

## 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, considerate le sole terapie finalizzate alla cura di patologie cardiache, restituisca, per ciascuna di esse, il nome della patologia e il farmaco più utilizzato per curarla. La soluzione proposta deve presupporre che, data una patologia cardiaca, tale farmaco possa non essere unico.

```
CREATE OR REPLACE VIEW TotaliTerapie AS
SELECT T.Patologia, T.Farmaco, COUNT(*) AS NumeroTerapie
FROM Terapia T INNER JOIN Patologia P ON T.Patologia = P.Nome
WHERE P.ParteCorpo = 'Cuore'
GROUP BY T.Patologia, T.Farmaco;

SELECT TT.Patologia, TT.Farmaco
FROM TotaliTerapie TT
WHERE TT.NumeroTerapie >= ALL (SELECT TT2.NumeroTerapie
                              FROM TotaliTerapie TT2
                              WHERE TT2.Patologia = TT.Patologia);
```

### Esercizio 2 [stessa soluzione per l'esercizio 3, anno accademico precedente]

Scrivere una query che restituisca nome, cognome e reddito dei pazienti di sesso femminile che al 15 Giugno 2010 risultavano affetti, oltre alle eventuali altre, da un'unica patologia cronica, con invalidità superiore al 50%, e non l'avevano mai curata con alcun farmaco fino a quel momento.

```
SELECT P.Nome, P.Cognome, P.Reddito
FROM Esordio E INNER JOIN Paziente P ON E.Paziente = P.CodFiscale
INNER JOIN Patologia PA ON E.Patologia = PA.Nome
WHERE E.DataEsordio < '2010-06-15'
AND PA.Invalidita > 50
AND E.DataGuarigione IS NULL -- ridondante, la malattia è cronica
AND E.Cronica = 'si'
```

```

        AND NOT EXISTS (SELECT *
                        FROM Terapia T
                        WHERE T.Paziente = E.Paziente
                           AND T.Patologia = E.Patologia
                           AND T.DataInizioTerapia < '2010-06-15'
                        )
GROUP BY E.Paziente, P.Nome, P.Cognome, P.Reddito
HAVING COUNT(*) = 1;

```

*A.A. precedente:*

Scrivere una query che restituisca, per tutte le patologie, il nome della patologia e il numero di pazienti di età superiore a quarant'anni che l'hanno contratta almeno due volte, sempre in forma non cronica.

```

SELECT D.Patologia, COUNT(DISTINCT D.Paziente) AS NumeroPazienti
FROM(
    SELECT E.Patologia, E.Paziente
    FROM Esordio E INNER JOIN Paziente P ON E.Paziente = P.CodFiscale
    WHERE P.DataNascita + INTERVAL 40 YEAR < CURRENT_DATE
        AND NOT EXISTS (SELECT *
                        FROM Esordio E2
                        WHERE E2.Paziente = E.Paziente
                           AND E2.Patologia = E.Patologia
                           AND E2.Cronica <> 'no')
    GROUP BY E.Patologia, E.Paziente
    HAVING COUNT(*) >= 2
) AS D
GROUP BY D.Patologia;

```

### Esercizio 3

Scrivere una query che restituisca, per tutte le patologie, nessuna esclusa, il nome della patologia e il numero di pazienti di età superiore a quarant'anni che l'hanno contratta almeno due volte, la seconda delle quali con gravità superiore alla prima, comunque sempre in forma non cronica.

```

CREATE OR REPLACE VIEW PrimoEsordio AS
SELECT E.Patologia, E.Paziente, MIN(E.DataEsordio) AS DataEsordio
FROM Esordio E INNER JOIN Paziente P ON E.Paziente = P.CodFiscale
WHERE P.DataNascita + INTERVAL 40 YEAR < CURRENT_DATE
    AND NOT EXISTS (SELECT *
                    FROM Esordio E2
                    WHERE E2.Paziente = E.Paziente
                       AND E2.Patologia = E.Patologia
                       AND E2.Cronica <> 'no')
GROUP BY E.Patologia, E.Paziente
HAVING COUNT(*) >= 2;

CREATE OR REPLACE VIEW PrimoEsordioGravita AS
SELECT PE.*, E.Gravita
FROM Esordio E NATURAL JOIN PrimoEsordio PE;

SELECT P.Nome, IF(D.Patologia IS NULL, 0, D.NumeroPazienti) AS TotPazienti
FROM Patologia P LEFT OUTER JOIN (
    SELECT E.Patologia, COUNT(DISTINCT E.Paziente) AS NumeroPazienti
    FROM PrimoEsordioGravita PEG INNER JOIN Esordio E

```

```

        ON PEG.Paziente = E.Paziente
        AND PEG.Patologia = E.Patologia
        AND PEG.DataEsordio < E.DataEsordio
WHERE DATEDIFF(E.DataEsordio, PEG.DataEsordio) =
    ( SELECT MIN(DATEDIFF(E2.DataEsordio, PEG2.DataEsordio))
      FROM PrimoEsordioGravita PEG2 INNER JOIN Esordio E2
        ON PEG2.Paziente = E2.Paziente
        AND PEG2.Patologia = E2.Patologia
        AND PEG2.DataEsordio < E2.DataEsordio
      WHERE E2.Paziente = E.Paziente
        AND E2.Patologia = E.Patologia
    )
    AND PEG.Gravita < E.Gravita
GROUP BY E.Patologia
) AS D
ON P.Nome = D.Patologia;

```

#### Esercizio 4

Scrivere una stored procedure che, ricevuto in ingresso il codice fiscale di un paziente e il nome di un principio attivo, blocchi immediatamente tutte le terapie attualmente in corso, impostando la data di fine terapia alla data corrente, qualora si stiano protrando per oltre una settimana, e il paziente abbia già effettuato in precedenza, comunque non oltre sei mesi prima, almeno tre terapie con lo stesso farmaco o con un farmaco contenente lo stesso principio attivo, di cui almeno una con posologia superiore a tre compresse al giorno. Al termine delle elaborazioni, la procedura deve restituire, nonché mostrare a video, un resoconto contenente le seguenti informazioni sulle terapie bloccate: codice fiscale del paziente, farmaco, durata della terapia interrotta, posologia, numero di terapie precedenti con posologia superiore a tre compresse al giorno.

```

DROP PROCEDURE IF EXISTS blocca_terapie;

DELIMITER $$

CREATE PROCEDURE blocca_terapie(IN codiceFiscale CHAR(16),
                                IN principioAttivo CHAR(100))
BEGIN

    -- variabili per il prelievo di un record da Terapia, e per il resoconto
    DECLARE finito INTEGER DEFAULT 0;
    DECLARE patologia VARCHAR(100) DEFAULT '';
    DECLARE farmaco VARCHAR(100) DEFAULT '';
    DECLARE durataTerapia INTEGER DEFAULT 0;
    DECLARE dataEsordio DATE;
    DECLARE dataInizioTerapia DATE;
    DECLARE posologia INTEGER DEFAULT 0;
    DECLARE terapiePrecPosologiaTre INTEGER DEFAULT 0;

    -- terapie relative al paziente e al principio attivo passati
    -- come parametro, iniziate non oltre sei mesi fa (si usa una table
    -- perché nella versione attuale di MySQL le view non possono essere
    -- parametrizzate utilizzando i parametri della procedure
    CREATE TABLE IF NOT EXISTS TerapiePrecPrincAttivo LIKE Terapia;

```

```

INSERT INTO TerapiePrecPrincAttivo
SELECT T.*
FROM Terapia T INNER JOIN Farmaco F ON T.Farmaco = F.NomeCommerciale
WHERE T.DataInizioTerapia >= CURRENT_DATE - INTERVAL 6 MONTH
      AND T.Paziente = codiceFiscale
      AND F.PrincipioAttivo = principioAttivo;

-- cursore per la selezione delle terapie da bloccare
BEGIN

DECLARE terapieDaBloccare CURSOR FOR
SELECT T.Patologia, T.Farmaco, T.DataEsordio, T.DataInizioTerapia,
      DATEDIFF(CURRENT_DATE, T.DataInizioTerapia) AS Durata,
      T.Posologia, (SELECT COUNT(*)
                    FROM TerapiePrecPrincAttivo TPPA
                    WHERE TPPA.DataInizioTerapia < T.DataInizioTerapia
                      AND Posologia > 3) AS TerapiePrecedenti
FROM Terapia T INNER JOIN Farmaco F ON T.Farmaco = F.NomeCommerciale
WHERE T.DataFineTerapia IS NULL
      AND F.PrincipioAttivo = principioAttivo
      AND T.DataInizioTerapia < CURRENT_DATE - INTERVAL 1 WEEK
      AND 3 <= (SELECT COUNT(*)
                FROM TerapiePrecPrincAttivo TPPA
                WHERE TPPA.DataInizioTerapia < T.DataInizioTerapia)
      AND EXISTS
        (SELECT *
         FROM TerapiePrecPrincAttivo TPPA
         WHERE TPPA.DataInizioTerapia < T.DataInizioTerapia
           AND TPPA.Posologia > 3);

DECLARE CONTINUE HANDLER FOR NOT FOUND SET finito = 1;

CREATE TEMPORARY TABLE IF NOT EXISTS _TerapieBloccate(
  ID INTEGER NOT NULL AUTO_INCREMENT,
  Paziente CHAR NOT NULL,
  Patologia CHAR(100) NOT NULL,
  Farmaco CHAR(100) NOT NULL,
  Durata INT(11) NOT NULL,
  Posologia INT(11) NOT NULL,
  TerapiePrecedentiPosTre INT(11) NOT NULL,
  PRIMARY KEY (ID)
) ENGINE=`InnoDB` DEFAULT CHARACTER SET latin1;

OPEN terapieDaBloccare;

-- ciclo che preleva una terapia da bloccare, la inserisce
-- nel risultato, e la blocca
prelievo: LOOP

  FETCH terapieDaBloccare INTO
  patologia, farmaco, dataEsordio, dataInizioTerapia, durataTerapia,
  posologia, terapiePrecPosologiaTre;

  IF finito = 1 THEN
    LEAVE prelievo;
  END IF;

```

```

-- inserimento della terapia bloccata nel risultato
INSERT INTO _TerapieBloccate(Paziente, Patologia, Farmaco, Durata,
                             Posologia, TerapiePrecedentiPostTre)
VALUES (codiceFiscale, patologia, farmaco, durataTerapia, posologia,
        terapiePrecPosologiaTre);

-- blocco effettivo della terapia
UPDATE Terapia SET DataFineTerapia = CURRENT_DATE
WHERE Paziente = codiceFiscale
      AND Patologia = patologia
      AND DataEsordio = dataEsordio
      AND Farmaco = farmaco
      AND DataInizioTerapia = dataInizioTerapia;

END LOOP prelievo;

CLOSE terapieDaBloccare;

END;

DROP TABLE IF EXISTS TerapiePrecPrincAttivo;

-- stampa a video delle terapie bloccate
SELECT * FROM _TerapieBloccate;

END; $$

DELIMITER ;

```

#### *A.A. precedente:*

Scrivere una query che blocchi immediatamente tutte le terapie attualmente in corso, impostando la data di fine terapia alla data corrente, qualora si stiano protrahendo per oltre una settimana, e il paziente abbia già effettuato in precedenza, almeno tre terapie con un farmaco contenente lo stesso principio attivo, di cui almeno una con posologia superiore a tre compresse al giorno.

```

UPDATE Terapia NATURAL JOIN(
  SELECT T.*
  FROM Terapia T INNER JOIN Farmaco F ON T.Farmaco = F.NomeCommerciale
  WHERE T.DataFineTerapia IS NULL
        AND DATEDIFF(CURRENT_DATE, T.DataInizioTerapia) > 7
        AND 3 <= (SELECT COUNT(*)
                  FROM Terapia T2 INNER JOIN Farmaco F2
                  ON T2.Farmaco = F2.NomeCommerciale
                  WHERE T2.Paziente = T.Paziente
                        AND F2.PrincipioAttivo = F.PrincipioAttivo
                        AND T2.DataInizioTerapia < T.DataInizioTerapia)
        AND EXISTS (SELECT *
                    FROM Terapia T2 INNER JOIN Farmaco F2
                    ON T2.Farmaco = F2.NomeCommerciale
                    WHERE T2.Paziente = T.Paziente
                          AND F2.PrincipioAttivo = F.PrincipioAttivo
                          AND T2.DataInizioTerapia < T.DataInizioTerapia
                          AND T2.Posologia > 3)
) AS D
SET DataFineTerapia = CURRENT_DATE;

```