

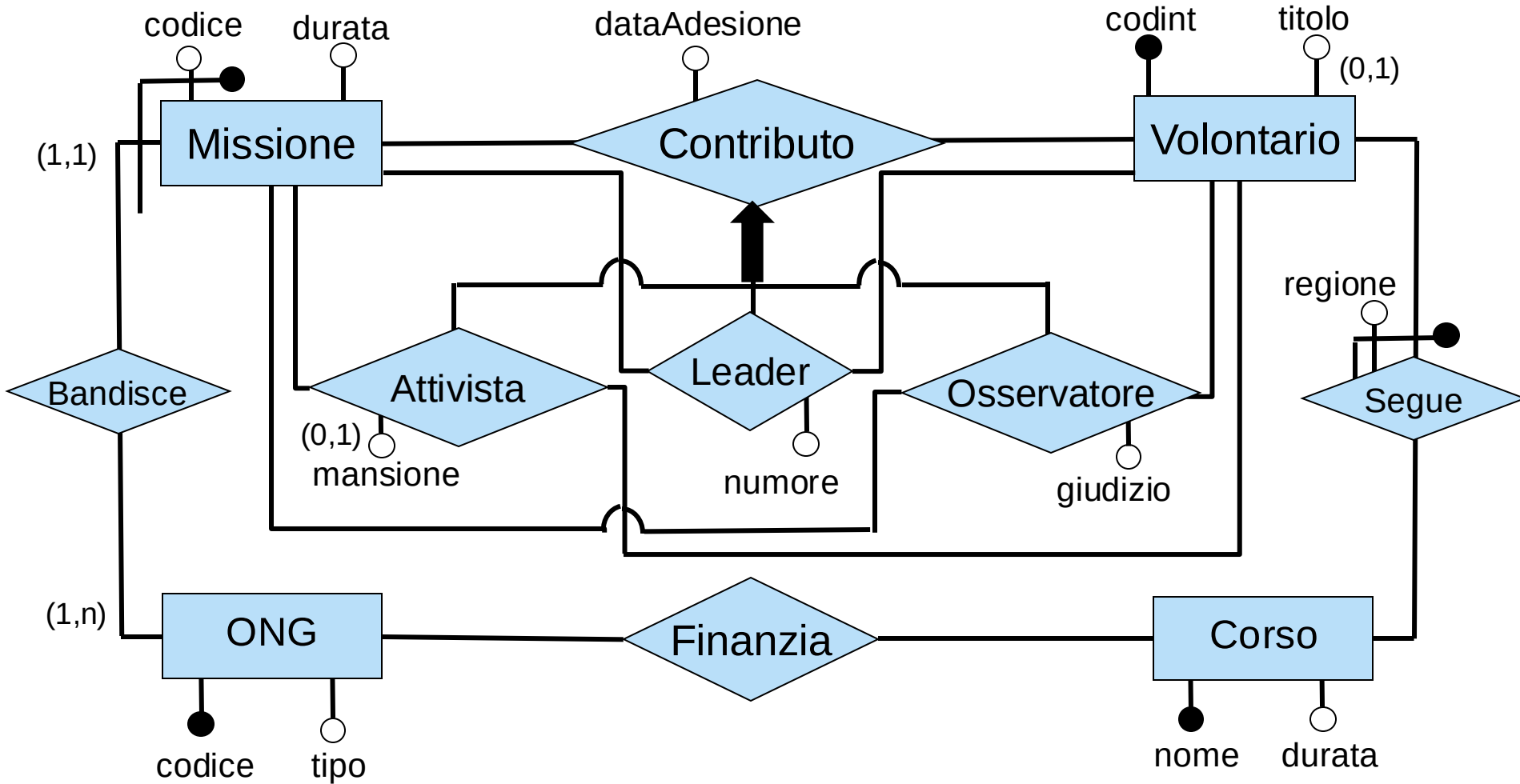
Basi di dati

**Soluzioni dei problemi proposti
nell'appello del 13-02-2025
Compito B**

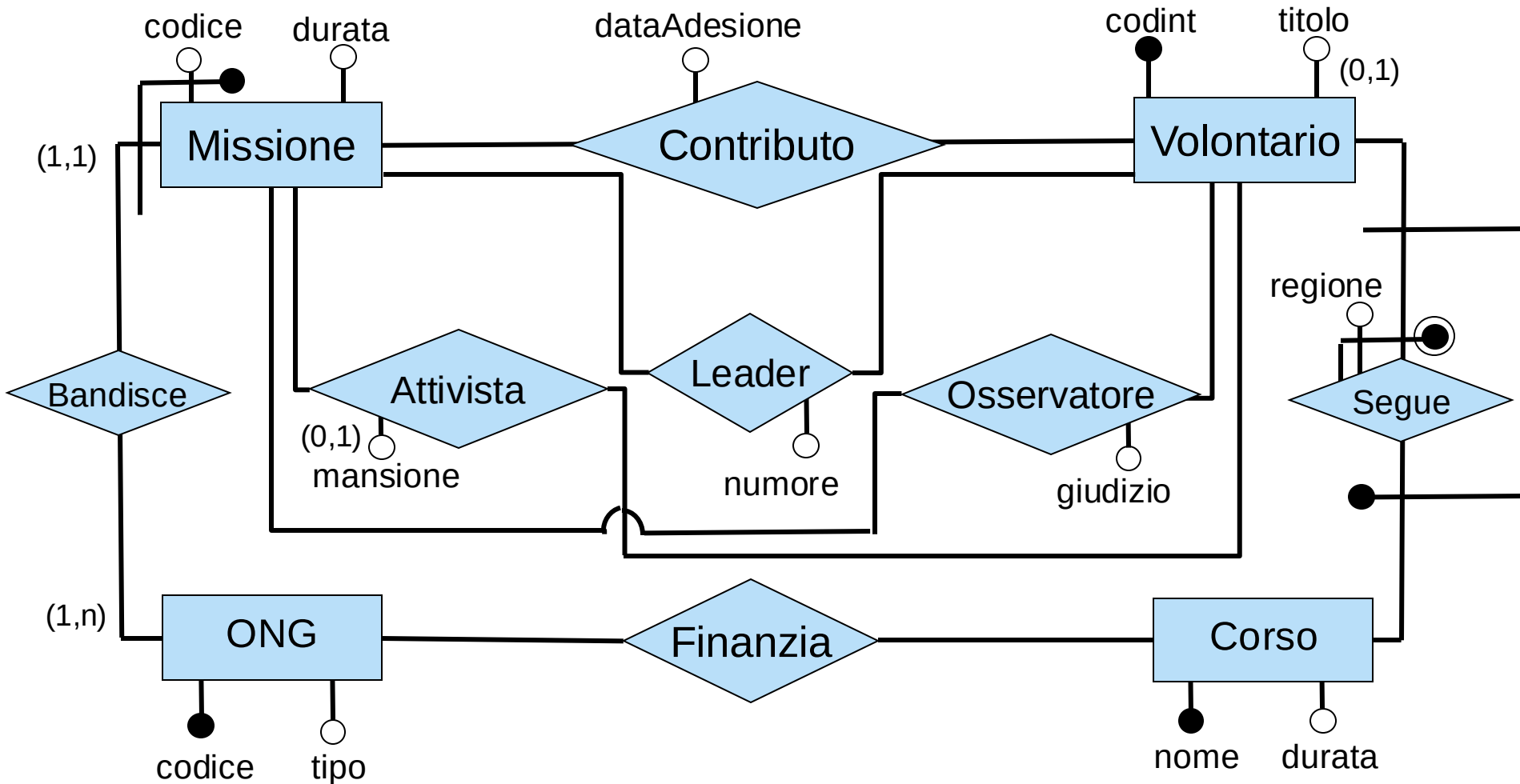
Maurizio Lenzerini

Anno Accademico 2024/25

Problema 1 – Schema ER



Problema 1 – Schema ER



Vincolo esterno: Attivista, Leader e Osservatore formano una partizione (disgiunta) di Contributo

Problema 2 – Traduzione diretta

Schema logico prodotto dalla traduzione diretta:

Missione(codice,ONG,durata)

foreign key: Missione[ONG] $\subseteq$ ONG[codice]

ONG(codice,tipo)

inclusione: ONG[codice] $\subseteq$ Missione[ONG]

Volontario(codint,titolo*)

Contributo(codice,ONG,volontario,dataAdesione)

foreign key: Contributo[codice,ONG] $\subseteq$ Missione[codice,ONG]

foreign key: Contributo[volontario] $\subseteq$ Volontario[codint]

vincolo esterno: Contributo[codice,ONG,colontario] $\subseteq$ Attivista[codice,ONG,colontario] $\cup$

Leader[codice,ONG,colontario] $\cup$ Osservatore[codice,ONG,colontario]

Attivista(codice,ONG,volontario,mansione*)

foreign key: Attivista[codice,ONG,Volontario] $\subseteq$ Contributo[codice,ONG,volontario]

vincolo esterno: Attivista[codice,ONG,Volontario] $\cap$ Leader{codice,ONG,Volontario} = $\emptyset$

vincolo esterno: Attivista[codice,ONG,Volontario] $\cap$ Osservato[codice,ONG,Volontario] = $\emptyset$

Leader(codice,ONG,volontario,numore)

foreign key: Leader[codice,ONG,Volontario] $\subseteq$ Contributo[codice,ONG,volontario]

vincolo esterno: Leader[codice,ONG,Volontario] $\cap$ Osservatore[codice,ONG,Volontario] = $\emptyset$

Osservatore(codice,ONG,volontario, giudizio)

foreign key: Attivista[codice,ONG,Volontario] $\subseteq$ Contributo[codice,ONG,volontario]

Corso(nome,durata)

Segue(volontario,regione,corso)

foreign key: Segue[volontario] $\subseteq$ Volontario[codint]

foreign key: Segue[corso] $\subseteq$ Corso[nome]

chiave: volontario, corso

Finanzia(corso,ONG)

foreign key: Finanzia[corso] $\subseteq$ Corso[nome]

foreign key: Finanzia[ONG] $\subseteq$ ONG[codice]

Problema 2 – Ristrutturazione dello schema logico

L'indicazione di progetto induce un accorpamento tra la tabella Contributo e le tabelle Attivista, Leader e Osservatore, debolmente accoppiate.

L'unico cambiamento che dobbiamo prevedere rispetto allo schema logico prodotto dalla traduzione diretta è che spariscono le tabelle Attivista, Leader e Osservatore, mentre la tabella Contributo diventa:

Contributo(codice,ONG,volontario,dataAdesione,attivista,mansione*,numore*,giudizio*)

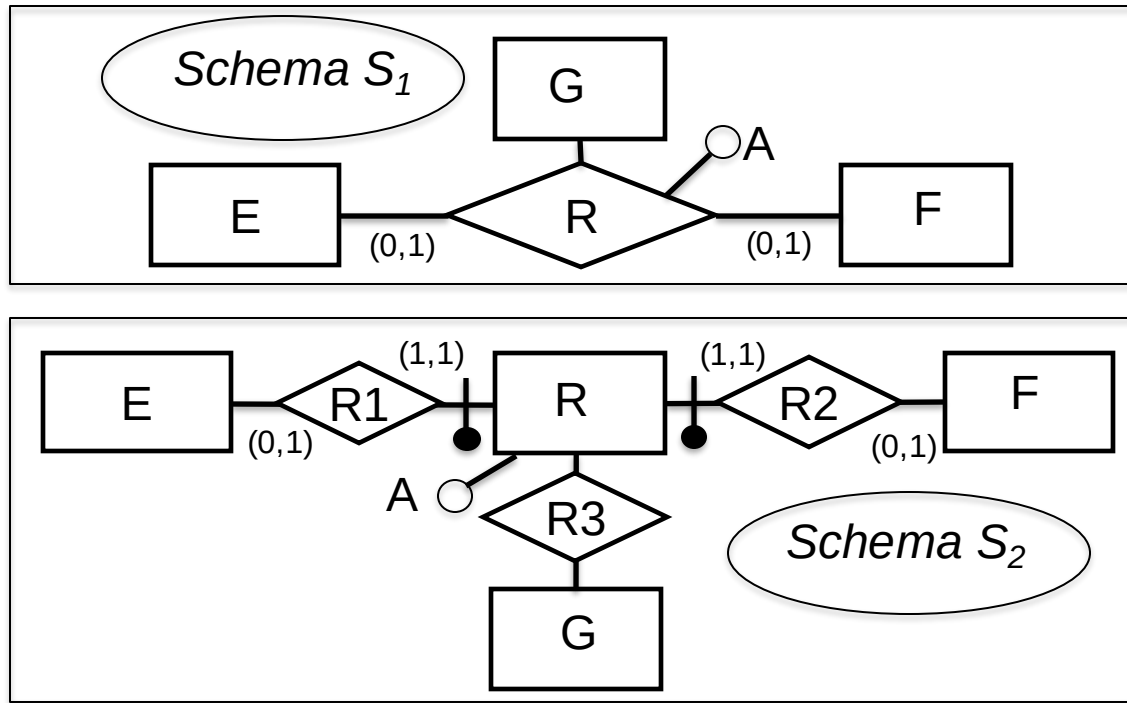
foreign key: Contributo[codice,ONG] $\subseteq$ Missione[codice,ONG]

foreign key: Contributo[volontario] $\subseteq$ Volontario[codint]

vincolo esterno: se attivista è false allora mansione è null

vincolo esterno: esattamente una di queste condizioni è true: attivista=true,
numore is not null, giudizio is not null

Problema 3: testo e soluzione



Si vuole sapere se gli schemi S_1 e S_2 mostrati qui a sinistra sono equivalenti, cioè se esiste una corrispondenza tra le istanze dei due schemi che specifichi come ogni istanza dell'uno si possa rappresentare correttamente mediante una istanza dell'altro. Se la risposta è positiva, motivarla in dettaglio. Se la risposta è negativa, mostrare una possibile istanza di uno dei due schemi che non potrà corrispondere ad alcuna istanza dell'altra, spiegando in dettaglio il motivo.

La risposta è negativa. Una delle ragioni è che nello schema S_1 ogni istanza di R è una terna di istanze di E , G ed F , mentre nello schema S_2 il concetto R è rappresentato come una entità che non ha partecipazione obbligatoria verso G . Consideriamo una istanza I di S_2 tale che:

$Istanze(I, E) = \{e\}$, $Istanze(I, F) = \{f\}$, $Istanze(I, G) = \{\}$, $Istanze(I, R) = \{r\}$, $Istanze(I, A) = \{<r, a>\}$,
 $Istanze(I, R_1) = \{<E:e, R:r>\}$, $Istanze(I, R_2) = \{<F:f, R:r>\}$, $Istanze(I, R_3) = \{\}$.

Notiamo che nella istanza I , $Istanze(I, G)$ è vuoto e quindi r non è legato ad alcuna istanza di G tramite R_3 . Quindi, qualunque funzione dalle istanze di S_2 alle istanze di S_1 che si volesse usare per dimostrare l'equivalenza di S_2 ed S_1 , quando applicata ad I , non potrebbe definire le istanze di R , perché in S_1 queste sono terne di istanze di E , G ed F , per le quali non è quindi ammessa l'assenza di una istanza di G .

Problema 4 – testo e soluzione

Si consideri la relazione Pista(codice,categoria), che memorizza i dati su piste di collaudo di veicoli, e la relazione Collaudo(veicolo,codice,data,marca,esito), che registra i collaudi subiti dai veicoli nelle piste nelle varie date, ciascuno con la marca del veicolo e l'esito.

- 4.1 Scrivere una query in algebra relazionale che per ogni categoria di pista, calcoli i veicoli che non hanno mai subito collaudi in piste di quella categoria.
- 4.2 Scrivere una query in SQL che per ogni marca di veicolo, calcoli qual'è la pista (o quali sono le piste se sono più d'una) maggiormente usata per collaudi di veicoli di quella marca.

(4.1)

$(\text{PROJ}_{\text{categoria}}(\text{Pista}) \text{ JOIN } \text{PROJ}_{\text{veicolo}}(\text{Collaudo})) - \text{PROJ}_{\text{categoria,veicolo}}(\text{Collaudo})$

(4.2)

with vista as

```
(select p.marca mar, c.codice pis, count(*) quanti
  from Pista p join Collaudo c on p.codice = c.codice
 group by p.marca, c.codice)
```

```
select v.mar, v.pis
```

```
from vista v
```

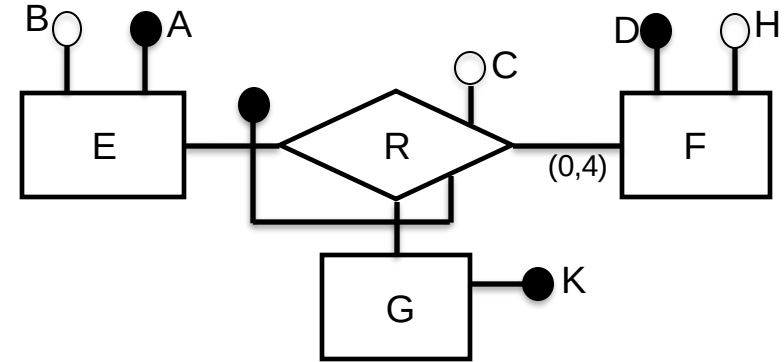
```
where v.quanti = (select max(v2.quanti)
```

```
  from vista v2
```

```
  where v2.mar=v1.mar and v2.pis=v1.pis)
```

Problema 5: testo e soluzione

Si consideri lo schema concettuale S mostrato qui a destra, per il quale sappiamo che, dopo il caricamento iniziale dei dati, non verranno eseguite operazioni di insert sulla relazione R. Si chiede di eseguire i vari passi della progettazione logica e produrre il corrispondente schema logico relazionale espresso nella nostra abituale rappresentazione. Si chiede poi di mostrare anche la definizione della tabella R in SQL, corredata da eventuali trigger



SOLUZIONE:

Lo schema concettuale ristrutturato è uguale a quello concettuale.

La definizione dello schema logico con la nostra usuale notazione è:

E(A,B)

F(D,H)

G(K)

R(EK,GK,FK,C)

foreign key R[EK] $\subseteq$ E[A]

foreign key R[FK] $\subseteq$ F[D]

foreign key R[GK] $\subseteq$ G[K]

vincolo esterno nessun valore di
appare più di 4 volte in R[FK]

Una definizione corretta di R in SQL è (aassumendo il tipo int per tutti gli attributi):

```
create table R (  
  FK int not null references E,  
  EK int not null references E,  
  GK int not null references G,  
  C int not null, primary key (EK,GK) )
```

create function aggiornaR() returns trigger as

\$\$

begin

if ((select count(*) from R where FK = new.FK) > 4)

then rollback; -- aggiornamento bloccato

else return new; -- aggiornamento da eseguire

end if;

end; \$\$ language plpgsql;

create trigger triggerForR after update on R for each
row execute procedure aggiornaR();