

## Esercizio 1:

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

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

*REPARTO*(Nome, Edificio, Piano, <sup>UNIQUE</sup>Primario);

*SI\_TROVA\_IN*(Città, Regione).

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 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 che un medico possa essere il primario di al più un reparto (quello al quale afferisce). Non si escluda la possibilità che due diversi reparti siano collocati nello stesso piano dello stesso edificio.

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 in cui sono presenti sia medici di sesso femminile che medici di sesso maschile, tutti nati dopo il 1960 (al più 59 anni);
- (FACOLTATIVO) il reparto (i reparti se più di uno) col numero più alto di medici di sesso femminile.

a. ~~SELECT NOME  
FROM REPARTO R  
WHERE EXIST ( SELECT \*  
FROM MEDICO  
WHERE GENERE = FEMMINA AND  
ANNONASCITA > 1960 AND  
REPARTO = R.NOME  
)  
AND EXIST ( SELECT \*  
FROM MEDICO  
WHERE GENERE = MASCHIO AND  
ANNONASCITA > 1960 AND  
REPARTO = R.NOME  
)~~

con questa hai anche  
oltre i 59 anni

a. SELECT NOME  
FROM REPARTO R  
WHERE EXIST ( SELECT \*  
FROM MEDICO M1, M2  
WHERE M1.GENERE = MASCHIO  
AND M2.GENERE = FEMMINA  
AND M1.REPARTO = R.NOME  
AND M2.REPARTO = R.NOME  
)  
AND 1960 < ALL ( SELECT ANNONASCITA  
FROM MEDICO  
WHERE REPARTO = R.NOME  
)

b. SELECT REPARTO, COUNT (\*) AS NUM\_FEM  
FROM MEDICO  
WHERE GENERE = FEMMINA  
GROUP BY REPARTO  
HAVING NUM\_FEM ≥ ( SELECT COUNT (\*)  
FROM MEDICO  
WHERE GENERE = FEMMINA  
GROUP BY REPARTO  
)

## 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):

(a) i reparti i cui medici risiedono in tutte e sole le città del Veneto.

tutte le città del Veneto

$CITTA\_VENETO \leftarrow \Pi_{CITTA} (\sigma_{REGIONE = VENETO} (SI\_TROVA\_IN))$

ma se ne elimino  
tutti i reparti

$REQUISITI (REPARTO, MEDICOID, CITTA\_RESIDENZA) \leftarrow \Pi_{REPARTO, MEDICOID} (MEDICO) \times CITTA\_VENETO$

$STATO\_DI\_FATTO \leftarrow \Pi_{REPARTO, MEDICOID, CITTA\_RESIDENZA} (MEDICO)$

$NO\_GOOD (NOME, MEDICO, CITTA) \leftarrow REQUISITI - STATO\_DI\_FATTO$

$TUTTO\_VENETO \leftarrow \Pi_{NOME} (REPARTO) - \Pi_{NOME} (NO\_GOOD)$

$NON\_VENETO (NOME) \leftarrow \Pi_{REPARTO} (\sigma_{REGIONE \neq VENETO} (MEDICO \bowtie SI\_TROVA\_IN))$   
CITTA = CITTA

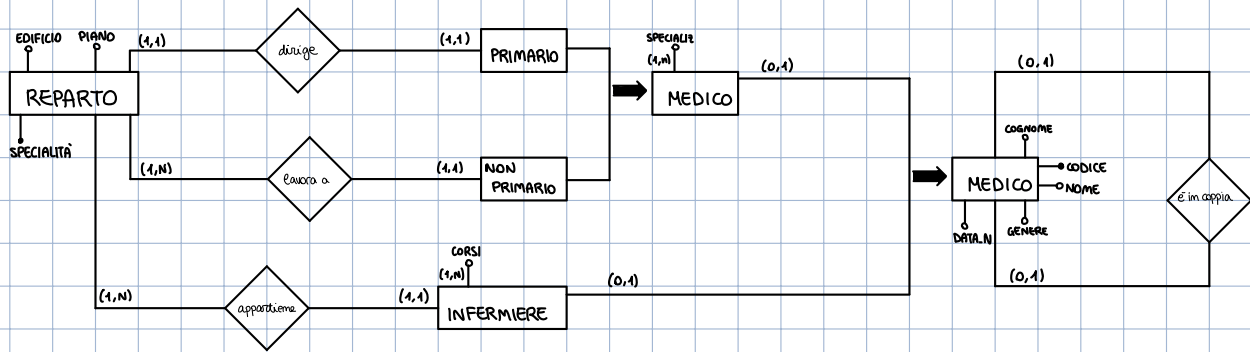
$S \leftarrow TUTTO\_VENETO - NON\_VENETO$

## Esercizio 3:

Si vuole realizzare una base di dati per la gestione di informazioni relative a reparti, medici e infermieri di un dato ospedale sulla base del seguente insieme di requisiti.

- L'ospedale sia organizzato in un certo numero di reparti. Ogni reparto sia caratterizzato da una specialità medica, che lo identifica univocamente (assumiamo che vi sia al più un reparto per ogni specialità medica), una collocazione (edificio e piano), un insieme di medici, un primario (scelto fra i medici del reparto) e un insieme di infermieri. Non escludiamo la possibilità che due reparti diversi siano collocati nello stesso piano dello stesso edificio.
- Ad ogni medico sia assegnato un codice, che lo identifica univocamente fra tutti i medici dell'ospedale. Ogni medico sia caratterizzato da un nome, un cognome, una o più specializzazioni, un genere (maschio o femmina), una data di nascita e il reparto cui appartiene (si assuma che ogni medico sia assegnato ad uno e un solo reparto).
- A ogni infermiere sia assegnato un codice, che lo identifica univocamente fra tutti gli infermieri dell'ospedale. Ogni infermiere sia caratterizzato da un nome, un cognome, un genere (maschio o femmina), una data di nascita, l'insieme dei corsi di formazione ai quali ha partecipato e il reparto cui appartiene (si assuma che ogni infermiere sia assegnato ad uno e un solo reparto).
- Si tenga traccia dei legami coniugali esistenti fra i membri del personale ospedaliero (medici e infermieri), comprendenti le coppie medico/medico, infermiere/infermiere e medico/infermiere.

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:

Si consideri la seguente istanza di base di dati contenente la tabella **MontagneVisitate**, con chiave primaria la *coppia* (nome, data).

MontagneVisitate	
nome	data
Celva	20/06/2020
Bondone	03/05/2020
Coglians	03/05/2020

Si consideri la seguente coppia di transazioni:

T1	T2
start transaction; insert into MontagneVisitate values('Matajur', '12/12/2019');  commit	start transaction; insert into MontagneVisitate values('Matajur', '12/12/2019');  commit;

Indicare quali sono i 4 livelli di isolamento previsti dallo standard SQL. Per ognuno di essi, indicare qual è l'esito della transazione **T2** qualora, al posto dei puntini, venga inserita l'istruzione **commit** e qualora, invece, venga inserita l'istruzione **rollback**.

i 4 livelli previsti sono: READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE

Se viene inserito COMMIT:

RV:	T2 si blocca perché il lock è in 2PL stretto su WRITE, e se T1 scrive allora T2 non riesce ad inserire → T2 fa rollback
RC:	
RR:	
SR:	

Se viene scritto ABORT:

RV:	T2 si blocca perché il lock è in 2PL stretto su WRITE, e se T1 fa ROLLBACK allora T2 scrive e inserisce → T2 fa commit
RC:	
RR:	
SR:	