

SOLUZIONI

Si consideri la realtà medica descritta dalla base di dati relazionale definita dal seguente schema:

PAZIENTE(CodFiscale, Cognome, Nome, Sesso, DataNascita, Citta, Reddito)
 MEDICO(Matricola, Cognome, Nome, Specializzazione, Parcella, Citta)
 FARMACO(NomeCommerciale, PrincipioAttivo, Costo, Pezzi)
 PATOLOGIA(Nome, ParteCorpo, SettoreMedico, Invalidita, PercEsenzione)
 INDICAZIONE(Farmaco, Patologia, DoseGiornaliera, NumGiorni, AVita)
 VISITA(Medico, Paziente, Data, Mutuata)
 ESORDIO(Paziente, Patologia, DataEsordio, DataGuarigione, Gravita, Cronica)
 TERAPIA(Paziente, Patologia, DataEsordio, Farmaco, DataInizioTerapia, DataFineTerapia, Posologia)

Risolvere i seguenti esercizi utilizzando la sintassi MySQL. La correttezza dei primi due esercizi è una condizione necessaria per la correzione dell'intero elaborato. Quelle che seguono sono possibili soluzioni degli esercizi proposti. Soluzioni alternative sono corrette purché producano lo stesso risultato e siano semanticamente equivalenti a quelle proposte.

Esercizio 1

Scrivere una query che restituisca nome e cognome del medico che, al 31/12/2014, aveva visitato un numero di pazienti superiore a quelli visitati da ciascun medico della sua stessa specializzazione.

```
CREATE OR REPLACE VIEW PazientiVisitati AS
SELECT V.Medico, M.Specializzazione,
       COUNT(DISTINCT V.Paziente) AS NumeroPazienti
FROM Visita V INNER JOIN Medico M ON V.Medico = M.Matricola
WHERE V.Data < '2014-12-31'
GROUP BY V.Medico;

SELECT M.Nome, M.Cognome
FROM Medico M INNER JOIN PazientiVisitati PV
ON M.Matricola = PV.Medico
WHERE PV.NumeroPazienti >= ALL
      (SELECT PV2.NumeroPazienti
FROM PazientiVisitati PV2
WHERE PV2.Specializzazione = PV.Specializzazione
);
```

Esercizio 2

Scrivere una query che restituisca per ciascun principio attivo, il nome del principio attivo e il nome commerciale di ogni farmaco utilizzato almeno una volta per tutte le patologie per le quali è indicato. Il risultato è formato da row (PrincipioAttivo, NomeCommerciale), una per ogni farmaco che rispetta la condizione.

```
SELECT F.PrincipioAttivo, T.Farmaco
FROM Terapia T INNER JOIN Farmaco F ON T.Farmaco = F.NomeCommerciale
GROUP BY T.Farmaco, F.PrincipioAttivo
HAVING COUNT(DISTINCT T.Patologia) =
```

```

        (SELECT COUNT(*)
         FROM Indicazione I
         WHERE I.Farmaco = T.Farmaco
        )
ORDER BY F.PrincipioAttivo;

```

/* ORDER BY non è necessario, ma aiuta a interpretare meglio il risultato richiesto a livello visivo */

Esercizio 3

Scrivere un trigger che impedisca l'inserimento di due terapie consecutive per lo stesso paziente, caratterizzate dallo stesso farmaco, con una posologia superiore al doppio rispetto alla precedente.

```

DROP TRIGGER IF EXISTS VietaTerapieConsecutive;

DELIMITER $$

CREATE TRIGGER VietaTerapieConsecutive
BEFORE INSERT ON Terapia
FOR EACH ROW
BEGIN
    SET @terapiaPrecedente =
        (SELECT COUNT(*)
         FROM Terapia T1
         WHERE T1.Paziente = NEW.Paziente
              AND T1.Farmaco = NEW.Farmaco
              AND T1.Posologia = 0.5*NEW.Posologia
              AND NOT EXISTS
                (SELECT *
                 FROM Terapia T2
                 WHERE T2.Paziente = T1.Paziente
                      AND T2.DataFineTerapia > T1.DataFineTerapia
                      AND T2.Farmaco = T1.Farmaco
                )
        );
    IF @terapiaPrecedente = 1 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Terapia non consentita';
    END IF;
END $$

DELIMITER ;

```

Esercizio 4

Al termine di Febbraio 2015, come ogni anno, le parcelle dei medici della clinica saranno aggiornate. La percentuale di aumento della parcella di un medico è pari alla percentuale di terapie prescritte dal medico nel 2014 che hanno condotto il paziente alla guarigione, rispetto a tutte le terapie da egli/ella prescritte nello stesso anno. Assumere che il medico che prescrive una terapia a un paziente sia il medico, la cui specializzazione è uguale al settore medico della patologia oggetto della terapia, dal quale il paziente è stato visitato da meno tempo prima dell'inizio della terapia stessa.

Scrivere una stored procedure aggiorna_parcelle che prenda come argomento un anno (in questo caso il 2014) e aggiorni, come descritto, la parcella di tutti i medici.

```

DROP PROCEDURE IF EXISTS aggiorna_parcelle;
DELIMITER $$
CREATE PROCEDURE aggiorna_parcelle(IN anno INTEGER)
BEGIN

DROP TEMPORARY TABLE IF EXISTS Successi;
CREATE TEMPORARY TABLE Successi(
    Matricola CHAR(100) NOT NULL,
    TotaleSuccessi INT(11) NOT NULL,
    TotalePrescrizioni INT(11) NOT NULL,
    PRIMARY KEY (Matricola)
);

INSERT INTO Successi
SELECT D.Matricola, SUM(D.Successo), COUNT(*)
FROM(
SELECT M.Matricola, IF(YEAR(T.DataFineTerapia) = anno, 1, 0) AS Successo
FROM Terapia T INNER JOIN Visita V USING(Paziente)
    INNER JOIN Patologia PA ON T.Patologia = PA.Nome
    INNER JOIN Medico M ON PA.SettoreMedico = M.Specializzazione
WHERE YEAR(T.DataInizioTerapia) = anno
    AND NOT EXISTS
        (SELECT *
         FROM Visita V2 INNER JOIN Medico M2 ON V2.Medico = M2.Matricola
         WHERE V2.Paziente = T.Paziente
            AND M2.Specializzazione = PA.SettoreMedico
            AND V2.Data BETWEEN V.Data AND T.DataInizioTerapia
        )
) AS D
GROUP BY D.Matricola;

UPDATE Medico M NATURAL JOIN Successi S
SET M.Parcella = M.Parcella + (S.Successi/S.Prescrizioni);

DROP TEMPORARY TABLE Successi;

END $$
DELIMITER ;

```

A.A. precedente:

Al termine di Febbraio 2015, come ogni anno, le parcelle dei medici della clinica saranno aggiornate. La percentuale di aumento della parcella di un medico è pari alla percentuale di terapie prescritte dal medico nel 2014 che hanno condotto il paziente alla guarigione, rispetto a tutte le terapie da egli/ella prescritte nello stesso anno. Assumere che il medico che prescrive una terapia a un paziente sia il medico, la cui specializzazione è uguale al settore medico della patologia oggetto della terapia, dal quale il paziente è stato visitato da meno tempo prima dell'inizio della terapia stessa. Scrivere una query che aggiorni, come descritto, la parcella di tutti i medici.

```

CREATE OR REPLACE VIEW Successi AS
SELECT D.Matricola, SUM(D.Successo), COUNT(*)
FROM(
SELECT M.Matricola, IF(YEAR(T.DataFineTerapia) = 2014, 1, 0) AS Successo
FROM Terapia T INNER JOIN Visita V USING(Paziente)
    INNER JOIN Patologia PA ON T.Patologia = PA.Nome
    INNER JOIN Medico M ON PA.SettoreMedico = M.Specializzazione
WHERE YEAR(T.DataInizioTerapia) = 2014

```

```
AND NOT EXISTS
  (SELECT *
   FROM Visita V2 INNER JOIN Medico M2 ON V2.Medico = M2.Matricola
   WHERE V2.Paziente = T.Paziente
        AND M2.Specializzazione = PA.SettoreMedico
        AND V2.Data BETWEEN V.Data AND T.DataInizioTerapia
   )
) AS D
GROUP BY D.Matricola;

UPDATE Medico M NATURAL JOIN Successi S
SET M.Parcella = M.Parcella + (S.Successi/S.Prescrizioni);
```