

# Esame di Basi di Dati

## A.A. 2020/2021 – Appello del 21/01/2021

### Problema 1

Si richiede di progettare lo schema concettuale Entità-Relazione relativo al sistema informativo di una banca, in accordo ai seguenti requisiti. Sono di interesse i prestiti di denaro che le agenzie della banca assegnano ai clienti della banca stessa. Sono ovviamente ammessi più prestiti allo stesso cliente, anche dalla stessa agenzia, ma la banca impedisce che ad un cliente venga assegnato, complessivamente dalle agenzie della banca stessa, più di un prestito al mese. Di ogni prestito interessa: la data in cui è stato assegnato, la somma di denaro prestata e il tasso di interesse iniziale. Di ogni agenzia interessa il codice identificativo e il capitale che gestisce. I clienti di interesse sono quelli che hanno avuto almeno un prestito da una qualunque agenzia e di ognuno di essi interessa il codice fiscale, l'età ed il sesso. Il tasso di interesse di un prestito può subire variazioni (positive o negative), ma non più di una al giorno. Al variare del tasso di interesse di un prestito, interessa registrare la data della variazione, il codice del dirigente della banca che ha approvato la variazione e l'ammontare della variazione.

### Problema 2

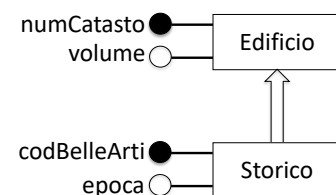
Si richiede di effettuare la progettazione logica, tenendo conto dell'indicazione che il numero di variazioni del tasso di interesse da registrare è dell'ordine delle centinaia di milioni, e che alle variazioni del tasso di interesse positive si accede sempre separatamente da quelle negative.

### Problema 3

Si consideri una base di dati che comprende le relazioni **Persona**(CF, dataNascita), **Nascita**(CF, comune) e **haLavorato**(CF, comune), dove la prima memorizza il codice fiscale (chiave primaria) e la data di nascita delle persone, mentre le altre due relazioni memorizzano, per ogni persona, rispettivamente il comune di nascita (se noto) ed i comuni in cui ha lavorato (una persona può non aver mai lavorato). Si sa che le uniche foreign key che sono soddisfatte nella base di dati sono quella da **Nascita**[CF] a **Persona**[CF] e quella da **haLavorato**[CF] a **Persona**[CF]. Si scrivano in SQL le seguenti query: (1) Calcolare *tutte* le persone, con il codice fiscale, la data di nascita e, se noto, anche il comune di nascita. (2) Calcolare il codice fiscale di *tutte* le persone che hanno lavorato *solo* nel comune di nascita. (3) Per *ogni* persona, calcolare il suo codice fiscale ed il numero di comuni in cui ha lavorato. Si scriva poi la query (2) anche in algebra relazionale.

### Problema 4

Si consideri lo schema concettuale mostrato qui a destra e si esegua la progettazione logica a partire da esso, tenendo conto del fatto che quando si accede ad un edificio si usa il numero catastale e si vuole spesso sapere se è storico o no, mentre quando si accede ad un edificio storico si usa il codice delle Belle Arti e si vuole spesso sapere il suo numero catastale.



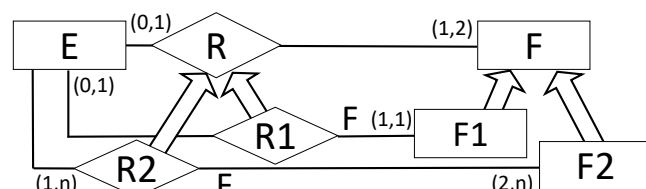
### Problema 5

Si consideri lo schema relazionale mostrato qui a destra e si risponda alle seguenti domande (1) quali sono i vincoli di chiavi che sono soddisfatti nella relazione **R** (in altre parole, quali sono le chiavi di **R**)? (2) Quali sono i vincoli di superchiave che sono soddisfatti nella relazione **R** (in altre parole, quali sono le superchiavi di **R**)? (3) Quali sono i vincoli di chiave esterna (foreign key) che sono soddisfatti nella base di dati?

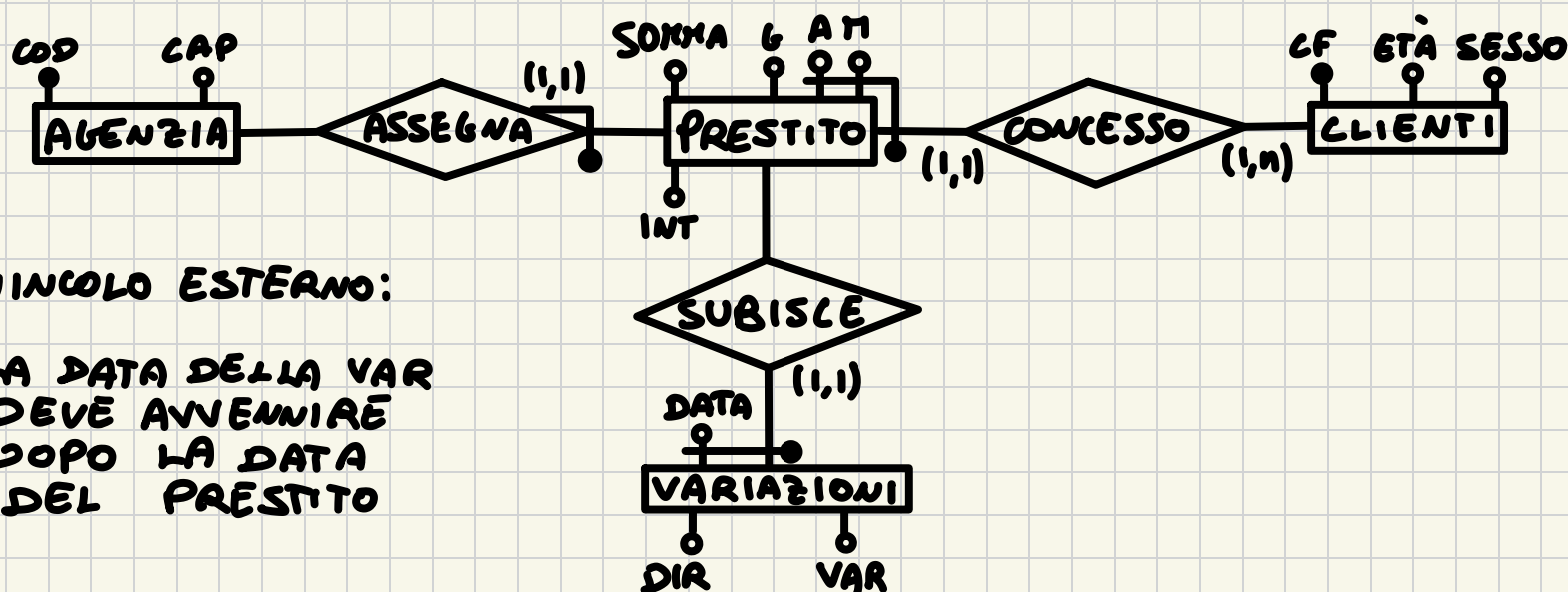
R				S
A	B	C	D	E
8	8	7	1	8
8	8	8	2	9
null	7	6	3	10
8	6	8	4	

### Problema 6

Si consideri lo schema concettuale *S* mostrato qui a destra e si risponda alla seguente domanda: esiste una istanza dello schema *S* in cui esiste una istanza di **F1** che è anche istanza di **F2**? Se la risposta è positiva, mostrare una istanza di *S* con la caratteristica richiesta; se la risposta è negativa motivare in dettaglio la risposta.



## PROBLEMA 1



VINCOLO ESTERNO:

LA DATA DELLA VAR  
DEVE AVVENIRE  
DOPO LA DATA  
DEL PRESTITO

## PROBLEMA 2

AGENZIA (CODICE, CAPITALE)

PRESTITO (CLIENTE, MESE, ANNO, GIORNO, SOMMA, INT)

FOREIGN KEY: PRESTITO[CLIENTE]  $\subseteq$  CLIENTE[CF]

FOREIGN KEY: PRESTITO[CL, M, A]  $\subseteq$  ASSEGNA[CL, M, A]

ASSEGNA (CLIENTE, ANNO, MESE, AGENZIA)

FOREIGN KEY: ASS[CL, A, M]  $\subseteq$  PRESTITO[CL, A, M]

FOREIGN KEY: ASS[AG]  $\subseteq$  AG[CODICE]

CLIENTE (CF, ETÀ, SESSO)

INCLUSIONE: CL[CF]  $\subseteq$  PRESTITO[CL]

VARIAZIONI (DATA, ANNO, MESE, CLIENTE, DIRETTORE, VARIAZIONE)

FOREIGN KEY: VAR[ANNO, MESE, CLIENTE]  $\subseteq$  PREST[A, M, CL]

VINCOLO DI DOMINIO: VARIAZIONE  $< 0$

VINCOLO ESTERNO: VARIAZIONE[DATA]  $>$  PRESTITO[DATA]



VARIAZIONI NEGATIVE (DATA, ANNO, MESE, CLIENTE, DIRETTORE, VARIAZIONE)

FOREIGN KEY: VARNEG[ANNO, MESE, CLIENTE]  $\subseteq$  PREST[A, M, CL]

VINCOLO DI DOMINIO: VARIAZIONE  $< 0$

VINCOLO ESTERNO: VARIAZIONENEG[DATA]  $>$  PRESTITO[DATA]

VARIAZIONI POSITIVE (DATA, ANNO, MESE, CLIENTE, DIRETTORE, VARIAZIONE)

FOREIGN KEY: VARPOS[ANNO, MESE, CLIENTE]  $\subseteq$  PREST[A, M, CL]

VINCOLO DI DOMINIO: VARIAZIONI  $> 0$

VINCOLO: VARNEG[DATA, A, M, CL]  $\cap$  VARPOS[D, A, M, CL] =  $\emptyset$

VINCOLO ESTERNO: VARIAZIONEPOS[DATA]  $>$  PRESTITO[DATA]

### PROBLEMA 3

1) SELECT DISTINCT P.CF, P.DATANASCITA, N.COMUNE  
FROM PERSONA P JOIN NASCITA N ON P.CF = N.CF  
  
UNION

SELECT P.CF, P.DATANASCITA, NULL  
FROM PERSONA  
WHERE P.CF NOT IN (SELECT CF FROM NASCITA)

2) SELECT CF  
FROM PERSONA

EXCEPT

SELECT H.CF  
FROM HALAVORATO H JOIN NASCITA N ON H.CF = N.CF  
WHERE H.COMUNE != N.COMUNE

3) SELECT H.CF, COUNT(\*)  
FROM HALAVORATO H  
GROUP BY H.CF

UNION

SELECT P.CF, 0  
FROM PERSONA P  
WHERE P.CF NOT IN (SELECT CF FROM HALAVORATO)

2) PROJ<sub>CF</sub> (PERSONA)

PROJ<sub>CF</sub> (NASCITA JOIN<sub>C=CF AND COMUNE != M</sub> REN<sub>C=CF, π=COMUNE</sub> (HALAVORATO))

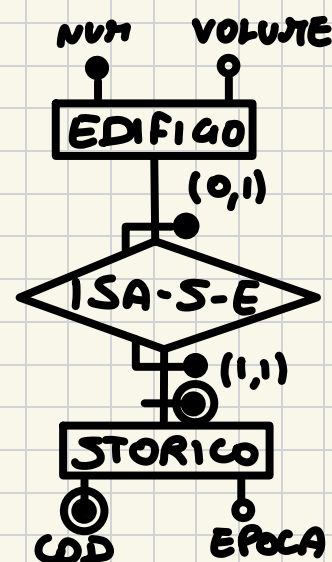
### PROBLEMA 4

EDIFICIO(NUM, VOLUME)

STORICO(COD, EPOCA)  
FK: ST[COD] ≤ ISA[ST]

ISA-S-E(STORICO, EDIFICIO)  
FK: ISA[ST] ≤ ST[COD]  
FK: ISA[ED] ≤ ED[NUM]  
CHIAVE: EDIFICIO

EDIFICIO(NUM, VOL, STORICO\*)



## PROBLEMA 5

- 1) LE CHIAVI DI R SONO:  $\{D\}, \{B, C\}$
- 2) SUPERCHIAVI:  $\{A, B, C, D\}, \{D\}, \{A, D\}, \{B, D\}, \{C, D\}$   
 $\{B, C, D\}, \{A, B, D\}, \{A, C, D\}, \{A, B, C\}$
- 3) FOREIGN KEY:  $R[A] \subseteq S[E]$

## PROBLEMA 6

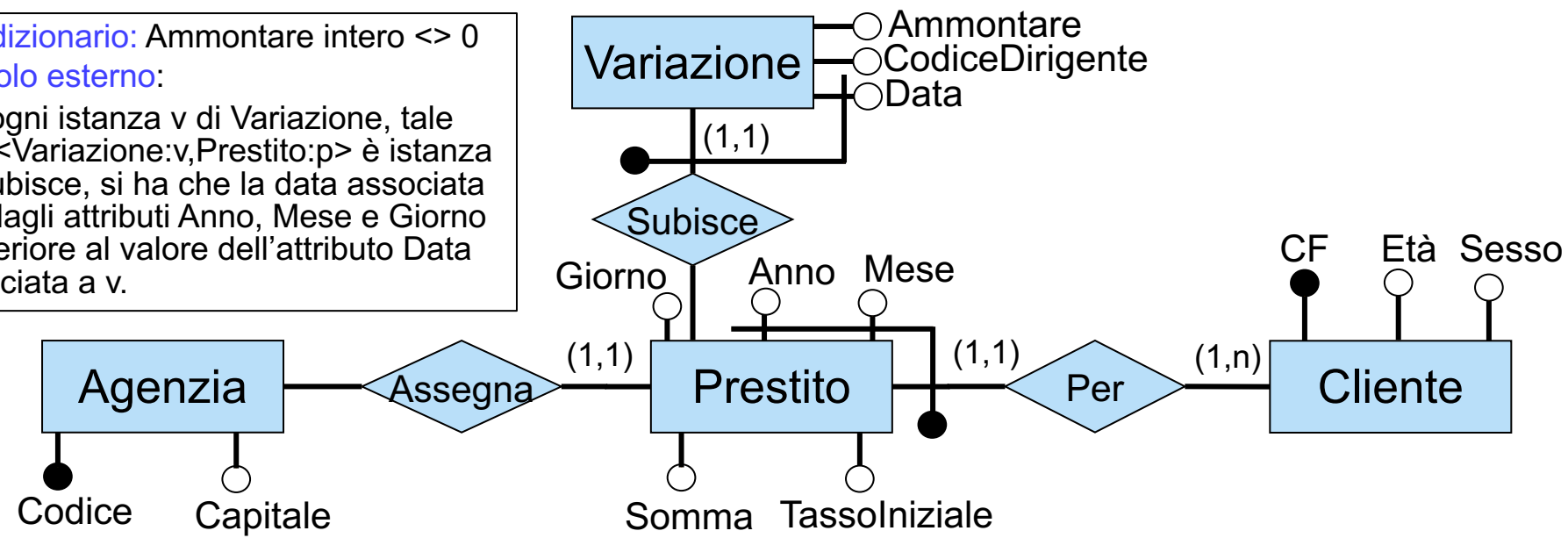
## Problema 1 e 2

### Nota per gli studenti:

- L'assenza dei vincoli esterni nel compito NON determina perdita di punti
- La presenza dei vincoli esterni nel compito determina un guadagno di punti

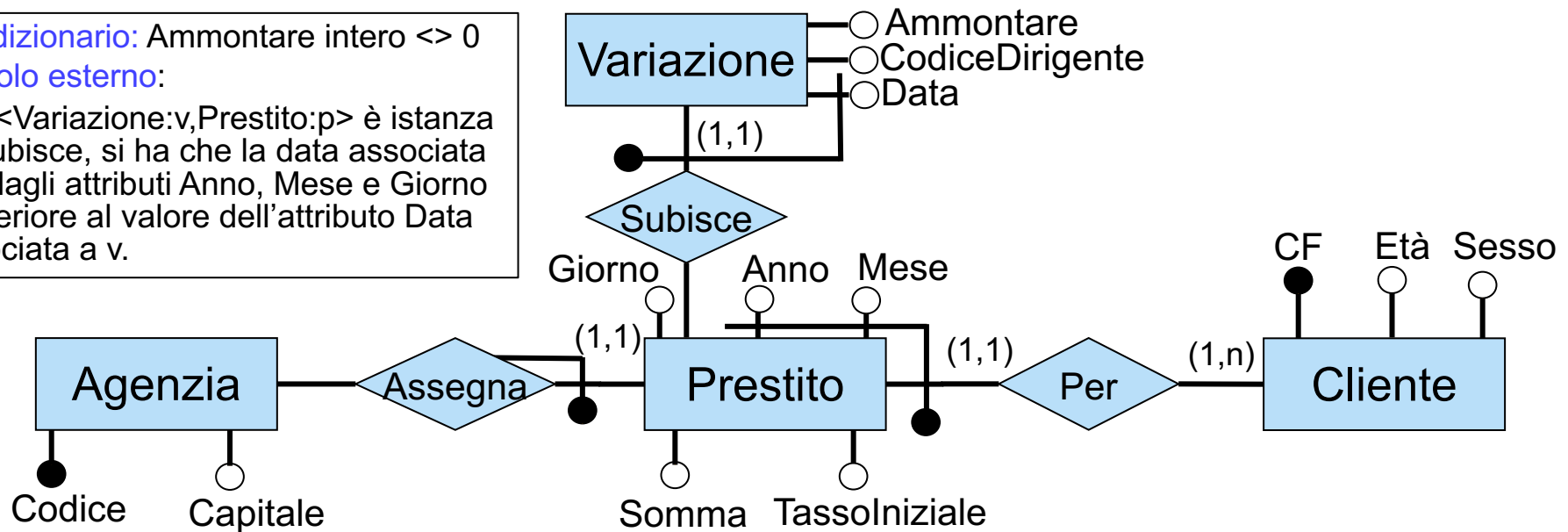
# Problema 1 e 2 – Schema ER e sua ristrutturazione

**Nel dizionario:** Ammontare intero <> 0  
**Vincolo esterno:**  
per ogni istanza v di Variazione, tale che <Variazione:v,Prestito:p> è istanza di Subisce, si ha che la data associata a p dagli attributi Anno, Mese e Giorno è inferiore al valore dell'attributo Data associata a v.



Lo schema ristrutturato è praticamente identico allo schema concettuale:

**Nel dizionario:** Ammontare intero <> 0  
**Vincolo esterno:**  
che <Variazione:v,Prestito:p> è istanza di Subisce, si ha che la data associata a p dagli attributi Anno, Mese e Giorno è inferiore al valore dell'attributo Data associata a v.



## Problema 2 – Traduzione diretta e ristrutturazione schema logico

### Schema relazionale prodotto dalla traduzione diretta:

Agenzia(codice, capitale)  
Assegna(agenzia, clienteprestito, anno, mese)  
foreign key: Assegna[agenzia]  $\subseteq$  Agenzia[codice]  
foreign key: Assegna[clienteprestito, anno, mese]  $\subseteq$  Prestito[cliente, anno, mese]  
Prestito(cliente, anno, mese, giorno, somma, tassoiniziale)  
foreign key: Prestito[cliente, anno, mese]  $\subseteq$  Assegna[clienteprestito, anno, mese]  
foreign key: Prestito[cliente]  $\subseteq$  Cliente[cf]  
Cliente(cf, età, sesso)  
inclusione: Cliente[cf]  $\subseteq$  Prestito[cliente]  
Variazione(cliente, anno, mese, data, ammontare, codicedirigente)  
foreign key: Variazione[cliente, anno, mese]  $\subseteq$  Prestito[cliente, anno, mese]  
vincolo di dominio: ammontare  $\neq 0$   
vincolo esterno: in ogni tupla del join naturale con Prestito,  $\langle \text{giorno}, \text{mese}, \text{anno} \rangle$  è inferiore a data

### Ristrutturazione dello schema relazionale - lo schema ristrutturato è identico a prima, con l'unica differenza che Variazione viene sostituita dalle due tabelle:

VariazionePositiva(cliente, anno, mese, data, ammontare, codicedirigente)  
foreign key: VariazionePositiva[cliente, anno, mese]  $\subseteq$  Prestito[cliente, anno, mese]  
vincolo di dominio: ammontare  $> 0$   
vincolo esterno: in ogni tupla del join naturale con Prestito, giorno, mese, anno è inferiore a data  
vincolo di disgiunzione: VariazionePositiva[cliente, anno, mese, data]  $\cap$  VariazioneNegativa[cliente, anno, mese, data] =  $\emptyset$   
VariazioneNegativa(cliente, anno, mese, data, ammontare, codicedirigente)  
foreign key: VariazioneNegativa[cliente, anno, mese]  $\subseteq$  Prestito[cliente, anno, mese]  
vincolo di dominio: ammontare  $< 0$   
vincolo esterno: in ogni tupla del join naturale con Prestito,  $\langle \text{giorno}, \text{mese}, \text{anno} \rangle$  è inferiore a data

# Problema 3 – soluzione

Query 1:

```
select CF, datanascita, comune
from Persona left outer join Nascita on Persona.CF = Nascita.CF
```

Query 2:

```
select p.CF
from Persona
where CF not in (select h.CF
                 from haLavorato h
                 where h.comune not in (select comune
                                         from Nascita
                                         where CF = h.CF))
```

Query 2 in algebra relazionale:

$\text{PROJ}_{CF}(\text{Persona}) -$   
 $\text{PROJ}_{CF}(\text{Nascita JOIN}_{C=CF \text{ and } comune \neq M} \text{REN}_{C \leftarrow CF, M \leftarrow comune}(\text{haLavorato}))$

Query 3:

```
select CF, count(*)
from haLavorato
group by CF
union
select CF, 0
from Persona
where CF not in (select CF from haLavorato)
```

oppure →

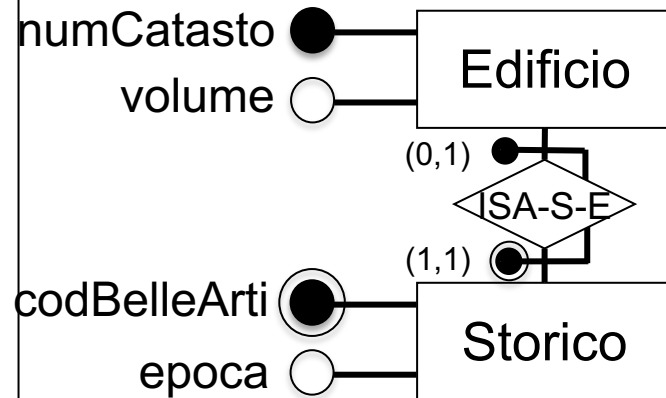
Query 3:

```
select CF, count(comune)
from Persona left outer join haLavorato
group by CF
```



## Problema 4 – soluzione

## Schema ER ristrutturato:



## Schema logico prodotto della traduzione diretta:

```
Edificio(numCatasto,volume)
ISA-S-E(codBelleArti,numCatasto)
    foreign key: ISA-S-E[numCatasto]  $\subseteq$  Edificio[numCatasto]
    foreign key: ISA-S-E[codBelleArti]  $\subseteq$  Storico[codBelleArti]
    chiave: numCatasto
Storico(codBelleArti,epoca)
    foreign key: Storico[codBelleArti]  $\subseteq$  ISA-S-E[codBelleArti]
```

## Ristrutturazione dello schema logico:

- 1) accorpamento di ISA-S-E con Storico (accoppiamento forte)
- 2) Edificio e la nuova tabella sono accoppiate debolmente e si può quindi portare nella tabella Edificio il flag che indica se un edificio è storico.

Edificio(numCatasto,volume,flagStorico)  
     **vincolo esterno:** per ogni tupla t di Edificio, t.flagStorico = true se e solo se  
                                 t.numCatasto in Storico[numCatasto]

Storico(codBelleArti,epoca,numCatasto)  
     **foreign key:** Storico[numCatasto]  $\subseteq$  Edificio[numCatasto]  
     **chiave:** numCatasto

## Problema 5 e 6 – soluzione

### Problema 5

- 1) Le chiavi di R sono:  $\{D\}$ ,  $\{B,C\}$
- 2) Le superchiavi di R sono:  $\{D\}$ ,  $\{D,A\}$ ,  $\{D,B\}$ ,  $\{D,C\}$ ,  $\{D,A,B\}$ ,  $\{D,A,C\}$ ,  $\{D,B,C\}$ ,  $\{D,A,B,C\}$ ,  $\{A,B,C\}$
- 3) L'unico vincolo di foreign key soddisfatto nella base di dati è  $R[A] \subseteq S[E]$

### Problema 6

Una istanza dello schema concettuale S in cui una istanza dell'entità F1 è anche istanza dell'entità F2 esiste, ad esempio la seguente istanza I:

$Istanze(I,F) = \{ f \}$

$Istanze(I,F1) = \{ f \}$

$Istanze(I,F2) = \{ f \}$

$Istanze(I,E) = \{ e1, e2 \}$

$Istanze(I,R1) = \{ \langle E:e1, F:f \rangle \}$

$Istanze(I,R2) = \{ \langle E:e1, F:f \rangle, \langle E:e2, F:f \rangle \}$

$Istanze(I,R) = \{ \langle E:e1, F:f \rangle, \langle E:e2, F:f \rangle \}$