

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 la dose giornaliera media dei farmaci indicati per la cura di sole patologie intestinali.

```
SELECT AVG(I1.DoseGiornaliera)
FROM Indicazione I1
WHERE I1.Farmaco NOT IN(
  SELECT I2.Farmaco
  FROM Indicazione I2 INNER JOIN Patologia P ON I2.Patologia = P.Nome
  WHERE P.ParteCorpo <> 'Intestino'
);
```

Esercizio 2

Scrivere una query che restituisca, per il sesso maschile e per quello femminile, rispettivamente, il numero di pazienti attualmente affetti da ipertensione, trattata con lo stesso farmaco da più di venti anni.

```
SELECT P.Sesso, COUNT(DISTINCT P.CodFiscale)
FROM Esordio E INNER JOIN Paziente P ON E.Paziente = P.CodFiscale
WHERE E.DataGuarigione IS NULL
  AND E.Patologia = 'Ipertensione'
  AND EXISTS(
    SELECT *
    FROM Terapia T
    WHERE T.Paziente = E.Paziente
      AND T.Patologia = E.Patologia
      AND T.DataEsordio = E.DataEsordio
      AND T.DataFineTerapia IS NULL
      AND T.DataInizioTerapia < CURRENT_DATE - INTERVAL 20 YEAR
  )
GROUP BY P.Sesso;
```

Esercizio 3

Scrivere una query che, considerate le sole patologie muscolari, elimini gli esordi conclusi con guarigione relativi a pazienti che hanno contratto, e curato con successo, almeno due di tali patologie.

```
DELETE E.*
FROM Esordio E INNER JOIN Patologia PA ON E.Patologia = PA.Nome
    NATURAL JOIN(
        SELECT E2.Paziente
        FROM Esordio E2 INNER JOIN Patologia P ON E2.Patologia = P.Nome
        WHERE E2.DataGuarigione IS NOT NULL
            AND P.ParteCorpo = 'Muscoli'
        GROUP BY E2.Paziente
        HAVING COUNT(DISTINCT E2.Patologia) >= 2
    ) AS D
WHERE PA.ParteCorpo = 'Muscoli'
    AND E.DataGuarigione IS NOT NULL;
```

Esercizio 4

Negli ultimi mesi, la direzione della clinica è interessata al fenomeno della resistenza alle terapie per la patologia influenzale. I pazienti target sono gli anziani aventi più di ottanta anni, affetti da almeno due patologie croniche. Dato un paziente target, interessano i suoi esordi di influenza degli ultimi tre anni. Considerato un esordio i , sia T il numero di terapie effettuate per curarlo, e sia d_{ij} la durata della terapia j relativa all'esordio i . La resistenza della patologia nell'esordio i è espressa da: $r_i = \frac{1}{T} \sum_{j=1}^T d_{ij}$. Supponendo che gli esordi di influenza del paziente considerato siano E , se è $r_1 < r_2 < \dots < r_E$, allora vi è una resistenza ai farmaci per il trattamento dell'influenza, e il tasso di resistenza è quantificabile come

$$TDR = \frac{\bar{r}}{\sum_{i=1}^E (r_i - \bar{r})^2},$$

dove $\bar{r} = \frac{1}{E} \sum_{i=1}^E r_i$. Scrivere una function per il calcolo del TDR , e il codice per il deferred full refresh mensile di una materialized view contenente il codice fiscale di un paziente e il relativo TDR .

```
DROP FUNCTION IF EXISTS tdr;
DELIMITER $$

CREATE FUNCTION tdr(codiceFiscale CHAR(16))
RETURNS DOUBLE NOT DETERMINISTIC

BEGIN

    DECLARE data_nascita DATE;
    DECLARE patologie_croniche INTEGER DEFAULT 0;
    DECLARE esordi_influenza INTEGER DEFAULT 0;

    DECLARE r_mean DOUBLE DEFAULT 0;
    DECLARE somma_scostamenti DOUBLE DEFAULT 0;

    SELECT P.DataNascita INTO data_nascita
    FROM Paziente P
    WHERE P.CodFiscale = codiceFiscale;

    SELECT COUNT(*) INTO patologie_croniche
    FROM Esordio E
    WHERE E.Paziente = codiceFiscale
        AND E.Cronica = 'si';
```

```

SELECT COUNT(*) INTO esordi_influenza
FROM Esordio E
WHERE E.Paziente = codiceFiscale
      AND E.Patologia = 'Influenza'
      AND YEAR(E.DataEsordio) BETWEEN YEAR(CURRENT_DATE)-2
                                AND YEAR(CURRENT_DATE);

/* Se il paziente argomento della function non è un paziente target
   la function restituisce -1 */
IF data_nascita + INTERVAL 80 YEAR >= CURRENT_DATE
   OR patologie_croniche < 2 THEN
   RETURN -1;
END IF;

/* Se il paziente non ha contratto l'influenza negli ultimi tre anni
   la function restituisce 0 */
IF esordi_influenza = 0 THEN
   RETURN 0;
END IF;

BEGIN

  DECLARE ri_prec DOUBLE DEFAULT 0;
  DECLARE ri_succ DOUBLE DEFAULT 0;
  DECLARE ri_sum DOUBLE DEFAULT 0;
  DECLARE ri_scostamento DOUBLE DEFAULT 0;
  DECLARE finito INTEGER DEFAULT 0;

  /* Il resultset assegnato al cursore contiene, per ciascun esordio i
     del paziente, la relativa resistenza ri */

  DECLARE resistenze CURSOR FOR
  SELECT AVG(DATEDIFF(IFNULL(T.DataFineTerapia, CURRENT_DATE),
                      T.DataInizioTerapia)) ri
  FROM Esordio E NATURAL JOIN Terapia T
  WHERE E.Paziente = codiceFiscale
        AND E.Patologia = 'Influenza'
        AND YEAR(E.DataEsordio) BETWEEN YEAR(CURRENT_DATE)-2
                                AND YEAR(CURRENT_DATE)
  GROUP BY E.DataEsordio;

  DECLARE CONTINUE HANDLER FOR NOT FOUND SET finito = 1;

  SELECT SUM(D.ri)/esordi_influenza INTO r_mean
  FROM(
    SELECT AVG(DATEDIFF(IFNULL(T.DataFineTerapia, CURRENT_DATE),
                        T.DataInizioTerapia)) ri
    FROM Esordio E NATURAL JOIN Terapia T
    WHERE E.Paziente = codiceFiscale
          AND E.Patologia = 'Influenza'
          AND YEAR(E.DataEsordio) BETWEEN YEAR(CURRENT_DATE)-2
                                AND YEAR(CURRENT_DATE)

    GROUP BY E.DataEsordio
  )AS D;

```

```

OPEN resistenze;

scan: LOOP

    FETCH resistenze INTO ri_succ;

    IF finito = 1 THEN
        LEAVE scan;
    END IF;

    IF ri_succ > ri_prec THEN
        BEGIN
            SET ri_scostamento = ri_succ - r_mean;
            /* La funzione POW(value,exp) calcola value^exp */
            SET somma_scostamenti =
                somma_scostamenti + POW(ri_scostamento,2);
            SET ri_prec = ri_succ;
        END;
    ELSE RETURN 0;
    END IF;

END LOOP scan;

CLOSE resistenze;
END;

RETURN r_mean/somma_scostamenti;

END $$

DELIMITER ;

DROP TABLE IF EXISTS MV_TassiResistenza;
CREATE TABLE MV_TassiResistenza (
    'CodiceFiscale' CHAR(16) NOT NULL,
    'TDR' DOUBLE NOT NULL,
    PRIMARY KEY ('CodiceFiscale')
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

DELIMITER $$

CREATE EVENT aggiornaTDR
ON SCHEDULE EVERY 1 MONTH
DO
BEGIN
    REPLACE INTO MV_TassiResistenza
    SELECT P.CodFiscale, tdr(P.CodFiscale)
    FROM Paziente P
    WHERE P.DataNascita + INTERVAL 80 YEAR < CURRENT_DATE
        AND 2 <= (SELECT COUNT(*)
                FROM Esordio E
                WHERE E.Paziente = P.CodFiscale
                AND E.Cronica = 'si'
                );
END $$

DELIMITER;

```

A.A. precedente:

Negli ultimi mesi, la direzione della clinica è interessata al fenomeno della resistenza alle terapie per la patologia influenzale. I pazienti target sono gli anziani aventi più di ottanta anni, affetti da almeno due patologie croniche. Dato un paziente target, interessano i suoi esordi di influenza degli ultimi tre anni. Considerato un esordio i , sia T il numero di terapie effettuate per curarlo, e sia d_{ij} la durata della terapia j relativa all'esordio i . La resistenza della patologia nell'esordio i è espressa da: $r_i = \frac{1}{T} \sum_{j=1}^T d_{ij}$. Creare una tabella ridondante RESISTENZA, contenente, per ciascun esordio di ogni paziente target, il suo codice fiscale e il relativo tasso di resistenza. Scrivere poi un event che aggiorni mensilmente la tabella RESISTENZA.

```
CREATE TABLE Resistenza (  
  'CodiceFiscale' CHAR(16) NOT NULL,  
  'TassoResistenza' DOUBLE NOT NULL,  
  PRIMARY KEY ('CodiceFiscale, TassoResistenza')  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
CREATE OR REPLACE VIEW PazientiTarget AS  
SELECT E.Paziente  
FROM Paziente P INNER JOIN Esordio E ON P.CodFiscale = E.Paziente  
WHERE P.DataNascita + INTERVAL 80 YEAR <= CURRENT_DATE  
      AND E.Cronica = 'si'  
GROUP BY E.Paziente  
HAVING COUNT(*) >= 2;  
  
DELIMITER $$  
CREATE EVENT AggiornaResistenza  
ON SCHEDULE EVERY 1 MONTH  
DO  
BEGIN  
  TRUNCATE TABLE Resistenza;  
  INSERT INTO Resistenza  
  SELECT D.Paziente, AVG(D.dij)  
  FROM(  
    SELECT T.Paziente, T.DataEsordio,  
           DATEDIFF(IF(T.DataFineTerapia IS NOT NULL,  
                     T.DataFineTerapia, CURRENT_DATE)  
                   ,  
                   T.DataInizioTerapia) AS dij  
  FROM Terapia T NATURAL JOIN PazientiTarget  
  WHERE T.Patologia = 'Influenza'  
        AND YEAR(T.DataEsordio) BETWEEN YEAR(CURRENT_DATE)-2  
        AND YEAR(CURRENT_DATE)  
  ) AS D  
  GROUP BY D.Paziente, D.DataEsordio  
END $$  
DELIMITER ;
```