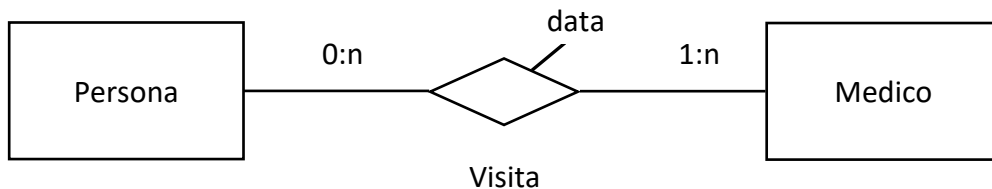


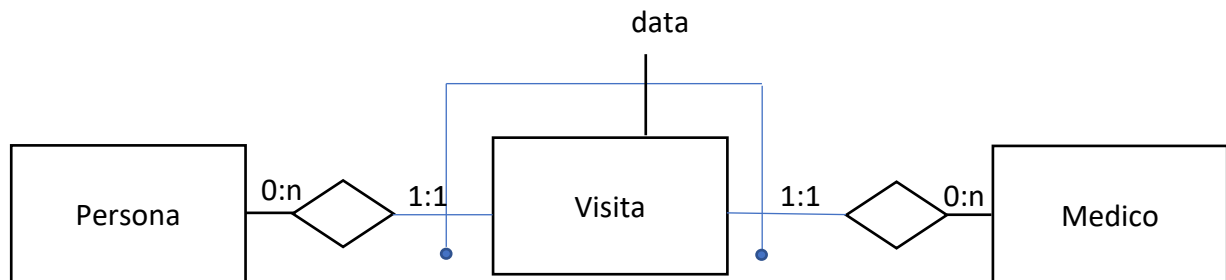
## Basi di Dati – Prova B - 3 luglio 2021

Tempo a disposizione: 1 ora

**Esercizio 1.** Si vuole modificare il seguente schema ER in maniera tale che sia possibile rappresentare più visite di una persona con lo stesso medico in date diverse



Soluzione



**Esercizio 2.** Dato il seguente schema relazionale

**PERSONA**(CF, NOME, ETÀ, SESSO)  
**CONIUGE**(CONIUGE1, CONIUGE2, ATTUALE)  
**AUTORE**(CF\*, NUMLIBRI)  
**LIBRO**(COD, TITOLO)  
**HALETTO**(PERS\*, LIBRO\*, ANNO)  
**HASCRITTO**(AUTORE\*, LIBRO\*, ANNO)

- scrivere una espressione in Algebra Relazionale per trovare i codici fiscali delle persone che NON hanno letto alcun libro nel 2020

$$\pi_{CF} Person - \pi_{CF}(\rho_{CF \leftarrow Pers}(\sigma_{anno=2020} HaLetto))$$

- formulare una interrogazione SQL per trovare i codici fiscali degli autori che hanno scritto almeno due libri in un anno

```

SELECT Autore
FROM HaScritto As X
WHERE Exists
    (SELECT *
     FROM HaScritto As Y
     WHERE X.autore=Y.autore and X.libro <> Y.libro and X.anno=Y.anno)
    
```

Esistono varie formulazioni alternative, quale, ad esempio, quella basata sull'utilizzo esplicito del prodotto cartesiano con l'uso di sinonimi

3. scrivere una asserzione che verifichi la seguente condizione: se una persona p1 ha *attualmente* coniuge p2, allora non esiste un'altra persona p3 che ha *attualmente* lo stesso coniuge p2 - cioè, se la tupla  $\langle p1, p2, \text{vero} \rangle \in \text{Coniuge}$  allora  $\langle p3, p2, \text{vero} \rangle \notin \text{Coniuge}$

```
CREATE ASSERTION ppp
CHECK NOT EXISTS
  (Select *
   From Coniuge as X
   Where EXISTS
     (Select *
      From Coniuge as Y
      Where X.Coniuge1 <> Y.Coniuge1 and X.coniuge2=Y.coniuge2 and
            X.attuale=vero and Y.attuale=vero))
```

**Esercizio 3.** Si consideri la seguente esecuzione concorrente: (1) dire se è serializzabile; (2) mostrare l'esecuzione risultante dalla applicazione del protocollo 2PL.

T1	T2
read(X)	
write(X)	
	read(Y)
	write(Y)
	read(X)
	write(X)
read(Y)	
write(Y)	
Commit	Commit

- 1) Non è c-serializzabile, come è facile vedere dal grafo dei conflitti.
- 2) Applicazione del 2PL

T1	T2
rlock(X)	
read(X)	
wlock(X)	
write(X)	
	rlock(Y)
	read(Y)
	wlock(Y)
	write(Y)
	rlock(X)
	wait
rlock(Y)	wait
wait	wait
wait (Y)	

L'esecuzione va in deadlock.