

## 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)

Esprimere le seguenti richieste in linguaggio SQL.

Quelle che seguono sono possibili soluzioni delle query proposte. Soluzioni diverse sono corrette purché producano lo stesso risultato.

### Esercizio 1

Considerato ciascun principio attivo, indicarne il nome e il costo medio al pezzo fra tutti i farmaci che lo contengono.

```
SELECT F.PrincipioAttivo, AVG(F.Costo/F.Pezzi) AS CostoMedioPezzo
FROM Farmaco F
GROUP BY F.PrincipioAttivo;
```

### Esercizio 2

Indicare nome e cognome dei pazienti che hanno contratto almeno due volte tutte le patologie intestinali.

```
CREATE OR REPLACE VIEW DoppiEsordi AS
SELECT E.Patologia, E.Paziente
FROM Patologia PA INNER JOIN Esordio E
    ON PA.Nome = E.Patologia
WHERE PA.ParteCorpo = 'Intestino'
GROUP BY E.Patologia, E.Paziente
HAVING COUNT(*) >= 2;

SELECT P.Nome, P.Cognome
FROM DoppiEsordi DE INNER JOIN Paziente P
    ON DE.Paziente = P.CodFiscale
GROUP BY DE.Paziente, P.Nome, P.Cognome
HAVING COUNT(*) = (SELECT COUNT(*)
    FROM Patologia
    WHERE ParteCorpo = 'Intestino');
```

### Esercizio 3

Indicare le patologie esordite esclusivamente in forma cronica, curate con il farmaco *Lyrica*.

```
SELECT DISTINCT E.Patologia
FROM Esordio E NATURAL JOIN Terapia T
WHERE T.Farmaco = 'Lyrica'
      AND E.Patologia NOT IN( SELECT E2.Patologia
                              FROM Esordio E2
                              WHERE E2.Cronica = 'no');
```

### Esercizio 4

Scrivere un trigger che, in risposta all'aggiornamento della gravità di un esordio, elimini l'esordio stesso, e ne inserisca un altro caratterizzato dalla nuova gravità, dalla data dell'aggiornamento come data di esordio, e dagli stessi valori dell'esordio eliminato per i restanti attributi.

```
DELIMITER $$
CREATE TRIGGER ModificaGravita BEFORE UPDATE ON Esordio
FOR EACH ROW
BEGIN
    IF (NEW.Gravita <> OLD.Gravita AND NEW.Paziente = OLD.Paziente
        AND NEW.Patologia = OLD.Patologia AND NEW.DataGuarigione = NULL
        AND NEW.DataEsordio = OLD.DataEsordio AND NEW.Cronica = OLD.Cronica)
    THEN
        INSERT INTO Esordio
        VALUES (OLD.Paziente, OLD.Patologia, CURRENT_DATE,
                NULL, NEW.Gravita, OLD.Cronica);

        DELETE FROM Esordio
        WHERE (Paziente, Patologia, DataEsordio) =
              (OLD.Paziente, OLD.Patologia, OLD.DataEsordio);

        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Si è tentato un aggiornamento non permesso.
                            L'esordio è stato cancellato e sostituito.';
    END IF;
END $$
DELIMITER ;
```

*A.A. precedente:*

Cancellare ogni esordio non ancora concluso, qualora lo stesso paziente abbia contratto successivamente la stessa patologia, ma in forma più grave.

```
DELETE E.*
FROM Esordio AS E NATURAL JOIN
    (SELECT Paziente, Patologia, DataEsordio
     FROM Esordio AS E1
     WHERE E1.DataGuarigione IS NULL
          AND EXISTS (SELECT *
                     FROM Esordio E2
                     WHERE E2.Paziente = E1.Paziente
                          AND E2.Patologia = E1.Patologia
                          AND E2.DataEsordio > E1.DataEsordio
                          AND E2.Gravita > E1.Gravita)
    ) AS D;
```

## Esercizio 5

Indicare la specializzazione medica che, considerate le visite effettuate dai suoi medici dall'anno 2010 a oggi, ha totalizzato ogni anno un numero di pazienti visitati per ciascuna città mai inferiore all'anno precedente.

```
CREATE OR REPLACE VIEW Anni AS
SELECT DISTINCT YEAR(Data) AS Anno
FROM Visita
WHERE YEAR(Data) >= 2010;
```

```
CREATE OR REPLACE VIEW Specializzazioni AS
SELECT DISTINCT Specializzazione
FROM Medico;
```

```
CREATE OR REPLACE VIEW CittaPazienti AS
SELECT DISTINCT Citta
FROM Paziente;
```

```
CREATE OR REPLACE VIEW AnnoSpecializzazioneCitta AS
SELECT *
FROM Anni CROSS JOIN Specializzazioni CROSS JOIN CittaPazienti;
```

```
CREATE OR REPLACE VIEW ParzialeVisite AS
SELECT YEAR(V.Data) AS Anno, M.Specializzazione, P.Citta,
       COUNT(DISTINCT V.Paziente) AS NumeroPazienti
FROM Visita V INNER JOIN Paziente P ON V.Paziente = P.CodFiscale
     INNER JOIN Medico M ON V.Medico = M.Matricola
WHERE YEAR(V.Data) >= 2010
GROUP BY YEAR(V.Data), M.Specializzazione, P.Citta;
```

```
CREATE OR REPLACE VIEW VisiteAnnoSpecializzazioneCitta AS
SELECT *
FROM ParzialeVisite
UNION
SELECT ASCI.Anno, ASCI.Specializzazione, ASCI.Citta, 0
FROM ParzialeVisite PV NATURAL RIGHT OUTER JOIN
     AnnoSpecializzazioneCitta ASCI
WHERE PV.Anno IS NULL;
```

```
SELECT D.Specializzazione
FROM (
    SELECT V1.Specializzazione
    FROM VisiteAnnoSpecializzazioneCitta V1 INNER JOIN
         VisiteAnnoSpecializzazioneCitta V2
         ON (V2.Anno = V1.Anno + 1
             AND V2.Specializzazione = V1.Specializzazione
             AND V2.Citta = V1.Citta)
    WHERE V2.NumeroPazienti >= V1.NumeroPazienti
    GROUP BY V1.Anno, V1.Specializzazione
    HAVING COUNT(*) = (SELECT COUNT(DISTINCT Citta)
                       FROM Paziente)
) AS D
GROUP BY D.Specializzazione
HAVING COUNT(*) = (SELECT COUNT(*)
                   FROM Anni);
```