenotazione(**Codice**, DataOraPrenotazione, DataOraPartenza, DataOraArrivo, Costo,  
NomeLuogoPartenza, CittàPartenza, NomeLuogoArrivo, CittàArrivo,  
NomeCompagnia, NomeChiCompra, CognomeChiCompra,  
DataNascitaChiCompra)

Sorgente A:

Viaggio(**IdViaggio**, NomeLuogoPartenza, NomeCittàPartenza, NomeLuogoDestinazione,  
NomeCittàDestinazione, NomeCompagnia, DataOraPartenza)

**[ViaggioA,**

**ViaggioA(A, B, C, D, E, F, G) :-**

**Prenotazione(J, L, G, M, N, B, C, D, E, F, O, P, Q) ]**

Mapping diretto.

## **Mapping LAV – DB 5 – Sorgente B**

Globale:

Cliente(**Nome, Cognome, DataNascita**, Tel)

Sorgente B:

Cliente(**Nome, Cognome, DataNascita**, Tel)

**[ClienteB, ClienteB(X, Y, Z, K) :- Cliente(X, Y, Z, K)]**

Mapping diretto.

---

Globale:

Prenotazione(**Codice**, DataOraPrenotazione, DataOraPartenza, DataOraArrivo, Costo,  
NomeLuogoPartenza, CittàPartenza, NomeLuogoArrivo, CittàArrivo,  
NomeCompagnia, NomeChiCompra, CognomeChiCompra,  
DataNascitaChiCompra)

Sorgente B:

Prenotazione(**Codice**, DataOraPrenotazione, NomeCompagnia, Prezzo, NomeChiCompra,  
CognomeChiCompra, DataNascitaChiCompra)

**[PrenotazioneB,**

**PrenotazioneB(A, B, C, D, E, F, G) :-**

**Prenotazione(A, B, X, Y, D, Z, K, J, L, C, E, F, G) ]**

Tutte le informazioni per lo schema della sorgente sono presenti nel globale.

---

Globale:

Prenotazione(**Codice**, DataOraPrenotazione, DataOraPartenza, DataOraArrivo, Costo,  
NomeLuogoPartenza, CittàPartenza, NomeLuogoArrivo, CittàArrivo,  
NomeCompagnia, NomeChiCompra, CognomeChiCompra,  
DataNascitaChiCompra)

Aerea(**Nome**, Nazione)

Sorgente B:

PreAerea(**CodPrenotazione**, DataOraPartenza, NomeAereoportoPartenza,  
NomeCittàPartenza, DataOraArrivo, NomeAereoportoArrivo, NomeCittàArrivo)

**[PreAereaB,**

**PreAereaB(A, B, C, D, E, F, G) :-**

**Prenotazione(A, M, B, E, N, C, D, F, G, O, P, Q, R), Aerea(O, S) ]**

La relazione PreAerea rappresenta le prenotazioni aeree. Queste informazioni possono essere prese dalle relazioni del globale *Prenotazione* e *Aerea*.

---

Globale:

Prenotazione(**Codice**, DataOraPrenotazione, DataOraPartenza, DataOraArrivo, Costo,  
NomeLuogoPartenza, CittàPartenza, NomeLuogoArrivo, CittàArrivo,  
NomeCompagnia, NomeChiCompra, CognomeChiCompra,  
DataNascitaChiCompra)

Navale(**Nome**)

Sorgente B:

PreNavale(**CodPrenotazione**, DataOraPartenza, NomePortoPartenza,  
NomeCittàPartenza, DataOraArrivo, NomePortoArrivo, NomeCittàArrivo)

**[PreNavaleB,**

**PreNavaleB(A, B, C, D, E, F, G) :-**

**Prenotazione(A, M, B, E, N, C, D, F, G, O, P, Q, R), Navale(O) ]**

La relazione PreNavale rappresenta le prenotazioni navali. Queste informazioni possono essere prese dalle relazioni del globale Prenotazione e Navale.

Globale:

Prenotazione(**Codice**, DataOraPrenotazione, DataOraPartenza, DataOraArrivo, Costo,  
NomeLuogoPartenza, CittàPartenza, NomeLuogoArrivo, CittàArrivo,  
NomeCompagnia, NomeChiCompra, CognomeChiCompra,  
DataNascitaChiCompra)

Ferroviaria(**Nome**)

Sorgente B:

PreFerroviaria(**CodPrenotazione**, DataOraPartenza, NomeStazionePartenza,  
NomeCittàPartenza, DataOraArrivo, NomeStazioneArrivo, NomeCittàArrivo)

**[PreFerroviariaB,**

**PreFerroviariaB(A, B, C, D, E, F, G) :-**

**Prenotazione(A, M, B, E, N, C, D, F, G, O, P, Q, R), Ferroviaria(O) ]**

La relazione PreFerroviaria rappresenta le prenotazioni ferroviarie. Queste informazioni possono essere prese dalle relazioni del globale Prenotazione e Ferroviaria.

Globale:

Aereoporto(**NomeLuogo, NomeCittà**)

Sorgente B:

Aereoporto(**Nome, NomeCittà**)

**[AereoportoB,**

**AereoportoB(A, B) :- Aereoporto(A, B) ]**

Mapping diretto.

Globale:

Porto(**NomeLuogo, NomeCittà**)

Sorgente B:

Porto(**Nome, NomeCittà**)

**[PortoB, PortoB(A, B) :- Porto(A, B) ]**

Mapping diretto.

---

Globale:

Stazione(**NomeLuogo, NomeCittà**)

Sorgente B:

Stazione(**Nome, NomeCittà**)

**[StazioneB, StazioneB(A, B) :- Stazione(A, B) ]**

Mapping diretto.

---

Globale:

Città(**Nome**, Nazione)

Sorgente B:

Città(**Nome**, Nazione)

**[CittàB, CittàB(A, B) :- Città(A, B) ]**

Mapping diretto.

---

Globale:

Prenotazione(**Codice**, DataOraPrenotazione, DataOraPartenza, DataOraArrivo, Costo,  
NomeLuogoPartenza, CittàPartenza, NomeLuogoArrivo, CittàArrivo,  
NomeCompagnia, NomeChiCompra, CognomeChiCompra,  
DataNascitaChiCompra)

PerChi(**CodicePrenotazione, Nome, Cognome, DataNascita**)

Aerea(**Nome**, Nazione)

Sorgente B:

PerChi(**CodicePrenotazioneAereo, Nome, Cognome, DataNascita**)

**[PerChiB,**

**PerChiB(A, R, Q, P) :-**

**PerChi(A, R, Q, P), Prenotazione(A, M, B, E, N, C, D, F, G, O, P, Q, R),**

**Aerea(O, S) ]**

Nella sorgente B l'informazione che indica a chi è rivolta una prenotazione è mantenuta solo per le prenotazioni aeree. Dato che nel globale questa relazione mantiene i riferimenti per qualsiasi tipo di viaggio bisogna estrarre solo le prenotazioni che afferiscono ad una compagnia aerea.

---

Globale:

Pacchetto(CodicePrenotazione, Data, Durata, Albergo, CittàAlbergo)

Sorgente B:

Pacchetto(CodicePrenotazione, Data, Durata, NomeAlbergo, CittàAlbergo)

**[PacchettoB, PacchettoB(A, B, C, D, E) :- Pacchetto(A, B, C, D, E) ]**

Mapping diretto.

---

Globale:

Albergo(NomeAlbergo, NomeCittà, Categoria)

Sorgente B:

Albergo(NomeAlbergo, NomeCittà, Categoria)

**[AlbergoB, AlbergoB(A, B, C) :- Albergo(A, B, C)]**

Mapping diretto.



## Indice

Tesina per il corso di “Seminari di Ingegneria del Software” .....	1
Integrazione dei dati e Mapping.....	2
La problematica dell’integrazione .....	2
Formalizzazione del mapping .....	3
Mapping GAV.....	3
Mapping LAV .....	4
Mapping GLAV .....	4
GAV vs LAV .....	4
Lavoro svolto .....	6
Mapping DB 3 (Invenzioni brevettate) .....	7
Specifiche:.....	7
Diagramma ER – DB 3 – Globale .....	8
Diagramma ER Ristrutturato – DB 3 – Globale .....	9
Schema logico – DB 3 – Globale.....	10
Diagramma ER – DB 3 – Sorgente A .....	11
Diagramma ER Ristrutturato – DB 3 – Sorgente A.....	12
Schema logico – DB 3 – Sorgente A .....	13
Diagramma ER – DB 3 – Sorgente B .....	14
Diagramma ER Ristrutturato – DB 3 – Sorgente B .....	15
Schema logico – DB 3 – Sorgente B.....	16
Mapping GAV – DB 3 .....	17
Mapping LAV – DB 3 – Sorgente A .....	19
Mapping LAV – DB 3 – Sorgente B.....	21
Mapping DB 4 (Ordini di un'azienda) .....	23
Specifiche:.....	23
Diagramma ER – DB 4 – Globale .....	24
Diagramma ER Ristrutturato – DB 4 – Globale .....	25
Schema logico – DB 4 – Globale.....	26
Diagramma ER – DB 4 – Sorgente A .....	27
Diagramma ER Ristrutturato – DB 4 – Sorgente A.....	28
Schema logico – DB 4 – Sorgente A .....	29
Diagramma ER – DB 4 – Sorgente B .....	30
Diagramma ER Ristrutturato – DB 4 – Sorgente B .....	31
Schema logico – DB 4 – Sorgente B.....	32
Mapping GAV – DB 4 .....	33
Mapping LAV – DB 4 – Sorgente A .....	36
Mapping LAV – DB 4 – Sorgente B.....	38
Mapping DB 5 (Agenzia di viaggio).....	40
Specifiche:.....	40
Considerazioni: .....	40
Diagramma ER – DB 5 – Globale .....	41
Diagramma ER Ristrutturato – DB 5 – Globale .....	42
Schema logico – DB 5 – Globale.....	43
Diagramma ER – DB 5 – Sorgente A .....	45
Diagramma ER Ristrutturato – DB 5 – Sorgente A.....	46
Schema logico – DB 5 – Sorgente A .....	47
Diagramma ER – DB 5 – Sorgente B .....	49
Diagramma ER Ristrutturato – DB 5 – Sorgente B .....	50
Schema logico – DB 5 – Sorgente B.....	51

---

Mapping GAV – DB 5 .....	53
Mapping LAV – DB 5 – Sorgente A .....	60
Mapping LAV – DB 5 – Sorgente B.....	64
Indice.....	69