

Esame di Basi di Dati

A.A. 2022/2023 – Appello del 20/01/2023 – Compito B

Problema 1

Si richiede di progettare lo schema concettuale Entità-Relazione di un'applicazione relativa alle ristrutturazioni di edifici da parte di ditte costruttrici, in accordo ai seguenti requisiti. Di ogni ditta costruttrice interessa la partita IVA (identificativo), l'anno di inizio attività ed i progetti di ristrutturazione da essa avviati. Ogni progetto riguarda un edificio, prevede un architetto che seguirà il progetto, ha un ritorno economico previsto ed ha una durata. L'architetto designato al momento dell'avvio di un progetto viene detto "promotore" del progetto stesso. Di ogni architetto interessa il codice fiscale (identificativo) e gli anni di anzianità. Si noti che, dato un edificio, ogni ditta costruttrice può avviare al massimo un progetto di ristrutturazione per quell'edificio. Un progetto può subire rielaborazioni (al massimo una al mese) al fine di registrare una variazione (diversa da 0) del ritorno economico ed eventualmente cambiare l'architetto che seguirà il progetto. Per ogni rielaborazione entrambe queste informazioni, ossia la variazione del ritorno economico e l'eventuale nuovo architetto, sono di interesse per l'applicazione, oltre al mese in cui tale rielaborazione viene sottoscritta. Ogni edificio è identificato da un codice ed appartiene ad una categoria. Degli edifici pubblici interessa anche il valore immobiliare (se noto) ed ognuno di essi può essere oggetto di al massimo un progetto di ristrutturazione.

Problema 2

Si richiede di effettuare la progettazione logica per l'applicazione citata nel problema 1, tenendo conto delle seguenti indicazioni: (i) quando si accede ad un progetto di ristrutturazione si vuole sempre sapere l'architetto promotore e quando si accede ad una rielaborazione del progetto si vuole sempre sapere qual è l'eventuale nuovo architetto che seguirà il progetto stesso; (ii) quando si accede ad un edificio si vuole sempre sapere se è pubblico oppure no e, nel caso in cui lo sia, qual è l'eventuale valore immobiliare.

Problema 3

Riferendosi all'applicazione menzionata nel problema 1 e 2, illustrare in che modo si deve adattare lo schema logico (se necessario, presentando anche frammenti di codice SQL, i cui eventuali errori sintattici saranno ignorati) alle seguenti ulteriori indicazioni di progetto: (i) non sono ammesse modifiche ai progetti di ristrutturazione, a parte ovviamente l'aggiunta di loro rielaborazioni; (ii) quando si elimina un edificio dalla base di dati occorre eliminare tutti i progetti di ristrutturazione che lo riguardano; (iii) quando si inserisce nella base di dati una rielaborazione di un progetto P , l'eventuale nuovo architetto che seguirà il progetto non può avere un numero di anni di anzianità inferiore a quelli dell'architetto promotore del progetto stesso.

Problema 4

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.

Problema 5

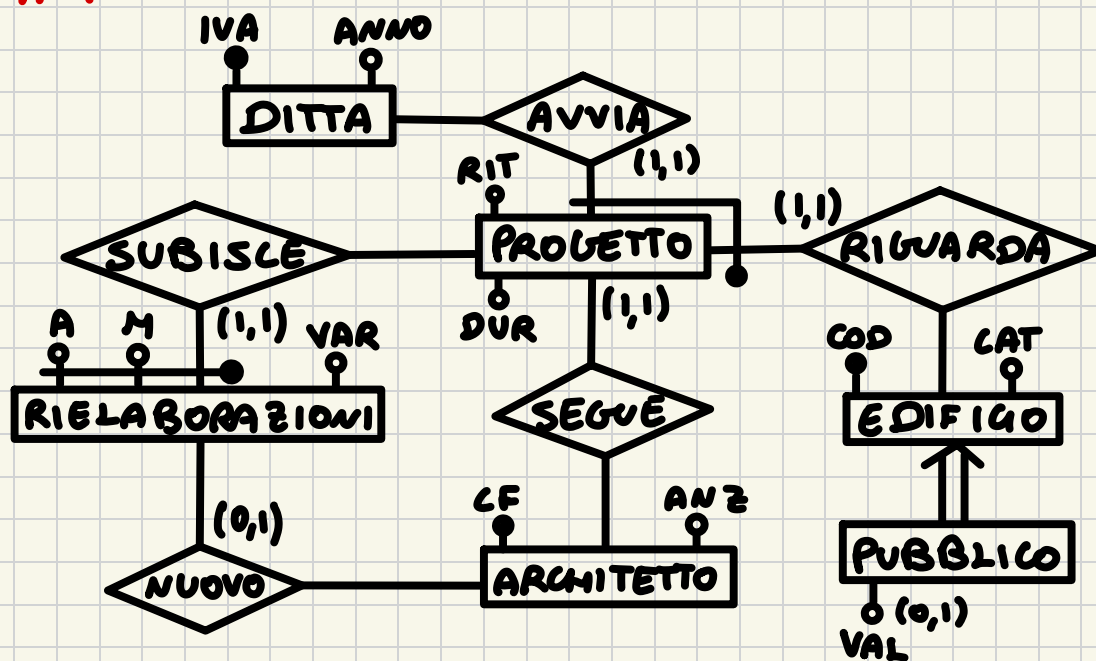
Dare le definizioni di (i) vincolo di integrità, (ii) vincolo di tupla, (iii) superchiave di una relazione e (iv) chiave di una relazione. Considerare poi il seguente schema di relazione (i cui attributi sono di tipo intero):

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

vincolo di tupla: $A = C + D$

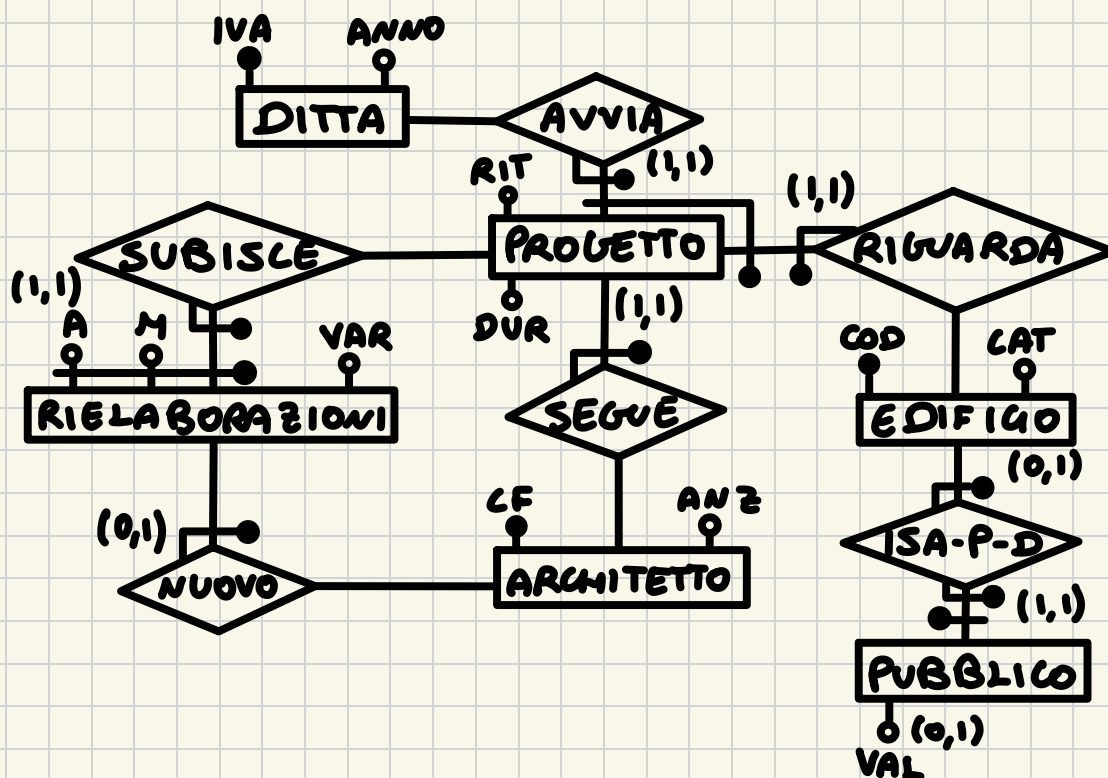
e fornire la lista di tutte le superchiavi per R . In altre parole, si chiede di mostrare tutti gli insiemi di attributi di R che, in ogni relazione coerente con lo schema R , soddisfano la condizione di superchiave. Per ogni elemento della lista occorre anche specificare la ragione per la quale quell'elemento è una superchiave per R , e, in ultimo, motivare il perché non vi sono altre superchiavi rispetto a quelle indicate nella lista.

PROBLEMA 1



OGNI ISTANZA DI EDIFICIO PUBBLICO PARTECIPA AL MASSIMO AD UNA ISTANZA DI RIGUARDA

PROBLEMA 2



OGNI ISTANZA DI EDIFICIO CHE PARTECIPA AD IS-P-E PARTECIPA AL MASSIMO AD UNA ISTANZA DI RIGUARDA

DITTA (IVA, ANNO)

PROGETTO (DITTA, EDIFICIO, RITORNO, DURATA)

FOREIGN KEY: PROG [DITTA] \subseteq DITTA [IVA]

FOREIGN KEY: PROG [EDIFICIO] \subseteq EDI [COD]

FOREIGN KEY: PROG [DITTA, EDIFICIO] \subseteq SEGUE [DITTA, EDIFICIO]

VINCOLO: OGNI ISTANZA DI PUBBLICO PUÒ PART UNA SOLA VOLTA A PROGETTO

EDIFICIO (COD, CAT)

ARCHITETTO (CF, ANZ)

PUBBLICO (EDIFICIO, VALORE*)

FOREIGN KEY: PUB [ED] \subseteq ED [COD]

SEGUE (DITTA, EDIFICIO, ARCHITETTO)

FOREIGN KEY: SE [DITTA, EDI] \subseteq PROG [DITTA, EDI]

FOREIGN KEY: SE [ARC] \subseteq ARC [CF]

RIELABORAZIONI (DITTA, EDIFICIO, ANNO, MESE, VARIAZIONE)

FOREIGN KEY: RIEL [DITTA, EDIF] \subseteq PROG [DITTA, EDIF]

NUOVO (DITTA, EDIFICIO, ANNO, MESE, ARCHITETTO)

FOREIGN KEY: N [DITTA, EDIF, ANNO, MESE] \subseteq RIEL [DITTA, EDIF, A, M]

FOREIGN KEY: N [ARC] \subseteq ARC [CF]



ii ALLORPIAMO PROGETTO CON SEGUE

PROGETTO (DITTA, EDIFICIO, RITORNO, DURATA, ARCHITETTO)

FOREIGN KEY: PROG [DITTA] \subseteq DITTA [IVA]

FOREIGN KEY: PROG [EDIFICIO] \subseteq EDI [COD]

FOREIGN KEY: PROG [ARC] \subseteq ARC [CF]

VINCOLO: OGNI ISTANZA DI PUBBLICO PUÒ PART UNA SOLA VOLTA A PROGETTO

ALLORPIAMO RIELABORAZIONI CON NUOVO

RIELABORAZIONI (DITTA, EDIFICIO, ANNO, MESE, VARIAZIONE, NUOVO ARC*)

FOREIGN KEY: RIEL [DITTA, EDIF] \subseteq PROG [DITTA, EDIF]

FOREIGN KEY: RIEL [NUOVO ARC] \subseteq ARC [CF]

ii EDIFICIO (COD, CAT, FLAG PUBBLICO, VALORE*)

PROBLEMA 3

1) 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) FOREIGN KEY (EDIFICIO) REFERENCES EDIFICIO ON DELETE CASCADE

3) CREATE OR REPLACE FUNCTION INSERT_RIELAB() RETURNS TRIGGER AS

```
$$ BEGIN
```

```
IF (SELECT ANZ  
    FROM ARCHITETTO  
    WHERE CF = NEW.NUOVO_ARCH) >=
```

```
(SELECT A.ANZ  
    FROM ARCHITETTO A JOIN PROGETTO P  
    ON A.CF = P.ARCHITETTO  
    WHERE P.DITTA = NEW.DITTA AND P.EDIFICIO = NEW.EDIFICIO)
```

```
THEN RETURN NEW;  
ELSE RETURN NULL;  
END IF;  
END;
```

```
$$ LANGUAGE PLPGSQL;
```

```
CREATE TRIGGER TRIGGER_INSERT_RIELAB() BEFORE INSERT ON  
RIELABORAZIONE FOR EACH ROW EXECUTE PROCEDURE  
INSERT_RIELAB();
```

PROBLEMA 4

```
SELECT P.DITTA, E.CATEGORIA, COUNT(*) AS X
FROM ((PROGETTO P JOIN EDIFICIO E ON P.EDIFICIO = E.COD)
      JOIN RIELABORAZIONI R ON R.DITTA = P.DITTA
      AND R.EDIFICIO = P.EDIFICIO)
WHERE R.NUOVOARCHITETTO IS NOT NULL
GROUP BY P.DITTA, E.CATEGORIA

UNION

SELECT P.DITTA, E.CATEGORIA, 0 AS X
FROM PROGETTO P JOIN EDIFICIO E ON P.EDIFICIO = E.COD
WHERE (P.DITTA, E.CATEGORIA) NOT IN
      (SELECT DITTA, EDIFICIO
       FROM RIELABORAZIONI
       WHERE NUOVOARC IS NOT NULL)

GROUP BY P.DITTA, E.CATEGORIA
```

PROBLEMA 5

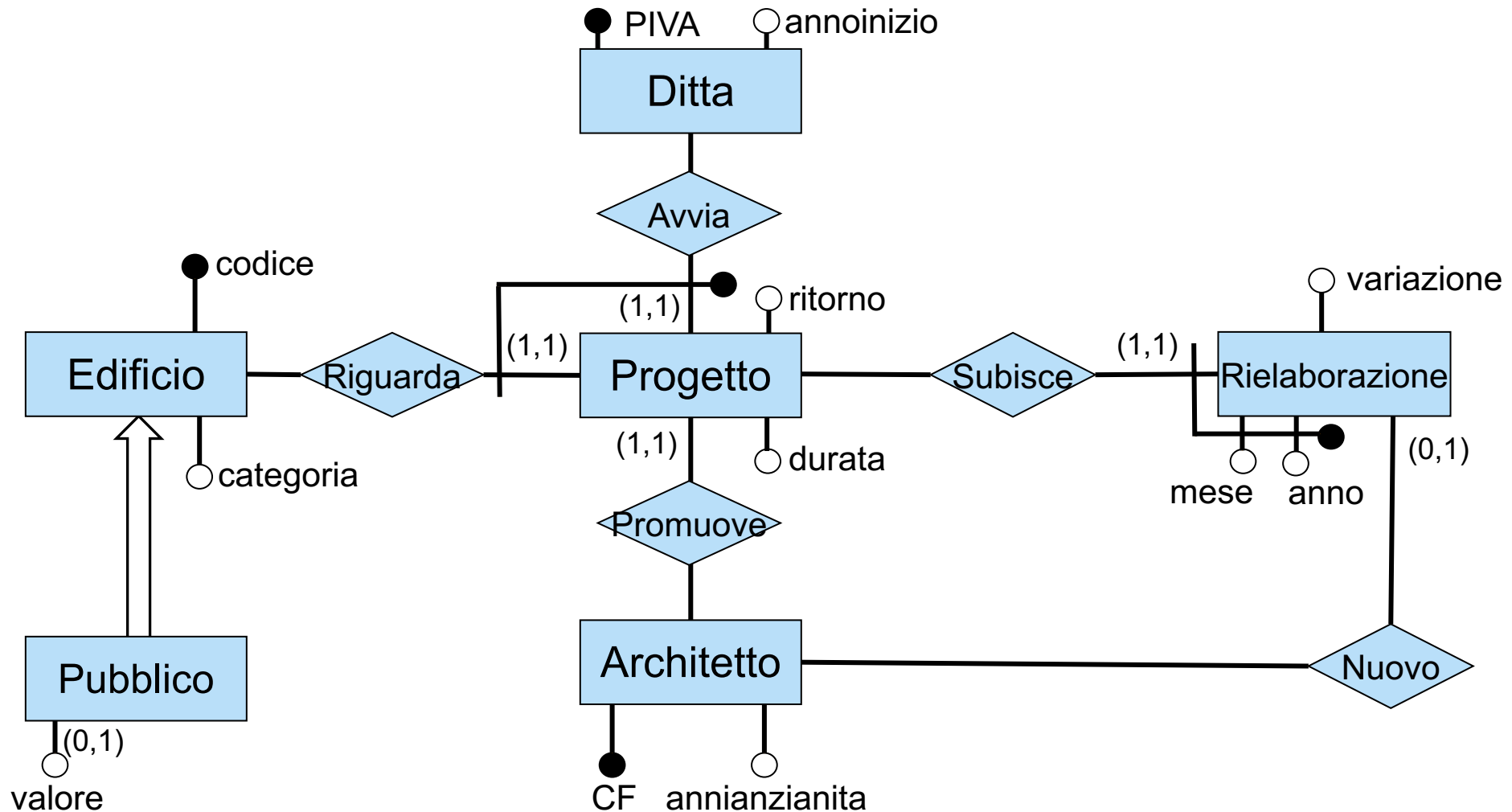
- 1) UN VINCOLO DI INT È UNA CONDIZIONE CHE SI ESPRIME A LIVELLO DI SCHEMA E CHE DEVE ESSERE SODDISFATTA DA OGNI ISTANZA DELLA BASE DI DATI
- 2) UN VINCOLO DI TUPLA ESPRIME UNA CONDIZIONE CHE DEVE ESSERE SODDISFATTA PER OGNI TUPLA DELLA RELAZIONE
- 3) SIA K UN INSIEME NON VUOTO DI ATTRIBUTI DI UNA RELAZIONE, K SODDISFA LA CONDIZIONE DI SUPERCHIAVE SE NON ESISTONO DUE TUPLE τ_1 E τ_2 TALE CHE $\tau_1[A_1] = \tau_2[A_1] \dots \tau_1[A_n] = \tau_2[A_n]$.
- 4) K SODDISFA LA CONDIZIONE DI CHIAVE IN R SE K È UNA SUPERCHIAVE MINIMALE IN R , OVE SE NESSUN SOTTOINSIEME DI K È UNA SUPERCHIAVE IN R .

$R(A, B, C, D)$ VINCOLO DI TUPLA: $A = C + D$

SUPERCHIAVI: $\{A, C\}, \{A, B, C\}, \{A, C, D\}, \{A, B, C, D\}$
 $\{A, D\}, \{A, B, D\}, \{C, D\}, \{B, C, D\}$

Problema 1 – Schema ER

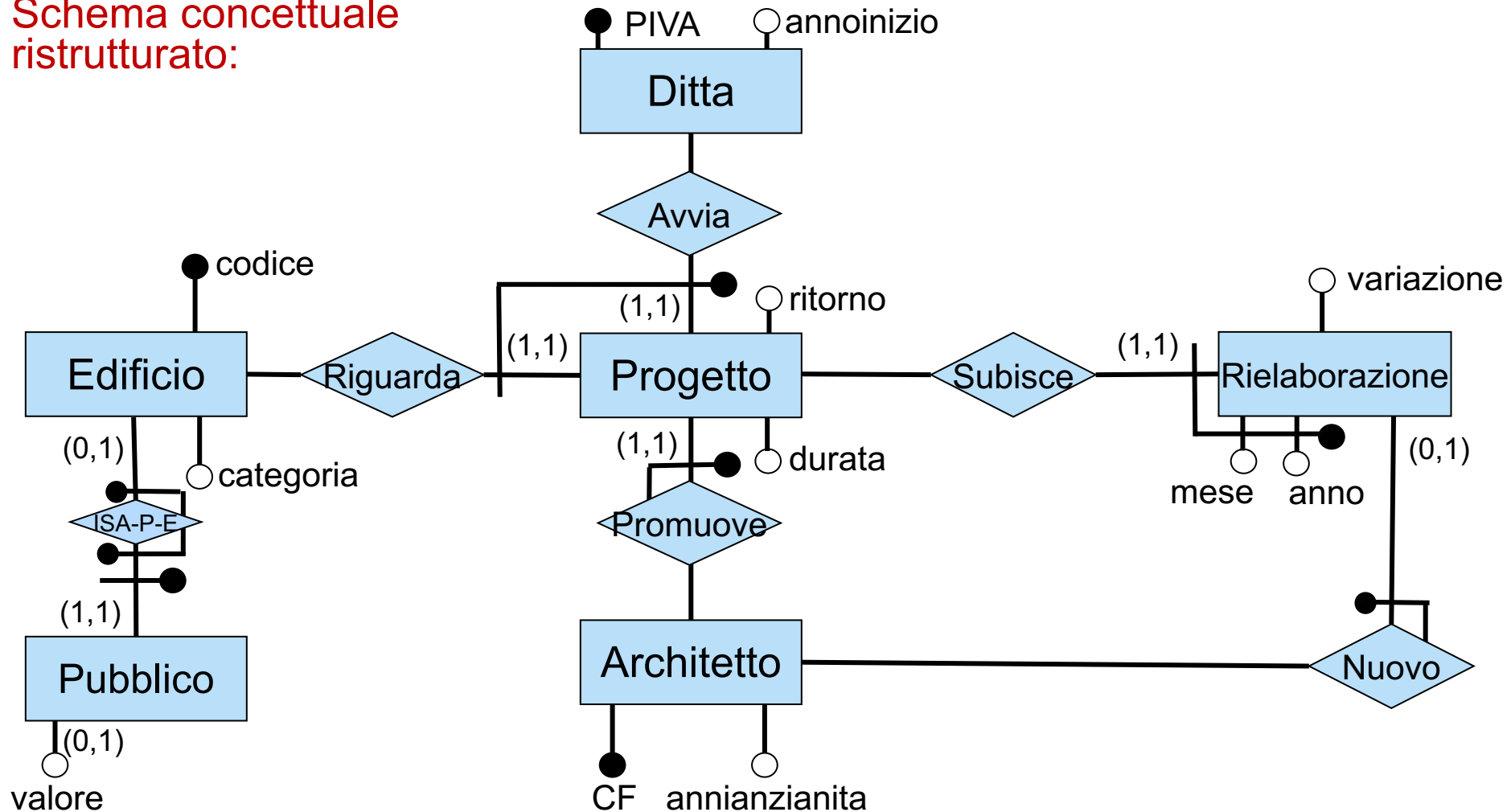
Schema concettuale:



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 $\neq 0$

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 $A = C + D$.