

# Compito di Basi di dati

22 luglio 2020

## Esercizio 1:

Sia dato il seguente schema relazionale relativo a reparti e medici:

*REPARTO*(Nome, Edificio, Piano, Primario);

*MEDICO*(MedicoId, Nome, Cognome, Specializzazione, Genere, AnnoNascita, Reparto, CittàResidenza);

*SI.TROVA.IN*(Città, Regione).

Si assuma che ogni reparto sia identificato univocamente dal suo nome e sia caratterizzato dalla sua collocazione (edificio e piano) e dal capo reparto (primario). Si assuma, inoltre, che un medico possa essere il primario di al più un reparto (quello al quale afferisce). Si assuma, infine, che ogni piano disponibile di ogni edificio sia assegnato ad un unico reparto.

Si assuma che ogni medico sia identificato da un codice, che lo individua univocamente fra tutti i medici dell'ospedale, e sia caratterizzato da un nome, un cognome, una specializzazione (per semplicità, assumiamo di registrare una e una sola specializzazione per ogni medico), un genere (maschio o femmina), un anno di nascita, un reparto di appartenenza (ogni medico sia assegnato ad uno e un solo reparto) e una città di residenza.

Si assuma, infine, che la tabella *SI.TROVA.IN* contenga tutte e sole le città italiane già capoluogo di provincia. Città e regioni siano identificate univocamente dal loro nome.

Definire preliminarmente le chiavi primarie, le eventuali altre chiavi candidate e, se ve ne sono, le chiavi esterne delle relazioni date. Successivamente, formulare opportune interrogazioni in SQL che permettano di determinare quanto richiesto (senza usare l'operatore *CONTAINS* e usando solo se e quando necessario le funzioni aggregate):

- i reparti tali che (i) tutti i medici afferenti risiedano in città della regione *Piemonte* e (ii) almeno uno di loro risieda nella città *Torino*;
- i reparti con lo stesso numero di medici del reparto di terapia intensiva.

## Esercizio 2:

Con riferimento all'Esercizio 1, formulare un'interrogazione in algebra relazionale che permetta di determinare quanto richiesto (senza usare l'operatore di divisione e usando solo se necessario le funzioni aggregate):

- i reparti i cui medici risiedono solo in città del Veneto, ma non in tutte.

## Esercizio 3:

Si vuole realizzare una base di dati per la gestione di informazioni relative a un corso di studio di una università sulla base del seguente insieme di requisiti.

- Ogni insegnamento sia identificato univocamente da un codice numerico e sia caratterizzato da un nome, un insieme di prerequisiti (insieme di altri insegnamenti che devono essere già stati superati per poter sostenere l'esame dell'insegnamento) e il professore che tiene le lezioni (ogni insegnamento sia tenuto da un solo professore).
- Ogni studente sia identificato univocamente da un numero di matricola e sia caratterizzato da un nome, un cognome, un indirizzo di posta elettronica, un recapito postale e una data di nascita. Ogni studente abbia uno o più esami inseriti nel suo piano degli studi e abbia superato un certo numero di essi (non si escluda l'eventualità che non ne abbia ancora superato alcuno). Per ogni insegnamento del quale lo studente ha già superato l'esame, si registrino la data in cui ha sostenuto con successo l'esame e il voto ottenuto.

- I professori siano suddivisi in professori ordinari e professori associati. Ogni professore sia identificato univocamente dal suo codice fiscale e sia caratterizzato da un nome, un cognome, un numero di cellulare, il dipartimento cui afferisce e gli insegnamenti che tiene (uno o più). Ogni dipartimento sia identificato univocamente da un nome e sia caratterizzato da un recapito postale. Di ogni professore ordinario vogliamo registrare la data in cui ha preso servizio in tale veste (il primo giorno nel ruolo di professore ordinario).

Si definisca uno schema Entità-Relazioni che descriva il contenuto informativo del sistema, illustrando con chiarezza le eventuali assunzioni fatte. Lo schema dovrà essere completato con attributi ragionevoli per ciascuna entità (identificando le possibili chiavi) e relazione. Vanno specificati accuratamente i vincoli di cardinalità e partecipazione di ciascuna relazione. Si definiscano anche eventuali regole di gestione (regole di derivazione e vincoli di integrità) necessarie per codificare alcuni dei requisiti attesi del sistema.

#### Esercizio 4:

Definire la nozione di schedule *seriale* e di schedule *serializzabile* di un insieme di transazioni concorrenti. Esistono schedule seriali non serializzabili? In caso affermativo, fornire un esempio; in caso negativo, spiegare il motivo.

Si considerino ora le seguenti transazioni:

$T_0$ :  $r_0(x) \ w_0(x) \ r_0(y) \ w_0(y)$ ;

$T_1$ :  $r_1(x) \ w_1(x)$ .

Definire la nozione di serializzabilità *rispetto alle viste*. Successivamente, indicare (spiegandone il motivo) quali fra i seguenti schedule sono serializzabili rispetto alle viste:

$S_0$ :  $r_0(x) \ r_1(x) \ w_1(x) \ w_0(x) \ r_0(y) \ w_0(y)$ ;

$S_1$ :  $r_0(x) \ w_0(x) \ r_0(y) \ r_1(x) \ w_1(x) \ w_0(y)$ ;

$S_2$ :  $r_0(x) \ w_0(x) \ r_0(y) \ w_0(y) \ r_1(x) \ w_1(x)$ .