

Tesina per il corso di “Seminario di Ingegneria del Software”

Antonello Scarano

SOMMARIO

Tesina per il corso di “Seminario di Ingegneria del Software”	1
Introduzione ed illustrazione.....	3
Introduzione ed illustrazione.....	3
GAV e LAV in breve	4
Esercizio 1.....	6
Diagramma ER Ristrutturato – DB 1 – Globale	9
Diagramma ER Ristrutturato – DB 1 – Sorgente B.....	13
Schema logico Globale - 1	14
Schema logico A - 1	15
Schema logico B - 1	16
Mapping GAV.....	17
Mapping LAV 1A	18
Mapping LAV 1B	19
Esercizio 2.....	20
Diagramma ER Ristrutturato – DB 2 – Sorgente B.....	28
Schema logico Globale - 2.....	29
Schema logico A - 2.....	30
Schema logico B - 2.....	31
Mapping GAV 2.....	32
Diagramma ER Ristrutturato – DB 6 – Globale	40
Schema logico – DB 6 – Globale.....	42
Diagramma ER – DB 6 – Sorgente A	44
Diagramma ER Ristrutturato – DB 6 – Sorgente A.....	46
Schema logico – DB 6 – Sorgente A	48
Diagramma ER – DB 6 – Sorgente B	50
Diagramma ER Ristrutturato – DB 6 – Sorgente B.....	51
Schema logico – DB 6 – Sorgente B.....	52
Mapping GAV.....	53
Mapping LAV 3A	55
Mapping LAV 3B	56
Osservazioni sui Mapping.....	57
Conclusioni	60

Introduzione ed illustrazione

Il nostro compito è stato quello di integrare basi di dati al fine di chiarire ed approfondire i concetti studiati durante il corso.

Le tecniche di cui ci siamo avvalsi per l'integrazione dei dati sono quelle offerte da GAV e LAV. Il nostro lavoro è consistito nello svolgere separatamente cinque compiti d'esame di Basi di Dati, incontrarci per progettare assieme i rispettivi schemi globali e passare poi alla fase vera e propria di mapping.

L'intento era quella di usare come sorgenti gli schemi che avevamo progettato separatamente per poi integrarli con il globale. Il problema è sorto quando abbiamo constatato che gli schemi che avevamo realizzato erano troppo simili, da non permettere un esercizio valido di integrazione. Ciò deriva dal fatto che i temi d'esame che abbiamo svolto, come del resto tutti quelli tra i quali potevamo scegliere, erano molto dettagliato, lasciando poco spazio a variazioni che poi sarebbe stato interessante analizzare e di conseguenza integrare. Abbiamo pertanto deciso, per meglio inquadrare una situazione più realistica, di assegnarci delle specifiche più lasse e meno dettagliate in modo da far emergere quelle differenze che sarebbero state poi analizzate ed integrate. Pertanto i primi esempi che andremo a fornire sono quelli che abbiamo ricavato dai compiti d'esame che abbiamo svolto, a differenza degli ultimi due che sono quelli che abbiamo inventato noi.

Nell' integrazione dei dati un ruolo importante lo gioca anche la trattazione dei vincoli che si incontrano via via nell' analisi delle varie basi di dati: noi nello sviluppo dei nostri esercizi abbiamo deciso di non ignorare tali vincoli, ma abbiamo lavorato sotto un'assunzione abbastanza forte e cioè che i vari inserimenti nelle tabelle vengono effettuati lecitamente (senza violare in alcun caso i vincoli in questione).

GAV e LAV in breve

GAV e LAV forniscono un modo generale, un meccanismo, per rappresentare le correlazioni tra schema globale e schemi locali, attraverso una sintassi che consente 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. Si definisce, dunque, il sistema di integrazione attraverso *Mapping* formali tra le sorgenti.

E' possibile specificare due 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.

Il tipo di mapping utilizzato incide pesantemente sulla metodologia di elaborazione delle query. Infatti, il passaggio cruciale consiste nel decidere come scomporre la query sullo schema globale in sotto-query sugli schemi locali; ciò dipende fortemente dal significato che si dà alle sorgenti rispetto allo schema globale. Altre problematiche possono nascere in presenza di limitazioni sull'accesso ai dati sorgente.

Nel seguito utilizzeremo la notazione della programmazione logica in quanto più intuitiva per esprimere i concetti di interesse.

Si noti bene che nell'utilizzo dei mapping LAV si suppone di avere già uno schema globale che contiene le informazioni di interesse, e l'obiettivo è quello di far corrispondere i dati nelle sorgenti coinvolte ai dati che ci si aspetta dallo schema globale. Questo è il procedimento inverso del ben più intuitivo mapping GAV. Tuttavia entrambi presentano pregi e difetti:

Pregi e difetti dei mapping LAV:

- Consentono un'alta modularità e riusabilità: se cambia una sorgente è necessario modificare solo la sua definizione
- Facile estensibilità: aggiungere una sorgente al sistema di integrazione consiste solo

nel definire una nuova vista

- Le relazioni interschema tra le sorgenti possono essere dedotte dai mapping
- La valutazione delle query è difficile (è necessaria una riformulazione delle query a run-time)

Pregi e difetti dei mapping GAV:

- Quando una sorgente cambia è necessario modificare il mapping o ripensare lo schema globale.
- L'elaborazione delle query in genere è molto semplice (unfolding)
- E' necessario derivare le relazioni interschema tra le varie sorgenti

Esercizio 1

Si richiede di progettare lo schema concettuale Entità-Relazione di un'applicazione relativa ai collaudi condotti da una azienda internazionale P produttrice di veicoli. L'azienda P in questione vuole tenere traccia di un insieme di informazioni concernenti i collaudi effettuati sui diversi prototipi di veicolo, descritte nel seguito. Ogni prototipo è identificato da un codice ed ha un tempo di sviluppo (in mesi) ed un costo di produzione. Per ogni prototipo l'azienda P ha effettuato uno o più collaudi. Di ogni collaudo, che è relativo ad uno ed un solo prototipo, interessano la data di inizio, l'ora di inizio (in ore e minuti) e la durata (in minuti). Sullo stesso prototipo non possono essere effettuati più collaudi contemporaneamente (ovvero lo stesso giorno alla stessa ora di inizio). I collaudi sono di due tipi: su pista e su banco collaudo. Di ogni collaudo su pista interessa la pista su cui è stato effettuato e la temperatura esterna.

Ogni collaudo su pista è associato ad esattamente un collaudo su banco, ed interessa sapere quale. Ovviamente, il collaudo su pista e l'associato collaudo su banco devono essere relativi allo stesso prototipo. Di ogni collaudo su banco interessa il nome del collaudatore e il banco prova utilizzato, che è unico. Ogni banco prova ha un codice identificativo ed interessa il suo livello di qualità. Inoltre, per ogni banco prova interessano anche le nazioni presso le quali è stato omologato, con l'indicazione dell'anno di omologazione. Le nazioni rilasciano anche autorizzazioni a far circolare i prototipi nei propri territori, e tali autorizzazioni hanno un certo costo. Ciascun prototipo deve avere almeno una autorizzazione. Infine, di ogni nazione interessa il nome, la capitale, e, se disponibile, il numero di telefono per richiedere le autorizzazioni di circolazione dei prototipi.

Le specifiche sono molto stringenti e dettagliate e gli schemi che abbiamo progettato, anche se ognuno per proprio conto, sono risultati assolutamente uguali.

E neppure sullo schema globale è stato possibile apportare nessuna modifica, dal momento che ciò avrebbe significato un inserimento di informazioni che dalle sorgenti non si sarebbe

potuto ricavare in nessun modo; avremmo ottenuto un mapping banale alla stessa maniera di quello che abbiamo ottenuto attenendoci alle specifiche progettuali.

L'unica conseguenza banale che si ricava è chiaramente che in presenza di fonti omogenee il lavoro di mapping è completamente azzerato e che l'informazione in più rispetto alle sorgenti in uno schema globale chiaramente non può essere ricavata dalle sorgenti, lasciando lacunoso il popolamento della base di dati globale.

Diagramma ER – DB 1 – Globale

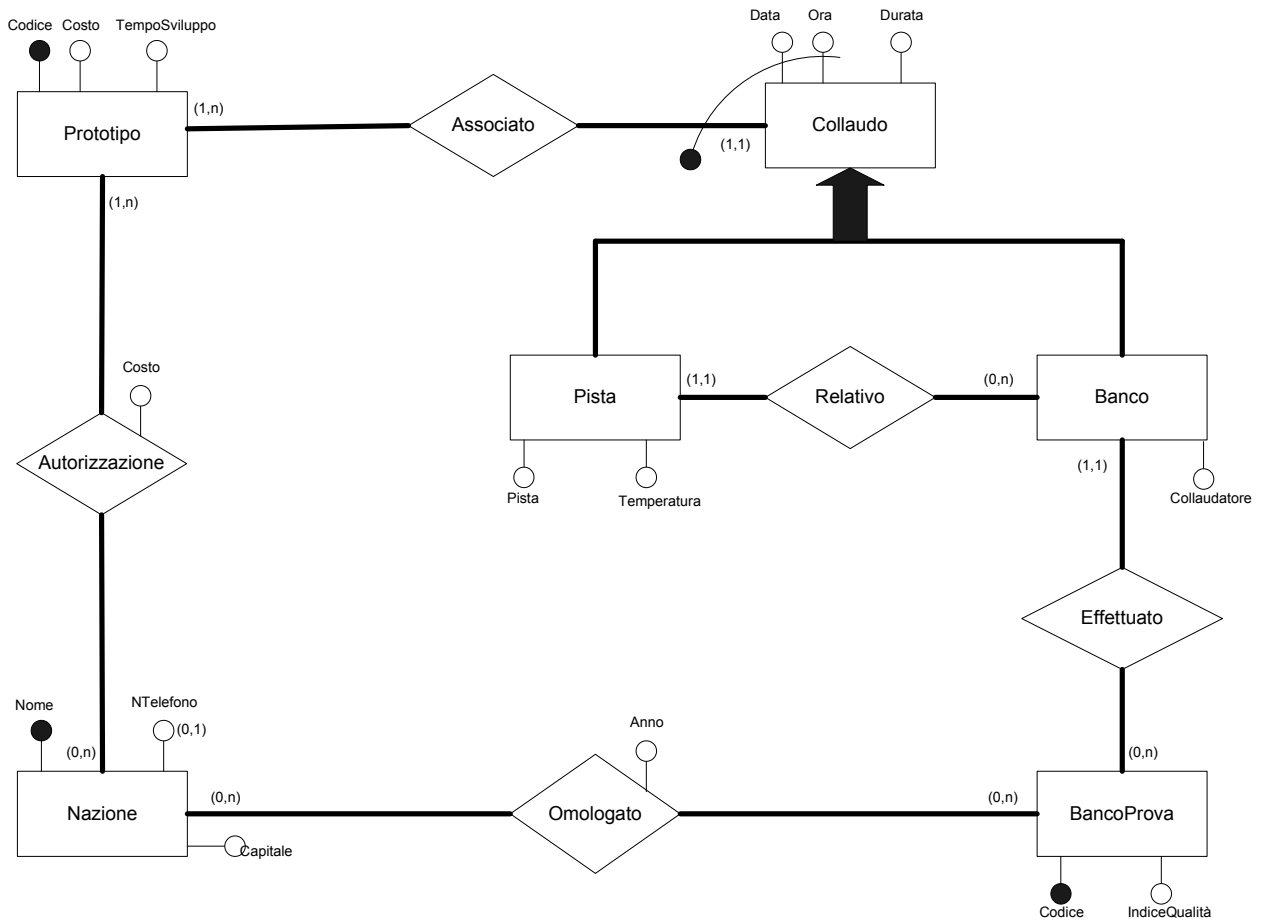


Diagramma ER Ristrutturato – DB 1 – Globale

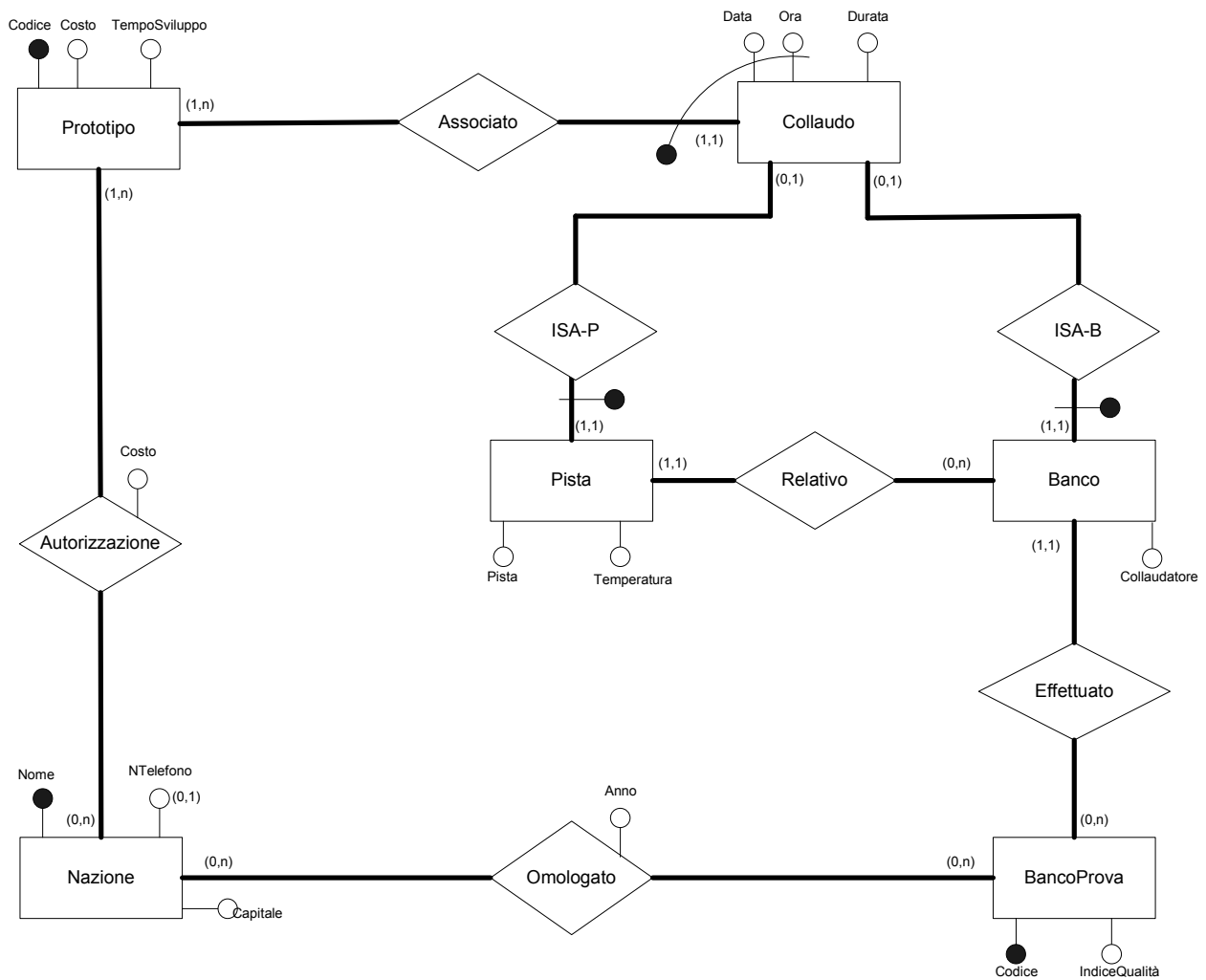


Diagramma ER – DB 1 – Sorgente A

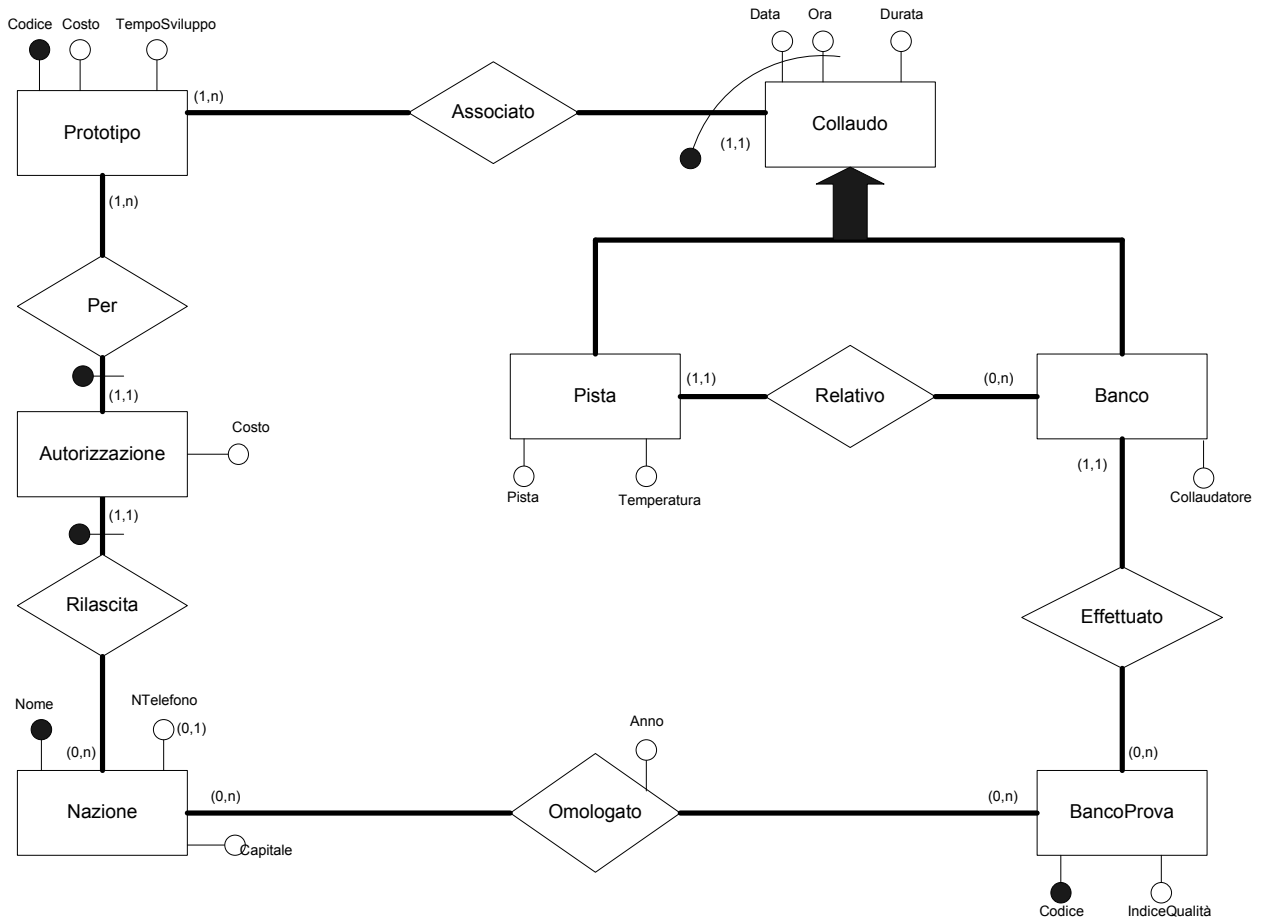


Diagramma ER Ristrutturato – DB 1 – Sorgente A

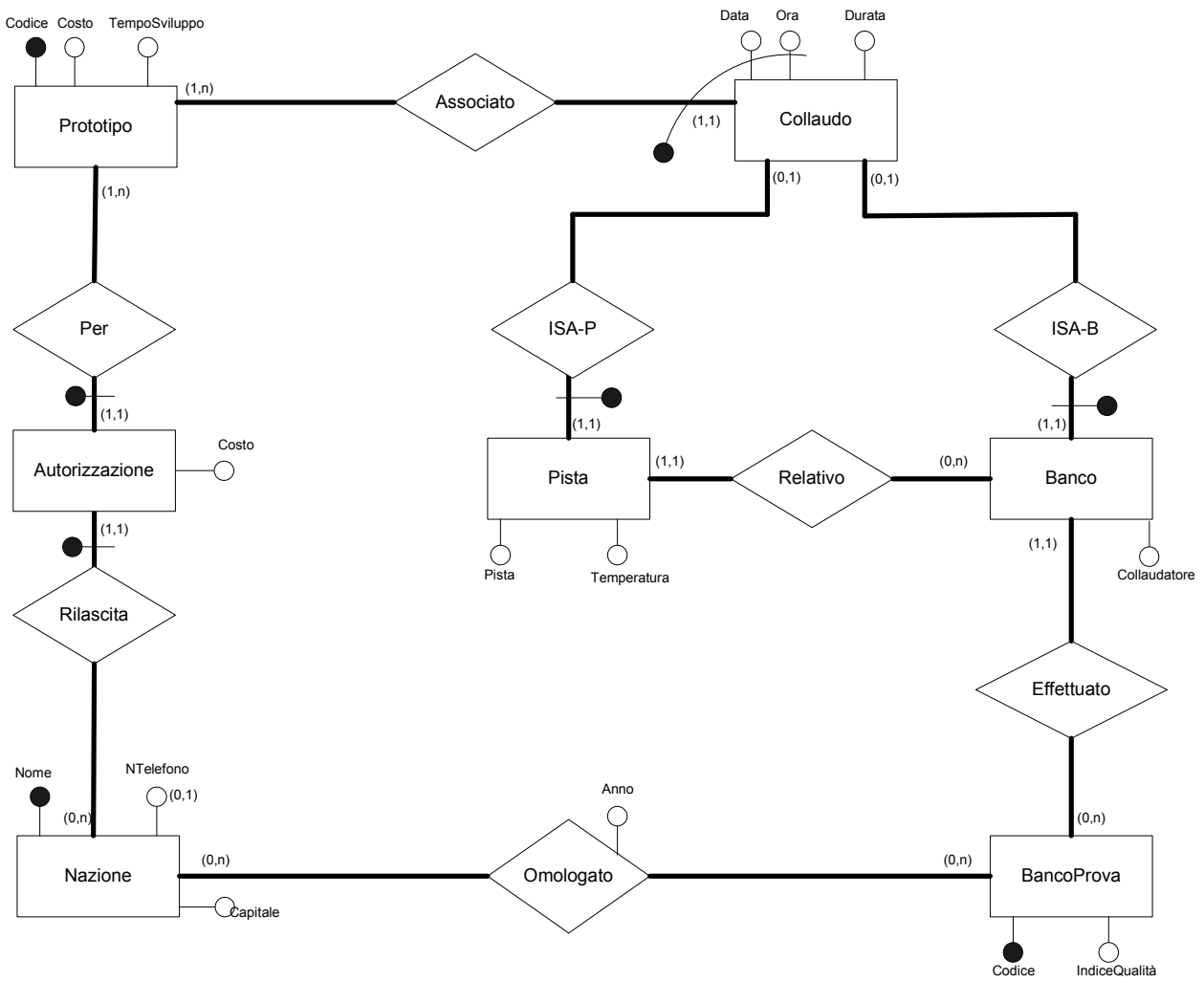


Diagramma ER – DB 1 – Sorgente B

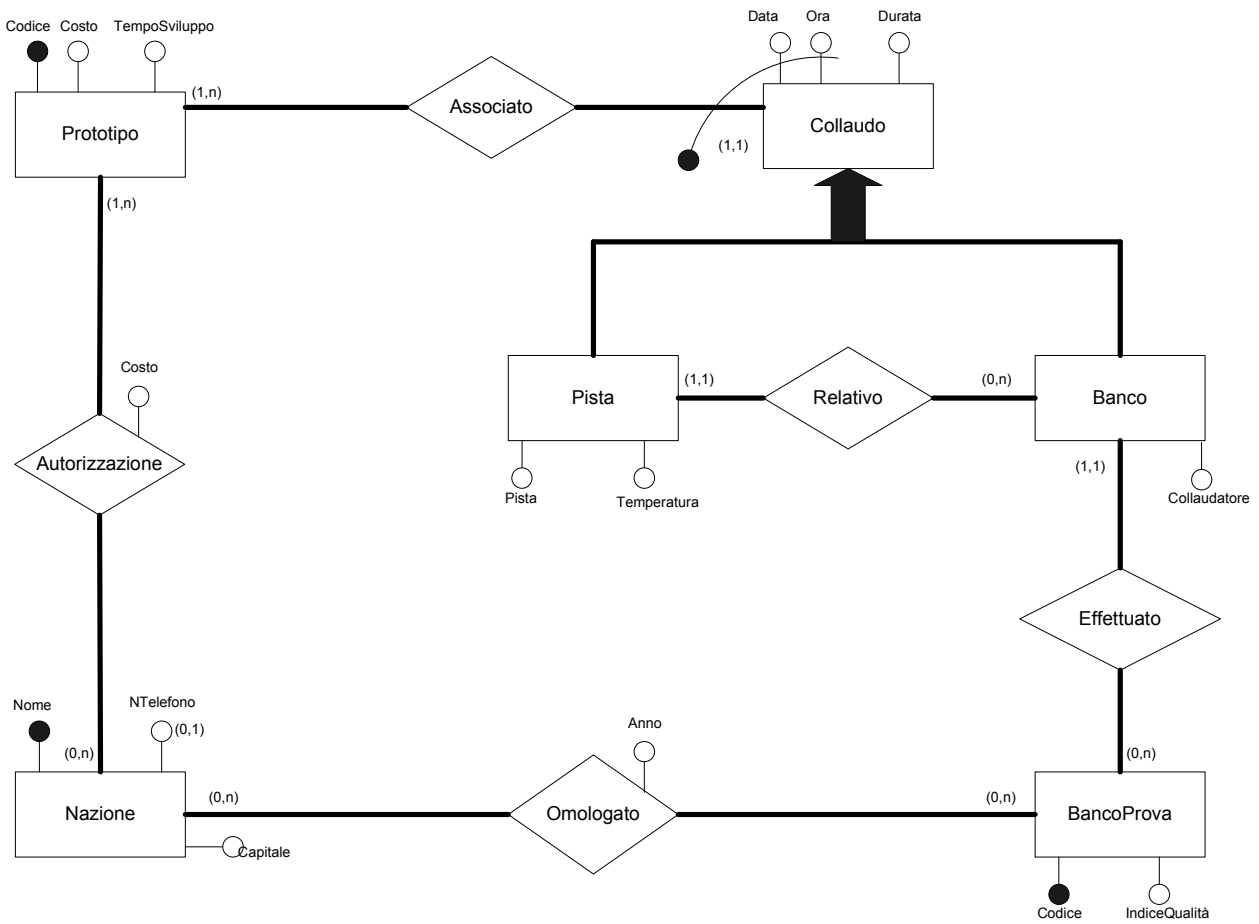
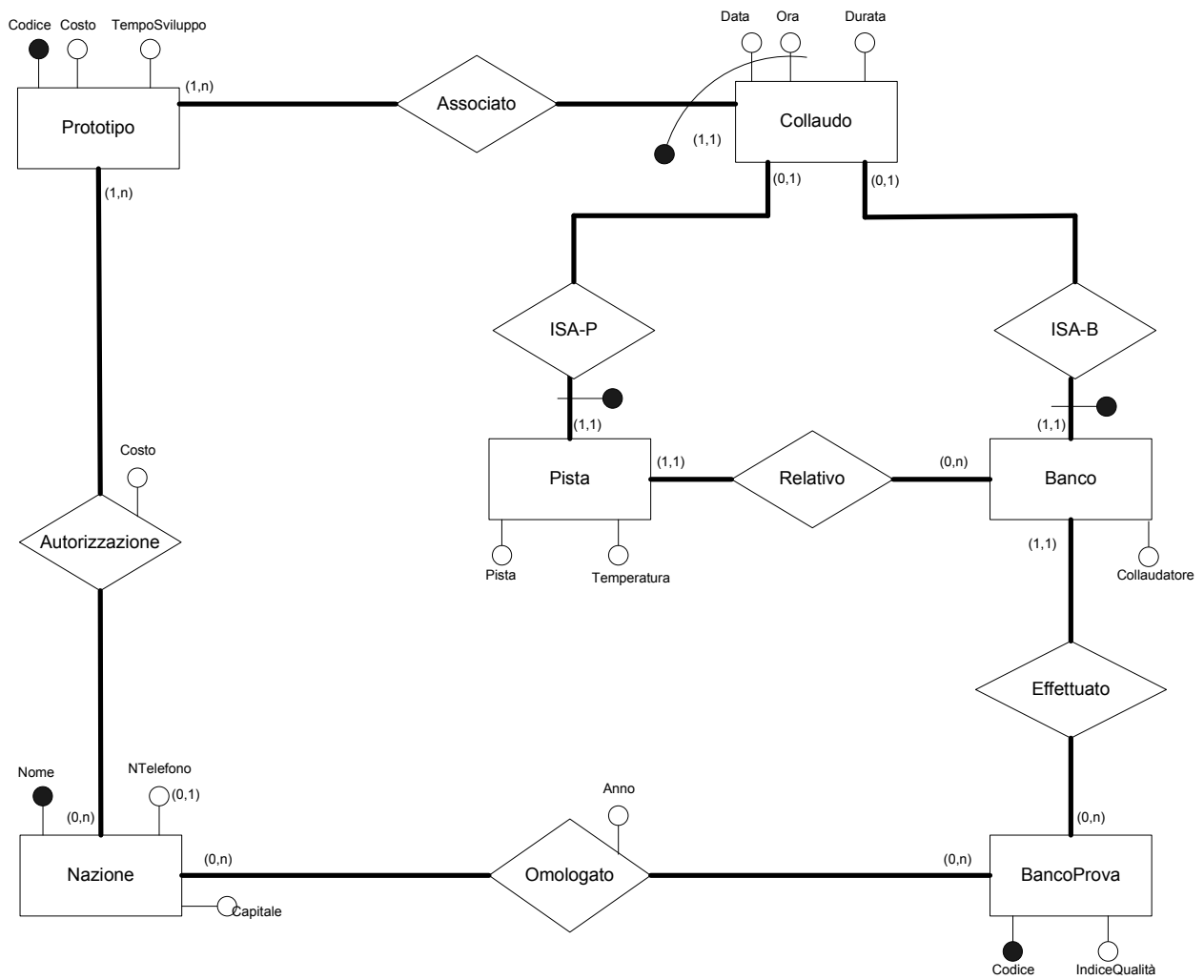


Diagramma ER Ristrutturato – DB 1 – Sorgente B



Schema logico Globale - 1

Prototipo(**Codice**, Costo, TempoSviluppo)

Collaudo(**Prototipo, Data, Ora**, Durata)

Pista(**Prototipo, Data, Ora**, Pista, Temperatura, ProtCollBanco, DataCollBanco, OraCollBanco)

Banco(**Prototipo, Data, Ora**, Collaudatore, Banco)

BancoProva(**Codice**, IndiceQualita)

Omologato(**Banco, Nazione**, Anno)

Nazione(**Nome**, Capitale, NTelefono*)

Autorizzazione(**Nazione, Prototipo**, Costo)

Schema logico A - 1

PrototipoA(Codice, Costo, TempoSviluppo)

CollaudoA(Prototipo, Data, Ora, Durata)

PistaA(Prototipo, Data, Ora, Pista, Temperatura, ProtCollBanco,
DataCollBanco, OraCollBanco)

BancoA(Prototipo, Data, Ora, Collaudatore, Banco)

BancoProvaA(Codice, IndiceQualita)

OmologatoA(Banco, Nazione, Anno)

NazioneA(Nome, Capitale, NTelefono*)

AutorizzazioneA(Nazione, Prototipo, Costo)

Schema logico B - 1

PrototipoB(Codice, Costo, TempoSviluppo)

CollaudoB(Prototipo, Data, Ora, Durata)

PistaB(Prototipo, Data, Ora, Pista, Temperatura, ProtCollBanco,
DataCollBanco, OraCollBanco)

BancoB(Prototipo, Data, Ora, Collaudatore, Banco)

BancoProvaB(Codice, IndiceQualita)

OmologatoB(Banco, Nazione, Anno)

NazioneB(Nome, Capitale, NTelefono*)

AutorizzazioneB(Nazione, Prototipo, Costo)

Mapping GAV

[Prototipo, Prototipo(X,Y,Z) :- PrototipoA(X,Y,Z)
Prototipo(X,Y,Z) :- PrototipoB(X,Y,Z)]

[Collaudo, Collaudo(X,Y,Z,K) :- CollaudoA(X,Y,Z,K)
Collaudo(X,Y,Z,K) :- CollaudoB(X,Y,Z,K)]

[Pista, Pista (X,Y,Z,K,Q,W,H,J) :- Pista A(X,Y,Z,K,Q,W,H,J)
Pista (X,Y,Z,K,Q,W,H,J) :- Pista B(X,Y,Z,K,Q,W,H,J)]

[Banco, Banco(X,Y,Z,K,Q) :- BancoA(X,Y,Z,K,Q)
Banco(X,Y,Z,K,Q) :- BancoB(X,Y,Z,K,Q)]

[BancoProva, BancoProva(X,Y,Z) :- BancoProvaA(X,Y,Z)
BancoProva(X,Y,Z) :- BancoProvaB(X,Y,Z)]

[Omologato, Omologato(X,Y,Z) :- OmologatoA(X,Y,Z)
Omologato(X,Y,Z) :- OmologatoB(X,Y,Z)]

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

[Autorizzazione, Autorizzazione(X,Y,Z) :- AutorizzazioneA(X,Y,Z)
Autorizzazione(X,Y,Z) :- AutorizzazioneB(X,Y,Z)]

Mapping LAV 1A

[Prototipo, PrototipoA(X,Y,Z) :- Prototipo(X,Y,Z)]

[Collaudo, CollaudoA(X,Y,Z,K) :- Collaudo(X,Y,Z,K)]

[Pista, PistaA(X,Y,Z,K,Q,W,H,J) :- Pista(X,Y,Z,K,Q,W,H,J)]

[Banco, BancoA(X,Y,Z,K,Q) :- Banco(X,Y,Z,K,Q)]

[BancoProva, BancoProvaA(X,Y,Z) :- BancoProva(X,Y,Z)]

[Omologato, OmologatoA(X,Y,Z) :- Omologato(X,Y,Z)]

[Nazione, NazioneA(X,Y,Z) :- Nazione(X,Y,Z)]

[Autorizzazione, AutorizzazioneA(X,Y,Z) :- Autorizzazione(X,Y,Z)]

Mapping LAV 1B

[Prototipo, PrototipoB(X,Y,Z) :- Prototipo(X,Y,Z)]

[Collaudo, CollaudoB(X,Y,Z,K) :- Collaudo(X,Y,Z,K)]

[Pista, PistaB(X,Y,Z,K,Q,W,H,J) :- Pista(X,Y,Z,K,Q,W,H,J)]

[Banco, BancoB(X,Y,Z,K,Q) :- Banco(X,Y,Z,K,Q)]

[BancoProva, BancoProvaB(X,Y,Z) :- BancoProva(X,Y,Z)]

[Omologato, OmologatoB(X,Y,Z) :- Omologato(X,Y,Z)]

[Nazione, NazioneB(X,Y,Z) :- Nazione(X,Y,Z)]

[Autorizzazione, AutorizzazioneB(X,Y,Z) :- Autorizzazione(X,Y,Z)]

Esercizio 2

Si richiede di progettare lo schema concettuale Entità-Relazione di un'applicazione relativa alle prenotazioni di posti in teatro. Di ogni teatro interessa il nome (identificativo), l'anno di fondazione, ed il comune in cui è situato. Di ogni comune interessa il nome (unico nella regione), la regione, ed il salario annuo medio dei suoi abitanti. Ogni posto di un teatro è identificato dal numero di sedia unico nell'ambito della fila del teatro stesso, ed è caratterizzato da un numero intero che indica la categoria (platea, galleria, ecc.). Di ogni prenotazione di posti interessa il codice (identificativo), la persona che l'ha effettuata, la data in cui è stata effettuata, e la somma pagata. Ogni prenotazione è o singola o collettiva. Ogni prenotazione singola è relativa ad una data per un teatro, e dà diritto all'assegnazione di esattamente un posto di quel teatro per quella data. Di ogni prenotazione singola interessa il posto di teatro assegnato, la data per la quale è stato assegnato il posto, ed anche la persona (una ed una sola) che occuperà il posto assegnato. Ovviamente un posto di teatro in una certa data può essere assegnato al massimo una volta. Ogni prenotazione collettiva è relativa invece ad un certo numero di posti, tutti per la stessa data e per lo stesso teatro. Di ogni prenotazione collettiva interessa quindi il numero di posti richiesti (non quali, o per quali persone), la data ed il teatro per i quali sono stati richiesti, ed anche lo sconto ottenuto sul prezzo della prenotazione. Di ogni persona interessa il codice fiscale (identificativo), la data di nascita, il sesso, il comune di nascita, ed il comune di residenza, con la data dalla quale la persona vi risiede.

Specifiche così stringenti hanno naturalmente dato luogo da parte nostra a due schemi (che poi abbiamo adottato come sorgenti) estremamente simili tra loro. Da qui, di conseguenza mapping fundamentalmente uguali, a meno di differenze tra nomi di attributi e loro ordine. Abbiamo comunque trovato il modo di riformulare lo schema logico in maniera tale da poter fare delle considerazioni valide su quanto svolto, come meglio specificheremo in seguito.

Diagramma ER – DB 2 – Globale

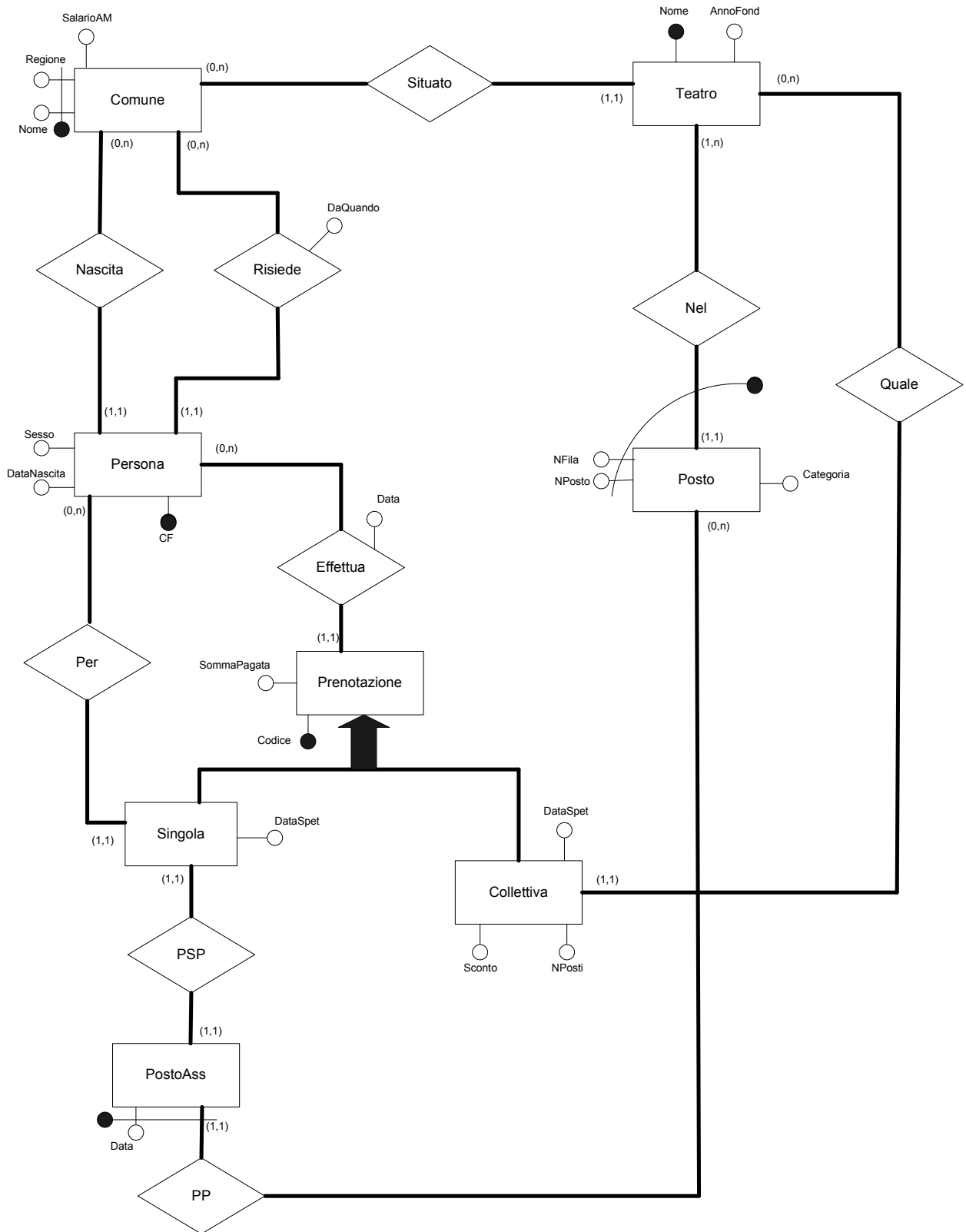


Diagramma ER Ristrutturato – DB 2 – Globale

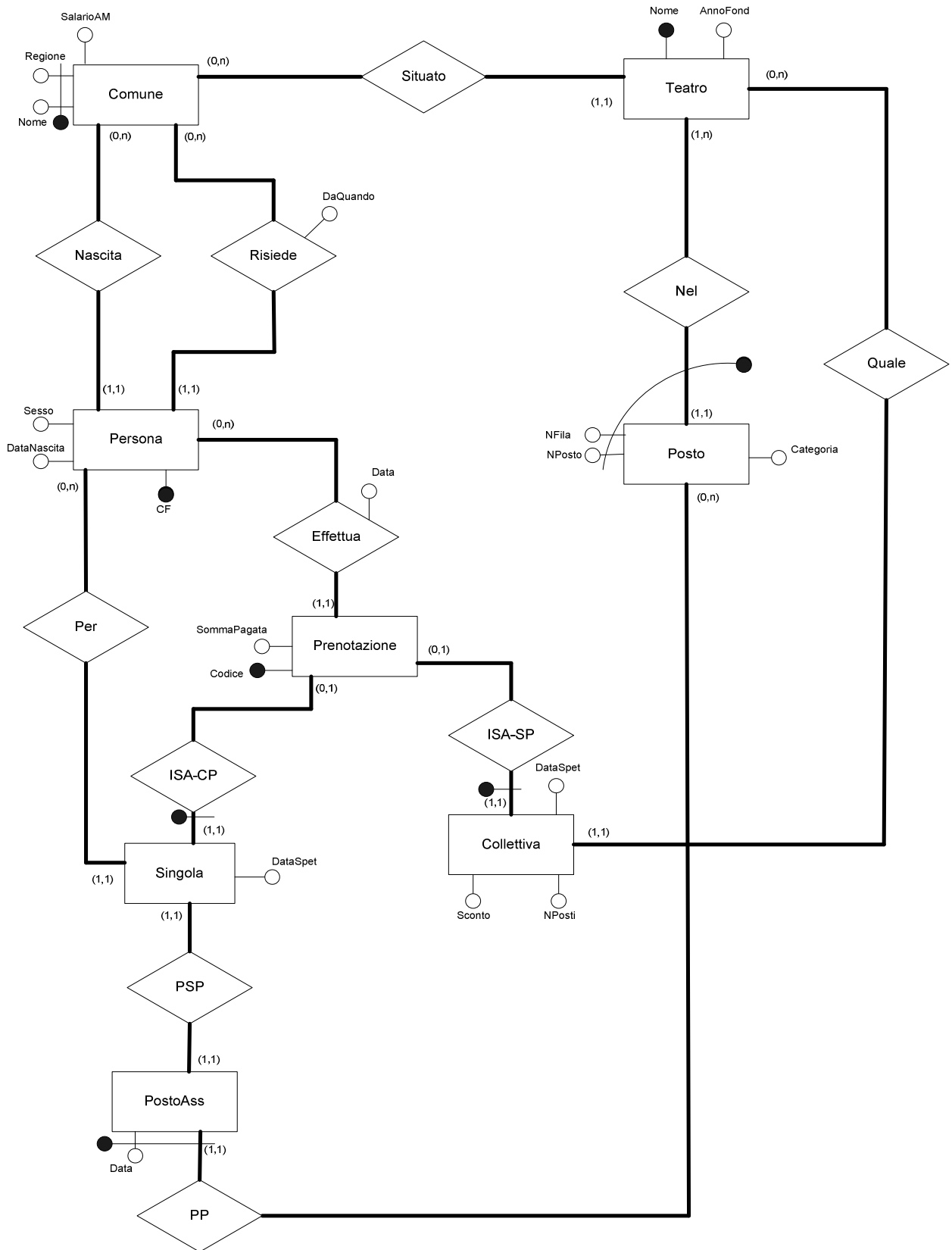


Diagramma ER – DB 2 – Sorgente A

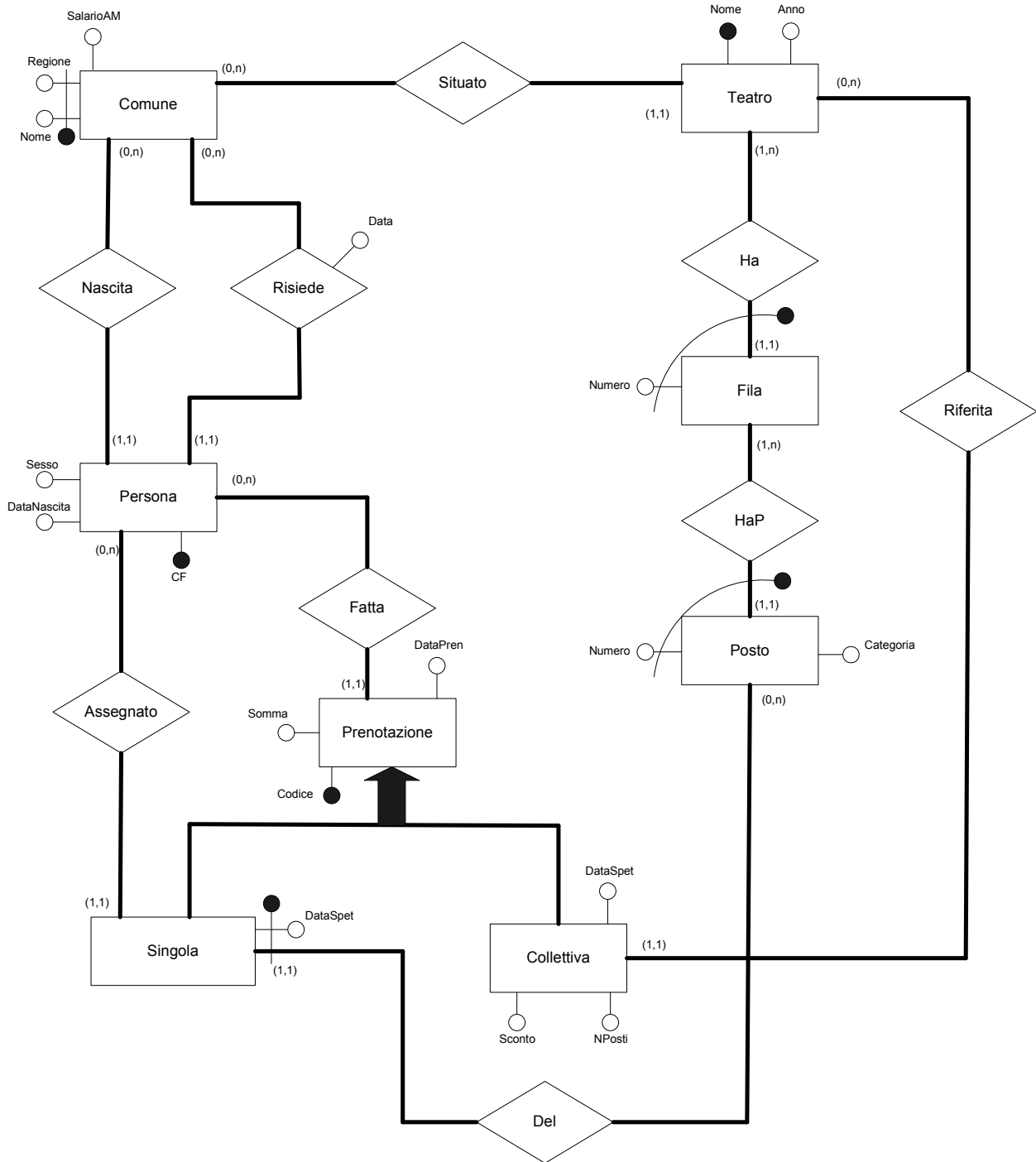


Diagramma ER Ristrutturato – DB 2 – Sorgente A

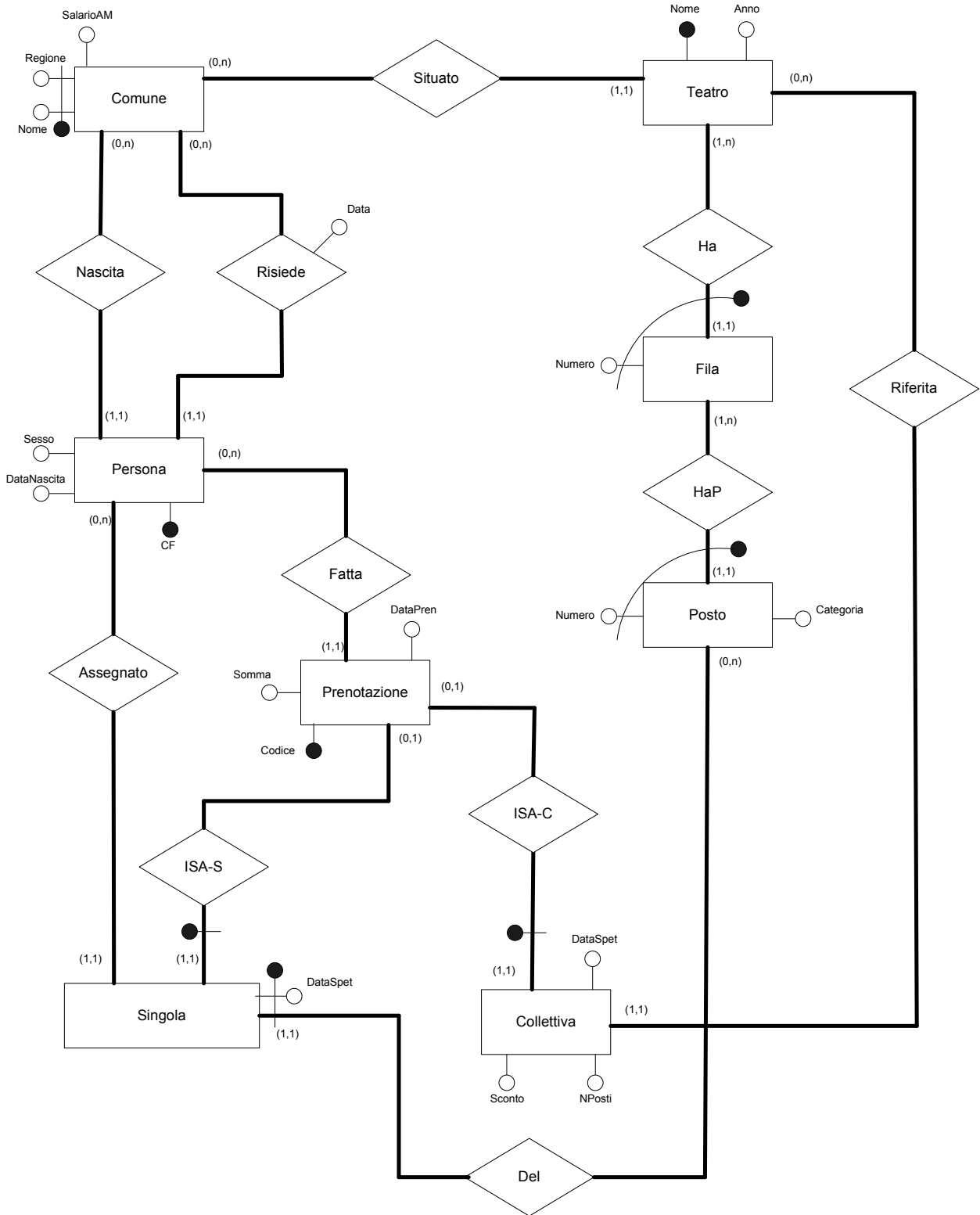


Diagramma ER – DB 2 – Sorgente B

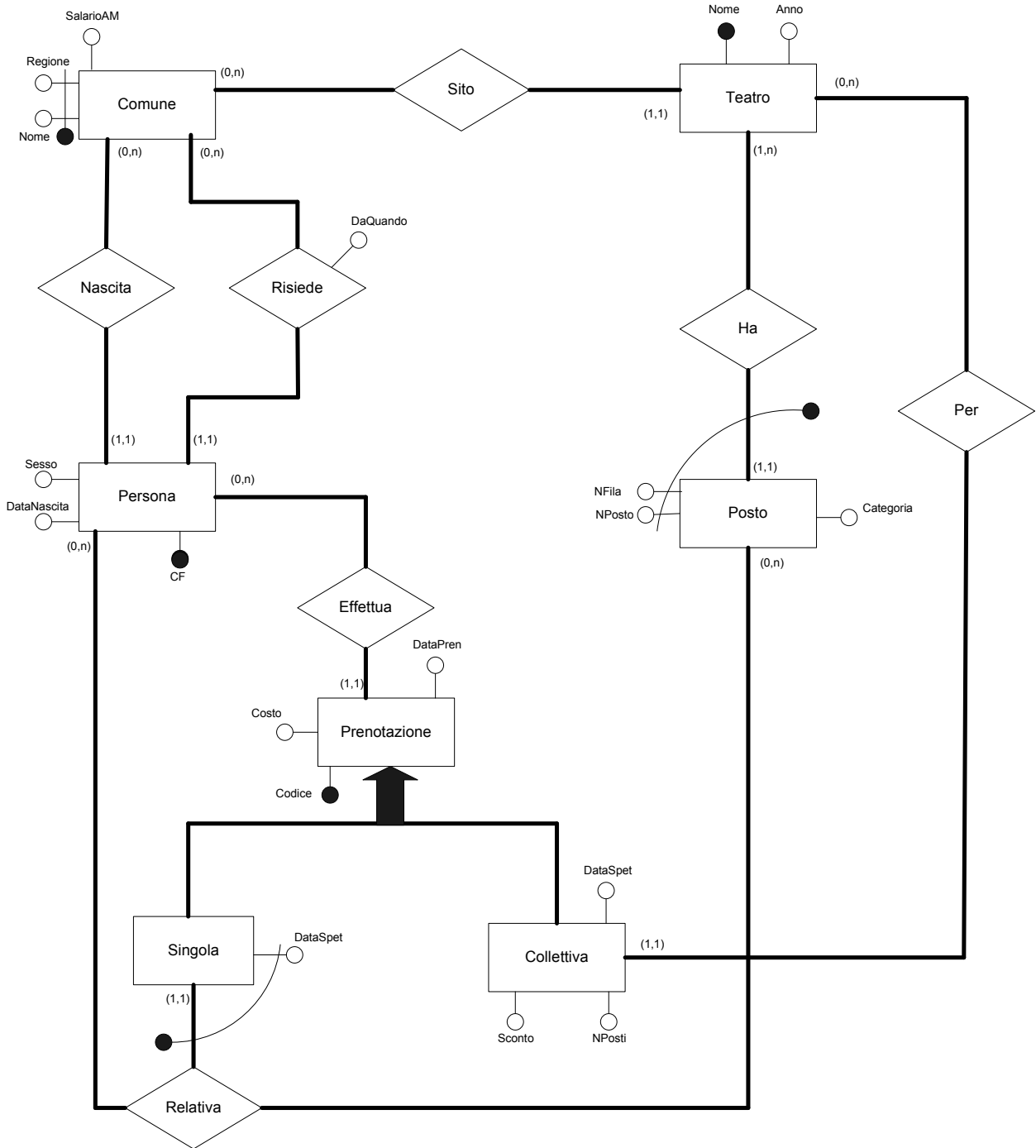
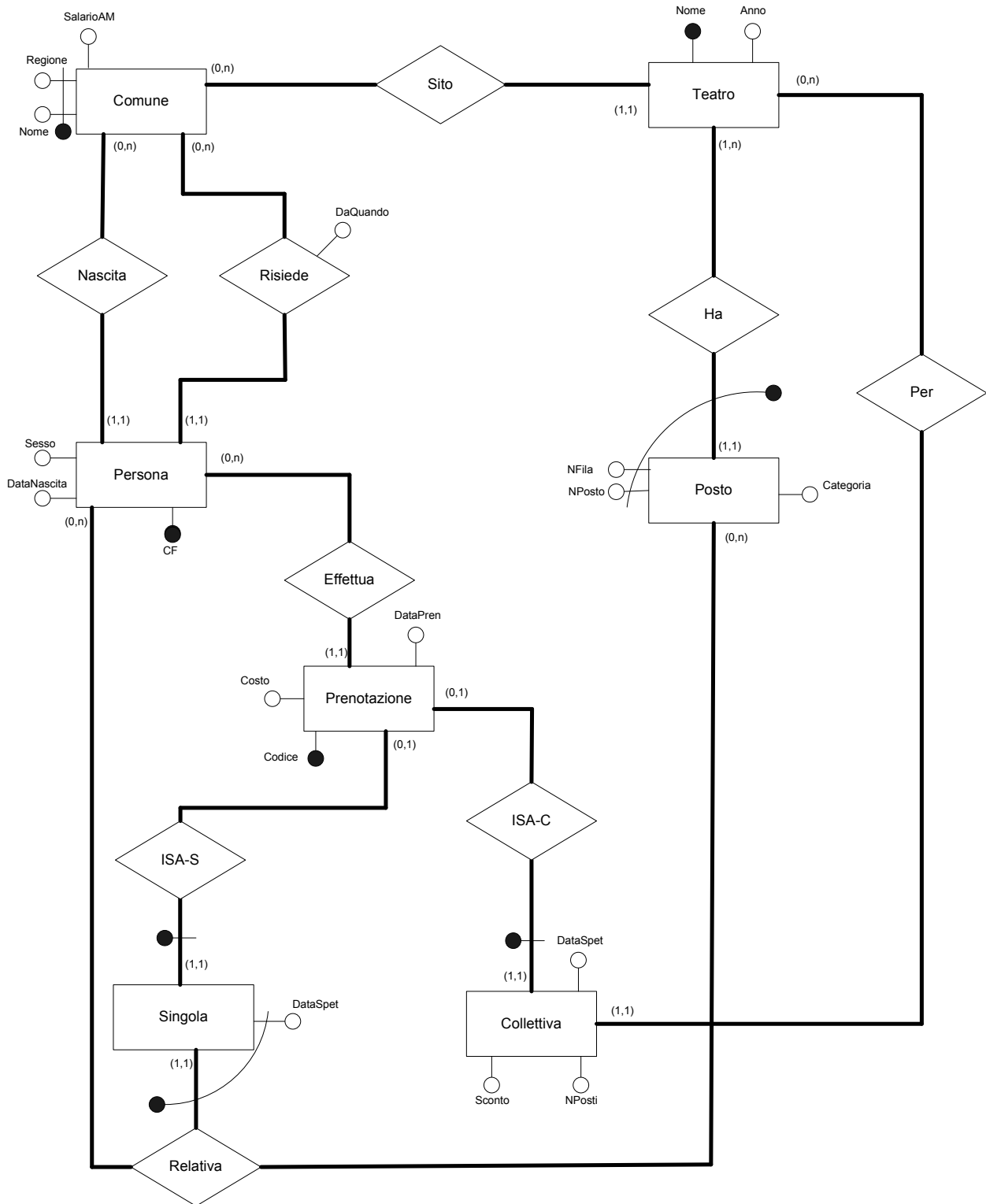


Diagramma ER Ristrutturato – DB 2 – Sorgente B



Schema logico Globale - 2

Prenotazione(Codice, SommaPagata)

Effettua(Prenotazione, Persona, Data)

Collettiva(Codice, Sconto, Data, Num)

Singola(Codice)

Per(Singola, Persona)

Persona(CF, DataN, Sesso)

Risiede(Persona, NomeComune, RegioneComune, Da)

NatoA(Persona, NomeComune, RegioneComune)

Comune(Nome, Regione, SalarioMedio)

Situato(Teatro, NomeComune, RegioneComune)

Teatro(Nome, AnnoFond)

Posto(Num, Fila, Teatro, Categoria)

PostoAssegnato(Num, Fila, Teatro, Data)

PSP(Num, Fila, Teatro, Data, Singola)

Chiave > singola

Quale(Collettiva, Teatro)

Schema logico A - 2

Prenotazione(Codice, Somma, DataPren, CF)

Collettiva(Codice, DataSpet, Sconto, NPosti, NomeTeatro)

Singola(Codice, DataSpet, NumFila, NumPosto, NomeTeatro, CF)

Persona(CF, DataNascita, Sesso, ComuneDiNascita, RegioneDiNascita)

Risiede(CFPersona, Data, NomeComune, RegioneComune)

Comune(Nome, Regione, SalarioMedio)

Teatro(Nome, Anno, NomeComune, RegioneComune)

Fila(Numero, Teatro)

Posto(Numero, NFila, Teatro, Categoria)

Schema logico B - 2

Prenotazione(Codice, Costo, DataPren, CF)

Collettiva(Codice, DataSpet, Sconto, NPosti, NomeTeatro)

Singola(Codice, DataSpet, NFile, NPosto, Teatro, CF)

Persona(CF, DataNascita, Sesso, ComuneDiNascita, RegioneDiNascita)

Risiede(CFPersona, DaQuando, NomeComune, RegioneComune)

Comune(Nome, Regione, SalarioMedio)

Teatro(Nome, Anno, NomeComune, RegioneComune)

Posto(NFile, NPosto, Teatro, Categoria)

Mapping GAV 2

[Prenotazione, Prenotazione(X,Y) :- PrenotazioneA(X,Y,Z,K)
Prenotazione(X,Y) :- PrenotazioneB(X,Y,Z,K)]

[Effettua, Effettua(X,Z,Y) :- PrenotazioneA(X,A,Y,Z)
Effettua(X,Z,Y) :- PrenotazioneB(X,A,Y,Z)]

[Collettiva, Collettiva(X,Z,Y,K) :- CollettivaA(X,Y,Z,K,A)
Collettiva(X,Z,Y,K) :- CollettivaB(X,Y,Z,K,A)]

[Singola, Singola(X) :- SingolaA(X,Y,Z,K,A,B)
Singola(X) :- SingolaB(X,Y,Z,K,A,B)]

[Per, Per(X,B) :- SingolaA(X,Y,Z,K,A,B)
Per(X,B) :- SingolaB(X,Y,Z,K,A,B)]

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

[Risiede, Risiede(X,Z,K,Y) :- RisiedeA(X,Y,Z,K)
Risiede(X,Z,K,Y) :- RisiedeB(X,Y,Z,K)]

[NatoA, NatoA(X,K,A) :- PersonaA(X,Y,Z,K,A)
NatoA(X,K,A) :- PersonaB(X,Y,Z,K,A)]

[Comune, Comune(X,K,A) :- ComuneA(X,K,A)
Comune(X,K,A) :- ComuneB(X,K,A)]

[Situato, Situato(X,A,Y) :- TeatroA(X,K,A,Y)
Situato(X,A,Y) :- TeatroB(X,K,A,Y)]

[Teatro, Teatro(X,K) :- TeatroA(X,K,A,Y)
Teatro(X,K) :- TeatroB(X,K,A,Y)]

[Posto, Posto(X,Y,Z,K) :- PostoA(X,Y,Z,K)
Posto(X,Y,Z,K) :- PostoB(Y,X,Z,K)]

[PostoAssegnato, PostoAssegnato(K,Z,A,Y) :- SingolaA(X,Y,Z,K,A,B)
PostoAssegnato(K,Z,A,Y) :- SingolaB(X,Y,Z,K,A,B)]

[PSP, PSP(K,Z,A,Y,X) :- SingolaA(X,Y,Z,K,A,B)
PSP(K,Z,A,Y,X) :- SingolaB(X,Y,Z,K,A,B)]

[Quale, Quale(X,K) :- CollettivaA(X,Y,Z,K)
Quale(X,K) :- CollettivaB(X,Y,Z,K)]

Mapping LAV 2A

[PrenotazioneA, PrenotazioneA(X,Y,Z,K) :- Prenotazione(X,Y),Effettua(X,Z,K)]

[CollettivaA, CollettivaA(X,Y,Z,K,A) :- Collettiva(X,Z,Y,K),Quale(X,A)]

[SingolaA, SingolaA(X,Y,Z,K,A,B) :- PSP(K,Z,A,Y,X), Per(X,B)]

[PersonaA, PersonaA(X,Y,Z,K,A) :- Persona(X,Y,Z), NatoA(X,,K,A)]

[RisiedeA, RisiedeA(X,Y,Z,K) :- Risiede(X,Z,K,Y)]

[ComuneA, ComuneA(X,Y,Z) :- Comune(X,Y,Z)]

[TeatroA, TeatroA(X,Y,Z,K) :- Teatro(X,Y) ,Situato(X,Z,K)]

[FilaA, FilaA(X,Y) :- Posto(A,X,Y,K)]

[PostoA, PostoA(X,Y,Z,K) :- Posto (X,Y,Z,K)]

Mapping LAV 2B

[PrenotazioneB, PrenotazioneB(X,Y,Z,K) :- Prenotazione(X,Y),Effettua(X,Z,K)]

[CollettivaB, CollettivaB(X,Y,Z,K,A) :- Collettiva(X,Z,Y,K),Quale(X,A)]

[SingolaB, SingolaB(X,Y,Z,K,A,B) :- PSP(K,,Z,A,Y,X), Per(X,B)]

[PersonaB, PersonaB(X,Y,Z,K,A) :- Persona(X,Y,Z), NatoA(X,,K,A)]

[RisiedeB, RisiedeB(X,Y,Z,K) :- Risiede(X,Z,K,Y)]

[ComuneB, ComuneB(X,Y,Z) :- Comuna(X,Y,Z)]

[TeatroB, TeatroB(X,Y,Z,K) :- Teatro(X,Y) ,Situato(X,Z,K)]

[PostoB, PostoB(X,Y,Z,K) :- Posto (X,Y,Z,K)]

Commenti ed osservazioni al Mapping

Il tentativo che abbiamo cercato di fare in questo esercizio, al fine di renderlo caso di studio, è stato quello di ottenere, mediante una progettazione “mirata”, uno schema globale il più dettagliato possibile; il dettagliamento da parte nostra voleva essere reso mediante una minore espressività delle tabelle: abbiamo cercato di suddividere il più possibile l'informazione in più tabelle dello schema. Il tentativo era quello di trarre informazioni utili sul “comportamento” dei mapping in una situazione del genere, sia quello di vagliare la possibilità di trarre informazioni, sulle relazioni intrinseche tra sorgenti e schema globale, dai mapping stessi.

Come si può facilmente capire, la stesura del mapping GAV è immediata dal momento che esso è concepito come una vista sulle sorgenti.

Le tabelle che si cerca di popolare in questo modo, essendo meno espressive di quelle alle quali fanno riferimento, non saranno mai carenti di informazione: la granularità delle tabelle e di ciò che esse vogliono esprimere non rappresenta un ostacolo; semmai il problema è rappresentato dalla difficoltà del reperimento di tali informazioni, qualora esse esistano, oppure dall'assenza completa delle stesse (come abbiamo osservato dal caso di studio dell'esercizio successivo).

Dunque lo scoglio è rappresentato dalla dispersione dell'informazione o dalla sua mancanza. Per il resto come si può leggere dalla stesura del mapping non abbiamo riscontrato nessuna difficoltà. Possiamo dire che abbiamo effettuato un mapping 1 a 1 con semplici query sulle nostre sorgenti.

Il mapping LAV, rappresentando il problema opposto a quello del GAV, pensavamo potesse riservare più sorprese: dal momento che il GAV è stato eseguito con sostanziale semplicità, il LAV poteva offrirci qualche spunto interessante, analizzando lo stesso caso di studio dal punto di vista opposto.

Ma fin dalle prime query abbiamo potuto osservare che si trattava semplicemente di eseguire qualche join sulle tabelle sorgenti.

E non poteva che essere così: come già ci siamo soffermati in precedenza il problema si presenta “solo” in caso di differenza o mancanza di quello espresso tra le tuple delle varie fonti e dello schema globale.

Abbiamo ,però, potuto intuire che un’attenta disamina dei mapping GAV e LAV può fornire informazione su quello che si sta cercando di mappare: ad esempio attraverso i due LAV si può osservare che l’entità PostoAssegnato non ha luogo di esistere, dal momento che in nessuno dei due schemi sorgente, che pur hanno la stessa quantità d’informazione, è stata mappata. Si può anche notare, sempre al momento della stesura del mapping, che l’informazione è “spalmata” su troppe tabelle nello schema globale.

Il compito che ci eravamo prefissi con questo esempio è stato dunque svolto, riuscendo a carpire il modo di estrarre informazioni dall’analisi di un mapping.

Esercizio 3

Questo è uno degli esercizi che abbiamo svolto assegnandoci delle specifiche non troppo dettagliate: ci siamo determinati delle linee guida alle quali attenerci, per evitare di ritrovarci a dover lavorare con schemi troppo differenti o per nulla somiglianti. Abbiamo voluto trattare un' applicazione che gestisse le partite di un torneo di calcio.

Abbiamo concordato di dover trattare degli argomenti quali: la squadra, il giocatore, la partita. Agendo in questa maniera volevamo meglio descrivere quello che poi in realtà succede: il disaccoppiamento della progettazione delle varie fonti ma non solo; la mancanza di riferimenti precisi e la necessità di inseguire vantaggi particolari. Così facendo abbiamo ottenuto il risultato sperato: basi di dati differenti ma riguardanti lo stesso argomento. Su schemi di questo tipo ha un senso andare ad effettuare un' integrazione delle fonti.

Gli schemi realizzati con questa politica sono due, e per meglio affrontare le problematiche che l' integrazione presenta, abbiamo deciso di descrivere uno schema globale nel primo caso (questo che vado a trattare in seguito) molto dettagliato e che volesse contenere tutte le informazioni delle due fonti; nel secondo invece si verifica l'opposto: uno schema globale meno dettagliato delle fonti.

Il nostro intento è stato quello di porre l'attenzione nel primo caso sul mapping GAV, nel secondo su quello LAV.

Ciò accade perché nel primo caso con GAV vogliamo ottenere informazioni da fonti meno dettagliate: in effetti è insolito adottare una politica GAV quando si sa già che le fonti sono meno dettagliate.

Il caso opposto è quello in cui le fonti (più dettagliate) sono viste di uno schema globale in un certo senso più povero di informazione. E' chiaro che il tentativo è puramente didattico e che nella realtà la bravura del progettista sta proprio nello scegliere e valorizzare al meglio la politica che si vuole adottare (a ragion veduta).

Diagramma ER – DB 6 – Globale

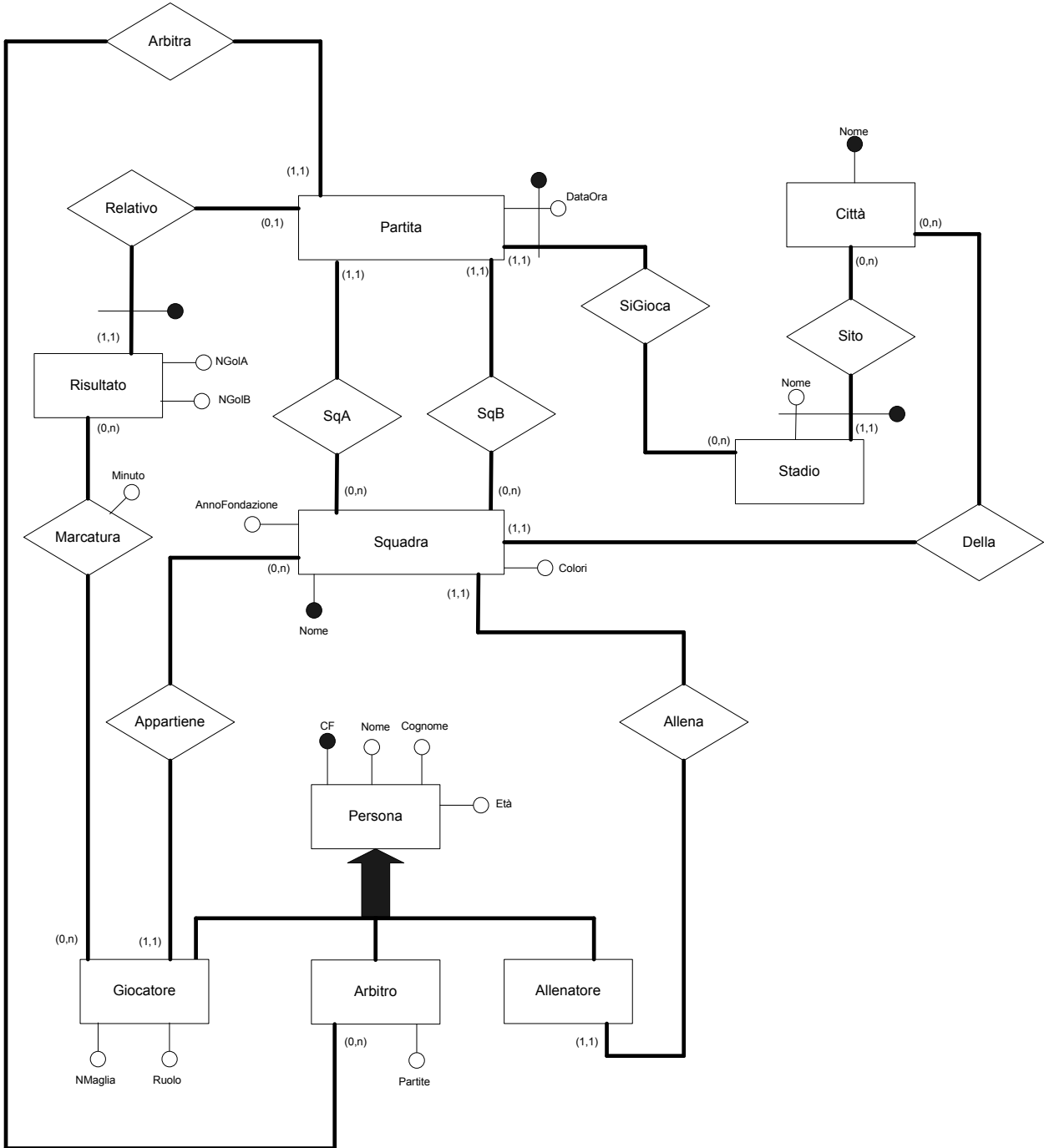
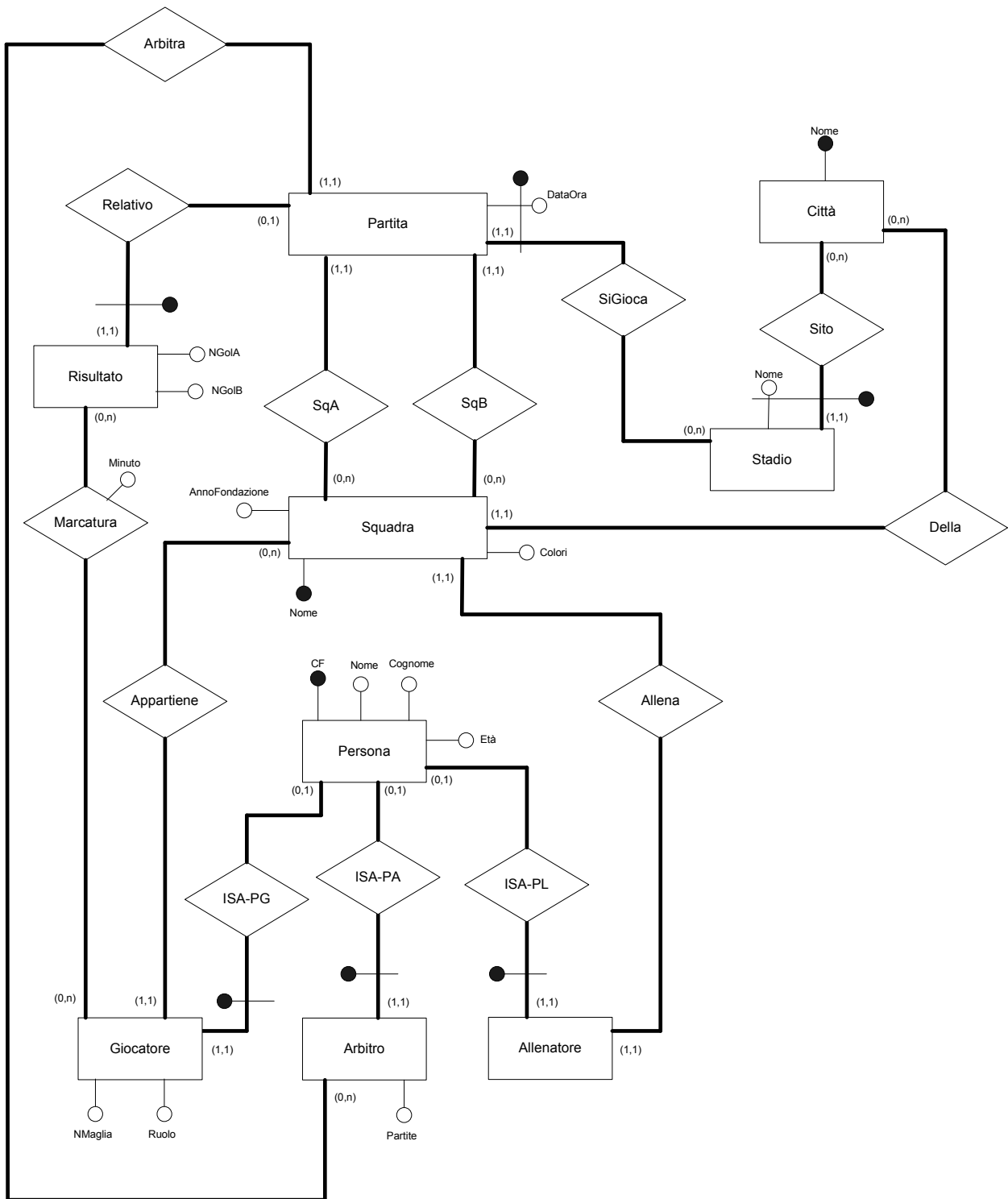


Diagramma ER Ristrutturato – DB 6 – Globale



Vincoli:

- Ogni istanza di Persona partecipa ad esattamente una sola istanza di ISA-PG, ISA-PA o ISA-PL.
- In ogni istanza di marcatura il giocatore interessato deve appartenere ad una squadra della partita a cui si riferisce la marcatura.

Schema logico – DB 6 – Globale

Partita(DataOra, Stadio, Città, SquadraA, SquadraB, Arbitro)

foreign key: Partita[Stadio, Città] \subseteq Stadio[Nome, Città]

foreign key: Partita[SquadraA] \subseteq Squadra[Nome]

foreign key: Partita[SquadraB] \subseteq Squadra[Nome]

foreign key: Partita[Arbitro] \subseteq Arbitro[CF]

Squadra(Nome, AnnoFondazione, Colori, Città, Allenatore)

foreign key: Squadra[Città] \subseteq Città[Nome]

foreign key: Squadra[Allenatore] \subseteq Allenatore[CF]

Risultato(DataOraPartita, StadioPartita, CittàPartita, NGoIA, NGoIB)

foreign key: Risultato[DataOraPartita, StadioPartita, CittàPartita] \subseteq Partita[DataOra, Stadio, Città]

Persona(CF, Nome, Cognome, Età)

Giocatore(CF, NMaglia, Ruolo, NomeSquadra)

foreign key: Giocatore[CF] \subseteq Persona[CF]

foreign key: Giocatore[NomeSquadra] \subseteq Squadra[Nome]

Arbitro(CF, Partite)

foreign key: Arbitro[CF] \subseteq Persona[CF]

Allenatore(CF)

foreign key: Allenatore[CF] \subseteq Persona[CF]

inclusione: Allenatore[CF] \subseteq Squadra[Allenatore]

Stadio(**Nome, Citta**)

foreign key: Stadio[Citta] \subseteq Citta[Nome]

Citta(**Nome**)

Marcatura(**DataOraPartita, StadioPartita, CittàPartita, CFMarcatore, Minuto**)

foreign key: Marcatura[DataOraPartita, StadioPartita, CittàPartita] \subseteq

Risultato[DataOraPartita, StadioPartita, CittàPartita]

foreign key: Marcatura[CFMarcatore] \subseteq Giocatore[CF]

Diagramma ER – DB 6 – Sorgente A

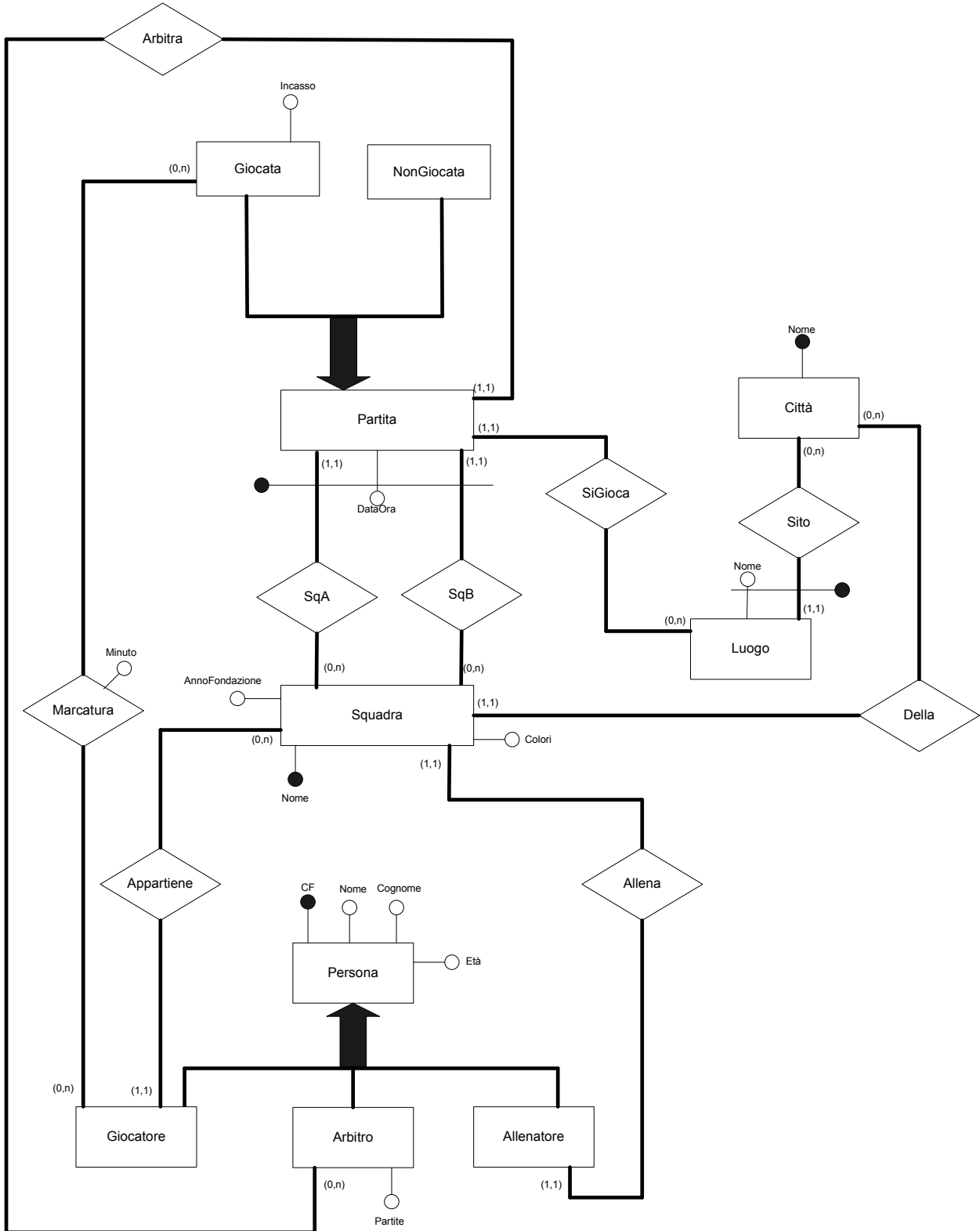
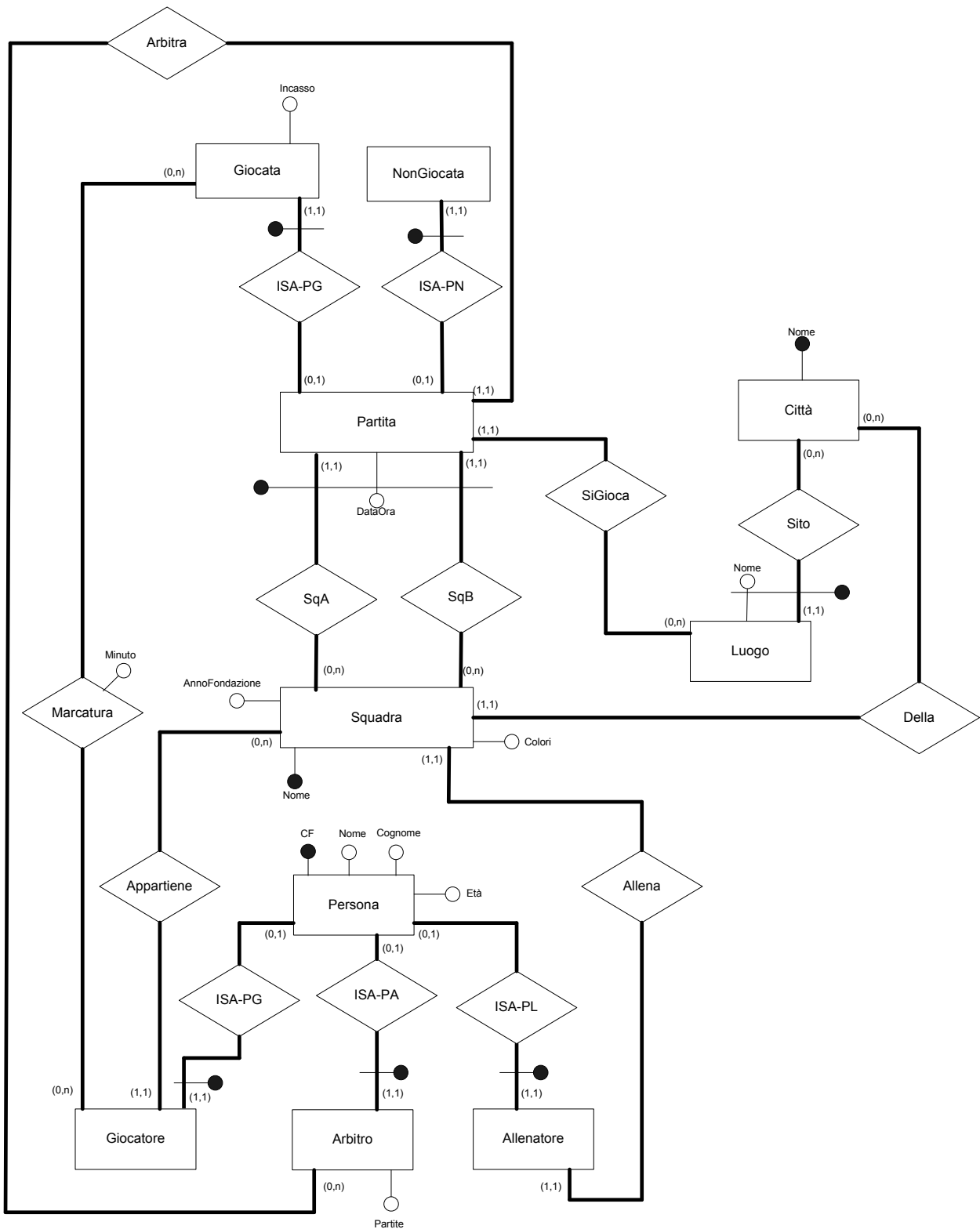


Diagramma ER Ristrutturato – DB 6 – Sorgente A



Schema logico – DB 6 – Sorgente A

Partita(**DataOra**, **SquadraA**, **SquadraB**, Arbitro, Luogo, Citta)

foreign key: Partita[SquadraA] \subseteq Squadra[Nome]

foreign key: Partita[SquadraB] \subseteq Squadra[Nome]

foreign key: Partita[Luogo, Citta] \subseteq Luogo[Nome, Citta]

foreign key: Partita[Arbitro] \subseteq Arbitro[CF]

Squadra(**Nome**, AnnoFondazione, Colori, Città, Allenatore)

foreign key: Squadra[Città] \subseteq Citta[Nome]

foreign key: Squadra[Allenatore] \subseteq Allenatore[CF]

Persona(**CF**, Nome, Cognome, Età)

Giocatore(**CF**, NomeSquadra)

foreign key: Giocatore[CF] \subseteq Persona[CF]

foreign key: Giocatore[NomeSquadra] \subseteq Squadra[Nome]

Arbitro(**CF**, Partite)

foreign key: Arbitro[CF] \subseteq Persona[CF]

Allenatore(**CF**)

foreign key: Allenatore[CF] \subseteq Persona[CF]

inclusione: Allenatore[CF] \subseteq Squadra[Allenatore]

Luogo(**Nome**, **Citta**)

foreign key: Luogo[Citta] \subseteq Citta[Nome]

Citta(**Nome**)

Giocata(**DataOra, SquadraA, SquadraB**, Incasso)

foreign key: Giocata[DataOra, SquadraA, SquadraB] \subseteq Partita[DataOra, SquadraA, SquadraB]

NonGiocata(**DataOra, SquadraA, SquadraB**)

foreign key: NonGiocata[DataOra, SquadraA, SquadraB] \subseteq Partita[DataOra, SquadraA, SquadraB]

Marcatura(**DataOra, SquadraA, SquadraB, Minuto, CFGiocatore**)

foreign key: Marcatura[DataOra, SquadraA, SquadraB] \subseteq Giocata[DataOra, SquadraA, SquadraB]

foreign key: Marcatura[CFGiocatore] \subseteq Giocatore[CF]

Diagramma ER – DB 6 – Sorgente B

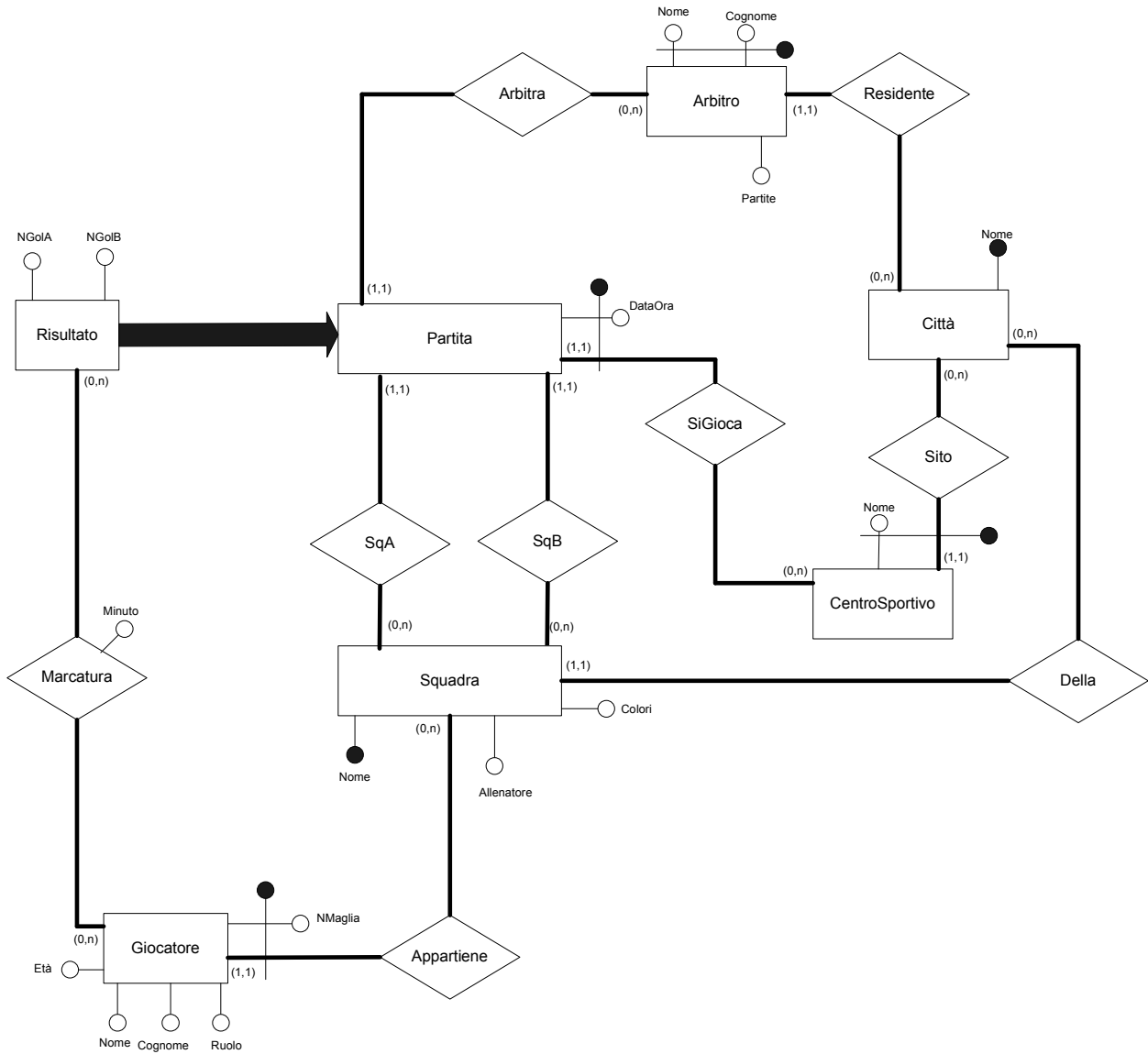
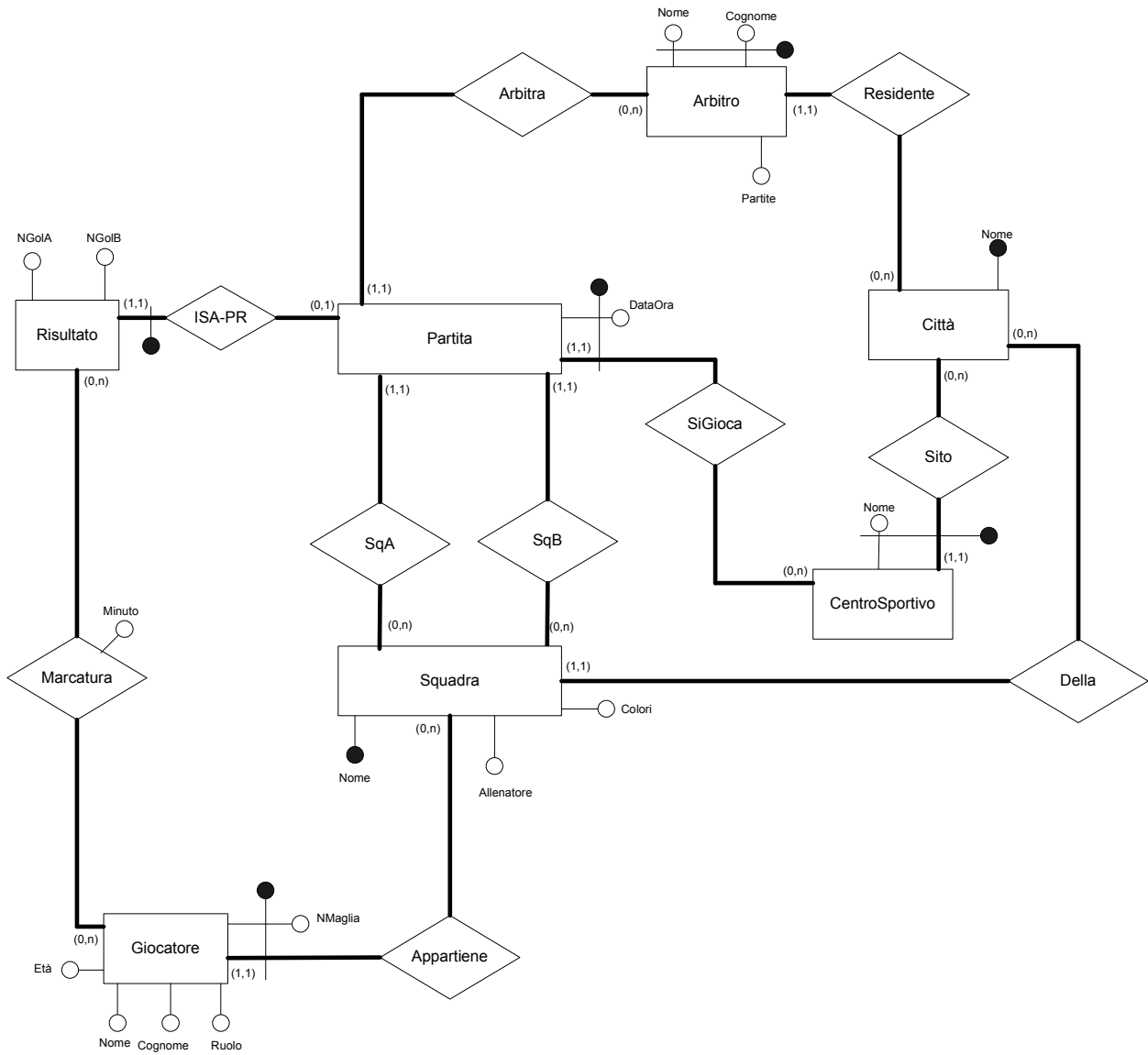


Diagramma ER Ristrutturato – DB 6 – Sorgente B



Schema logico – DB 6 – Sorgente B

Partita(**DataOra, Stadio, Città**, SquadraA, SquadraB, NomeArbitro, CognomeArbitro)

foreign key: Partita[SquadraA] ⊆ Squadra[Nome]

foreign key: Partita[SquadraB] ⊆ Squadra[Nome]

foreign key: Partita[NomeArbitro, CognomeArbitro] ⊆ Arbitro[Nome, Cognome]

Squadra(**Nome**, AnnoFondazione, Colori, Città, Allenatore)

foreign key: Squadra[Città] ⊆ Citta[Nome]

Risultato(**DataOraPartita, StadioPartita, CittàPartita**, NGoIA, NGoIB)

foreign key: Risultato[DataOraPartita, StadioPartita, CittàPartita] ⊆ Partita[DataOra, Stadio, Città]

Giocatore(**NMaglia, NomeSquadra**, Nome, Cognome, Ruolo, Eta)

foreign key: Giocatore[NomeSquadra] ⊆ Squadra[Nome]

Arbitro(**Nome, Cognome**, Partite, Città)

foreign key: Arbitro[Città] ⊆ Citta[Nome]

CentroSportivo(**Nome, Citta**)

foreign key: CentroSportivo[Citta] ⊆ Citta[Nome]

Citta(**Nome**)

Marcatura(**DataOraPartita, StadioPartita, CittàPartita, NMagliaMarcatore, NomeSquadraMarcatore, Minuto**)

foreign key: Marcatura[DataOraPartita, StadioPartita, CittàPartita] ⊆ Risultato[DataOraPartita, StadioPartita, CittàPartita]

foreign key: Marcatura[NMagliaMarcatore, NomeSquadraMarcatore] ⊆ Giocatore[NMaglia, NomeSquadra]

Mapping GAV

1-[Partita, Partita(X, Y, Z, A, B, C) :- PartitaA(X, A, B, C, Y, Z)
Partita(X, Y, Z, A, B, C) :- PartitaB(X, Y, Z, A, B, D, E)]

2-[Squadra, Squadra(X, Y, Z, A, B) :- SquadraA(X, Y, Z, A, B)
Squadra(X, Y, Z, A, B) :- SquadraB(X, Y, Z, A, B)]

3-[Risultato, Risultato(X, O, P, F, G) :- Vedi **
Risultato(X, Y, Z, A, B) :- RisultatoB(X, Y, Z, A, B)]

** Le query sono:

[NumeroGolA: F = Count(
Marcatura(X, Y, Z, M, A),
Partita(X, Y, Z, N, O, P),
Giocatore(A, Y))]

[NumeroGolB: G = Count(
Marcatura(X, Y, Z, M, A),
Partita(X, Y, Z, N, O, P),
Giocatore(A, Z))]

4-[Persona, Persona(X, Y, Z, A) :- PersonaA(X, Y, Z, A)
Persona(X, Y, Z, A) :- GiocatoreB(B, C, Y, Z, D, A)*
Persona(X, Y, Z, A) :- ArbitroB(Y, Z, E, F)*]

5-[Giocatore, Giocatore(X, Y, Z, A) :- GiocatoreA(X, A)
Giocatore(X, Y, Z, A) :- GiocatoreB(Y, A, B, C, Z, P) *]

6-[Arbitro, Arbitro(X, Y) :- ArbitroA(X, Y)
Arbitro(X, Y) :- ArbitroB(A, C, Y, D) *]

*Dalla sorgente B si prendono tutti gli arbitri e giocatori che non sono in A e viene autogenerato un CF fittizio.

7-[Allenatore, Allenatore (X) :- AllenatoreA(X)]

8-[Stadio, Stadio(X, Y) :- LuogoA(X, Y)
Stadio(X, Y) :- CentroSportivoB(X, Y)]

9-[Citta, Citta(X) :- CittaA(X)
Citta(X) :- CittaB(X)]

10

[Marcatura, Marcatura(A, H, I, D, E) :- MarcaturaA(A, B, C, D, E), Partita(A, B, C, G, H, I)
Marcatura(A, B, C, D, E) :- MarcaturaB(A, B, C, G, H, E), Persona(D,G,H,P)*]

*Il codice fiscale deve essere lo stesso generato in precedenza per persona. Deve essere verificato Persona(D,G,H,P).

Mapping LAV 3A

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

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

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

[Giocatore, Giocatore(X, Y) :- Giocatore(X, A, Z, Y)] perdo A,Z

[Arbitro, Arbitro(X, Y) :- Arbitro(X, Y)]

[Allenatore, Allenatore(X) :- Allenatore(X)]

[Luogo, Luogo(X, Y) :- Stadio(X, Y)]

[Citta, Citta(X) :- Citta(X)]

1A [Giocata, Giocata(X, K, J) :- Partita(X, A, B, K, J, C), Risultato(X, A, B, D, E)]

2A [NonGiocata, NonGiocata(X, K, J) :- Partita(X, A, B, K, J, C) / (Partita(X, A, B, K, J, C), Risultato(X, A, B, D, E))]

[Marcatura, Marcatura(X, A, B, J, K) :- Marcatura(X, Y, Z, K, J), Partita(X, Y, Z, A, B, C)]

Mapping LAV 3B

[Partita, Partita(X, Y, Z, A, B, C, D) :- Partita(X, Y, Z, A, B, K), Persona(K, C, D, E)]

1B [Squadra, Squadra(X, Y, Z, K, A) :- Squadra(X, Y, Z, K, B), Persona(B, N, C, E),
A = NomeCognome(N, C)]

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

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

2B [Arbitro, Arbitro(A, B, Y, H) :- Arbitro(X, Y), Persona(X, A, B, C)]

[CentroSportivo, CentroSportivo(A, B) :- Stadio(A, B)]

[Citta, Citta(A) :- Citta(A)]

[Marcatura, Marcatura(A, B, C, F, H, E) :- Marcatura(A, B, C, D, E), Giocatore(D, F, G, H)]

Osservazioni sui Mapping

Nel caso di questo esercizio ho ritenuto opportuno numerare uno per uno i passaggi del mapping GAV per poterli commentare singolarmente, in quanto alcuni necessitano delle spiegazioni particolari.

1: I campi contenuti nelle entità sorgenti sono identici a quelli del globale, perciò a meno di una semplice inversione di ordini di alcuni campi, il mapping è diretto.

Possiamo tuttavia osservare che la chiave della sorgente A, differisce da quella dello schema globale, ma questo non crea nessun problema, del momento che effettuiamo ogni inserimento sotto l'assunzione che sia fatto in maniera lecita senza contravvenire a nessun vincolo.

2: Le entità coinvolte nel mapping sono identiche.

3: A prima impressione il mapping sulla fonte A sembrerebbe almeno incompleto, dal momento che non si possiedono i campi che contengono il Numero dei gol marcati dalle due squadre coinvolte. Effettivamente tale informazione è reperibile all'interno della base di dati A; infatti quello che abbiamo fatto è stato produrre una query sulla sorgente A che contasse il numero di gol marcati dalla squadra A nella specifica partita, e con un'altra query contare quelli marcati dalla squadra B nella stessa partita. Le informazioni in questo modo ottenute le abbiamo usate per integrare quelle già in nostro possesso e mappare per intero tutti i campi dell'entità risultato. Il mapping con la sorgente B è immediato.

4: Il mapping con la sorgente A è immediato in quanto le entità sono identiche.

La problematica che la sorgente B presenta è quella della differenza dei campi tra la sorgente B e lo schema globale delle rispettive chiavi. Dal momento che l'informazione che è chiave all'interno dello schema globale non è reperibile in nessun modo nella sorgente B, non avrebbe alcun senso effettuare un inserimento mancante del campo chiave: abbiamo

pensato, dunque, che l'unica maniera di proseguire, aggirando il problema, se non si vogliono perdere tutte le informazioni contenute nelle entità da mappare, è quella di andare ad auto generare un codice unico che possa sostituire l'informazione della chiave.

L'ultima osservazione da fare è che i dati per popolare l'entità Persona del globale, le andiamo a reperire dalle tabelle Arbitro e Giocatore della sorgente B: è chiaro che noi siamo a conoscenza del significato logico del contenuto delle tabelle.

5: Vale anche qui il ragionamento fatto al punto 4. Il mapping dalla fonte B manca del campo chiave nello schema globale.

Anche nel caso del mapping della fonte A ci troviamo di fronte a mancanza di informazione, ma non di campi chiave. Il problema è facilmente risolvibile in due modi: il primo, ignorando gli inserimenti su quei campi; il secondo di "forzare" a NULL gli inserimenti sui campi dei quali non possediamo informazione, con un mapping del tipo

Giocatore(X, NULL, NULL, A) :- GiocatoreA(X, A).

In realtà un mapping di questo genere non è più un GAV, ma un GLAV. Per questa ragione, anche valutando migliore la seconda ipotesi, per attenerci ad un mapping di tipo GAV abbiamo scelto la prima possibilità.

6: Questo punto presenta le stesse problematiche affrontate nei punti 4 e 5, pertanto la risoluzione sarà la stessa.

7: Il mapping si può effettuare solo sulla prima fonte in quanto, a differenza dei casi delle entità "Arbitro" e "Giocatore", nella seconda fonte non disponiamo dell'informazione necessaria. In realtà la potremmo estrarre dell'entità squadra, ma non abbiamo ritenuto opportuno farlo: questa informazione la possiamo reperire all'interno del globale già nell'entità "Squadra". Dover adottare la politica vista nei casi precedenti in questo caso non è conveniente.

8: Il mapping è diretto in quanto abbiamo le stesse entità da mappare.

9: Il mapping è diretto in quanto abbiamo le stesse entità da mappare.

10: In questo caso il mapping sulla fonte A si ottiene con un semplice join sulle opportune entità. Il secondo mapping, invece, si può ottenere in maniera diretta a patto che il codice fiscale sia lo stesso generato in precedenza per l'inserimento in "Persona": deve essere verificato Persona(D,G,H,P), MarcaturaB(A, B, C, G, H, E) e GiocatoreB(G,H,M,N,O,P).

Sui mapping LAV invece non abbiamo riscontrato nessuna particolarità o difficoltà da dover commentare: si è trattato semplicemente di mappare in maniera diretta le varie entità o semplicemente di effettuare dei join per reperire le informazioni necessarie per effettuare gli inserimenti.

In effetti l'esercizio lo avevamo impostato in maniera tale da soffermare l'attenzione sulla parte GAV. Tuttavia ritengo opportuno spiegare meglio i seguenti passaggi:

1A La partita giocata, concettualmente, è quella per la quale è stato determinato un risultato; per questo motivo nel mapping, anche se in un primo momento sembra di avere a disposizione tutte le informazioni nella tabella "Partita", bisogna andare a fare un controllo sulla tabella "Risultato" al fine di verificare se tale partita è stata effettivamente giocata.

2A Le partite non giocate sono ricavate selezionando tutte le partite che non hanno un risultato associato. Questo viene ottenuto come la differenza tra l'insieme di tutte le partite e le partite giocate.

1B Nella tabella "Squadra" della sorgente B con il campo "Allenatore", intendiamo il nome e cognome dell'allenatore. Questo spiega la necessità del join per reperire tale informazione all'interno della schema globale.

2B Si ha la perdita di informazione: non si mappa il campo "Città" all'interno dello schema globale.

Conclusioni

In definitiva il lavoro che abbiamo svolto ci ha fatto capire la rilevanza che avrebbe nel panorama attuale dell'affinamento delle tecniche che abbiamo utilizzato per portare a termine il nostro compito. In fin dei conti GAV e LAV rappresentano le due facce di una stessa medaglia e lì dove terminano i pregi dell'uno cominciano i difetti dell'altro. Abbiamo capito che essi sono la visione dello stesso problema da due punti di vista opposti, ma che ciò nonostante non sempre forniscono una soluzione al problema che si vuole affrontare; a riprova di quanto detto è stata introdotto il metodo GLAV che rappresenta un ibrido dei due modelli già analizzati. Esso appare il giusto compromesso tra i due.

In generale, da quanto abbiamo potuto osservare, è opportuno studiare sempre bene il panorama nel quale ci si appresta ad effettuare integrazione:

il GAV potremmo definirlo abbastanza sfacciato, dal momento che si assume di poter descrivere lo schema globale come vista sulle sorgenti; a ben pensare ci troviamo di fronte ad un'asserzione realizzabile solo in casi molto fortunati, ma nella maggior parte delle occasioni troppo ottimistica, dal momento che il grado di dettagliamento delle fonti in genere è più basso di quello dello schema globale.

In realtà, anche se il meno intuitivo, il LAV è il modello che più si avvicina alla descrizione della realtà nella quale si opera, almeno dal punto di vista teorico, dal momento che si assume di poter descrivere le fonti come viste sullo schema globale.

Ma neanche in questo caso siamo sicuri di poter operare di fronte ad ogni occorrenza delle basi locali.

Il compromesso è dunque, come già detto il GLAV che già intuitivamente permette una maggiore elasticità nel reperimento delle informazione all'interno delle basi di dati.

Chiaramente fonti troppo eterogenee con difficoltà possono essere integrate; sta dunque alla bravura del progettista evitare di incorrere in queste situazioni o per lo meno limitarle il più possibile. Ho notato, inoltre, che qualora si volesse automatizzare il meccanismo di integrazione dei dati, qualunque fosse la politica scelta, si incorrerebbe in una problematica abbastanza seria: spesso bisogna essere a conoscenza del rapporto effettivo dei dati con la realtà, in modo da poter fare delle assunzioni sui dati che effettivamente si stanno andando ad integrare.