

Appello svolto il 01/06/2021

Correzione del 07/06/2021

## Esercizi

1. Dato il seguente schema logico relazionale<sup>1</sup>

MEDICI(codice, cognome, nome)  
PAZIENTI(codice\_SSN, cognome, nome, data\_nascita)  
MEDICINALI(codice, nome, principio\_attivo, marca)  
VISITE(medico<sub>fk</sub>, paziente<sub>fk</sub>, data, diagnosi, medicinale<sub>fk</sub>)

rispondere alle seguenti domande (implementando le interrogazioni in linguaggio SQL).

- (a) Elencare i medici che sono omonimi (stesso cognome e stesso nome) di almeno un paziente.
- (b) Elencare i pazienti visitati almeno una volta e ai quali non sono mai stati prescritti medicinali.
- (c) Eliminare dal database le visite effettuate a pazienti nati prima dell'anno 1950.
- (d) Elencare, in ordine alfabetico, i medici che nel 2020 hanno prescritto medicinali di una ed una sola marca.
- (e) Sapendo che le relazioni MEDICI, PAZIENTI e VISITE hanno cardinalità  $m, p, v \in \mathbb{N}$ , indicare (motivando la risposta) le cardinalità minime e massime delle seguenti interrogazioni:

- i. `select p.cognome, p.nome, v.data`  
`from pazienti p, visite v where v.paziente = p.codice_SSN;`
- ii. `select m.cognome`  
`from medici m, pazienti p where m.nome = p.nome;`

(a) `SELECT DISTINCT m.*`  
`FROM medici m, pazienti p`  
`WHERE m.cognome = p.cognome`  
`AND m.nome = p.nome;`

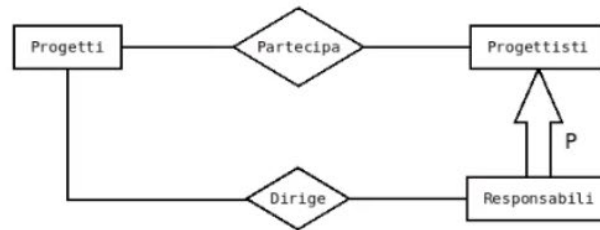
(b) `SELECT p.*`  
`FROM pazienti p`  
`WHERE p.codice IN (SELECT paziente FROM visite)`  
`AND p.codice NOT IN (SELECT paziente FROM visite`  
`WHERE medicinale IS NOT NULL)`

(c) `DELETE FROM visite`  
`WHERE paziente IN (SELECT codice FROM pazienti`  
`WHERE data_nascita < '1950-01-01')`

(d) `SELECT m.* FROM medici m, visite v, medicinali md`  
`WHERE m.codice = v.medico`  
`AND v.medicinale = md.codice`  
`AND v.data BETWEEN '2020-01-01' AND '2020-12-31'`  
`AND v.data BETWEEN >= '2020-01-01' AND v.data <= '2020-12-31'`  
`AND EXTRACT (year FROM v.data)=2020`  
`GROUP BY m.codice, m.cognome, m.nome`  
`HAVING COUNT (DISTINCT md.marca)=1`  
`ORDER BY m.cognome, m.nome`

(e) (i) `min = v` , `max = v`

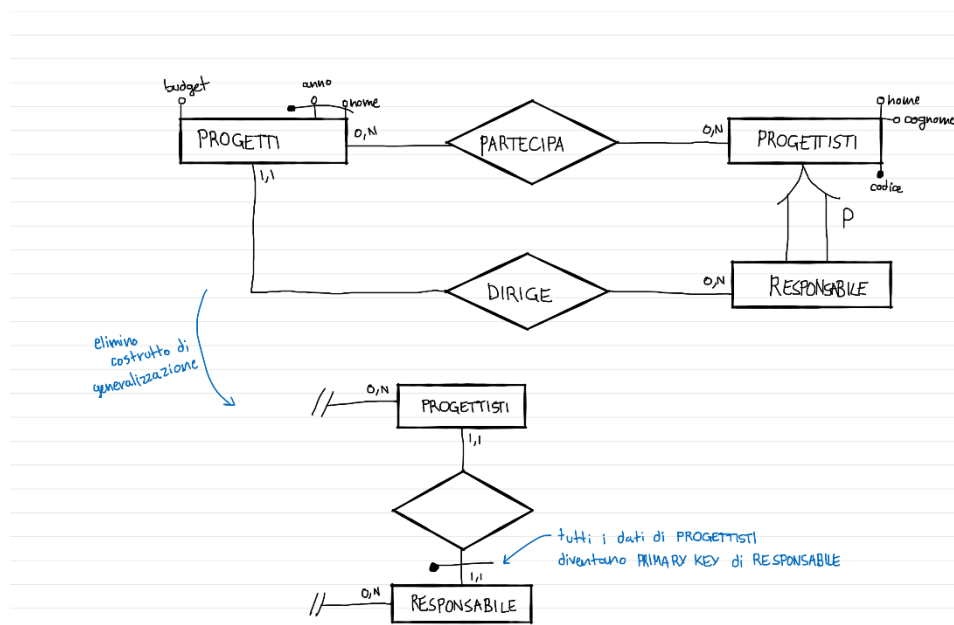
(ii) `min = 0` , `max = m*p`



2. Completare la bozza di schema ER in figura (indicando gli attributi, gli identificatori e le cardinalità mancanti) sapendo che:

- i progetti sono identificati da anno di inizio e nome (più progetti iniziati in anni diversi possono avere lo stesso nome); per essi si tiene traccia del budget e sono diretti da un unico responsabile di progetto;
- per i progettisti, identificati da un codice, si conoscono nome e cognome; possono partecipare a più progetti e, se responsabili di progetto, dirigere più progetti;
- ad ogni progetto può partecipare un numero qualunque di progettisti.

Ristrutturare lo schema, eliminando il costrutto di generalizzazione in modo da mantenere distinte le entità progettisti e responsabili di progetto. Infine, tradurre lo schema ristrutturato in uno schema logico relazionale, codificando opportunamente i vincoli di integrità.



```

progettisti(codice, cognome, nome)
resp_proj(progettistifk)
progetti(anno, nome, budget, resp_projfk)
partecipa([anno, nome]fk, progettistafk) -> in una associazione molti a molti va fatta
    
```

3. Rispondere brevemente alle seguenti domande.

- (a) Definire il concetto di transazione. Cosa si intende per atomicità delle transazioni?
  - (b) Definire il concetto di conflitto tra due istruzioni in uno schedule. Uno schedule seriale può avere conflitti?
  - (c) Spiegare la differenza tra fallimento di sistema e disastro. Spiegare l'uso dei record di checkpoint nei file di log nella fase di ripristino successiva a: (a) un fallimento di sistema; (b) un disastro.
- (a) Sequenza di read (r) e write (w), operazioni DBMS, terminate da commit o abort(con roll-back).  
Le sequenze sono un tutt'uno (una o niente, atomo), se fallisce 1, allora tutto fallisce.
- (b) Un conflitto crea arco,  $t1 \rightarrow t2 \rightarrow t3 \rightarrow t1$  crea un grafo ciclico, conflitto.  
Stabiliamo se serializzabile. Uno schedule seriale può avere conflitti.
- (c) Fallimento per singola istruzione;  
fallimento di sistema, perdita della memoria volatile, restart DBMS;  
disastro, perdo contenuto nel disco (danno fisico).  
Record di checkpoint, transazione già commit sono su disco, non ci riguardano a patto sia salvo il disco (quindi no disastro, dove altrimenti non conterrebbero e servirebbe dump).  
Per il fallimento, rifare transazioni solo da checkpoint i poi (quelle dunque fallite).