

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 parte del corpo maggiormente colpita da patologie con invalidità superiore al 70%. In caso di pari merito, restituire tutte le parti del corpo.

```
CREATE OR REPLACE VIEW PatologieAltaInvalidita AS
SELECT Nome AS Patologia, ParteCorpo
FROM Patologia
WHERE Invalidita > 70;

SELECT PAI.ParteCorpo
FROM Esordio E NATURAL JOIN PatologieAltaInvalidita PAI
GROUP BY PAI.ParteCorpo
HAVING COUNT(*) >= ALL (
  SELECT COUNT(*) AS NumeroEsordi
  FROM Esordio E2 NATURAL JOIN PatologieAltaInvalidita PAI2
  GROUP BY PAI2.ParteCorpo);
```

Esercizio 2

Scrivere una query che restituisca il numero di terapie iniziate da ciascun paziente in ogni mese dell'anno. Nel risultato devono comparire tutti i pazienti e tutti i mesi dell'anno.

```
SELECT PM.CodFiscale, PM.Mese, IF(P.CodFiscale IS NULL, 0, COUNT(*))
FROM Terapia T RIGHT OUTER JOIN Paziente P ON T.Paziente = P.CodFiscale
  RIGHT OUTER JOIN (SELECT P.CodFiscale, D.Mese
                    FROM Paziente P CROSS JOIN
                    (SELECT DISTINCT MONTH(DataInizioTerapia) AS Mese
                     FROM Terapia) AS D
                   ) AS PM
  ON (PM.CodFiscale = P.CodFiscale
      AND PM.Mese = MONTH(T.DataInizioTerapia))
GROUP BY PM.CodFiscale, PM.Mese;
```

A.A. precedente:

Scrivere una query che restituisca, relativamente all'ultimo triennio, il numero di terapie iniziate da ciascun paziente. Nel risultato devono comparire tutti i pazienti.

```
SELECT P.CodFiscale, IF(D.Paziente IS NULL, 0, COUNT(*))
FROM Paziente P LEFT OUTER JOIN(
    SELECT T.Paziente
    FROM Terapia T
    WHERE YEAR(T.DataInizioTerapia) BETWEEN
        YEAR(CURRENT_DATE)-3 AND YEAR(CURRENT_DATE)-1
) AS D
ON P.CodFiscale = D.Paziente
GROUP BY P.CodFiscale;
```

Esercizio 3

Scrivere una query che restituisca, relativamente al mese di Giugno 2011, la percentuale d'incasso totale mensile dovuta alle visite nefrologiche. Non si usino view, né derived table.

```
SELECT (
    (SELECT SUM(M.Parcella)
    FROM Visita V INNER JOIN Medico M ON V.Medico = M.Matricola
    WHERE MONTH(V.Data) = 6 AND YEAR(V.Data) = 2011
    AND M.Specializzazione = 'Nefrologia')
    /
    SUM(M.Parcella)
) * 100
FROM Visita V INNER JOIN Medico M ON V.Medico = M.Matricola
WHERE MONTH(V.Data) = 6 AND YEAR(V.Data) = 2011;
```

Esercizio 4

Scrivere una stored procedure per l'inserimento di una nuova terapia. Nel caso in cui il paziente oggetto della terapia non abbia assunto in precedenza lo stesso principio attivo, la terapia non deve essere inserita e deve essere restituito un messaggio di errore del tipo: "Il paziente potrebbe essere allergico al principio attivo X". Sostituire X con il nome del principio attivo oggetto della terapia. La stored procedure non deve contenere istruzioni di tipo CREATE.

```
DROP PROCEDURE IF EXISTS inserisci_terapia;
DELIMITER $$
```

```
CREATE PROCEDURE inserisci_terapia(
    IN paziente CHAR(100), IN patologia CHAR(100), IN dataEsordio DATE,
    IN farmaco CHAR(100), IN posologia INTEGER)
```

```
BEGIN
```

```
    DECLARE sicuro INTEGER DEFAULT 0;
    DECLARE finito INTEGER DEFAULT 0;
    DECLARE principioAttivoCur CHAR(100) DEFAULT '';
    DECLARE principioAttivoPrescritto CHAR(100);
```

```

DECLARE terapiePassate CURSOR FOR
SELECT DISTINCT F.PrincipioAttivo
FROM Terapia T INNER JOIN Farmaco F ON T.Farmaco = F.NomeCommerciale
WHERE T.Paziente = paziente;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET finito = 1;

SET principioAttivoPrescritto = (SELECT PrincipioAttivo
                                FROM Farmaco
                                WHERE NomeCommerciale = farmaco);

OPEN terapiePassate;

prelievo: LOOP

    FETCH terapiePassate INTO principioAttivoCur;

    IF finito = 1 THEN
        LEAVE prelievo;
    END IF;

    IF principioAttivoCur = principioAttivoPrescritto THEN
        BEGIN
            SET sicuro = 1;
            LEAVE prelievo;
        END;
    END IF;

END LOOP prelievo;

CLOSE terapiePassate;

IF sicuro = 1 THEN
    INSERT INTO Terapia VALUES (paziente, patologia, dataEsordio,
                                farmaco, CURRENT_DATE, NULL, posologia);
ELSE
    SELECT CONCAT('Il paziente potrebbe essere allergico
                  al principio attivo ', principioAttivoPrescritto);
END IF;

END; $$
DELIMITER ;

```

A.A. precedente:

Scrivere una query che restituisca, per ogni paziente della clinica, nessuno escluso, codice fiscale, reddito, e il numero di volte in cui ha contratto patologie relative a settori medici per i quali non ha mai effettuato visite in precedenza.

```

SELECT P.CodFiscale, P.Reddito, IF(D.Paziente IS NULL, 0, COUNT(*))
FROM (
    SELECT E.Paziente
    FROM Esordio E INNER JOIN Patologia PA ON E.Patologia = PA.Nome
    WHERE NOT EXISTS (
        SELECT *
        FROM Visita V INNER JOIN Medico M ON V.Medico = M.Matricola

```

```
WHERE V.Paziente = E.Paziente
      AND M.Specializzazione = PA.SettoreMedico
      AND V.Data < E.DataEsordio
)
) AS D
RIGHT OUTER JOIN Paziente P ON D.Paziente = P.CodFiscale
GROUP BY P.CodFiscale;
```