

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, considerata ciascuna parte del corpo, ne restituisca il nome, e i principi attivi contenuti in farmaci indicati solamente per la cura di patologie a carico di tale parte del corpo.

```
SELECT DISTINCT F.PrincipioAttivo,  
                P.ParteCorpo  
FROM Indicazione I  
     INNER JOIN  
     Patologia P ON I.Patologia = P.Nome  
     INNER JOIN  
     Farmaco F ON I.Farmaco = F.NomeCommerciale  
GROUP BY I.Farmaco  
HAVING COUNT(DISTINCT P.ParteCorpo) = 1;
```

Esercizio 2

Scrivere una query che elenchi nome e cognome dei pazienti oggi maggiorenni che, al 5 Settembre 2015, erano stati visitati da tutti gli oculisti della clinica, tranne eventualmente uno, e, qualora esista, il cognome di tale oculista.

```
CREATE OR REPLACE VIEW Oculisti AS  
SELECT M.Matricola,  
       M.Nome,  
       M.Cognome  
FROM Medico M  
WHERE M.Specializzazione = 'Oculistica';  
  
SET@num_oculisti = (  
    SELECT COUNT(*)  
    FROM Oculisti  
);
```

```

CREATE OR REPLACE VIEW PazientiVisitatiTuttiQuasiOculisti AS
SELECT V.Paziente,
       COUNT(DISTINCT V.Medico) AS Quanti
FROM Visita V
     INNER JOIN
     Oculisti O ON V.Medico = O.Matricola
     INNER JOIN Paziente P ON V.Paziente = P.CodFiscale
WHERE P.DataNascita + INTERVAL 18 YEAR < CURRENT_DATE
GROUP BY V.Paziente
HAVING COUNT(DISTINCT V.Medico) >=
(
    SELECT COUNT(*) - 1
    FROM Oculisti
);

CREATE OR REPLACE VIEW MappingPazienteOculista AS
SELECT DISTINCT V.Medico,
               V.Paziente
FROM Visita V
     INNER JOIN
     Oculisti O ON V.Medico = O.Matricola;

CREATE OR REPLACE VIEW Combinazioni AS
SELECT O.Matricola AS Medico,
       PVTQ0.Paziente,
       PVTQ0.Quanti
FROM Oculisti O
     CROSS JOIN
     PazientiVisitatiTuttiQuasiOculisti PVTQ0;

SELECT P.Nome,
       P.Cognome,
       O.Cognome
FROM
(
    SELECT DISTINCT C.Paziente,
                   C.Quanti,
                   IF(MP0.Medico IS NULL, C.Medico, NULL) AS Medico
    FROM Combinazioni C
         LEFT OUTER JOIN
         MappingPazienteOculista MP0
         ON C.Paziente = MP0.Paziente
            AND C.Medico = MP0.Medico
) AS D
INNER JOIN
Paziente P ON D.Paziente = P.CodFiscale
LEFT OUTER JOIN
Oculisti O ON D.Medico = O.Matricola
WHERE (
    D.Quanti = @num_oculisti
    AND D.Medico IS NULL
)
OR
(
    D.Quanti = @num_oculisti - 1
    AND D.Medico IS NOT NULL
);

```

A.A. precedente (2013-2014)

Scrivere una query che elenchi nome e cognome dei pazienti oggi maggiorenni che, al 5 Settembre 2015 erano stati visitati da tutti gli oculisti della clinica.

```

CREATE OR REPLACE VIEW Oculisti AS
SELECT M.Matricola,
       M.Nome,
       M.Cognome
FROM Medico M
WHERE M.Specializzazione = 'Oculistica';

SELECT P.Nome,
       P.Cognome
FROM Visita V
     INNER JOIN
     Oculisti O ON V.Medico = O.Matricola
     INNER JOIN
     Paziente P ON V.Paziente = P.CodFiscale
WHERE P.DataNascita + INTERVAL 18 YEAR < CURRENT_DATE
GROUP BY V.Paziente
HAVING COUNT(DISTINCT V.Medico) =
(
    SELECT COUNT(*)
    FROM Oculisti
);

```

Esercizio 3

Scrivere una stored procedure `report_spese` che riceva in ingresso tre parametri: il codice fiscale di un paziente i , il nome di un settore medico s e un parametro booleano (tinyint) ssn . La stored procedure deve restituire la spesa totale [e media] giornaliera sostenuta attualmente dal paziente p per le terapie in corso del settore medico s . Le spese sopra descritte sono calcolate in modo diverso dipendentemente dal valore di ssn . In particolare, se $ssn = 1$ la stored procedure restituisce le varie spese al netto della percentuale di esenzione, ove prevista, altrimenti l'esenzione è ignorata. Alla percentuale di esenzione associata alla patologia j , deve essere sommato un coefficiente $C_{i,s}$ dipendente dal reddito R_i del paziente e dal numero di patologie croniche $P_{i,s}$ attinenti al settore medico s , da cui è affetto il paziente i , secondo la seguente espressione: $P_{i,s} = C_{i,s} / (0.01 \cdot R_i)$.

```

DROP PROCEDURE IF EXISTS report_spese;

DELIMITER $$

CREATE PROCEDURE report_spese(IN _paz CHAR(100),
                             IN _spec CHAR(100),
                             IN _ssn TINYINT,
                             OUT totale_ DOUBLE,
                             OUT media_ DOUBLE)
BEGIN

    DECLARE nTerapie INTEGER DEFAULT 0;
    DECLARE spesaTotale DOUBLE DEFAULT 0;

    DECLARE C_is INTEGER DEFAULT 0;
    DECLARE R_i DOUBLE DEFAULT 0;
    DECLARE P_is DOUBLE DEFAULT 0;

    DECLARE costoUnitarioFarmaco DOUBLE DEFAULT 0;
    DECLARE esenzione DOUBLE DEFAULT 0;

    DECLARE costoLordoTerapia DOUBLE DEFAULT 0;
    DECLARE costoNettoTerapia DOUBLE DEFAULT 0;

    DECLARE finito INTEGER DEFAULT 0;

```

```

/* controllo validità paziente e settore medico*/
IF NOT EXISTS
(
    SELECT *
    FROM Paziente
    WHERE CodFiscale = _paz
)
OR
NOT EXISTS
(
    SELECT *
    FROM Patologia
    WHERE SettoreMedico = _spec
)
THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Paziente o settore medico non validi';
END IF;

/*
apertura di un blocco per il vincolo di posizione del cursore:
non si può dichiarare un cursore se non immediatamente dopo le DECLARE
*/
BEGIN
    DECLARE dettaglioTerapie
    CURSOR FOR
    SELECT F.Costo/F.Pezzi AS CostoUnitario,
           T.Posologia,
           P.PercEsenzione
    FROM Terapia T
    INNER JOIN
        Patologia P ON T.Patologia = P.Nome
    INNER JOIN
        Farmaco F ON T.Farmaco = F.NomeCommerciale
    WHERE T.Paziente = _paz
           AND P.SettoreMedico = _spec
           AND T.DataFineTerapia IS NULL;

    DECLARE CONTINUE HANDLER
    FOR NOT FOUND
    SET finito = 1;

    SET R_i =
    (
        SELECT P.Reddito
        FROM Paziente P
        WHERE P.CodFiscale = _paz
    );

    SET C_is =
    (
        SELECT COUNT(*)
        FROM Esordio E
        INNER JOIN
            Patologia P ON E.Patologia = P.Nome
        WHERE E.Paziente = _paz
              AND P.SettoreMedico = _spec
              AND E.Cronica = 'si'
    );

    SET P_is = C_is/(0.01*R_i);

    OPEN dettaglioTerapie;

```

```

scan: LOOP

    FETCH dettaglioTerapie INTO costoUnitarioFarmaco,
                                posologia,
                                esenzione;

    IF finito = 1 THEN
        LEAVE scan;
    END IF;

    SET costoLordoTerapia = costoUnitarioFarmaco*posologia;

    /* calcolo del costo terapia CON esenzione, ove prevista */
    IF _ssn = 1 THEN
        BEGIN
            IF esenzione = 0 THEN
                SET costoNettoTerapia = costoLordoTerapia;
            ELSE
                SET costoNettoTerapia = costoLordoTerapia -
                                        costoLordoTerapia*(esenzione + P_is);
            END IF;
            SET spesaTotale = spesaTotale + costoNettoTerapia;
        END;

    /* calcolo del costo terapia SENZA esenzione */
    ELSEIF _ssn = 0 THEN
        SET spesaTotale = spesaTotale + costoLordoTerapia;

    /* ssn non valido in input */
    ELSE
        BEGIN
            SET totale_ = NULL;
            SET media_ = NULL;
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Parametro _ssn non valido';
        END;
    END IF;

    SET nTerapie = nTerapie + 1;

END LOOP scan;

CLOSE dettaglioTerapie;

/* impostazione del risultato */
SET totale_ = spesaTotale;
SET media_ = spesaTotale/nTerapie;
END;
END $$
DELIMITER ;

/*
*****
ESEMPIO DI CHIAMATA (parametri 'bbe1', 'Ortopedia', 1)
*****
*/
SET @totale = 0;
SET @media = 0;

CALL report_spese('bbe1', 'Ortopedia', 1, @totale, @media);
SELECT @totale, @media;

```