

# Esame di Basi di Dati

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

### Problema 1

Si richiede di progettare lo schema concettuale Entità-Relazione di un'applicazione relativa ai contratti di un insieme di aziende, in accordo ai seguenti requisiti. Di ogni azienda interessa la partita IVA (identificativo), il capitale sociale ed i contratti da essa stipulati. Ogni contratto ha per oggetto un servizio (chiamato "servizio nativo" del contratto), è stipulato da un'azienda fornitrice e da un'azienda beneficiaria ed ha associato un costo ed una penale. Si noti che un servizio può essere il servizio nativo di un qualunque numero di contratti (anche 0) e che ogni azienda non può essere fornitrice di più di un contratto con lo stesso servizio nativo. Un contratto può subire delle revisioni (al massimo una al mese) al fine di registrare un incremento (maggiore di 0) del costo ed eventualmente cambiare il servizio oggetto del contratto. Per ogni revisione di contratto entrambe queste informazioni, ossia l'incremento del costo e l'eventuale nuovo servizio, sono di interesse per l'applicazione, oltre alla data in cui tale revisione viene sottoscritta. Ogni servizio è identificato da un codice ed appartiene ad una categoria. Le consulenze sono particolari servizi, per le quali vale la condizione che ogni consulenza è il servizio nativo di al massimo un contratto e può avere associato un costo standard.

### 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 contratto si vuole sempre sapere l'azienda beneficiaria e quando si accede ad una revisione di un contratto si vuole sempre sapere qual è l'eventuale nuovo servizio stabilito in tale revisione; (ii) quando si accede ad un servizio si vuole sempre sapere se è di consulenza oppure no e, nel caso in cui lo sia, qual è l'eventuale costo standard.

### 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) quando si elimina un contratto dalla base di dati occorre eliminare tutte le revisioni che lo riguardano; (ii) non sono ammesse modifiche ai contratti, a parte ovviamente l'aggiunta di loro revisioni; (iii) quando si inserisce nella base di dati una revisione di un contratto che ha come servizio nativo una consulenza, l'eventuale nuovo servizio previsto dalla revisione deve essere anch'esso una consulenza.

### Problema 4

Riferendosi allo schema logico prodotto per il problema 2, scrivere una query SQL che per ogni contratto  $C$  restituisca la partita IVA dell'azienda fornitrice di  $C$ , il codice del servizio nativo di  $C$ , la categoria del servizio nativo di  $C$  ed il numero delle revisioni subite da  $C$  che hanno cambiato il servizio oggetto del contratto.

### Problema 5

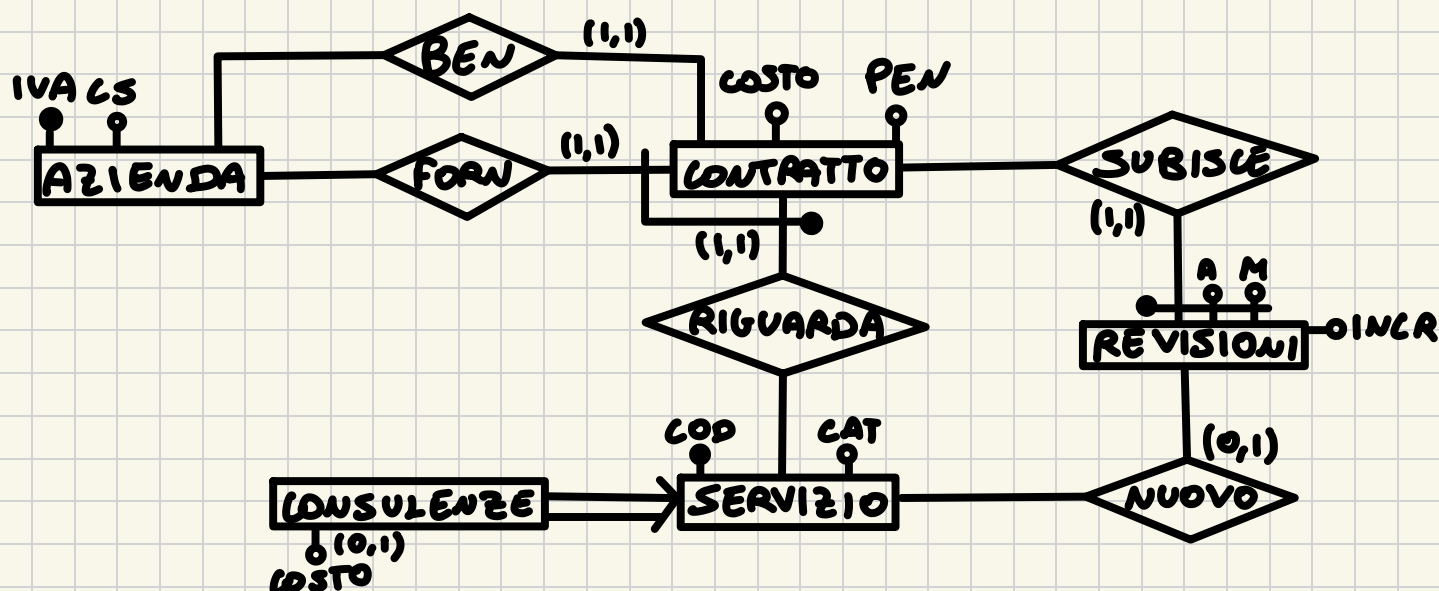
Dare le definizioni di (i) vincolo di integrità, (ii) vincolo di tupla, (iii) superchiave di una relazione e (iv) chiave di una relazione. Considerare poi il seguente schema di relazione

$R(A, B, C, D)$

vincolo intra-relazionale: non esistono  $t_1, t_2 \in R$  tali che  $t_1[D] = t_2[D]$  e  $t_1[C] \neq t_2[C]$

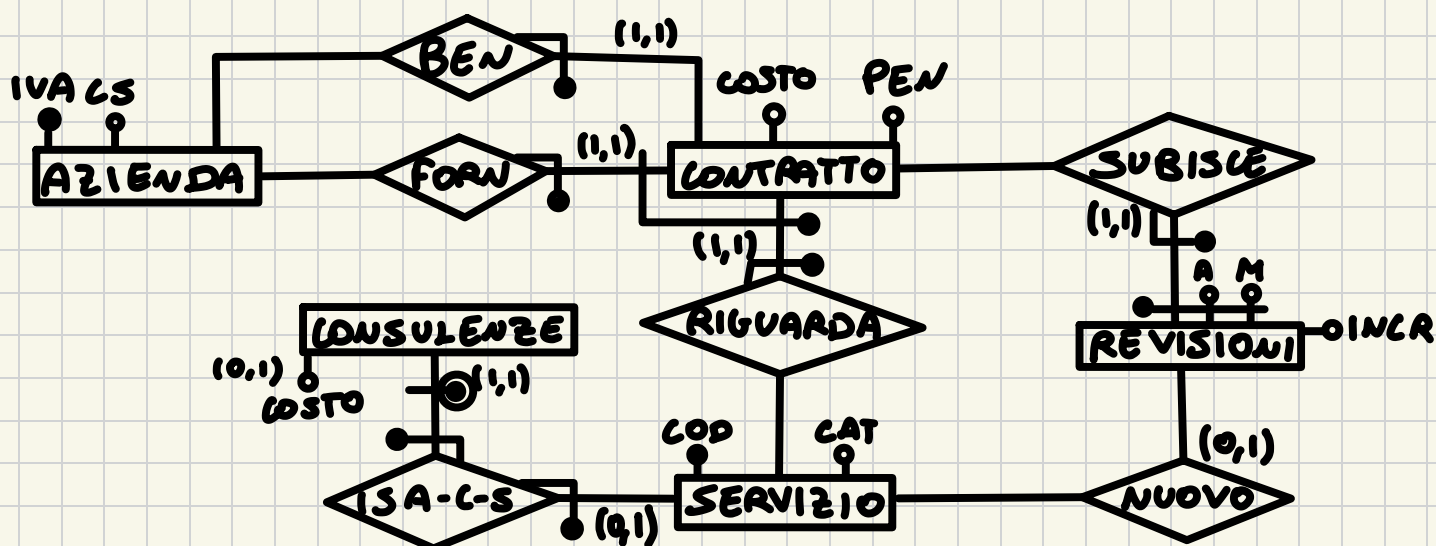
e fornire la lista di tutte le superchiavi per  $R$ . In altre parole, si chiede di mostrare tutti gli insiemi di attributi di  $R$  che, in ogni relazione coerente con lo schema  $R$ , soddisfano la condizione di superchiave. Per ogni elemento della lista occorre anche specificare la ragione per la 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 CONSULENZA PARTECIPA AL MASSIMO AD UNA ISTANZA DI RIGUARDA

## PROBLEMA 2



OGNI ISTANZA CHE PARTECIPA AD ISA-C-S PARTECIPA AL MASSIMO AD UNA DI RIGUARDA

AZIENDA (IVA, CS)

CONTRATTO (FORN, SERVIZIO, COSTO, PENALE)

FK: CONTR [FORN]  $\subseteq$  AZIENDA [IVA]

FK: CONTR [SERV]  $\subseteq$  SERV [COD]

FK: CONTR [FORN, SERV]  $\subseteq$  BEN [FORN, SERV]

VINCOLO: OGNI ISTANZA DI CONSULENZA PUÒ PART AD UNA SOLA DI CONTRATTO

BENEFICIARIA (FORN, SERV, AZIENDA)

FK: BEN [FORN, SERV]  $\subseteq$  CONTR [FORN, SERV]

FK: BEN [AZIENDA]  $\subseteq$  AZIENDA [IVA]

SERVIZIO (COD, CATEGORIA)

CONSULENZA (SERV, COSTO\*)  
FK: CONS [SERV]  $\subseteq$  SERV [COD]

REVISIONI (ANNO, MESE, FORN, SERV, INCR)  
FK: REV [FORN, SERV]  $\subseteq$  CONTR [FORN, SERV]

NUOVO (FORN, SERV, ANNO, MESE, NUOV SERV)  
FK: NUOVO [FORN, SERV, ANNO, MESE]  $\subseteq$  CONTR [FORN, SERV, ANNO, MESE]  
FK: NUOVO [NSERV]  $\subseteq$  SERV [COD]



i) ACCORPIAMO CONTRATTO CON BEN, REVISIONE CON NUOVO

CONTRATTO (FORN, SERVIZIO, COSTO, PENALE, BEN)  
FK: CONTR [FORN]  $\subseteq$  AZIENDA [IVA]  
FK: CONTR [SERV]  $\subseteq$  SERV [COD]  
FK: CONTR [BEN]  $\subseteq$  AZIENDA [IVA]

VINCOLO: OGNI ISTANZA DI CONSULENZA PUÒ PART AD UNA SOLA DI CONTRATTI

REVISIONI (ANNO, MESE, FORN, SERV, INCR, NUOV SERV\*)  
FK: REV [FORN, SERV]  $\subseteq$  CONTR [FORN, SERV]  
FK: REV [NOV SERV]  $\subseteq$  SERV [COD]

ii) SERVIZIO (COD, CATEGORIA, FLAGCONS, COSTO\*)

### PROBLEMA 3

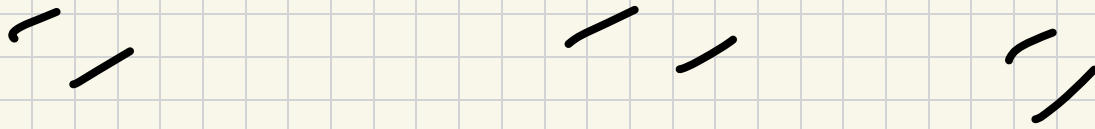
i) FOREIGN KEY (FORN, SERV) REFERENCES CONTRATTO ON DELETE CASCADE

2) CREATE OR REPLACE FUNCTION BLOCCA.UPDATE.CONTRATTO()  
RETURNS TRIGGER AS

```
$$ BEGIN RETURN NULL;  
END;  
$$ LANGUAGE PLPGSQL;
```

CREATE TRIGGER TRIGGER.UPDATE.CONTRATTO BEFORE UPDATE ON  
CONTRATTO FOR EACH ROW EXECUTE PROCEDURE  
BLOCCA.UPDATE.CONTRATTO();

3) CREATE OR REPLACE FUNCTION INSERIMENTO\_REV() RETURNS TRIGGER AS  
\$\$ BEGIN



#### PROBLEMA 4

```
SELECT C.FORN, C.SERV, S.CATEGORIA, COUNT(*)  
FROM CONTATTO C JOIN SERVIZIO S ON C.SERVIZIO = S.COD  
WHERE (C.FORN, C.SERV) IN (SELECT A.FORN, A.SERV  
                           FROM REVISIONE R  
                           WHERE R.NUOVOSERV IS NOT NULL)  
GROUP BY C.FORN, C.SERV, S.CAT
```

#### 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  $\tau_1$  E  $\tau_2$  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  $R$ , OÙ SE NESSUN SOTTOINSIEME DI  $K$  È UNA SUPERCHIAVE IN  $R$

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

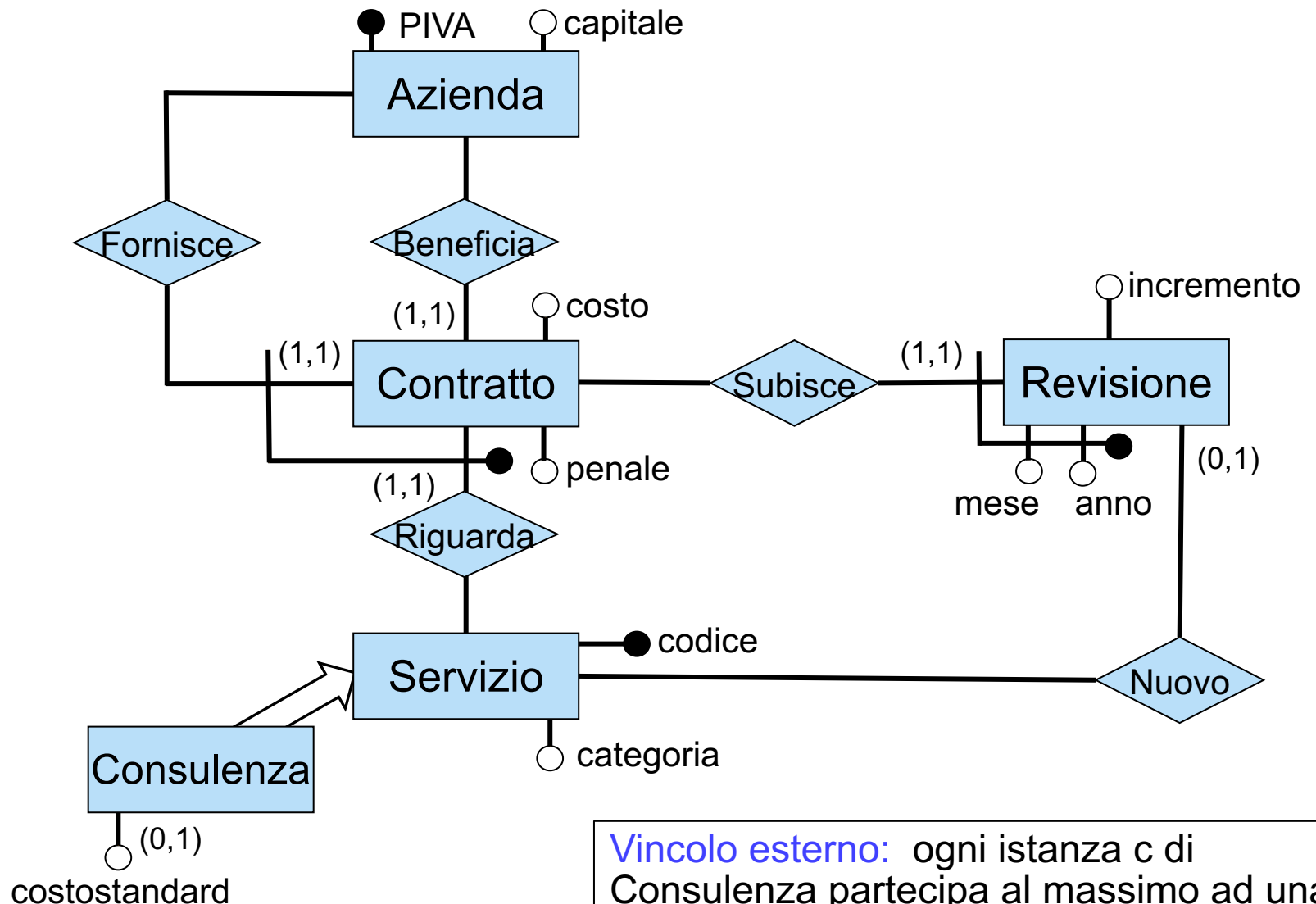
vincolo intra-relazionale: non esistono  $t_1, t_2 \in R$  tali che  $t_1[D] = t_2[D]$  e  $t_1[C] \neq t_2[C]$

$\{A, C\}$  CHIAVE  $\rightarrow$  SUPERCHIAVE

$\{A, B, C, D\}, \{A, B, C\}, \{A, C, D\}$  SUPERCHIAVE

# Problema 1 – Schema ER

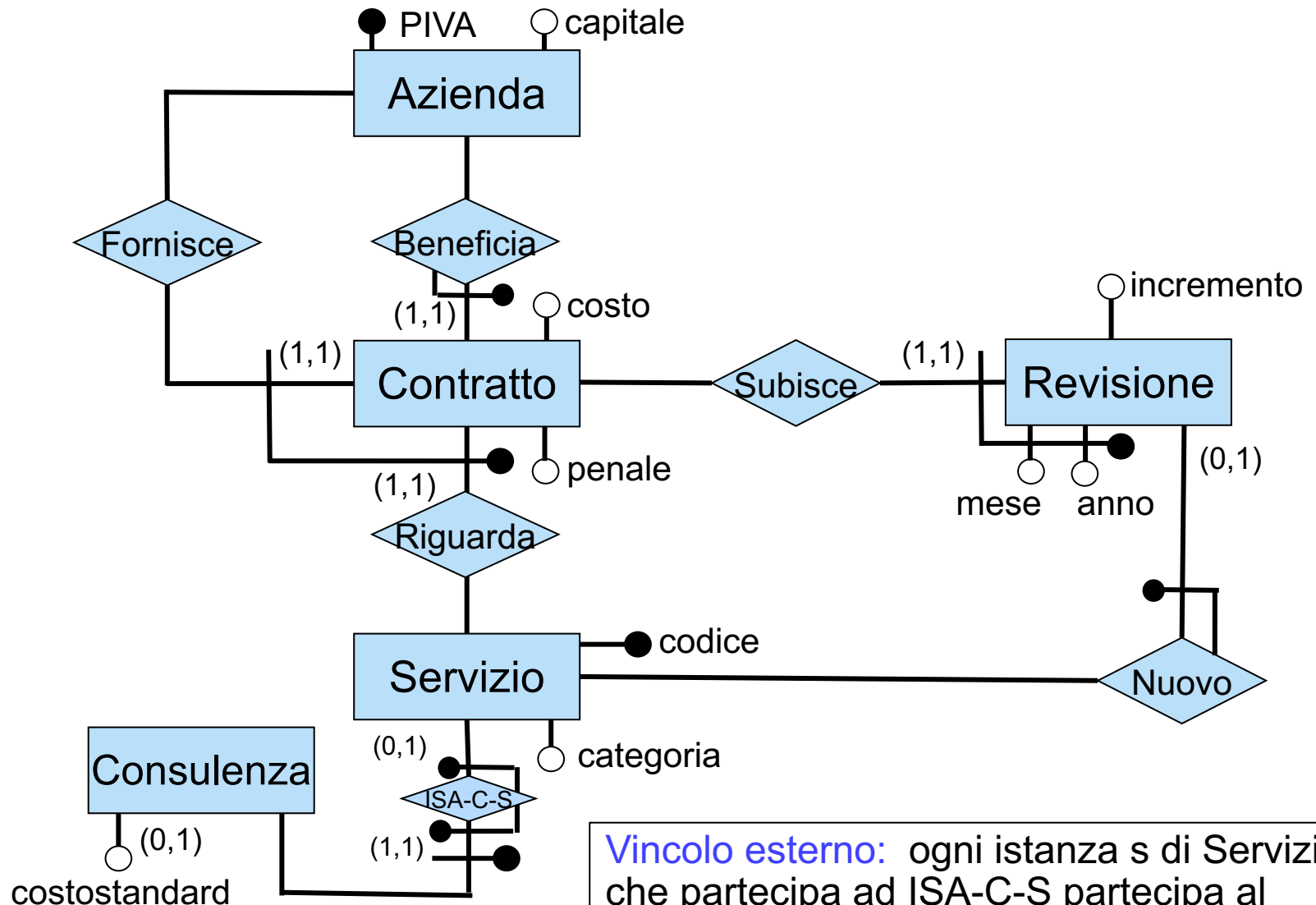
Schema concettuale:



**Vincolo esterno:** ogni istanza c di Consulenza partecipa al massimo ad una istanza di Riguarda.  
**Vincolo di dominio:** incremento > 0

# Problema 2 – Ristrutturazione Schema ER

Schema concettuale ristrutturato:



**Vincolo esterno:** ogni istanza s di Servizio che partecipa ad ISA-C-S partecipa al massimo ad una istanza di Riguarda.

**Vincolo di dominio:** incremento > 0

## Problema 2 – Traduzione diretta

Schema logico prodotto  
dalla traduzione diretta:

Azienda(piva, capitale)

Contratto(fornitore, servizionativo, costo, penale)

foreign key: Contratto[fornitore]  $\subseteq$  Azienda[piva]

foreign key: Contratto[servizionativo]  $\subseteq$  Servizio[codice]

foreign key: Contratto[fornitore, servizionativo]  $\subseteq$  Beneficia[fornitorecontratto, servizionativocontratto]

vincolo inter-relazionale: per ogni tupla t1 di Contratto tale che t1[servizionativo] è in Consulenza[codice],  
non esiste altra tupla t2 di Contratto tale che t2[servizionativo] = t1[servizionativo]

Beneficia(fornitorecontratto, servizionativocontratto, beneficiario)

foreign key: Beneficia[fornitorecontratto, servizionativocontratto]  $\subseteq$  Contratto[fornitore, servizionativo]

foreign key: Beneficia[beneficiario]  $\subseteq$  Azienda[piva]

Servizio(codice, categoria)

Consulenza(codice, costostandard\*)

foreign key: Consulenza[codice]  $\subseteq$  Servizio[codice]

Revisione(fornitorecontratto, servizionativocontratto, mese, anno, incremento)

foreign key: Revisione[fornitorecontratto, servizionativocontratto]  $\subseteq$  Contratto[fornitore, servizionativo]

vincolo di dominio: incremento > 0

Nuovo(fornitorecontratto, servizionativocontratto, mese, anno, servizio)

foreign key: Nuovo[fornitorecontratto, servizionativocontratto, mese, anno]  $\subseteq$

Revisione[fornitorecontratto, servizionativocontratto, mese, anno]

foreign key: Nuovo[servizio]  $\subseteq$  Servizio[codice]



## Problema 2 – Ristrutturazione dello schema logico

### Schema logico prodotto dalla ristrutturazione:

1. La prima indicazione di progetto induce un accorpamento tra Contratto e Beneficia, fortemente accoppiate.
2. Sempre la prima indicazione induce anche un accorpamento tra Revisione e Nuovo, debolmente accoppiate.
3. La seconda indicazione induce un accorpamento tra Servizio e Consulenza, debolmente accoppiate.

Azienda(piva, capitale)

Contratto(fornitore, servizionativo, costo, penale, beneficiario)

**foreign key:** Contratto[fornitore]  $\subseteq$  Azienda[piva]

**foreign key:** Contratto[servizionativo]  $\subseteq$  Servizio[codice]

**foreign key:** Contratto[beneficiario]  $\subseteq$  Azienda[piva]

**vincolo inter-relazionale:** per ogni tupla t1 di Contratto tale che esiste una tupla t2 di Servizio con  
t2[codice] = t1[servizionativo] e t2[flagconsulenza] = true, non esiste altra tupla  
t3 di Contratto tale che t2[servizionativo] = t1[servizionativo]

Servizio(codice, categoria, flagconsulenza, costostandard\*)

Revisione(fornitorecontratto, servizionativocontratto, mese, anno, incremento, nuovoservizio\*)

**foreign key:** Revisione[fornitorecontratto, servizionativocontratto]  $\subseteq$  Contratto[fornitore, servizionativo]

**foreign key:** Revisione[nuovoservizio]  $\subseteq$  Servizio[codice]

**vincolo di dominio:** incremento > 0

Definiamo anche le viste per ricostruire le relazioni accorpate:

View Beneficia(f,s,b) = select fornitore, servizionativo, beneficiario from Contratto

View Consulenza(c,s) = select codice, costostandard from Servizio where flagconsulenza

View Nuovo(f,s,m,a,n) = select fornitorecontratto, servizionativocontratto, mese, anno, nuovoservizio  
from Revisione

## Problema 3 – Ulteriori indicazioni di progetto

1. La prima indicazione induce la seguente definizione del vincolo di foreign key da Revisione a Contratto:  
`foreign key (fornitorecontratto, servizionativocontratto) references Contratto on delete cascade`
2. La seconda indicazione induce la definizione di questo trigger:  
`create or replace function blocca_update_contratto() returns trigger as  
$$ BEGIN RETURN NULL; END;  
$$ language plpgsql;`  
  
`create trigger trigger_update_contratto before update on Contratto  
for each row execute procedure blocca_update_contratto();`
3. La terza indicazione induce la definizione di questo trigger:  
`create or replace function inserimento_revisione() returns trigger as  
$$ BEGIN  
    IF (select flagconsulenza from Servizio where codice = NEW.servizionativocontratto)  
        and not  
        (select flagconsulenza from Servizio where codice = NEW.nuovoservizio)  
    THEN RETURN NULL;  
    ELSE RETURN NEW;  
    END IF;  
END;  
$$ language plpgsql;`  
  
`create trigger trigger_inserimento_revisione before insert on Revisione  
for eachrow execute procedure inserimento_revisione();`

## Problema 4 – testo e soluzione

**Testo:** Riferendosi allo schema logico prodotto per il problema 2, scrivere una query SQL che per ogni contratto C restituisca la partita IVA dell'azienda fornitrice di C, il codice del servizio nativo di C, la categoria del servizio nativo di C ed il numero delle revisioni subite da C che hanno cambiato il servizio oggetto del contratto.

### Soluzione:

```
select c.fornitore, c.servizionativo, s.categoria, 0
from Contratto c join Servizio s on c.servizionativo = s.codice
where (c.fornitore,c.servizionativo) not in
      (select r.fornitorecontratto, r.servizionativocontratto
       from Revisione r
       where r.nuovoservizio is not null)
union
select c.fornitore, c.servizionativo, s.categoria, count(*)
from Contratto c join Servizio s on c.servizionativo = s.codice
where (c.fornitore,c.servizionativo) in
      (select r.fornitorecontratto, r.servizionativocontratto
       from Revisione r
       where r.nuovoservizio is not null)
group by c.fornitore, c.servizionativo, s.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 intra-relazionale: non esistono  $t_1, t_2 \in R$  tali che  $t_1[D] = t_2[D]$  e  $t_1[C] \neq t_2[C]$

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 il vincolo intra-relazionale si può esprimere così: **per qualunque coppia di tuple  $t_1, t_2 \in R$ , se  $t_1[D] = t_2[D]$ , allora  $t_1[C] = t_2[C]$** . Da questo si evince immediatamente che D si comporta come C rispetto alla condizione di superchiave. In particolare, si può verificare che  $\{A, D\}$  è una superchiave (e anche una chiave) per R semplicemente notando che se  $\{A, D\}$  non fosse una superchiave, allora esisterebbe una relazione r coerente con R che contiene due tuple  $t_1, t_2$  diverse tra loro tali che  $t_1[A] = t_2[A]$  e  $t_1[D] = t_2[D]$ . Ma questo implicherebbe  $t_1[A] = t_2[A]$  e  $t_1[C] = t_2[C]$ , contraddicendo il fatto che  $\{A, C\}$  rispetta la condizione di superchiave in r. Possiamo quindi concludere che anche le seguenti sono superchiavi di R, visto che contengono la chiave  $\{A, D\}$ :  $\{A, D\}$ ,  $\{A, D, B\}$ .

# 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

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

il che significa che è una superchiave per R ogni insieme di attributi di R che contiene A e C, oppure contiene A e D.

Consideriamo un qualunque insieme S di attributi di R che non contiene entrambi A e C e non contiene entrambi A e D. Se S è vuoto, non è una superchiave per definizione. Altrimenti, la ragione per cui S non è una superchiave per R è che possiamo definire una relazione r coerente con R che contiene due tuple diverse tra loro  $t_1, t_2$  che coincidono negli attributi in S, dimostrando che S non soddisfa la condizione di superchiave in r (il che appunto implica che S non è superchiave per R).

Per scegliere  $t_1$  e  $t_2$  distinguiamo due casi.

1. **S contiene D** (e quindi non contiene A). Scegliamo un valore V e per ogni attributo  $x \in S$  poniamo  $t_1[x] = t_2[x] = V$  e anche  $t_1[C] = t_2[C]$ . Scegliamo due valori W1 e W2 diversi tra loro e per ogni attributo  $y \notin S$  diverso da C poniamo  $t_1[y] = W1$  e  $t_2[y] = W2$ . È facile verificare che abbiamo  $t_1[A] \neq t_2[A]$ ,  $t_1[D] = t_2[D]$  e  $t_1[C] = t_2[C]$  e quindi r soddisfa sia la condizione di superchiave per  $\{A,C\}$ , sia il vincolo intra-relazionale.
2. **S non contiene D**. Scegliamo ancora un valore V e per ogni attributo  $x \in S$  poniamo  $t_1[x] = t_2[x] = V$ . Scegliamo due valori W1 e W2 diversi tra loro e per ogni attributo  $y \notin S$  poniamo  $t_1[y] = W1$  e  $t_2[y] = W2$ . Siccome  $D \notin S$ , abbiamo  $t_1[D] \neq t_2[D]$  e quindi r soddisfa il vincolo intra-relazionale. Per quanto riguarda la condizione di superchiave per  $\{A,C\}$ , valgono le seguenti osservazioni: se  $C \notin S$ , allora abbiamo  $t_1[C] \neq t_2[C]$ , altrimenti  $C \in S$  e allora  $A \notin S$  e quindi abbiamo  $t_1[A] \neq t_2[A]$ . In entrambi i casi r soddisfa la condizione di superchiave per  $\{A,C\}$ .