

**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 del primo esercizio 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 elimini tutti gli esordi di otite contratta e curata con successo prima di cinque anni fa, relativi ai soli pazienti che hanno contratto nuovamente, negli ultimi cinque anni, la stessa patologia.

```
DELETE E.*
FROM Esordio E
  NATURAL JOIN
  (
    SELECT *
    FROM Esordio E2
    WHERE E2.Patologia = 'Otite'
      AND E2.DataGuarigione IS NOT NULL
      AND YEAR(E2.DataGuarigione) < YEAR(CURRENT_DATE) - 5
      AND EXISTS
        (
          SELECT *
          FROM Esordio E3
          WHERE E3.Paziente = E2.Paziente
            AND E3.Patologia = E2.Patologia
            AND YEAR(E3.DataEsordio) > YEAR(CURRENT_DATE) - 5
        )
  ) AS Target;
```

**Esercizio 2 (anni accademici precedenti senza la parte fra parentesi quadre)**

Scrivere una query che, considerati i soli pazienti affetti da ipertensione cronica da almeno dieci anni trattata al massimo con due farmaci diversi, indichi il nome commerciale del farmaco mediamente più utilizzato per curare le altre patologie cardiache croniche. [In caso di pari merito, il risultato deve essere vuoto.]

```

CREATE OR REPLACE VIEW PazientiTarget AS
SELECT DISTINCT E.Paziente
FROM Esordio E
     NATURAL JOIN
     Terapia T
WHERE E.Patologia = 'Ipertensione'
     AND E.Cronica = 'si'
     AND E.DataEsordio + INTERVAL 10 YEAR < CURRENT_DATE
GROUP BY E.Paziente, E.DataEsordio
HAVING COUNT(DISTINCT T.Farmaco) <= 2;

CREATE OR REPLACE VIEW TotaliUtilizzi AS
SELECT T.Patologia, T.Farmaco,
     COUNT(DISTINCT T.Paziente) NPazienti
FROM Esordio E
     NATURAL JOIN
     PazientiTarget PT
     NATURAL JOIN
     Terapia T
     INNER JOIN
     Patologia P ON T.Patologia = P.Nome
WHERE P.ParteCorpo = 'Cuore'
     AND P.Nome <> 'Ipertensione'
     AND E.Cronica = 'si'
GROUP BY T.Patologia, T.Farmaco;

SELECT TU.Farmaco
FROM TotaliUtilizzi TU
GROUP BY TU.Farmaco
HAVING AVG(TU.NPazienti) > ALL          /*sostituire con >=ALL per AA prec*/
(
  SELECT AVG(TU2.NPazienti)
  FROM TotaliUtilizzi TU2
  WHERE TU2.Farmaco <> TU.Farmaco /*linea da eliminare per AA prec*/
  GROUP BY TU2.Farmaco
);

```

### Esercizio 3

Un'analisi interna alla clinica ha lo scopo di esaminare il fenomeno del rischio di cronicizzazione delle patologie. Nell'ambito dell'analisi, un farmaco è considerato tanto potente quante patologie ad alta invalidità (superiore al 70%) è in grado di curare. Data una patologia, il rischio di cronicità di un paziente per tale patologia è espressa, utilizzando l'insieme  $\mathcal{T}$  delle terapie del paziente per curare la patologia, come la media normalizzata in  $[0,1]$  dei prodotti  $r_t = posologia_t \cdot potenza\_farmaco_t \cdot durata_t$ , dove  $t \in \mathcal{T}$ . Scrivere una function che, dato un paziente e una patologia, restituisca il relativo rischio di cronicità. Successivamente, implementare una stored procedure che, utilizzando la function, restituisca, dato un paziente, le prime tre patologie con rischio di cronicità superiore a 0.8, corredate dal relativo rischio cronicità, rispettando la seguente formattazione: "patologia<sub>1</sub>-rischio<sub>1</sub>, patologia<sub>2</sub>-rischio<sub>2</sub>, patologia<sub>3</sub>-rischio<sub>3</sub>". Se vi sono più di tre patologie con rischio di cronicità superiore a 0.8 la stringa restituita deve terminare con "-HR", a significare alto rischio cronicità, e la stored procedure deve inserire tutte le coppie patologia-rischio in una temporary table.

```

DROP FUNCTION IF EXISTS Risk;

DELIMITER $$
CREATE FUNCTION Risk(_paziente CHAR(16), _patologia CHAR(100))
RETURNS DOUBLE NOT DETERMINISTIC

```

```

BEGIN

DECLARE rischio_ DOUBLE DEFAULT 0;

CREATE TEMPORARY TABLE IF NOT EXISTS PotenzaFarmaco
(
    Farmaco CHAR(100) NOT NULL,
    Potenza INT(11) NOT NULL
    PRIMARY KEY (Farmaco)
);

REPLACE INTO PotenzaFarmaco
SELECT I.Farmaco, COUNT(*) AS Potenza
FROM Patologia P
    INNER JOIN
    Indicazione I ON P.Nome = I.Patologia
    LEFT OUTER JOIN
    Terapia T USING (Patologia, Farmaco)
WHERE T.Paziente = _paziente
    AND T.Patologia = _patologia
    AND P.Invalidita > 70
GROUP BY I.Farmaco;

SELECT IF(
    COUNT(*) > 1, SUM(DD.Xi)/COUNT(*), -1
)
FROM
(
    SELECT (D.TassoRischio - MIN(D.TassoRischio))
        /
        (MAX(D.TassoRischio) - MIN(D.TassoRischio)) AS Xi
    FROM(
        SELECT T.Posologia
            *PF.Potenza
            *DATEDIFF(
                IFNULL(T.DataFineTerapia, CURRENT_DATE),
                T.DataInizioTerapia
            ) AS TassoRischio
        FROM Esordio E
            NATURAL JOIN
            Terapia T
            NATURAL JOIN
            PotenzaFarmaco PF
        WHERE E.Paziente = _paziente
            AND E.Patologia = _patologia
            AND E.Cronica = 'no'
            AND E.DataGuarigione IS NOT NULL
        ) AS D
    ) AS DD;

RETURN rischio_;

END $$

```

```

DROP PROCEDURE IF EXISTS CheckRisk $$

CREATE PROCEDURE CheckRisk(IN _paziente CHAR(16)
                           OUT patologieRischio_ CHAR(300))
BEGIN
    DECLARE finito INTEGER DEFAULT 0;
    DECLARE patologia CHAR(100) DEFAULT '';
    DECLARE rischioPatologia DOUBLE DEFAULT 0;
    DECLARE elenco CHAR(300) DEFAULT '';
    DECLARE NPatologie INTEGER DEFAULT 0;

    /* elenco patologie contratte con relativo rischio cronicità */
    DECLARE patologie CURSOR FOR
    SELECT DISTINCT E.Patologia,
                   Risk(_paziente, E.Patologia) AS Rischio
    FROM Esordio E
    WHERE E.Paziente = _paziente
           AND E.Cronica = 'no'
           AND E.DataGuarigione IS NOT NULL
           AND Risk(_paziente, E.Patologia) > 0.8
    ORDER BY Rischio DESC;

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET finito = 1;

    /* Temporary table per result set oltre tre patologie */
    DROP TEMPORARY TABLE IF EXISTS Cronicita;
    CREATE TEMPORARY TABLE Cronicita(
        Patologia CHAR(100) NOT NULL,
        Rischio DOUBLE NOT NULL
        PRIMARY KEY (Patologia)
    );

    OPEN patologie;

    scan: LOOP

        FETCH patologie INTO patologia;
        IF (finito = 1) THEN
            LEAVE scan;
        END IF;

        /* prima patologia */
        IF (NPatologie = 0)
            SET elenco = CONCAT(patologia, '-', rischioPatologia);

        /* patologie seguenti */
        ELSE
            /* Se non sono state concatenate 3 coppie occorre concatenare */
            IF (NPatologie <= 3) THEN
                SET elenco = CONCAT(elenco, ',', patologia, '-',
                                   rischioPatologia);
            END IF;

            /* Push nella temporary table */
            INSERT INTO Cronicita
            VALUES(patologia, rischioPatologia);
        END IF;
    END LOOP;

```

```

        SET NPatologie = NPatologie + 1;

END LOOP scan;

/* Se le patologie processate sono più di 3 si concatena "-HR" */
IF (NPatologie > 3) THEN
    SET elenco = CONCAT(elenco, '-HR');

    /* Altrimenti si elimina la temporary table*/
ELSE
    DROP TEMPORARY TABLE Cronicita;

END IF;

SET patologieRischio_ = elenco;

END $$
DELIMITER ;

```

#### A.A. precedenti al 2013-2014

Un'analisi interna alla clinica ha lo scopo di esaminare il fenomeno del rischio di cronicizzazione delle patologie. Nell'ambito dell'analisi, un farmaco è considerato tanto potente quante patologie ad alta invalidità (superiore al 70%) è in grado di curare. Data una patologia, il rischio dicronicità di un paziente per tale patologia è espressa, utilizzando l'insieme  $\mathcal{T}$  delle terapie del paziente per curare la patologia, come la media normalizzata in  $[0,1]$  dei prodotti  $r_t = posologia_t \cdot potenza\_farmaco_t \cdot durata_t$ , dove  $t \in \mathcal{T}$ . Scrivere una query che, considerato ciascun paziente e ciascuna patologia da egli/ella contratta almeno una volta, restituisca il codice fiscale del paziente, il nome della patologia e il relativo rischio di cronicità.

```

CREATE OR REPLACE VIEW PotenzaFarmaco AS
SELECT I.Farmaco, COUNT(*) AS Potenza
FROM Patologia P
    INNER JOIN
        Indicazione I ON P.Nome = I.Patologia
WHERE P.Invalidita > 70
GROUP BY I.Farmaco;

CREATE OR REPLACE VIEW Prodotti AS
SELECT E.Paziente,
       E.Patologia,
       T.Posologia*PF.Potenza
       *DATEDIFF(
           IFNULL(T.DataFineTerapia, CURRENT_DATE),
           T.DataInizioTerapia
       ) AS TassoRischio
FROM Esordio E
    NATURAL JOIN
        Terapia T
    NATURAL JOIN
        PotenzaFarmaco PF
WHERE E.Cronica = 'no'
    AND E.DataGuarigione IS NOT NULL;

```

```
CREATE OR REPLACE VIEW MaxMin AS
SELECT PR.Paziente,
       PR.Patologia,
       MAX(PR.TassoRischio) AS Massimo,
       MIN(PR.TassoRischio) AS Minimo
FROM Prodotti PR
GROUP BY PR.Paziente, PR.Patologia;

SELECT PR.Paziente,
       PR.Patologia,
       SUM(
         (PR.TassoRischio - MM.Minimo)
         /
         (MM.Massimo - MM.Minimo)
       )
       /COUNT(*) AS RischioCronicita
FROM Prodotti PR
   NATURAL JOIN
   MaxMin MM
GROUP BY PR.Paziente, PR.Patologia
HAVING COUNT(*) > 1;
```