

Basi di dati

Esercitazione 12 – Esercizi sulla ristrutturazione schema logico

Anno Accademico 2023/2024

Maurizio Lenzerini

Esercizio 1 (su ristrutturazione dello schema logico)

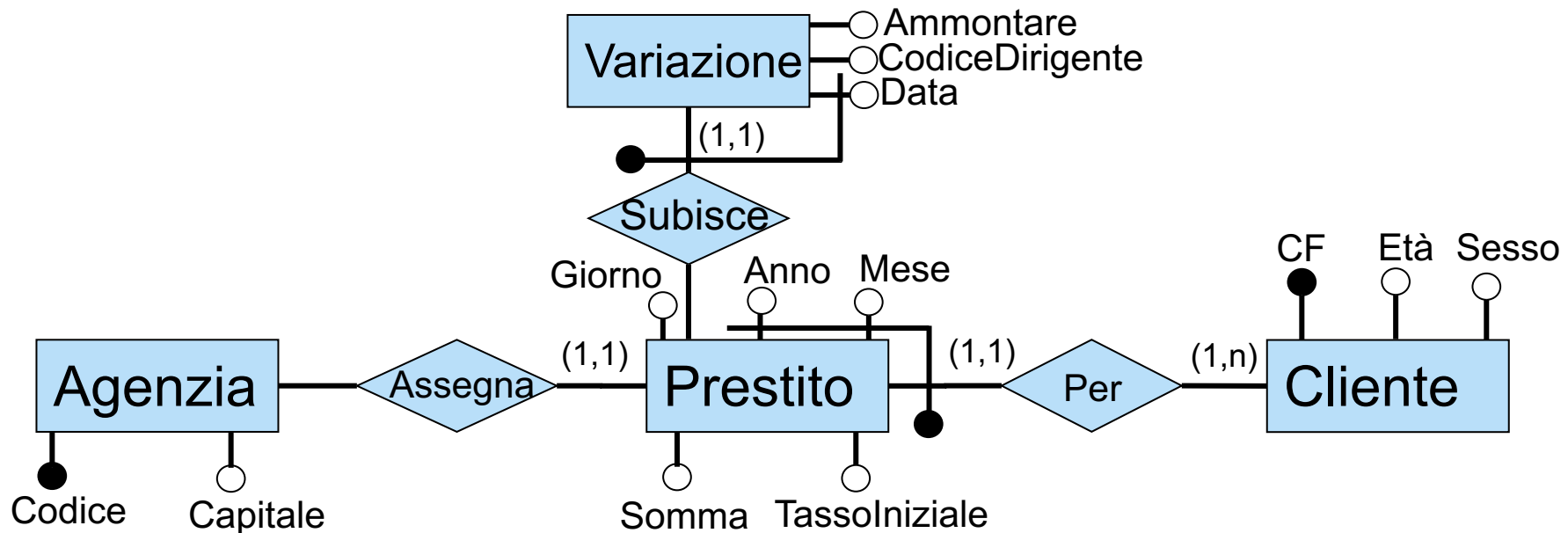
Considerare lo schema logico proveniente dalla traduzione diretta relativo al dominio dei prestiti bancari prodotto per l'esercizio 1 dell'esercitazione 11 ed effettuare la fase di ristrutturazione dello schema logico, tenendo conto delle seguenti indicazioni di progetto:

1. quando si accede ad un prestito si vuole sempre sapere l'agenzia che lo ha assegnato;
2. in più, quando si accede ad un prestito si vuole sempre anche sapere il capitale sociale dell'agenzia che lo ha assegnato.

Esercizio 1: riportiamo la specifica dei requisiti

Si richiede di progettare lo schema concettuale Entità-Relazione relativo al sistema informativo di una banca, in accordo ai seguenti requisiti. Sono di interesse i prestiti di denaro che le agenzie della banca assegnano ai clienti della banca stessa. Sono ovviamente ammessi più prestiti allo stesso cliente, anche dalla stessa agenzia, ma la banca impedisce che ad un cliente venga assegnato, complessivamente dalle agenzie della banca stessa, più di un prestito al mese. Di ogni prestito interessa: la data in cui è stato assegnato, la somma di denaro prestata e il tasso di interesse iniziale. Di ogni agenzia interessa il codice identificativo e il capitale che gestisce. I clienti di interesse sono quelli che hanno avuto almeno un prestito da una qualunque agenzia e di ognuno di essi interessa il codice fiscale, l'età ed il sesso. Il tasso di interesse di un prestito può subire variazioni (positive o negative), ma non più di una al giorno. Al variare del tasso di interesse di un prestito, interessa registrare la data della variazione, il codice del dirigente della banca che ha approvato la variazione e l'ammontare della variazione.

Esercizio 1: riportiamo lo schema concettuale



Frammento del dizionario dei dati

Definizione del dominio degli attributi:

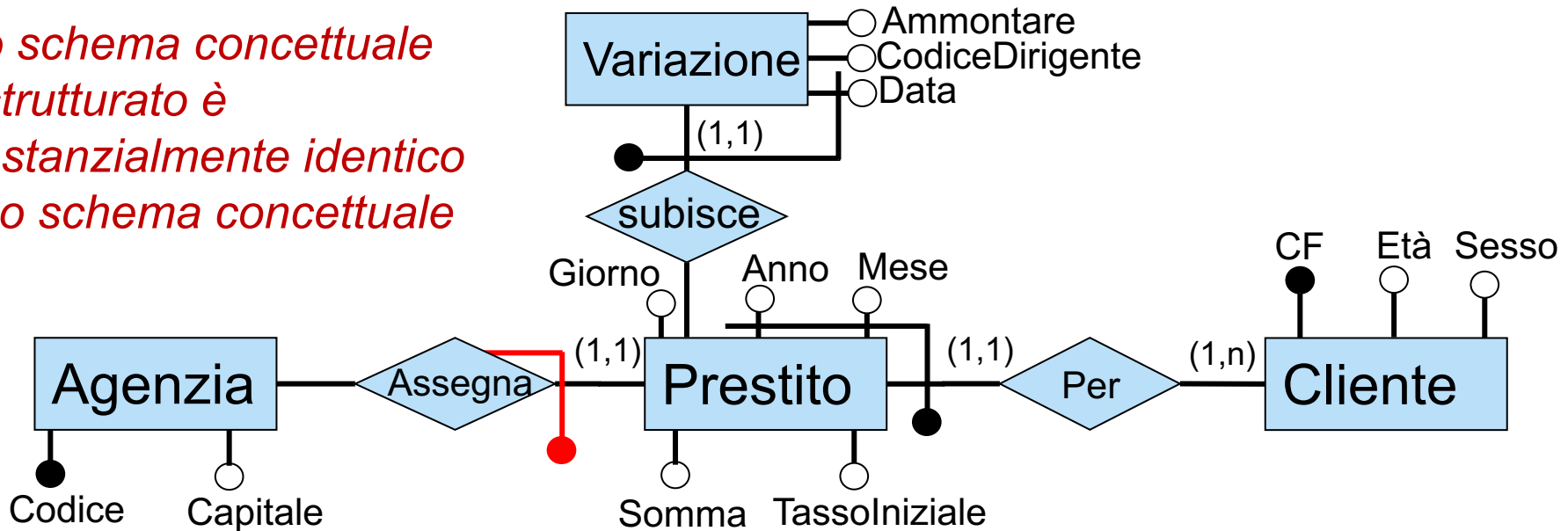
Ammontare: float $\neq 0$, Giorno: integer 1..31, Mese: integer 1..12, Anno: integer

Vincolo esterno:

- Per ogni istanza v di Variazione, tale che $\langle \text{Variazione}:v, \text{Prestito}:p \rangle$   istanza di Subisce, si ha che la data associata a p dagli attributi Anno, Mese e Giorno   inferiore al valore dell'attributo Data associata a v .

Soluzione esercizio 1: riportiamo lo schema concettuale ristrutturato

Lo schema concettuale ristrutturato è sostanzialmente identico allo schema concettuale



Frammento del dizionario dei dati

Definizione del dominio degli attributi:

Ammontare: float $\neq 0$, Giorno: integer 1..31, Mese: integer 1..12, Anno: Intero positivo

Vincolo esterno:

- Per ogni istanza v di Variazione tale che $\langle \text{Variazione}:v, \text{Prestito}:p \rangle$ è istanza di Subisce, si ha che la data associata a p dagli attributi Anno, Mese e Giorno è inferiore al valore dell'attributo Data associata a v .

Esercizio 1: riportiamo lo schema logico ottenuto dalla traduzione diretta

Agenzia(Codice, Capitale)

Cliente(CF, Età, Sesso)

inclusione: Cliente[CF] \subseteq Prestito[Cliente]

Variazione(Data, MeseP, AnnoP, ClienteP, CodiceDirigente, Ammontare)

foreign key: Variazione[MeseP,AnnoP,ClienteP] \subseteq Prestito[Mese,Anno,Cliente]

vincolo di dominio: Ammontare float \neq 0

Prestito(Giorno, Mese, Anno, Cliente, Somma, TassoIniziale)

foreign key: Prestito[Cliente] \subseteq Cliente[CF]

foreign key: Prestito[Mese,Anno,Cliente] \subseteq Assegna[MeseP,AnnoP,ClienteP]

vincoli di dominio: Giorno 1..31, Mese 1..12

Assegna(Agenzia, MeseP, AnnoP, ClienteP)

foreign key: Assegna[Agenzia] \subseteq Agenzia[Codice]

foreign key: Assegna[MeseP,AnnoP,ClienteP] \subseteq Prestito[Mese,Anno,Cliente]

Vincolo esterno:

- Per ogni tupla v di Variazione e per ogni tupla p di Prestito tali che v[MeseP,AnnoP,ClienteP] = p[Mese,Anno,Cliente], si ha che il valore di v[Data] è inferiore alla data corrispondente ai valori p[Giorno], p[Mese] e p[Anno]

Indicazioni di progetto:

1. quando si accede ad un prestito si vuole sempre sapere l'agenzia che lo ha assegnato;
2. in più, quando si accede ad un prestito si vuole sempre anche sapere il capitale sociale dell'agenzia che lo ha assegnato.

Soluzione esercizio 1

1. quando si accede ad un prestito si vuole sempre sapere l'agenzia che lo ha assegnato;

Per rispondere a questa indicazione possiamo operare l'accorpamento di Assegna a Prestito (le due tabelle sono fortemente accoppiate), ottenendo:

Agenzia(Codice, Capitale)

Cliente(CF, Età, Sesso)

inclusione: Cliente[CF] \subseteq Prestito[Cliente]

Variazione(Data, MeseP, AnnoP, ClienteP, CodiceDirigente, Ammontare)

foreign key: Variazione[MeseP, AnnoP, ClienteP] \subseteq Prestito[Mese, Anno, Cliente]

vincolo di dominio: Ammontare float \neq 0

Prestito(Giorno, Mese, Anno, Cliente, Somma, TassoIniziale, Agenzia)

foreign key: Prestito[Cliente] \subseteq Cliente[CF]

foreign key: Prestito[Agenzia] \subseteq Agenzia[Codice]

vincoli di dominio: Giorno 1..31, Mese 1..12, Anno intero positivo

Vincolo esterno:

- Per ogni tupla v di Variazione e per ogni tupla p di Prestito tali che $v[\text{MeseP}, \text{AnnoP}, \text{ClienteP}] = p[\text{Mese}, \text{Anno}, \text{Cliente}]$, si ha che il valore di v[Data] è inferiore alla data corrispondente ai valori p[Giorno], p[Mese] e p[Anno].

Soluzione esercizio 1

2. in più, quando si accede ad un prestito si vuole sempre anche sapere il capitale sociale dell'agenzia che lo ha assegnato.

Per rispondere a questa indicazione possiamo operare l'accorpamento di Prestito ad Agenzia (le due tabelle sono lascamente accoppiate), ottenendo:

Agenzia(Codice, Capitale)

Cliente(CF, Età, Sesso)

inclusione: Cliente[CF] \subseteq Prestito[Cliente]

Variazione(Data, MeseP, AnnoP, ClienteP, CodiceDirigente, Ammontare)

foreign key: Variazione[MeseP,AnnoP,ClienteP] \subseteq Prestito[Mese,Anno,Cliente]

vincolo di dominio: Ammontare float \neq 0

Prestito(Giorno, Mese, Anno, Cliente, Somma, TassoIniziale, Agenzia, Capitale)

foreign key: Prestito[Cliente] \subseteq Cliente[CF]

inclusione: Prestito[Agenzia,Capitale] \subseteq Agenzia[Codice,Capitale]

vincoli di dominio: Giorno 1..31, Mese 1..12, Anno intero positivo

Vincolo esterno:

- Per ogni tupla v di Variazione e per ogni tupla p di Prestito tali che $v[\text{MeseP}, \text{AnnoP}, \text{ClienteP}] = p[\text{Mese}, \text{Anno}, \text{Cliente}]$, si ha che il valore di v[Data] è inferiore alla data corrispondente ai valori p[Giorno], p[Mese] e p[Anno].

Soluzione esercizio 1

Lo schema finale, compreso delle definizioni di viste, è il seguente:

Agenzia(Codice, Capitale)

Cliente(CF, Età, Sesso)

inclusione: $\text{Cliente}[\text{CF}] \subseteq \text{Prestito}[\text{Cliente}]$

Variazione(Data, MeseP, AnnoP, ClienteP, CodiceDirigente, Ammontare)

foreign key: $\text{Variazione}[\text{MeseP}, \text{AnnoP}, \text{ClienteP}] \subseteq \text{Prestito}[\text{Mese}, \text{Anno}, \text{Cliente}]$

vincolo di dominio: Ammontare float $\neq 0$

Prestito(Giorno, Mese, Anno, Cliente, Somma, TassoIniziale, Agenzia, Capitale)

foreign key: $\text{Prestito}[\text{Cliente}] \subseteq \text{Cliente}[\text{CF}]$

inclusione: $\text{Prestito}[\text{Agenzia}, \text{Capitale}] \subseteq \text{Agenzia}[\text{Codice}, \text{Capitale}]$

vincoli di dominio: Giorno 1..31, Mese 1..12, Anno intero positivo

Vincolo esterno: Per ogni tupla v di Variazione e per ogni tupla p di Prestito tali che $v[\text{MeseP}, \text{AnnoP}, \text{ClienteP}] = p[\text{Mese}, \text{Anno}, \text{Cliente}]$, si ha che il valore di $v[\text{Data}]$ è inferiore alla data corrispondente ai valori $p[\text{Giorno}]$, $p[\text{Mese}]$ e $p[\text{Anno}]$

View Assegna(Agenzia, MeseP, AnnoP, ClienteP) as
select Agenzia, MeseP, AnnoP, ClienteP
from Prestito

View PrestitoOriginario(Giorno, Mese, Anno, Cliente, Somma, TassoIniziale) as
select Giorno, Mese, Anno, Cliente, Somma, TassoIniziale
from Prestito

Soluzione esercizio 1

Se nello schema concettuale ci fosse stata la cardinalità minima 1 nel ruolo Agenzia di Assegna (e quindi una inclusione nella tabella Agenzia verso la tabella Assegna) e se le indicazioni di progetto avessero specificato che alle agenzie e al capitale delle agenzie si accede solo quando si accede alle informazioni sui prestiti, si sarebbe potuto decidere di eliminare la tabella Agenzia (a quel punto giudicata inutile), aggiungendo però un vincolo che afferma che il valore dell'attributo Agenzia nella tabella Prestito **determina funzionalmente** il valore dell'attributo Capitale, ossia che non esistono due tuple nella tabella Prestito con stesso valore di Agenzia e valore diverso di Capitale. Lo schema finale sarebbe stato quindi:

Cliente(CF, Età, Sesso)

inclusione: $\text{Cliente}[\text{CF}] \subseteq \text{Prestito}[\text{Cliente}]$

Variazione(Data, MeseP, AnnoP, ClienteP, CodiceDirigente, Ammontare)

foreign key: $\text{Variazione}[\text{MeseP}, \text{AnnoP}, \text{ClienteP}] \subseteq \text{Prestito}[\text{Mese}, \text{Anno}, \text{Cliente}]$

vincolo di dominio: Ammontare float $\neq 0$

Prestito(Giorno, Mese, Anno, Cliente, Somma, TassoIniziale, Agenzia, Capitale)

foreign key: $\text{Prestito}[\text{Cliente}] \subseteq \text{Cliente}[\text{CF}]$

inclusione: $\text{Prestito}[\text{Agenzia}, \text{Capitale}] \subseteq \text{Agenzia}[\text{Codice}, \text{Capitale}]$

vincoli di dominio: Giorno 1..31, Mese 1..12, Anno intero positivo

dipendenza funzionale: $\text{Agenzia} \rightarrow \text{Capitale}$, ossia non esistono due tuple con stesso valore di Agenzia e diverso valore di Capitale.

Vincolo esterno: Per ogni tupla v di Variazione e per ogni tupla p di Prestito tali che $v[\text{MeseP}, \text{AnnoP}, \text{ClienteP}] = p[\text{Mese}, \text{Anno}, \text{Cliente}]$, si ha che il valore di $v[\text{Data}]$ è inferiore alla data corrispondente ai valori $p[\text{Giorno}]$, $p[\text{Mese}]$ e $p[\text{Anno}]$

View Agenzia(Codice, Capitale) as `select Agenzia, Capitale from Prestito`

View Assegna(Agenzia, MeseP, AnnoP, ClienteP) as `select Agenzia, MeseP, AnnoP, ClienteP from Prestito`

View PrestitoOriginario(Giorno, Mese, Anno, Cliente, Somma, TassoIniziale) as

`select Giorno, Mese, Anno, Cliente, Somma, TassoIniziale from Prestito`

Esercizio 2 (su ristrutturazione dello schema logico)

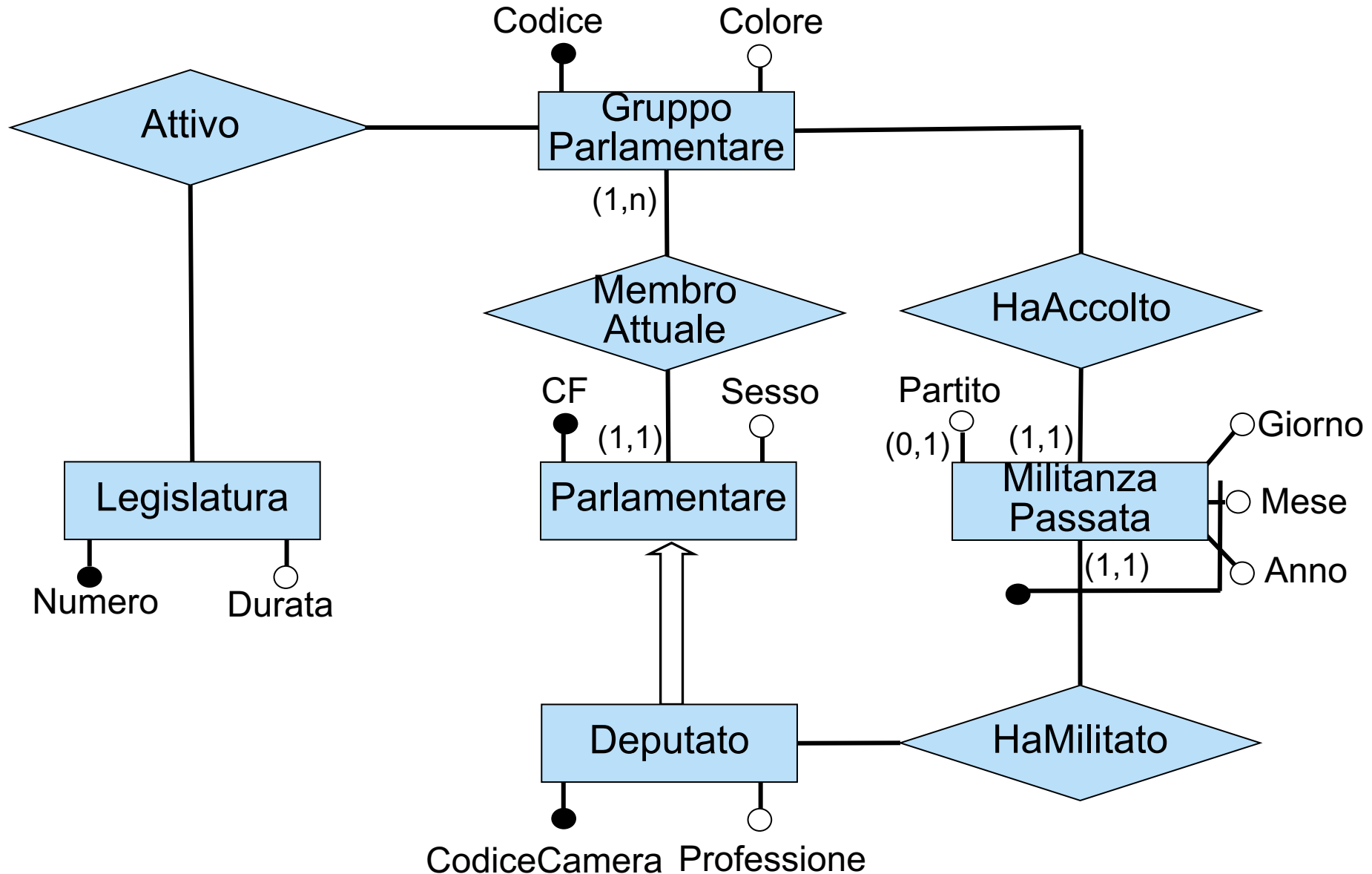
Considerare lo schema logico proveniente dalla traduzione diretta relativo al dominio dei **gruppi parlamentari** prodotto per l'esercizio 4 dell'esercitazione 10 ed effettuare la fase di ristrutturazione dello schema logico, tenendo conto delle seguenti indicazioni di progetto:

1. quando si accede ad un parlamentare si vuole sempre sapere il Gruppo parlamentare a cui appartiene;
2. quando si accede ad un deputato si vuole sempre sapere il suo codice fiscale ed il sesso, e quando si accede ad un parlamentare si vuole sempre sapere se è un deputato e, in caso, il suo codice camera la sua professione.

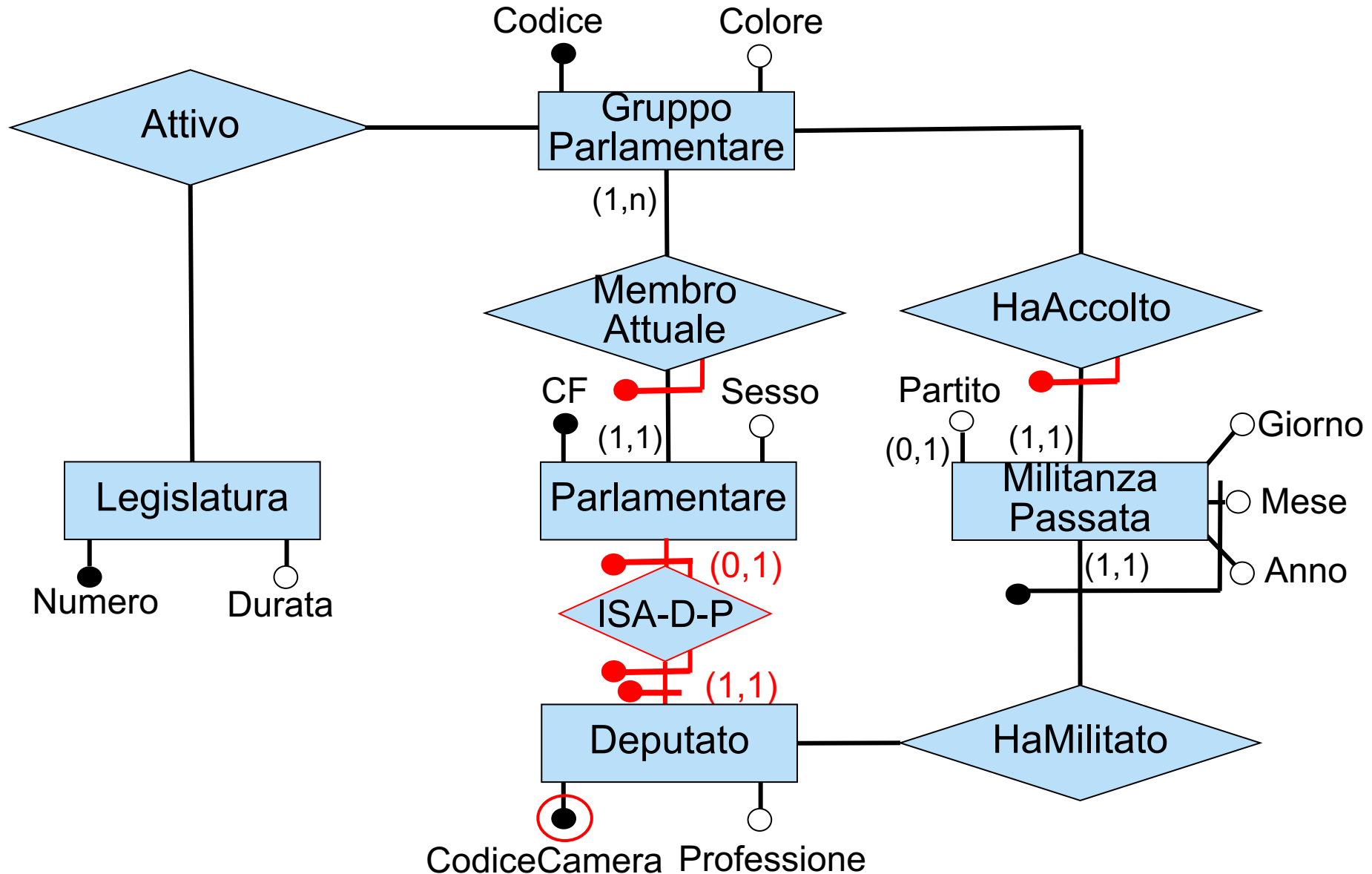
Esercizio 2: riportiamo la specifica dei requisiti

Ogni gruppo parlamentare ha un codice (identificativo), un colore politico ed un insieme non vuoto di parlamentari che ne sono attualmente membri. Di ogni gruppo parlamentare interessano anche le legislature (ciascuna con numero identificativo e durata in anni) nelle quali il gruppo è stato attivo. Di ogni parlamentare che attualmente risiede in parlamento interessa il codice fiscale (identificativo), il sesso ed il gruppo parlamentare (uno ed uno solo) di cui è attualmente membro. I deputati sono parlamentari particolari (risiedono nel ramo "Camera") e di ognuno di essi interessa anche il codice identificativo assegnato dalla Camera, la professione ed i gruppi parlamentari di cui è stato membro nel passato. Ogni volta che un deputato ha militato nel passato in un gruppo parlamentare che lo ha accolto tra i suoi membri, è stata registrata la data di ingresso ed il codice dell'eventuale partito politico di appartenenza in quella data. Entrambe queste informazioni sono di interesse nel nostro sistema informativo. Si noti che nessun deputato può cambiare gruppo parlamentare più di una volta al mese.

Esercizio 2: riportiamo lo schema concettuale



Esercizio 2: riportiamo lo schema concettuale ristrutturato



Esercizio 2 (su ristrutturazione dello schema logico)

Ricordiamo le indicazioni di progetto:

1. quando si accede ad un parlamentare si vuole sempre sapere il Gruppo parlamentare a cui appartiene;
2. quando si accede ad un deputato si vuole sempre sapere il suo codice fiscale ed il sesso, e quando si accede ad un parlamentare si vuole sempre sapere se è un deputato e, in caso, il suo codice camera la sua professione.

Esercizio 2: riportiamo lo schema logico ottenuto dalla traduzione diretta

GruppoParlamentare(Codice, Colore)

inclusione GruppoParlamentare[Codice] \subseteq MembroAttuale[Gruppo]

Attivo(Gruppo, Legislatura)

foreign key Attivo[Gruppo] \subseteq GruppoParlamentare[Codice]

foreign key Attivo[Legislatura] \subseteq Legislatura[Nnumero]

Legislatura(Numero, Durata)

MembroAttuale(Parlamentare, Gruppo)

foreign key MembroAttuale[Parlamentare] \subseteq Parlamentare[CF]

foreign key MembroAttuale[Gruppo] \subseteq GruppoParlamentare[Codice]

Parlamentare(CF, Sesso)

foreign key Parlamentare[CF] \subseteq MembroAttuale[Parlamentare]

Deputato(CodiceCamera, Professione)

foreign key Deputato[CodiceCamera] \subseteq ISA-D-P[Deputato]

ISA-D-P(Deputato, Parlamentare)

chiave Parlamentare

foreign key \subseteq ISA-D-P[Deputato] \subseteq Deputato[CodiceCamera]

foreign key \subseteq ISA-D-P[Parlamentare] \subseteq Parlamentare[CF]

MilitanzaPassata(Giorno, Mese, Anno, Deputato, Partito*)

foreign key MilitanzaPassata[Deputato] \subseteq Deputato[CodiceCamera]

foreign key MilitanzaPassata[Mese, Anno, Deputato] \subseteq HaAccolto[Mese, Anno, Deputato]

HaAccolto(Gruppo, Mese, Anno, Deputato)

foreign key HaAccolto[Gruppo] \subseteq GruppoParlamentare[Codice]

foreign key HaAccolto[Mese, Anno, Deputato] \subseteq MilitanzaPassata[Mese, Anno, Deputato]

Soluzione esercizio 2

1. quando si accede ad un parlamentare si vuole sempre sapere il Gruppo parlamentare a cui appartiene;

Per rispondere a questa indicazione possiamo operare l'accorpamento delle relazioni Parlamentare e MembroAttuale (fortemente accoppiate), ottenendo:

Parlamentare(CF, Sesso, Gruppo)

foreign key Parlamentare[Gruppo] \subseteq GruppoParlamentare[Codice]

e ricordando di modificare l'inclusione da GruppoParlamentare a MembroAttuale, che non è più nello schema:

GruppoParlamentare(Codice, Colore)

inclusione GruppoParlamentare[Codice] \subseteq Parlamentare[Gruppo]

Soluzione esercizio 2

- quando si accede ad un deputato si vuole sempre sapere il suo codice fiscale ed il sesso, e quando si accede ad un parlamentare si vuole sempre sapere se è un deputato e, in caso, il suo codice camera e la sua professione.

Per rispondere alla indicazione che quando si accede ad un deputato si vuole sempre sapere il suo codice fiscale possiamo dapprima operare l'accorpamento delle relazioni Deputato e ISA-D-P (fortemente accoppiate), ottenendo la relazione dove è disponibile il codice fiscale dei deputati (che è chiave in Deputato, perché Parlamentare era chiave in ISA-D-P):

Deputato(CodiceCamera, Professione, CF)
foreign key Deputato[CF] \subseteq Parlamentare[CF]
chiave CF

Soluzione esercizio 2

A questo punto, siccome quando si accede ad un deputato si vuole sempre sapere anche il sesso e quando si accede ad un parlamentare si vuole sempre sapere se è un deputato e, in caso, il suo codice camera e la sua professione, possiamo operare l'accorpamento della nuova relazione appena ottenuta

Deputato(CodiceCamera, Professione, CF)

foreign key Deputato[CF] \subseteq Parlamentare[CF]

chiave CF

con

Parlamentare(CF, Sesso, Gruppo)

(le due tabelle sono debolmente accoppiate), ottenendo:

Parlamentare(CF, Sesso, Gruppo, CodiceCamera*, Professione*)

foreign key Parlamentare[Gruppo] \subseteq GruppoParlamentare[Codice]

vincolo di tupla CodiceCamera is null se e solo se Professione is null

chiave CodiceCamera

Soluzione esercizio 2

Lo schema finale, compreso delle definizioni di viste, è il seguente:

GruppoParlamentare(Codice, Colore)

inclusione GruppoParlamentare[Codice] \subseteq Parlamentare[Gruppo]

Attivo(Gruppo, Legislatura)

foreign key Attivo[Gruppo] \subseteq GruppoParlamentare[Codice]

foreign key Attivo[Legislatura] \subseteq Legislatura[Numero]

Legislatura(Numero, Durata)

Parlamentare(CF, Sesso, Gruppo, CodiceCamera*, Professione*)

foreign key Parlamentare[Gruppo] \subseteq GruppoParlamentare[Codice]

vincolo di tupla CodiceCamera is null se e solo se Professione is null

chiave CodiceCamera

MilitanzaPassata(Giorno, Mese, Anno, Deputato, Partito*)

foreign key MilitanzaPassata[Deputato] \subseteq Parlamentare[CodiceCamera]

foreign key MilitanzaPassata[Mese, Anno, Deputato] \subseteq HaAccolto[Mese, Anno, Deputato]

HaAccolto(Gruppo, Mese, Anno, Deputato)

foreign key HaAccolto[Gruppo] \subseteq GruppoParlamentare[Codice]

foreign key HaAccolto[Mese, Anno, Deputato] \subseteq MilitanzaPassata[Mese, Anno, Deputato]

View Deputato(CodiceCamera, Professione) as

select CodiceCamera, Professione from Parlamentare where CodiceCamera is not null

View ISA-D-P(Deputato, Parlamentare) as select CodiceCamera, CF from Parlamentare

View MembroAttuale(Parlamentare, Gruppo) as select CF, Gruppo from Parlamentare

Soluzione esercizio 2

GruppoParlamentare(Codice, Colore)

inclusione GruppoParlamentare[Codice] \subseteq MembroAttuale[Gruppo]

Attivo(Gruppo, Legislatura)

foreign key Attivo[Gruppo] \subseteq GruppoParlamentare[Codice]

foreign key Attivo[Legislatura] \subseteq Legislatura[Numero]

Legislatura(Numero, Durata)

MembroAttuale(Parlamentare, Gruppo)

foreign key MembroAttuale[Parlamentare] \subseteq Parlamentare[CF]

foreign key MembroAttuale[Gruppo] \subseteq GruppoParlamentare[Codice]

Parlamentare(CF, Sesso)

foreign key Parlamentare[CF] \subseteq MembroAttuale[Parlamentare]

Deputato(CodiceCamera, Professione)

foreign key Deputato[CodiceCamera] \subseteq ISA-D-P[Deputato]

ISA-D-P(Deputato, Parlamentare)

chiave Parlamentare

foreign key \subseteq ISA-D-P[Deputato] \subseteq Deputato[CodiceCamera]

foreign key \subseteq ISA-D-P[Parlamentare] \subseteq Parlamentare[CF]

MilitanzaPassata(Giorno, Mese, Anno, Deputato, Partito*)

foreign key MilitanzaPassata[Deputato] \subseteq Deputato[CodiceCamera]

foreign key MilitanzaPassata[Mese,Anno,Deputato] \subseteq

HaAccolto[Mese,Anno,Deputato]

HaAccolto(Gruppo, Mese, Anno, Deputato)

foreign key HaAccolto[Gruppo] \subseteq GruppoParlamentare[Codice]

foreign key HaAccolto[Mese,Anno,Deputato] \subseteq

MilitanzaPassata[Mese,Anno,Deputato]

GruppoParlamentare(Codice, Colore)

inclusione GruppoParlamentare[Codice] \subseteq Parlamentare[Gruppo]

Attivo(Gruppo, Legislatura)

foreign key Attivo[Gruppo] \subseteq GruppoParlamentare[Codice]

foreign key Attivo[Legislatura] \subseteq Legislatura[Numero]

Legislatura(Numero, Durata)

Parlamentare(CF, Sesso, Gruppo, CodiceCamera*, Professione*)

foreign key Parlamentare[Gruppo] \subseteq GruppoParlamentare[Codice]

vincolo di tupla CodiceCamera is null se e solo se Professione is null

chiave CodiceCamera

MilitanzaPassata(Giorno, Mese, Anno, Deputato, Partito*)

foreign key MilitanzaPassata[Deputato] \subseteq

Parlamentare[CodiceCamera]

foreign key MilitanzaPassata[Mese,Anno,Deputato] \subseteq

HaAccolto[Mese,Anno,Deputato]

HaAccolto(Gruppo, Mese, Anno, Deputato)

foreign key HaAccolto[Gruppo] \subseteq GruppoParlamentare[Codice]

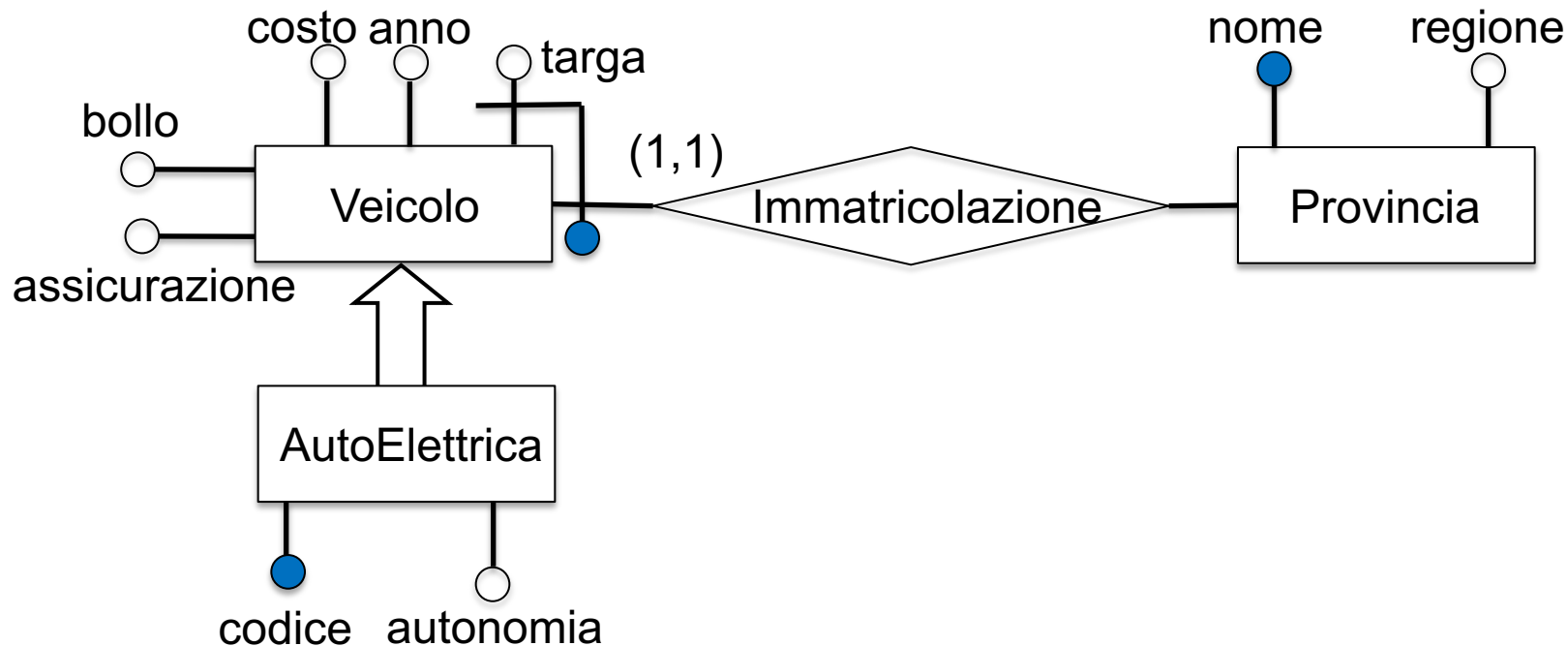
foreign key HaAccolto[Mese,Anno,Deputato] \subseteq

MilitanzaPassata[Mese,Anno,Deputato]

Esercizio 3: progettazione logica

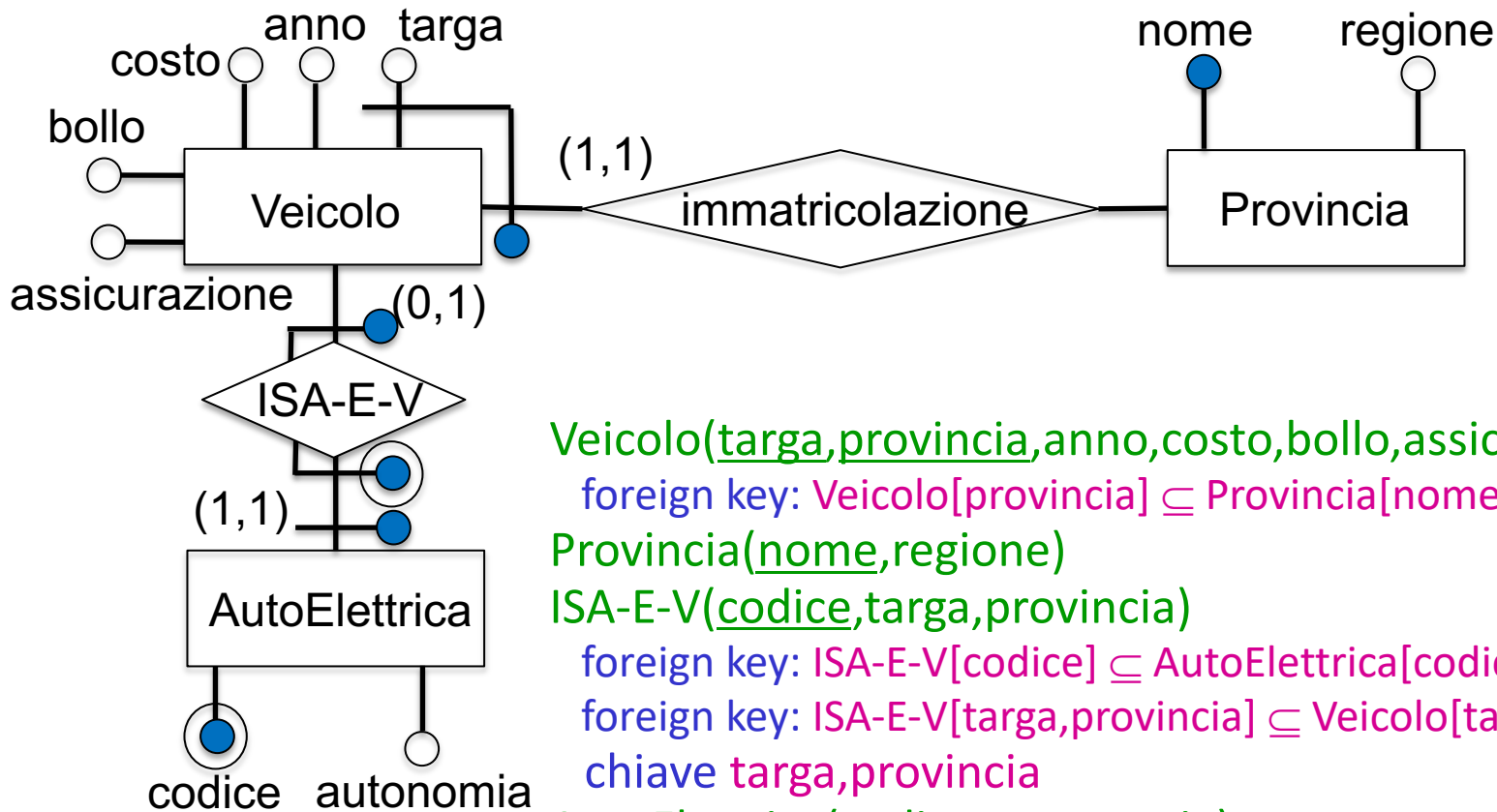
Partendo dal seguente schema concettuale qui sotto, si produca lo schema logico mediante la ristrutturazione dello schema concettuale e la traduzione diretta e poi, se necessario, si ristrutturi lo schema logico. Le indicazioni di progetto da tenere presente sono:

1. quando si accede alle auto elettriche si utilizza prevalentemente il codice;
2. ai dati fiscali (bollo e assicurazione) dei veicoli si accede separatamente rispetto ai dati commerciali (costo, anno di immatricolazione);
3. ai dati commerciali delle auto elettriche si accede separatamente rispetto ai dati commerciali degli altri veicoli;
4. quando si accede ai dati commerciali dei veicoli diversi dalle auto elettriche si vuole sempre conoscere la regione di immatricolazione;
5. quando si accede ai dati commerciali delle auto elettriche si vuole sempre conoscere la targa, la provincia, l'autonomia, il costo e l'anno di immatricolazione.



Esercizio 3: soluzione

Schema concettuale ristrutturato e schema logico: siccome quando si accede alle auto elettriche si utilizza prevalentemente il codice, scegliamo "codice" quale identificatore principale di AutoElettrica.



Veicolo(targa, provincia, anno, costo, bollo, assicurazione)

foreign key: Veicolo[provincia] \subseteq Provincia[nome]

Provincia(nome, regione)

ISA-E-V(codice, targa, provincia)

foreign key: ISA-E-V[codice] \subseteq AutoElettrica[codice]

foreign key: ISA-E-V[targa, provincia] \subseteq Veicolo[targa, provincia]

chiave targa, provincia

AutoElettrica(codice, autonomia)

foreign key: AutoElettrica[codice] \subseteq ISA-E-V[codice]

Esercizio 3: soluzione

Seguendo l'indicazione che ai dati fiscali (bollo e assicurazione) dei veicoli si accede separatamente rispetto ai dati commerciali (costo, anno) decomponiamo verticalmente Veicolo in VeicoloDatiFiscali e VeicoloDatiCommerciali.

VeicoloDatiFiscali(targa,provincia,bollo,assicurazione)

foreign key: VeicoloDatiFiscali[provincia] \subseteq Provincia[nome]

foreign key: VeicoloDatiFiscali[targa,provincia] \subseteq VeicoloDatiCommercio[targa,provincia]

VeicoloDatiCommercio(targa,provincia,anno,costo)

foreign key: VeicoloDatiCommercio[targa,provincia] \subseteq VeicoloDatiFiscali[targa,provincia]

Provincia(nome,regione)

ISA-E-V(codice,targa,provincia)

foreign key: ISA-E-V[codice] \subseteq AutoElettrica[codice]

foreign key: ISA-E-V[targa,provincia] \subseteq VeicoloDatiFiscali[targa,provincia]

chiave targa,provincia

AutoElettrica(codice,autonomia)

foreign key: AutoElettrica[codice] \subseteq ISA-E-V[codice]

Esercizio 3: soluzione

Seguendo l'indicazione che ai dati commerciali delle auto elettriche si accede separatamente rispetto ai dati commerciali degli altri veicoli, decomponiamo orizzontalmente `VeicoloDatiCommercio` in `VeicoloNonElettricoDatiCommercio` e `AutoElettricaDatiCommercio` (che diventa fortemente accoppiata con `ISA-E-V`).

`VeicoloDatiFiscali(targa,provincia,bollo,assicurazione)`

foreign key: `VeicoloDatiFiscali[provincia] ⊆ Provincia[nome]`

inclusione : `VeicoloDatiFiscali[targa,provincia] ⊆ VeicoloNonAutoElettricaDatiCommercio[targa,provincia]`
`U AutoElettricaDatiCommercio[targa,provincia]`

`VeicoloNonElettricoDatiCommercio(targa,provincia,anno, costo)`

foreign key: `VeicoloNonElettricoDatiCommercio[targa,provincia] ⊆ VeicoloDatiFiscali[targa,provincia]`

`AutoElettricaDatiCommercio(targa,provincia,anno, costo)`

foreign key: `AutoElettricaDatiCommercio[targa,provincia] ⊆ VeicoloDatiFiscali[targa,provincia]`

vincolo esterno: `AutoElettricaDatiCommercio[targa,provincia] ∩`
`VeicoloNonElettricoDatiCommercio[targa,provincia] = ∅`

foreign key: `AutoElettricaDatiCommercio[targa,provincia] ⊆ ISA-E-V[targa,provincia]`

`Provincia(nome, regione)`

`ISA-E-V(codice, targa, provincia)`

foreign key: `ISA-E-V[codice] ⊆ AutoElettrica[codice]`

foreign key: `ISA-E-V[targa,provincia] ⊆ AutoElettricaDatiCommercio[targa,provincia]`

chiave `targa,provincia`

`AutoElettrica(codice, autonomia)`

foreign key: `AutoElettrica[codice] ⊆ ISA-E-V[codice]`

Esercizio 3: soluzione

Seguendo l'indicazione che quando si accede ai dati commerciali dei veicoli diversi dalle auto elettriche si vuole sempre conoscere la regione di immatricolazione, accorpriamo VeicoloNonAutoElettricaDatiCommercio e Provincia (accoppiate lascamente), così da portare l'attributo regione in VeicoloNonAutoElettricaDatiCommercio;

VeicoloDatiFiscali(targa,provincia,bollo,assicurazione)

foreign key: VeicoloDatiFiscali[provincia] \subseteq Provincia[nome]

inclusione : VeicoloDatiFiscali[targa,provincia] \subseteq VeicoloNonElettricoDatiCommercio[targa,provincia] \cup
AutoElettricaDatiCommercio[targa,provincia]

VeicoloNonElettricoDatiCommercio(targa,provincia,anno,costo,regione)

foreign key: VeicoloNonElettricoDatiCommercio[targa,provincia] \subseteq VeicoloDatiFiscali[targa,provincia]

foreign key: VeicoloNonElettricoDatiCommercio[provincia,regione] \subseteq Provincia[nome,regione]

AutoElettricaDatiCommercio(targa,provincia,anno,costo)

foreign key: AutoElettricaDatiCommercio[targa,provincia] \subseteq VeicoloDatiFiscali[targa,provincia]

vincolo esterno: AutoElettricaDatiCommercio[targa,provincia] \cap VeicoloNonElettricoDatiCommercio[targa,provincia] = \emptyset

foreign key: AutoElettricaDatiCommercio[targa,provincia] \subseteq ISA-E-V[targa,provincia]

Provincia(nome,regione)

ISA-E-V(codice,targa,provincia)

foreign key: ISA-E-V[codice] \subseteq AutoElettrica[codice]

foreign key: ISA-E-V[targa,provincia] \subseteq VeicoloDatiFiscali[targa,provincia]

chiave targa,provincia

AutoElettrica(codice,autonomia)

foreign key: AutoElettrica[codice] \subseteq ISA-E-V[codice]

Esercizio 3: soluzione

Seguendo l'indicazione che quando si accede ai dati commerciali delle auto elettriche si vuole sempre conoscere la targa, la provincia, l'autonomia, il costo e l'anno di immatricolazione, accorpriamo AutoElettricaDatiCommercio con ISA-E-V (fortemente accoppiate) e la relazione ottenuta con AutoElettrica (anch'esse fortemente accoppiate).

VeicoloDatiFiscali(targa,provincia,bollo,assicurazione)

foreign key: VeicoloDatiFiscali[provincia] \subseteq Provincia[nome]

inclusione : VeicoloDatiFiscali[targa,provincia] \subseteq VeicoloNonAutoElettricaDatiCommercio[targa,provincia]
U AutoElettricaDatiCommercio[targa,provincia]

VeicoloNonElettricoDatiCommercio(targa,provincia,anno,costo,regione)

foreign key: VeicoloNonElettricoDatiCommercio[targa,provincia] \subseteq VeicoloDatiFiscali[targa,provincia]

foreign key: VeicoloNonElettricoDatiCommercio[provincia,regione] \subseteq Provincia[nome,regione]

Provincia(nome,regione)

AutoElettricaConDatiCommercio(codice,autonomia,targa,provincia,anno,costo)

foreign key: AutoElettricaDatiCommercio[targa,provincia] \subseteq VeicoloDatiFiscali[targa,provincia]

vincolo esterno: AutoElettricaDatiCommercio[targa,provincia] \cap
VeicoloNonElettricoDatiCommercio[targa,provincia] = \emptyset

chiave targa,provincia

View Veicolo(targa,provincia,anno,costo,bollo,assicurazione) as

select targa,provincia,anno,costo from

VeicoloDatiFiscali natural join (select targa,provincia,anno,costo from VeicoloNonElettricoDatiCommercio
union

select targa,provincia,anno,costo from AutoElettricaConDatiCommercio)

View ISA-E-V(codice,targa,provincia) as select codice,targa,provincia from AutoElettricaConDatiCommercio

Esercizio 3: soluzione

Confrontiamo lo schema ottenuto dalla traduzione diretta:

Veicolo(targa,provincia,anno, costo,bollo,assicurazione)

foreign key: **Veicolo**[provincia] \subseteq **Provincia**[nome]

Provincia(nome,regione)

ISA-E-A(codice,targa,provincia)

foreign key: **ISA-E-V**[codice] \subseteq **AutoElettrica**[codice]

foreign key: **ISA-E-V**[targa,provincia] \subseteq **Veicolo**[targa,provincia]

chiave **targa**,**provincia**

AutoElettrica(codice,autonomia)

foreign key: **AutoElettrica**[codice] \subseteq **ISA-E-V**[codice]

con quello ottenuto dalla ristrutturazione:

VeicoloDatiFiscali(targa,provincia,bollo,assicurazione)

foreign key: **VeicoloDatiFiscali**[provincia] \subseteq **Provincia**[nome]

inclusione : **VeicoloDatiFiscali**[targa,provincia] \subseteq **VeicoloNonAutoElettricaDatiCommercio**[targa,provincia]
 \cup **AutoElettricaDatiCommercio**[targa,provincia]

VeicoloNonElettricoDatiCommercio(targa,provincia,anno, costo,regione)

foreign key: **VeicoloNonElettricoDatiCommercio**[targa,provincia] \subseteq **VeicoloDatiFiscali**[targa,provincia]

foreign key: **VeicoloNonElettricoDatiCommercio**[provincia,regione] \subseteq **Provincia**[nome,regione]

Provincia(nome,regione)

AutoElettricaConDatiCommercio(codice,autonomia,targa,provincia,anno, costo)

foreign key: **AutoElettricaDatiCommercio**[targa,provincia] \subseteq **VeicoloDatiFiscali**[targa,provincia]

vincolo esterno: **AutoElettricaDatiCommercio**[targa,provincia] \cap
VeicoloNonElettricoDatiCommercio[targa,provincia] = \emptyset

chiave **targa**,**provincia**

Vengono da
Veicolo

Viene da AutoElettrica,
ISA-E-V e Veicolo