

# Basi di dati

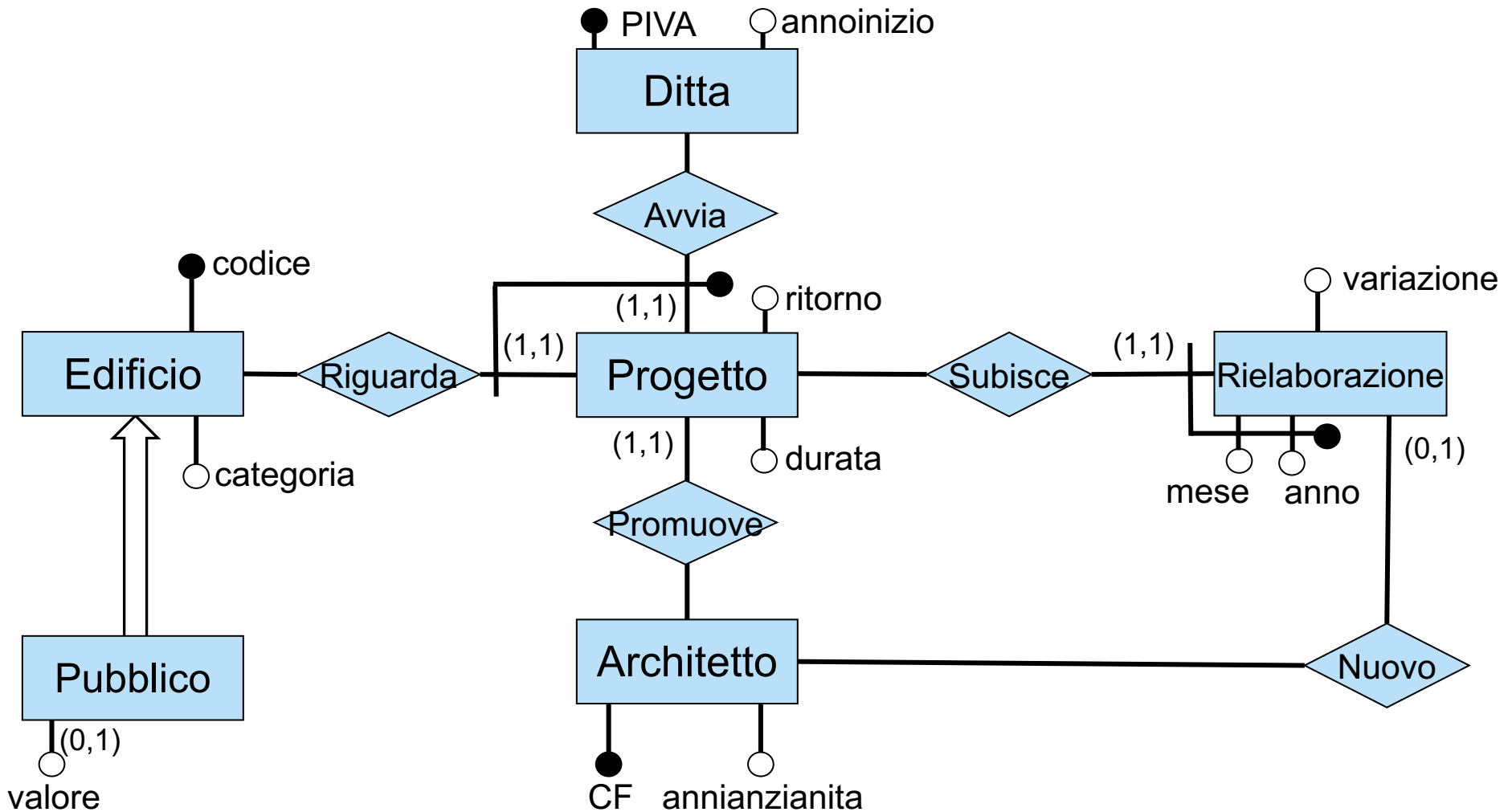
**Soluzioni dei problemi proposti  
nell'appello del 20-01-2023  
Compito B**

***Maurizio Lenzerini***

Anno Accademico 2022/23

# Problema 1 – Schema ER

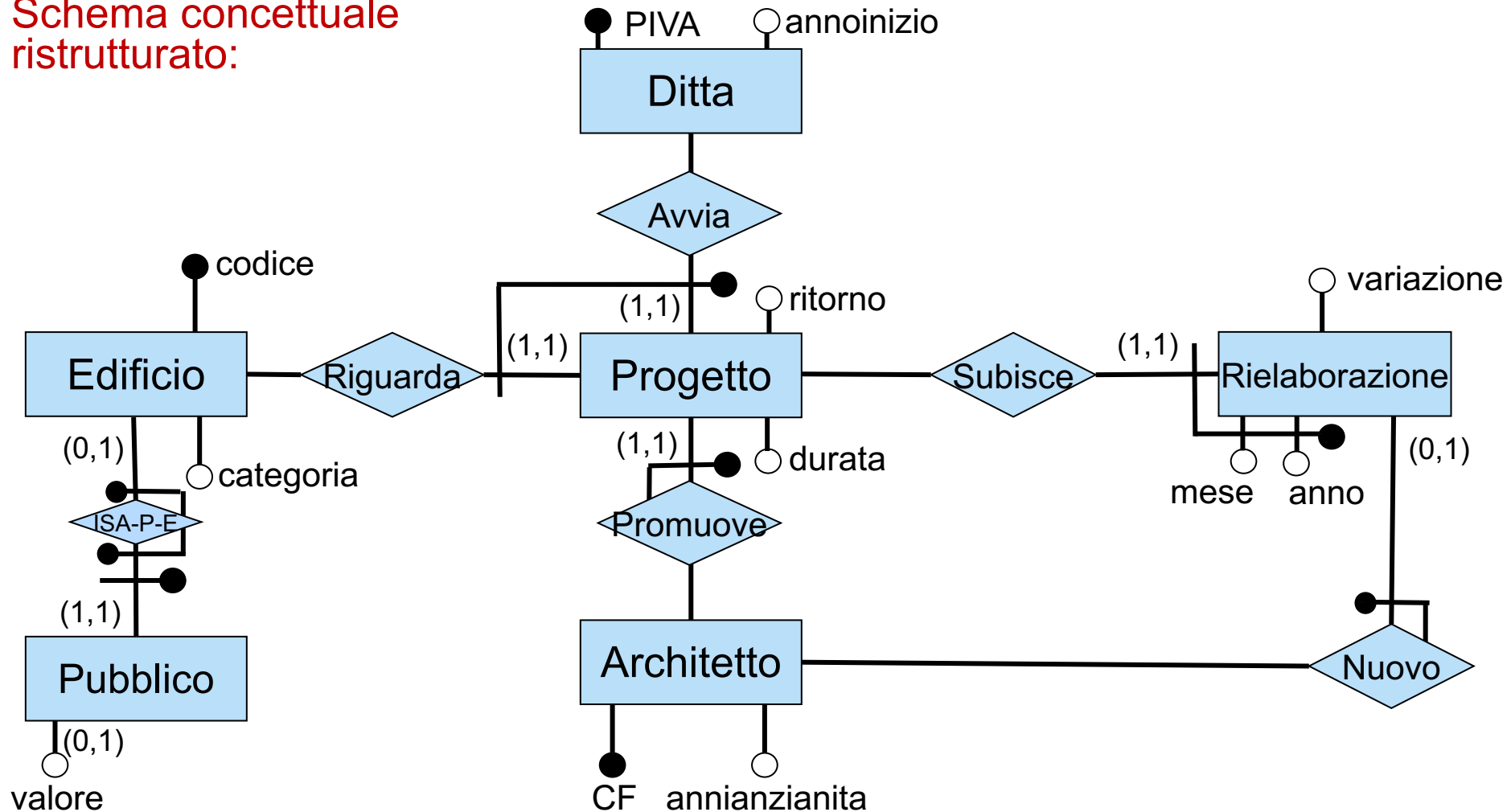
Schema concettuale:



**Vincolo esterno:** ogni istanza di Pubblico partecipa al massimo ad una istanza di Riguarda  
**Vincolo di dominio:** variazione  $\neq 0$

# Problema 2 – Ristrutturazione Schema ER

Schema concettuale ristrutturato:



**Vincolo esterno:** ogni istanza di Edificio che partecipa a ISA-P-E partecipa al massimo ad una istanza di Riguarda  
**Vincolo di dominio:** variazione  $\neq 0$

## Problema 2 – Traduzione diretta

Schema logico prodotto  
dalla traduzione diretta:

Ditta(piva,annoinizio)

Progetto(ditta,edificio,ritorno,durata)

foreign key: Progetto[ditta]  $\subseteq$  Ditta[piva]

foreign key: Progetti[edificio]  $\subseteq$  Edificio[codice]

foreign key: Progetto[ditta,edificio]  $\subseteq$  Promuove[dittaprogetto,edificioprogetto]

vincolo inter-relazionale: per ogni tupla t1 di Progetto tale che t1[edificio] è in Pubblico[codice],  
non esiste altra tupla t2 di Progetto tale che t2[edificio] = t1[edificio]

Promuove(dittaprogetto,edificioprogetto,architetto)

foreign key: Promuove[dittaprogetto,edificioprogetto]  $\subseteq$  Progetto[ditta,edificio]

foreign key: Promuove[architetto]  $\subseteq$  Architetto[CF]

Architetto(CF,annianzianita)

Edificio(codice,categoria)

Pubblico(codice,valore\*)

foreign key: Pubblico[codice]  $\subseteq$  Edificio[codice]

Rielaborazione(dittaprogetto,edificioprogetto,mese,anno,variazione)

foreign key: Rielaborazione[dittaprogetto,edificioprogetto]  $\subseteq$  Progetto[ditta,edificio]

vincolo di dominio: variazione  $\leq 0$

Nuovo(dittaprogetto,edificioprogetto,mese,anno,architetto)

foreign key: Nuovo[dittaprogetto,edificioprogetto,mese,anno]  $\subseteq$

Rielaborazione[dittaprogetto,edificioprogetto,mese,anno]

foreign key: Nuovo[architetto]  $\subseteq$  Architetto[CF]

## Problema 2 – Ristrutturazione dello schema logico

Schema logico prodotto  
dalla ristrutturazione:

1. La prima indicazione di progetto induce un accorpamento tra Progetto e Promuove, fortemente accoppiate.
2. Sempre la prima indicazione induce anche un accorpamento tra Rielaborazione e Nuovo, debolmente accoppiate.
3. La seconda indicazione induce un accorpamento tra Edificio e Pubblico, debolmente accoppiate.

Ditta(piva,annoinizio)

Progetto(ditta,edificio,ritorno,durata,architetto)

foreign key: Progetto[ditta]  $\subseteq$  Ditta[piva]

foreign key: Progetti[edificio]  $\subseteq$  Edificio[codice]

foreign key: Progetto[architetto]  $\subseteq$  Architetto[CF]

vincolo inter-relazionale: per ogni tupla t1 di Progetto tale che t1[edificio] è in Pubblico[codice], non esiste altra tupla t2 di Progetto tale che t2[edificio] = t1[edificio]

Architetto(CF,annianzianita)

Edificio(codice,categoria,flagpubblico,valore\*)

Rielaborazione(dittaprogetto,edificioprogetto,mese,anno,variazione,nuovoarchitetto\*)

foreign key: Rielaborazione[dittaprogetto,edificioprogetto]  $\subseteq$  Progetto[ditta,edificio]

vincolo di dominio: variazione  $\neq$  0

Definiamo anche le viste per ricostruire le relazioni accorpate:

View Promuove(d,e,a) = select ditta,edificio,architetto from Progetto

View Pubblico(c,v) = select codice,valore from Edificio where flagpubblico

View Nuovo(d,e,m,a,r) = select dittaprogetto,edificioprogetto,mese,anno,nuovoarchitetto  
from Rielaborazione

## Problema 3 – Ulteriori indicazioni di progetto

1. La prima indicazione induce la definizione di questo trigger:  
`create or replace function` blocca\_update\_progetto() `returns trigger as`  
`$$ BEGIN RETURN NULL; END;`  
`$$ language plpgsql;`  
  
`create trigger` trigger\_update\_progetto `before update on` Progetto  
`for each row execute procedure` blocca\_update\_progetto();
2. La seconda indicazione induce la seguente definizione del vincolo di foreign key da Progetto ad Edificio:  
`foreign key` (edificioprogetto) `references` Edificio `on delete cascade`

3. La terza indicazione induce la definizione di questo trigger:  
`create or replace function` inserimento\_rielaborazione() `returns trigger as`  
`$$ BEGIN`  
    IF (select annianzianita from Architetto where CF = NEW.nuovoarchitetto) <  
        (select annianzianita  
          from Architetto a join Progetto p on a.CF = p.architetto  
          where p.ditta = NEW.dittaprogetto and p.edificio = NEW.edificioprogetto)  
    THEN RETURN NULL;  
    ELSE RETURN NEW;  
    END IF;  
END;  
`$$ language plpgsql;`  
  
`create trigger` trigger\_inserimento\_rielaborazione `before insert on` Rielaborazione  
`for each row execute procedure` inserimento\_rielaborazione();

# Problema 4 – testo e soluzione

**Testo:** Riferendosi allo schema logico prodotto per il problema 2, scrivere una query SQL che per ogni progetto di ristrutturazione P restituisca la partita IVA della ditta costruttrice che ha avviato P, la categoria dell'edificio oggetto della ristrutturazione ed il numero di rielaborazioni di P che hanno cambiato l'architetto che segue il progetto.

## Soluzione:

```
select p.ditta, e.categoria, 0
from Progetto p join Edificio e on p.edificio = e.codice
where (p.ditta,p.edificio) not in
      (select r.dittaprogetto, r.edificioprogetto
       from Rielaborazione r
       where r.nuovoarchitetto is not null)
union
select p.ditta, e.categoria, count(*)
from Progetto p join Edificio e on p.edificio = e.codice
where (p.ditta,p.edificio) in
      (select r.dittaprogetto, r.edificioprogetto
       from Rielaborazione r
       where r.nuovoarchitetto is not null)
group by p.ditta, e.categoria
```

# Problema 5 – soluzione

## Prima parte

Relativamente alla prima parte del problema 5, ricordiamo solo la definizione di **superchiave** per lo schema di relazione R. Sia S un insieme non vuoto degli attributi di R.

1. Se r è una relazione coerente con R (ossia che soddisfa tutti i vincoli di R), S soddisfa la **condizione di superchiave in r** se non esistono due tuple in r che coincidono negli attributi S.
2. S è **una superchiave per R** se per ogni relazione r coerente con R, S soddisfa la condizione di superchiave in r.

## Seconda parte

Per la seconda parte dobbiamo individuare tutte le superchiavi per R, dove R è definita così:

$R(\underline{A}, B, \underline{C}, D)$

vincolo di tupla:  $A = C + D$

Siccome  $\{A, C\}$  è la chiave primaria per S, le seguenti sono chiaramente superchiavi di R:  $\{A, C\}$ ,  $\{A, C, B\}$ ,  $\{A, C, D\}$ ,  $\{A, C, B, D\}$ .

Il problema è adesso verificare se ci sono altre superchiavi per R. Notiamo che se r è una relazione coerente con R, date due diverse tuple t1 e t2 in r, se  $t1[C] = t2[C]$  e  $t1[D] = t2[D]$ , allora si ha anche  $t1[A] = t2[A]$ . Perciò, se  $\{C, D\}$  non fosse una superchiave in r, allora anche  $\{A, C\}$  non lo sarebbe, il che è una contraddizione, visto che  $\{A, C\}$  è la chiave primaria di R. Concludiamo quindi che  $\{C, D\}$  è una superchiave (e, in particolare, una chiave). Notiamo poi che il vincolo di tupla implica che per ogni tupla,  $C = A - D$ . Questo significa che, se r è una relazione coerente con R, date due diverse tuple t1 e t2 in r, se  $t1[D] = t2[D]$  e  $t1[A] = t2[A]$ , allora si ha anche  $t1[C] = t2[C]$ . Perciò, se  $\{A, D\}$  non fosse una superchiave in r, allora anche  $\{A, C\}$  non lo sarebbe, il che è una contraddizione, visto che  $\{A, C\}$  è la chiave primaria di R. Concludiamo quindi che anche  $\{A, D\}$  è una superchiave (e, in particolare, una chiave). A questo punto possiamo fare la lista di tutte le superchiavi di R:  $\{A, C\}$ ,  $\{A, C, B\}$ ,  $\{A, C, D\}$ ,  $\{A, C, B, D\}$ ,  $\{A, D\}$ ,  $\{A, D, B\}$ ,  $\{C, D\}$ ,  $\{C, D, B\}$ .



# Problema 5 – soluzione

*Terza parte (che nella correzione degli esami è stata considerata facoltativa)*

Dobbiamo infine motivare il perché non vi sono altre superchiavi per R. Notiamo che l'insieme di superchiavi che abbiamo individuato è tale per cui è una superchiave per R ogni insieme di attributi di R che contiene due attributi tra A,C e D.

Consideriamo un qualunque insieme S di attributi di R che non contiene due attributi tra A,C e D. Se S è vuoto, non è una superchiave per definizione. Assumiamo quindi S non vuoto e dimostriamo che S non è una superchiave per R definendo una relazione r coerente con R che contiene due tuple diverse tra loro t1,t2 che coincidono negli attributi in S, ossia dimostriamo che S non soddisfa la condizione di superchiave in r (il che appunto implica che S non è superchiave per R).

Per scegliere t1 e t2 distinguiamo due casi.

- 1. S non contiene alcuno tra gli attributi A,C,D.** Scegliamo un valore V e per ogni attributo  $x \in S$  poniamo  $t1[x] = t2[x] = V$ . Poniamo  $t1[A] = 0, t1[C] = 1, t1[D] = -1, t2[A] = 1, t2[C] = 3, t2[D] = -2$ . È immediato verificare che r soddisfa sia la condizione di superchiave per {A,C}, sia il vincolo di tupla  $A = C + D$ .
- 2. S contiene D uno tra gli attributi A,C,D.** Per B e per ogni attributo  $x \in S$  poniamo  $t1[x] = t2[x] = 0$ . Inoltre:
  1. Se S contiene A, allora poniamo  $t1[C] = 1, t1[D] = -1, t2[C] = -1, t2[D] = 1$ .
  2. Se S contiene C, allora  $t1[A] = 1, t1[D] = 1, t2[A] = -1, t2[D] = -1$ .
  3. Se S contiene D, allora  $t1[A] = 1, t1[C] = 1, t2[A] = -1, t2[C] = -1$ .

È immediato verificare che in tutti i casi r soddisfa sia la condizione di superchiave per {A,C}, sia il vincolo di tupla  $A = C + D$ .