

# Esame di Basi di Dati

## A.A. 2021/2022 – Appello del 30/05/2022

### Problema 1

Si richiede di progettare lo schema concettuale Entità-Relazione di un'applicazione relativa ai prestiti erogati da una banca a persone fisiche.

Di ogni prestito erogato interessa il codice (identificativo), l'ammontare, il tasso di interesse iniziale ed il titolare. Ogni prestito erogato è di uno tra i seguenti due tipi possibili: fido e mutuo. Di ogni fido interessa la data di erogazione, lo scopo per il quale è stato richiesto e le variazioni del tasso di interesse a cui è stato sottoposto, sapendo che tale tasso di interesse non può variare più di una volta al mese. Di ogni mutuo interessa l'anno di inizio e la durata. Alcuni mutui sono per la prima casa e di questi interessa l'eventuale sconto sul tasso di interesse.

Di ogni titolare interessa il codice fiscale, l'anno di nascita ed il sesso. Di ogni titolare che è un dipendente pubblico interessa anche il giorno ed il mese di nascita. Infine, si tenga conto del fatto che ad ogni titolare non possono essere erogati più di 2 mutui e ad ogni dipendente pubblico non può essere erogato più di 1 mutuo per la prima casa.

### Problema 2

Si richiede di effettuare la progettazione logica relativa al suddetto sistema informativo, producendo lo schema relazionale completo di vincoli, tenendo conto delle seguenti indicazioni: (i) ai dati relativi ai prestiti si accede sempre scegliendo di accedere o ai mutui o ai fidi, mai a entrambi; (ii) ogni volta che si accede ai dati di un mutuo si vuole sempre sapere se è un mutuo per prima casa o no ed in caso positivo si vuole sempre sapere l'eventuale sconto sul tasso di interesse.

### Problema 3

Si consideri la relazione `versamento(codice,cliente,mese,anno,ammontare)`, che memorizza i versamenti che un cliente fa su una banca, con codice identificativo del versamento (chiave primaria), cliente titolare del conto, mese e anno in cui è avvenuto il versamento ed ammontare in Euro del versamento stesso. Scrivere in SQL le seguenti query:

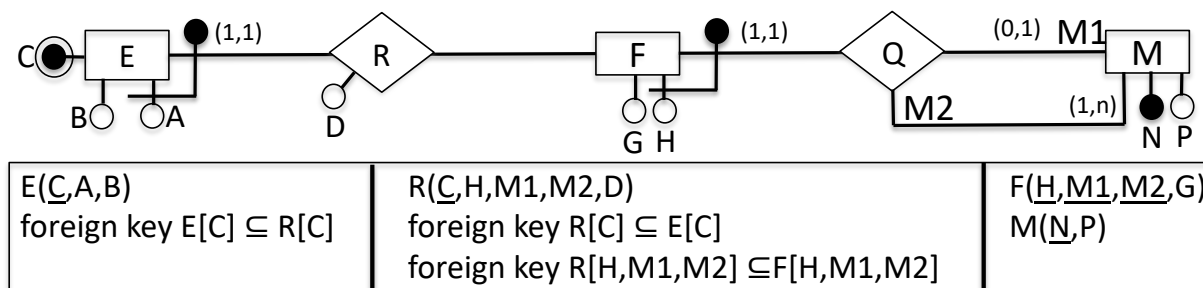
- Dato un mese di un anno (ad esempio, aprile 2020), mostrare per ogni cliente l'ammontare del conto in quel mese di quell'anno, che ovviamente dipenderà da tutti i versamenti fatti da quel cliente.
- Per ogni cliente e per ogni anno calcolare il mese (o i mesi) in cui è stato versato l'ammontare massimo di quell'anno.

### Problema 4

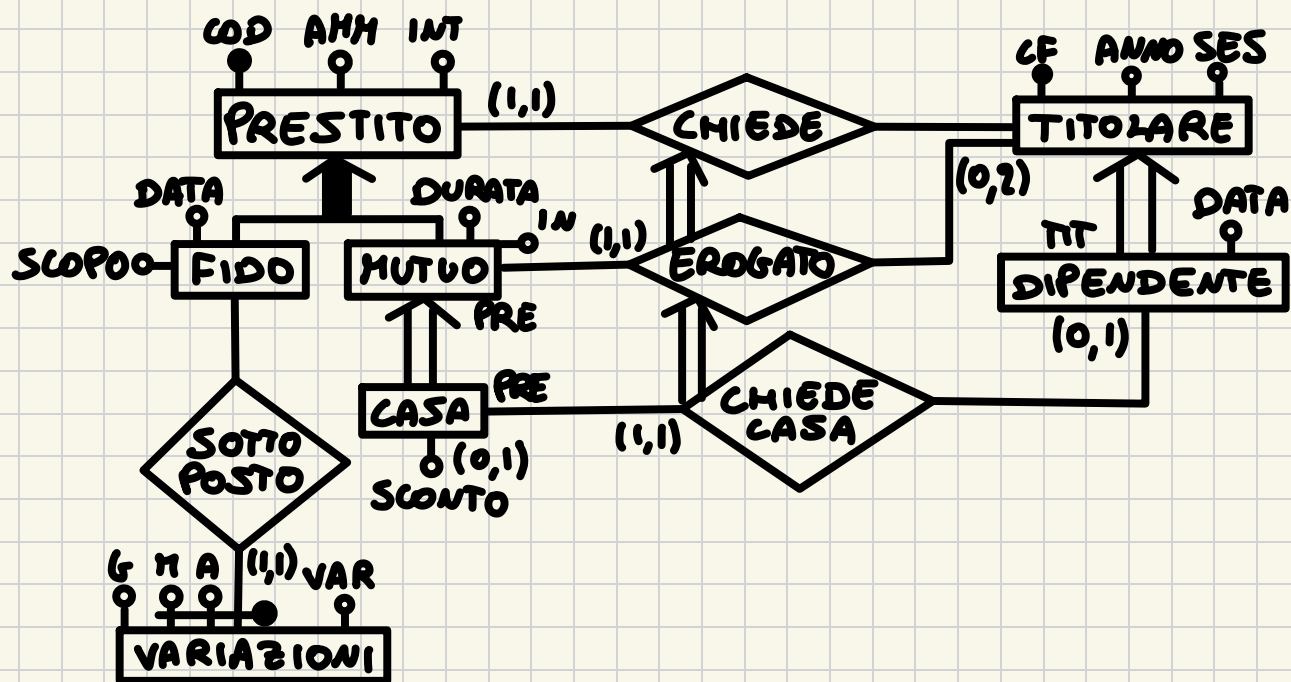
Sempre considerando la relazione `versamento(codice,cliente,mese,anno,ammontare)`, scrivere una query in algebra relazionale che per ogni cliente calcola i mesi per i quali il cliente ha effettuato almeno due versamenti con lo stesso ammontare (dove ovviamente un mese è identificato da numero del mese e anno)

### Problema 5

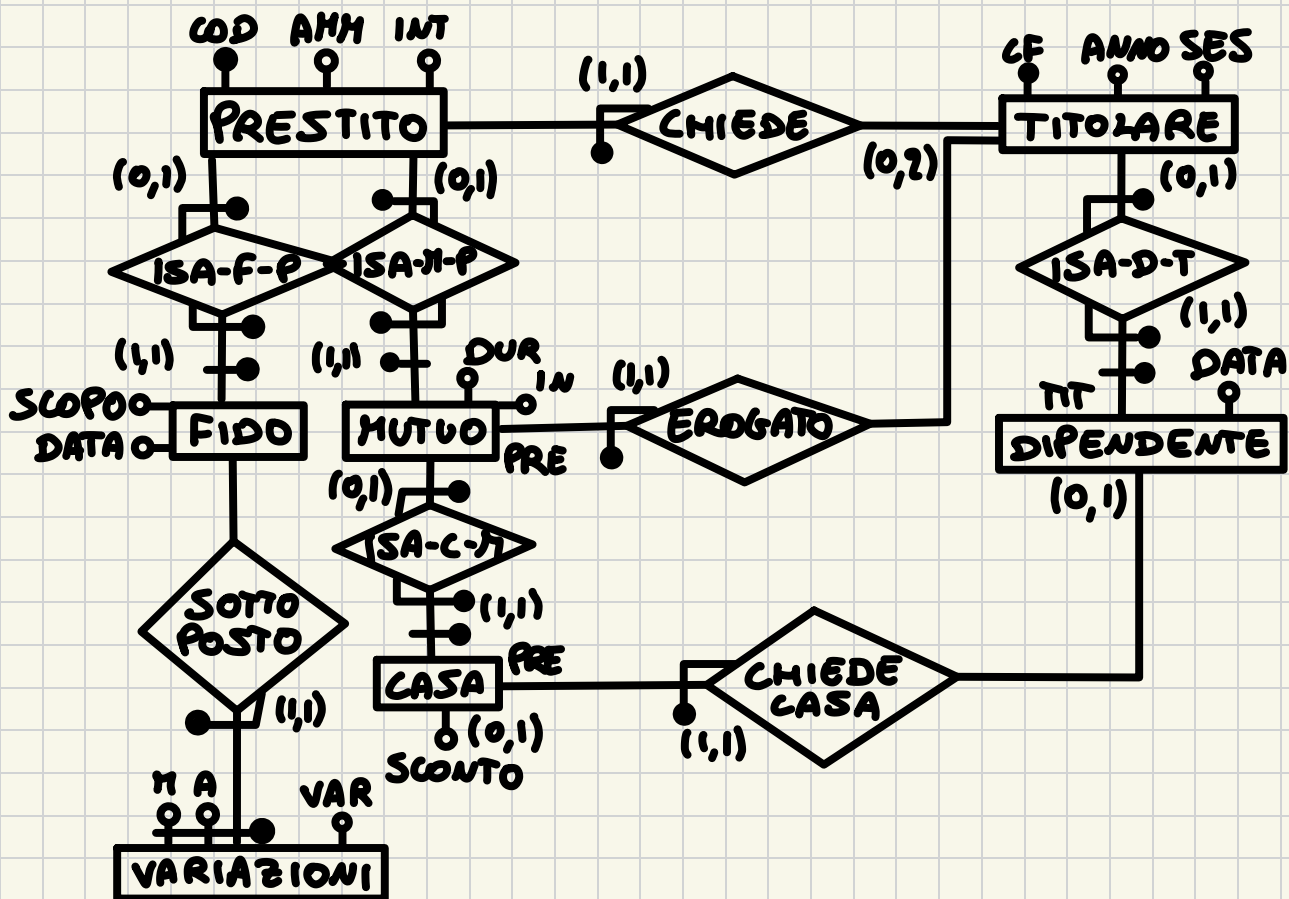
Considerare lo schema concettuale ristrutturato mostrato qui sotto ed il relativo schema relazionale, compreso di vincoli. Dire se lo schema relazionale è corretto. In caso di risposta affermativa, motivare tale risposta, ed in caso di risposta negativa dire come dovrebbe essere modificato lo schema relazionale per correggere gli errori.



## PROBLEMA 1



## PROBLEMA 2



VINCOLI ESTERNI:

$$(I, ISA-F-P) \cap (I, ISA-M-P) = \emptyset$$

PRESI  $m, p, x$ , SE  $\exists$  UNA ISTANZA DI CONCESSO  $\langle \text{TITOLARE}: x, \text{MUTUO}: m \rangle$  E ESISTE UNA ISTANZA DI  $ISA-M-P \langle \text{MUTUO}: m, \text{PRESTITO}: p \rangle$ , ALLORA  $\exists$  UNA ISTANZA DI CHIEDE CON  $\langle \text{PRESTITO}: p, \text{TITOLARE}: x \rangle$

PRESTITO (COD, ANH, INT)

FOREIGN KEY: P[COD]  $\subseteq$  CHIEDE[PRESTITO]

TITOLARE (CF, ANNO, SESSO)

CHIEDE (PRESTITO, TITOLARE)

FK: C[P]  $\subseteq$  P[COD]

FK: C[TIT]  $\subseteq$  TIT[CF]

FIDO (PREST, SCOPO, DATA)

FK: FIDO[P]  $\subseteq$  P[COD]

VINCOLO: FIDO[P]  $\cap$  MUTUO[P] =  $\emptyset$

MUTUO (PREST, DURATA, INIZIO)

FK: M[P]  $\subseteq$  P[COD]

FK: M[P]  $\subseteq$  ER[MUTUO]

VARIAZIONI (PRESTITO, MESE, ANNO, VAR)

FK: VAR[P]  $\subseteq$  P[COD]

PRIMA CASA (PRESTITO, SCONTO\*)

FK: PC[P]  $\subseteq$  P[COD]

FK: PC[P]  $\subseteq$  CC[PC]

DIPENDENTE (TIT, DATA)

FK: DIP[TIT]  $\subseteq$  TIT[CF]

EROGATO (MUTUO, TIT)

FK: ER[M]  $\subseteq$  M[P]

FK: ER[TIT]  $\subseteq$  TIT[CF]

FK: ER[M, TIT]  $\subseteq$  CHIEDE[P, TIT]

CHIEDE CASA (PRIMA CASA, DIP)

FK: CC[PC]  $\subseteq$  PC[P]

FK: CC[DIP]  $\subseteq$  DIP[TIT]

FK: CC[PC, DIP]  $\subseteq$  ER[M, TIT]

---

PRESTITO FIDO (COD, ANH, INT)

FK: PF[COD]  $\subseteq$  FIDO[COD]

PRESTITO MUTUO (COD, ANH, INT)

FK: PM[COD]  $\subseteq$  MUTUO[COD]

FK: PM[COD]  $\cup$  PF[COD]  $\subseteq$  CHIEDE[PRESTITO]

} PRESTITO

MUTUO ( PRESTITO/MUTUO, INIZIO, DURATA, SCONTO\*)

FK:  $\pi [PH] \subseteq PH [COD]$

FK: SELECT COD  
FROM MUTUO  
WHERE SCONTO  $\neq$  NULL  $\subseteq$  CHIEDE CASA

### PROBLEMA 3

1) SELECT CLIENTE, MESE, ANNO, SUM (AMMONTARE)  
FROM VERSAMENTO  
GROUP BY CLIENTE, MESE, ANNO

2) SELECT CLIENTE, ANNO, MESE  
FROM VERSAMENTO  
GROUP BY CLIENTE, ANNO, MESE  
HAVING (CLIENTE, ANNO, MAX (AMMONTARE)) IN (

SELECT CLIENTE, ANNO, MAX (AMMONTARE)  
FROM VERSAMENTO  
GROUP BY CLIENTE, ANNO  
)

### PROBLEMA 4

PROJ<sub>CLIENTE, MESE</sub> ( VERSAMENTO JOIN<sub>MESE=M  
AND  
ANNO=A  
AND  
CODICE $\neq$ L  
AND  
AMMONTARE=S  
AND  
CLIENTE=CL</sub> (REN<sub>m $\leftarrow$ MESE  
a $\leftarrow$ ANNO  
S $\leftarrow$ AMMONTARE  
C $\leftarrow$ CODICE  
CL $\leftarrow$ CLIENTE</sub> (VERSAMENTO)))