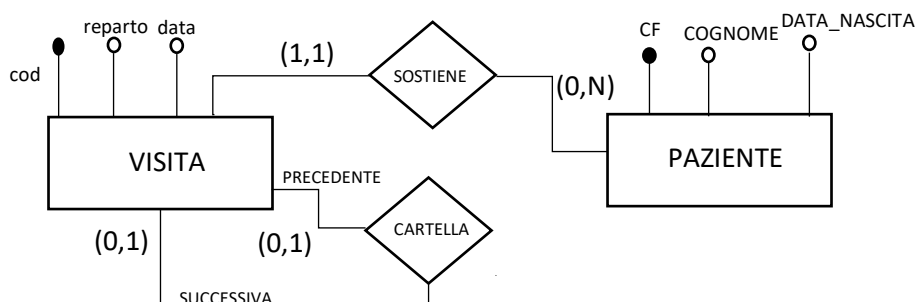


## Basi di Dati – Prova C - 3 luglio 2021

Tempo a disposizione: 1 ora

**ESERCIZIO 1.** Produrre uno schema relazionale equivalente al seguente schema ER.



- Visita(cod, reparto, ora, data, paz\*, successiva\*)
- Paziente(CF, cognome, dataNascita)

oppure

- Visita(cod, reparto, ora, data, paz\*)
- Paziente(CF, cognome, dataNascita)
- Cartella(prec\*, succ\*) con vincolo di unicità sia su prec sia su succ

**ESERCIZIO 2.** Sia data la seguente base di dati

- PERSONA(CF, NOME, ETÀ, SESSO)
- REGISTA(CF\*, NumFilm)
- FILM(COD, TITOLO, REGISTA\*)
- HAVISTO(PERS\*, FILM\*, ANNO)
- HADIRETTO(REGISTA\*, FILM\*, ANNO)

1. Scrivere una espressione in Algebra Relazionale per trovare i codici fiscali dei registi che non hanno diretto film nel 2020

$$\pi_{CF} Regista - \pi_{CF}(\rho_{CF \leftarrow regista}(\sigma_{anno=2020} HaDiretto))$$

2. Usare SQL per trovare i codici fiscali delle persone
  - a. che non hanno visto alcun film né nel 2020 né nel 2021
  - b. che hanno visto almeno due film nel 2020

Query a

```

SELECT CF
FROM Persona
WHERE CF NOT IN
  (SELECT pers
   FROM HaVisto
   WHERE Anno=20 OR Anno=21)
  
```

Query b

```
SELECT CF
FROM HaVisto as X
WHERE EXISTS
  (SELECT *
   FROM HaVisto as Y
   WHERE X.pers=Y.pers AND X.film <> Y.film AND X.anno=2020 AND Y.anno=2020)
```

Una possibile alternativa è data dalla seguente query SQL

```
SELECT pers
FROM HaVisto
WHERE anno=2020
GROUPBY pers
HAVING count(*)>1
```

3. Scrivere un trigger che aggiorna il numero di film diretti da un regista quando viene inserita una nuova tupla nella relazione HaDiretto

```
CREATE TRIGGER
AFTER Insert on HaDiretto
FOR EACH ROW
UPDATE Regista
SET NumFilm++
WHERE CF=NEW.Regista
```

**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(Z)	
write(Z)	
	read(X)
	write(X)
Commit	Commit

- 1) L'esecuzione è serializzabile  
2) Esecuzione sotto 2PL

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

	wlock(Y)
	write(Y)
rlock(Z)	
read(Z)	
wlock(Z)	
write(Z)	
unlock(X,Z)	
	rlock(X)
Commit	read(X)
	wlock(X)
	write(X)
	unlock(X,Y)
	Commit