

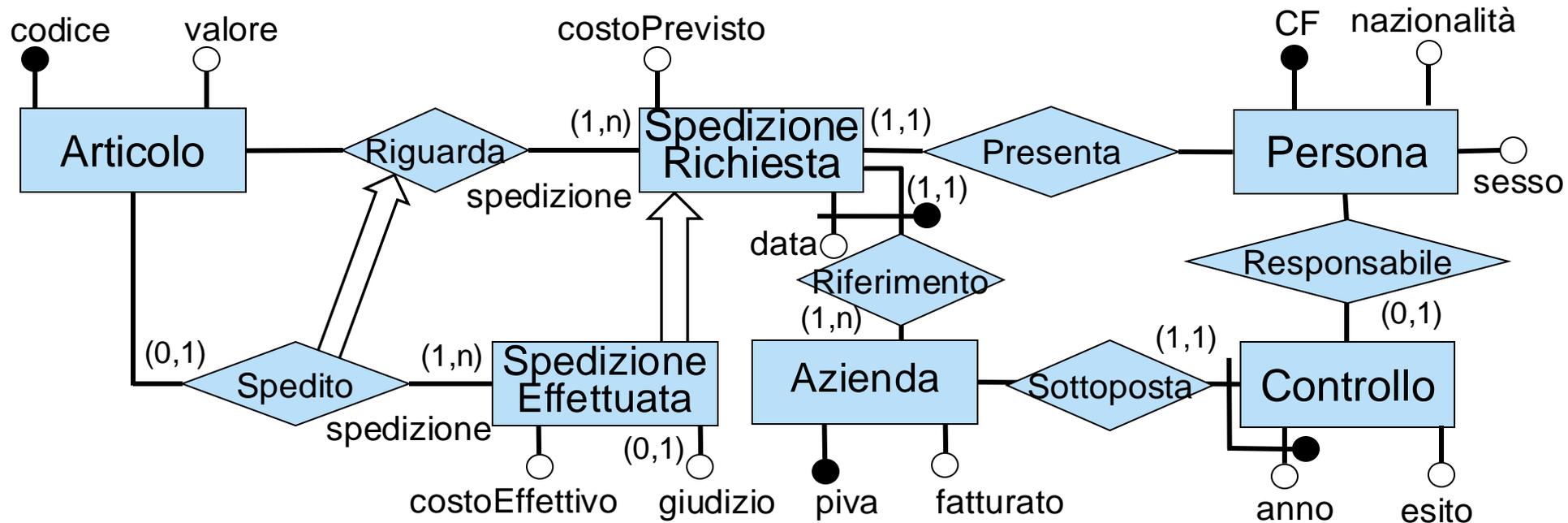
# **Basi di dati**

**Soluzioni dei problemi proposti  
nell'appello del 15-01-2025  
Compito A**

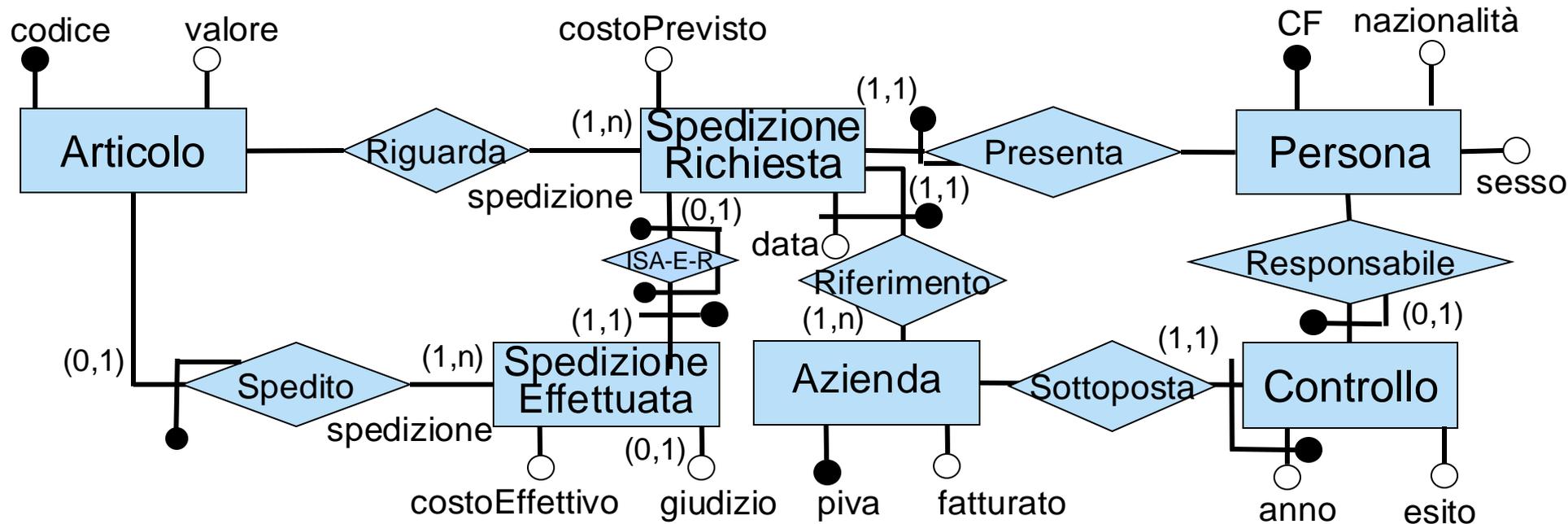
***Maurizio Lenzerini***

Anno Accademico 2024/25

# Problema 1 – Schema ER



# Problema 1 – Schema ER ristrutturato



## Vincolo esterno:

In ogni istanza  $I$  dello schema concettuale, per ogni tupla  $\langle \text{Articolo}:a, \text{SpedizioneEffettuata}:s \rangle$  in  $\text{Istanze}(I, \text{Spedito})$  tale che  $\langle \text{SpedizioneEffettuata}:s, \text{SpedizioneRichiesta}:r \rangle$  è in  $\text{Istanze}(I, \text{ISA-E-R})$ , la tupla  $\langle \text{Articolo}:a, \text{SpedizioneRichiesta}:r \rangle$  è in  $\text{Istanze}(I, \text{Riguarda})$

# Problema 2 – Traduzione diretta

Schema logico prodotto dalla traduzione diretta:

Articolo(codice,valore)

SpedizioneRichiesta(data,azienda,costoPrevisto)

foreign key: SpedizioneRichiesta[azienda]  $\subseteq$  Azienda[piva]

inclusione: SpedizioneRichiesta[data,azienda]  $\subseteq$  Riguarda[data,azienda]

foreign key: SpedizioneRichiesta[data,azienda]  $\subseteq$  Presenta[data,azienda]

Azienda(piva,fatturato)

inclusione: Azienda[piva]  $\subseteq$  SpedizioneRichiesta[azienda]

SpedizioneEffettuata(data,azienda,costoEffettivo,giudizio\*)

foreign key: SpedizioneEffettuata[data,azienda]  $\subseteq$  SpedizioneRichiesta[data,azienda]

inclusione: SpedizioneRichiesta[data,azienda]  $\subseteq$  Spedito[data,azienda]

Riguarda(articolo,data,azienda)

foreign key: Riguarda[articolo]  $\subseteq$  Articolo[codice]

foreign key: Riguarda[data,azienda]  $\subseteq$  SpedizioneRichiesta[data,azienda]

Spedito(articolo,data,azienda)

foreign key: Spedito[articolo,data,azienda]  $\subseteq$  Riguarda[articolo,data,azienda]

Persona(CF, sesso, nazionalità)

Presenta(data,azienda,persona)

foreign key: Presenta[data,azienda]  $\subseteq$  SpedizioneRichiesta[data,azienda]

foreign key: Presenta[persona]  $\subseteq$  Persona[CF]

Controllo(anno,azienda,esito)

foreign key: Controllo[azienda]  $\subseteq$  Azienda[piva]

Responsabile(anno,azienda,persona)

foreign key: Responsabile[anno,azienda]  $\subseteq$  Controllo[anno,azienda]

foreign key: Responsabile[persona]  $\subseteq$  Persona[CF]

## Problema 2 – Ristrutturazione dello schema logico

L'indicazione di progetto induce un accorpamento tra la tabella Controllo e la tabella Responsabile debolmente accoppiate, e un accorpamento per denormalizzazione tra la tabella così ottenuta (che chiamiamo ancora "Controllo") e la tabella Persona, al fine di portare anche l'attributo "nazionalità" all'interno della tabella Controllo. Il risultato è che la tabella Responsabile sparisce e la tabella Controllo diventa:

Controllo(anno,azienda,esito,persona\*,nazionalità\*)

**foreign key:** Controllo[azienda]  $\subseteq$  Azienda[piva]

**foreign key:** Controllo[persona,nazionalità]  $\subseteq$  Persona[CF,nazionalità]

**vincolo di tupla:** persona is null se e solo se nazionalità is null

La tabella Responsabile si può ricostruire attraverso la seguente vista:

**view** Responsabile(anno,azienda,persona) = PROJ<sub>anno,azienda,persona</sub>(Controllo)

## Problema 3 – testo e soluzione

(3.1) Scrivere una query in algebra relazionale che calcoli la partita IVA di tutte le aziende di trasporto perfette, dove un'azienda è perfetta se tutte le spedizioni richieste che fanno riferimento ad essa sono state effettuate.

(3.2) Scrivere una query in SQL che per ogni azienda di trasporto a e per ogni valore s di "sesso" restituisca la partita IVA dell'azienda a, il valore s ed il numero di controlli di qualità a cui l'azienda a è stata sottoposta con responsabile di sesso s.

(3.1)

$$\text{REN}_{\text{azienda}} \leftarrow \text{piva}(\text{PROJ}_{\text{piva}}(\text{Azienda})) - \text{PROJ}_{\text{azienda}}(\text{PROJ}_{\text{data,azienda}}(\text{SpedizioneRichiesta}) - \text{PROJ}_{\text{data,azienda}}(\text{SpedizioneEffettuata}))$$

(3.2)

```
select c.azienda, p.sesso, count(*)
from Controllo c join Persona p on c.persona = p.CF
group by c.azienda, p.sesso
union
select a.piva, p.sesso, 0
from Azienda a, Persona p
where (a.piva,p.sesso) not in (select c.azienda,p.sesso
                               from Controllo c join Persona p on c.persona = p.CF)
```

## Problema 4

4.1

Un vincolo di integrità è una condizione che si esprime a livello di uno schema  $S$  di basi di dati e che deve essere soddisfatta da tutte le istanze di  $S$ .

4.2

Siccome la relazione  $R(A,B,C,D)$  non può essere soggetta ad update, dobbiamo considerare solo cancellazioni ed inserimenti. Per quanto riguarda le cancellazioni, nessuna operazione di cancellazione su  $R$  può lasciare la base di dati in uno stato in cui il vincolo di dipendenza funzionale  $A \rightarrow B$  è violato. Rimane quindi da considerare l'operazione di inserimento.

Ovviamente, a fronte dell'inserimento di una tupla  $t = \langle a,b,c,d \rangle$  in  $R$  è possibile che la dipendenza funzionale sia violata e questo succede quando la base di dati prima dell'inserimento soddisfa questa condizione  $Z$ : in  $R$  è presente una tupla  $t'$  tale che  $t'.A = a$  e  $t'.B \neq b$ . La soluzione 4.a consiste nel definire un «before insert» trigger che realizza l'inserimento se e solo se la condizione  $Z$  non è soddisfatta nella base di dati prima dell'inserimento.

## Problema 4 (soluzione 4.a)

```
create table R (  
  a int not null, b int not null, c int not null, d int not null  
);
```

```
create function inserisciR() returns trigger as
```

```
$$
```

```
begin
```

```
  if (new.a in (select a from R where b <> new.b))
```

```
  then return null; -- inserimento bloccato
```

```
  else return new; -- inserimento da eseguire
```

```
  end if;
```

```
end;
```

```
$$ language plpgsql;
```

```
create trigger triggerForR before insert on R for each row  
execute procedure inserisciR();
```

## Problema 4 (soluzione 4.b)

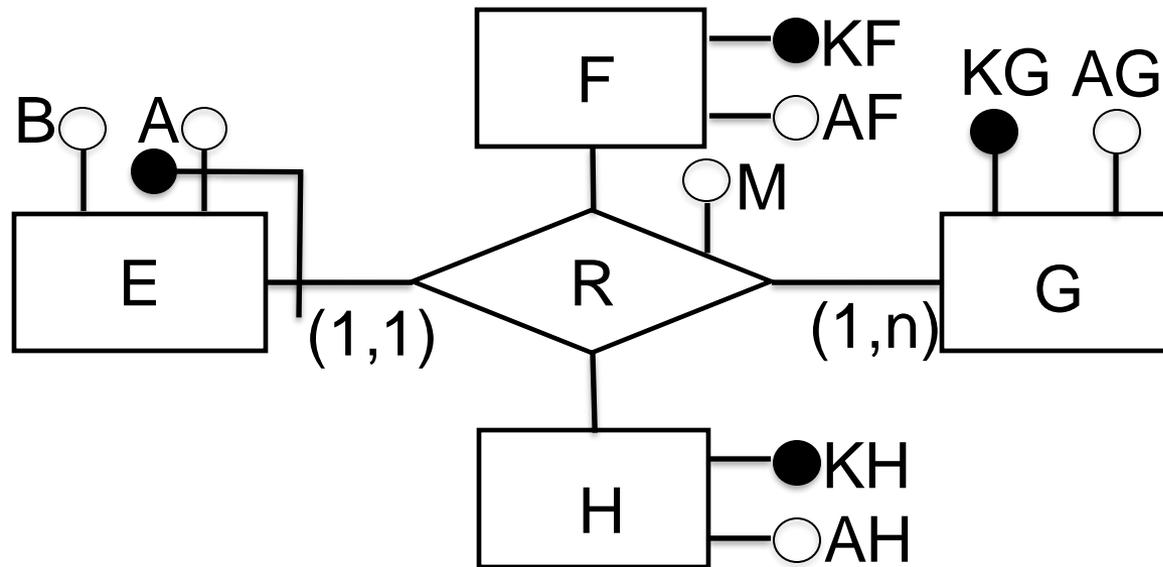
In questa soluzione, anch'essa corretta, definiamo un trigger di tipo after insert per bloccare l'inserimento nel caso in cui lo stato della base di dati non soddisfi il vincolo di dipendenza funzionale.

```
create table R (  
  a int not null, b int not null, c int not null, d int not null  
);
```

```
create function inserisciR() returns trigger as  
$$ begin  
  if exists (select *  
            from R r1 join R r2 on r1.a = r2.a  
            where r1.b <> r2.b)  
  then rollback;  
  else return new;  
  end if;  
end; $$ language plpgsql;
```

```
create trigger triggerForR after insert on R for each statement  
execute procedure inserisciR();
```

## Problema 5



5.1

La ristrutturazione non prevede alcuna operazione. La traduzione diretta produce questo schema relazionale:

$F(\underline{KF}, AF)$

$G(\underline{KG}, AG)$

inclusione  $G[KG] \subseteq E[G]$

$H(\underline{KH}, AH)$

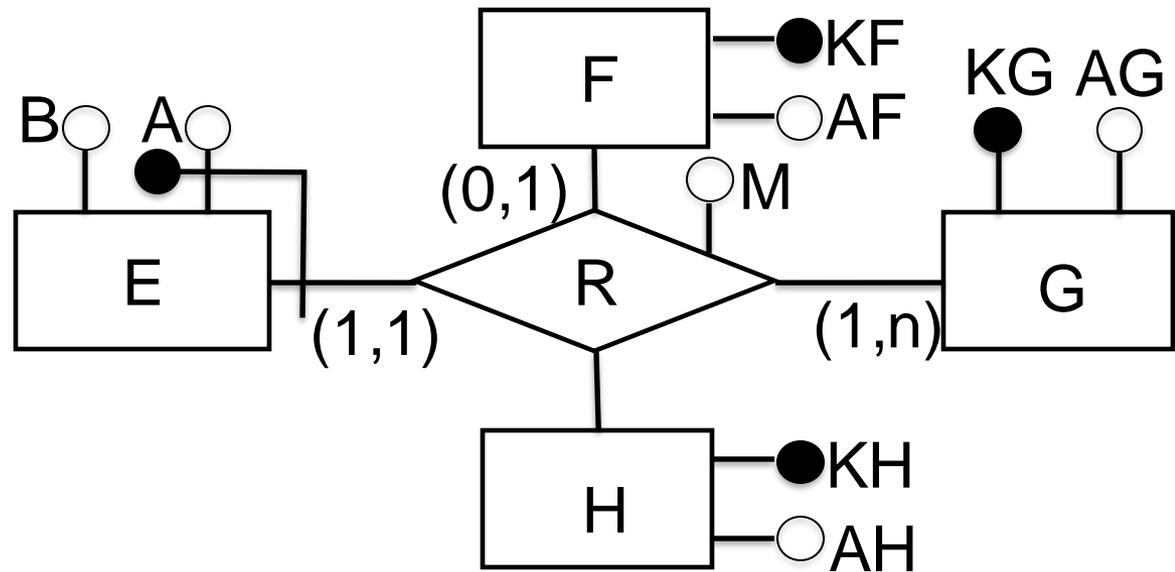
$E(\underline{B}, \underline{A}, \underline{F}, \underline{G}, \underline{H}, M)$

foreign key  $E[F] \subseteq F[KF]$

foreign key  $E[G] \subseteq G[KG]$

foreign key  $E[H] \subseteq H[KH]$

# Problema 5



5.2

Nel caso in cui lo schema di partenza fosse quello sopra, cioè con il vincolo (0,1) associato al ruolo F di R, nello schema concettuale ristrutturato l'entità E avrebbe come identificatore solo il ruolo E di R (vedi figura qui sotto) perché se ci fossero due istanze di E con la stessa partecipazione ad R, ci sarebbero due istanze di R con la stessa istanza di F nel ruolo F, contraddicendo il vincolo 1 di cardinalità massima sul ruolo F. In questo caso, la traduzione diretta produrrebbe questo schema relazionale:

$F(\underline{KF}, AF)$   
 $G(\underline{KG}, AG)$   
 inclusione  $G[KG] \subseteq E[G]$   
 $H(\underline{KH}, AH)$   
 $E(\underline{B}, A, \underline{F}, G, H, M)$   
 foreign key  $E[F] \subseteq F[KF]$   
 foreign key  $E[G] \subseteq G[KG]$   
 foreign key  $E[H] \subseteq H[KH]$

