

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

Indicare nome e cognome di ciascun medico che ha visitato tutti i pazienti della sua città.

```
SELECT M.Nome, M.Cognome
FROM Medico M
WHERE EXISTS (SELECT *
              FROM Paziente P1
              WHERE P1.Citta = M.Citta)
AND NOT EXISTS (SELECT *
               FROM Paziente P2
               WHERE P2.Citta = M.Citta
               AND NOT EXISTS (
                 SELECT *
                 FROM Visita V
                 WHERE V.Medico = M.Matricola
                       AND V.Paziente = P2.CodFiscale)
               );
```

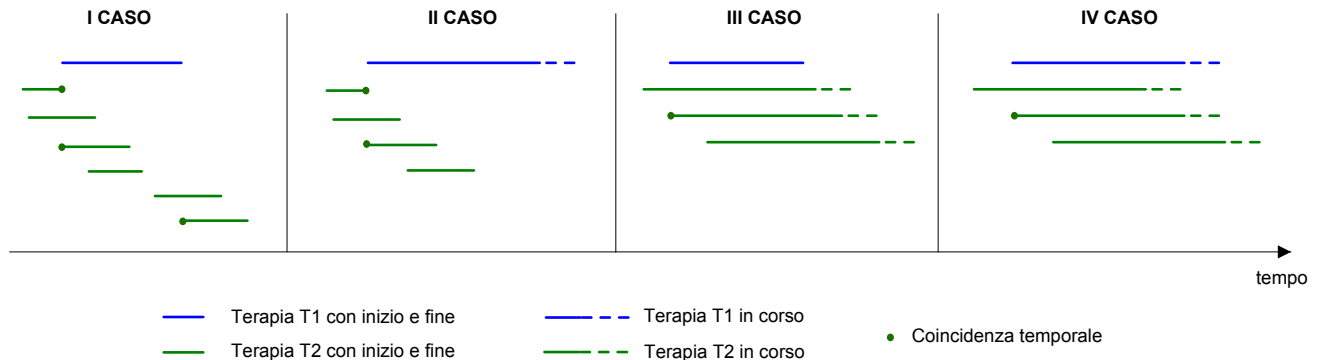
A.A. precedente:

Per ogni mese dell'anno, indicare il mese e il numero di pazienti visitati più di due volte.

```
SELECT D.MeseVisita, COUNT(*) AS NumeroPazienti
FROM (
  SELECT MONTH(V.Data) AS MeseVisita, V.Paziente
  FROM Visita V
  GROUP BY MONTH(V.Data), V.Paziente
  HAVING COUNT(*) > 2
) AS D
GROUP BY D.MeseVisita;
```

Esercizio 2

Indicare nome e cognome dei pazienti che hanno avuto, anche solo per un giorno, più terapie in corso contemporaneamente.



```
SELECT DISTINCT P.Nome, P.Cognome
FROM Terapia T1 INNER JOIN Terapia T2
  ON (T1.Paziente = T2.Paziente
      AND (T1.Patologia <> T2.Patologia
          OR T1.DataEsordio <> T2.DataEsordio
          OR T1.Farmaco <> T2.Farmaco
          OR T1.DataInizioTerapia <> T2.DataInizioTerapia)
      )
  INNER JOIN Paziente P ON P.CodFiscale = T1.Paziente
WHERE (T1.DataFineTerapia IS NOT NULL
      AND (T2.DataFineTerapia IS NOT NULL /*T1 e T2 terminate*/
          AND (T2.DataInizioTerapia BETWEEN
              T1.DataInizioTerapia AND T1.DataFineTerapia
              OR
              T2.DataFineTerapia BETWEEN
              T1.DataInizioTerapia AND T1.DataFineTerapia
          )
      )
      OR
      (T1.DataFineTerapia IS NULL
          AND (T2.DataFineTerapia IS NOT NULL /*T1 in corso, T2 terminata*/
              AND (T2.DataInizioTerapia >= T1.DataInizioTerapia
                  OR
                  T2.DataFineTerapia >= T1.DataInizioTerapia
              )
          )
      )
      OR
      (T1.DataFineTerapia IS NOT NULL
          AND T2.DataFineTerapia IS NULL /*T1 terminata, T2 in corso*/
          AND T2.DataInizioTerapia <= T1.DataFineTerapia
      )
      OR
      (T1.DataFineTerapia IS NULL
          AND T2.DataFineTerapia IS NULL /*T1 in corso, T2 in corso*/
      )
  );
```

Esercizio 3

Indicare il reddito massimo fra quelli di tutti i pazienti che, nell'anno 2011, hanno effettuato esattamente tre visite, ognuna delle quali con un medico avente specializzazione diversa dagli altri.

```
SELECT MAX(D.Reddito)
FROM(
    SELECT P.Reddito
    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) = 2011
    GROUP BY P.CodFiscale, P.Reddito
    HAVING COUNT(*) = 3 AND COUNT(DISTINCT M.Specializzazione) = 3
) AS D;
```

Esercizio 4

Creare un vincolo di integrità generico (mediante un trigger) per impedire che un medico possa visitare mensilmente più di due volte lo stesso paziente, qualora all'atto delle due visite già effettuate in un dato mese dal medico sul paziente, quest'ultimo non fosse affetto da alcuna patologia.

```
DELIMITER $$
CREATE TRIGGER VincoloVisiteMensiliMedicoPaziente
BEFORE INSERT ON Visita
FOR EACH ROW
BEGIN
    SET @NumeroVisiteMesePazienteSano =
        (SELECT COUNT(*)
         FROM Visita V
         WHERE V.Medico = NEW.Medico
             AND V.Paziente = NEW.Paziente
             AND MONTH(V.Data) = MONTH(CURRENT_DATE)
             AND YEAR(V.Data) = YEAR(CURRENT_DATE)
             AND NOT EXISTS (SELECT *
                             FROM Esordio E
                             WHERE E.Paziente = NEW.Paziente
                                 AND ((
                                     E.DataEsordio <= V.Data
                                     AND E.DataGuarigione IS NOT NULL
                                     AND E.DataGuarigione > V.Data
                                 )
                                OR
                                 (E.DataEsordio <= V.Data
                                  AND E.DataGuarigione IS NULL
                                 )
                             ))
        );
    IF @NumeroVisiteMesePazienteSano = 2 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Visita non inseribile!';
    END IF;
END $$
DELIMITER ;
```

A.A. precedente:

Indicare il numero di medici che hanno visitato almeno un paziente tre volte nello stesso mese benché tale paziente, nelle prime due visite, non fosse affetto da alcuna patologia.

```
SELECT COUNT(DISTINCT V1.Medico)
FROM Visita V1
WHERE 2 = (SELECT COUNT(*)
           FROM Visita V
           WHERE V.Data < V1.Data
                 AND V.Medico = V1.Medico
                 AND V.Paziente = V1.Paziente
                 AND MONTH(V.Data) = MONTH(V1.Data)
                 AND NOT EXISTS (SELECT *
                                FROM Esordio E
                                WHERE E.Paziente = V.Paziente
                                       AND E.DataEsordio <= V.Data
                                       AND (E.DataGuarigione > V.Data
                                            OR E.DataGuarigione IS NULL))
           )
);
```

Esercizio 5

Considerato ciascun farmaco per la cura di patologie gastroenterologiche, indicato per più di una patologia, ma di fatto assunto per curare un'unica patologia per oltre l'80% delle terapie basate su di esso iniziate negli ultimi dieci anni, mantenere nella tabella INDICAZIONE la sola indicazione del farmaco considerato riguardante tale unica patologia, eliminando tutte le altre.

Spiegazione del procedimento logico adottato nella soluzione proposta

Considerato un farmaco F , si denota con $I(F, P_i)$, l'indicazione del farmaco F per la patologia P_i . Per ogni farmaco F esiste un totale $T(F)$ di terapie basate su di esso. Inoltre, per ciascuna patologia P_i per la quale il farmaco F è indicato, esiste un sottotale $S(F, P_i)$ che esprime il numero di terapie basate sul farmaco F per curare la patologia P_i . Naturalmente, dato un farmaco F , si ha che $\sum_i S(F, P_i) = T(F)$.

I farmaci d'interesse sono i soli per i quali esiste almeno un'indicazione per patologie gastroenterologiche, indicati anche per la cura di almeno un'altra patologia, sia essa gastroenterologica o no. Questi farmaci sono recuperati dalla vista *FarmaciGastroPiuPatologie*, l'insieme di farmaci individuato dalla quale è indicato nel seguito con GPP . Considerato ciascun farmaco $F \in GPP$, la vista *TotaliTerapieFarmaco* contiene il totale $T(F)$ di ognuno di essi, e la vista *TotaliTerapieFarmacoPatologia* contiene tutti i relativi sottotali $S(F, P_i)$.

Fissato un farmaco $F \in GPP$, si costruisce la vista *TotaliSubtotali* definita dalla seguente espressione in algebra relazionale, dove si assume di proiettare tutto lo schema, omettendo così l'operatore Π :

$$TotaliSubtotali = TotaliTerapieFarmacoPatologia \bowtie_{(S(F, P_i) > 0.8 * T(F))} TotaliTerapieFarmaco.$$

In questo modo, mediante la vista *TotaliSubtotali*, dato un farmaco $F \in GPP$, è possibile individuare quali sottotali $S(F, P_i)$ superano la percentuale target, cioè $0.8 * T(F)$: tali sottotali costituiranno tuple joinabili. D'altra parte, considerando sempre lo stesso farmaco F , i sottotali che non raggiungono la percentuale target sono completati a destra da valori *NULL*.

Se un dato farmaco F , fra i suoi sottotali $S(F, P_i)$, ne ha esattamente uno che supera la percentuale target (joinabile), ogni altro sottotale che non la supera (non joinabile, ma mantenuto dal join esterno sinistro), è relativo a un'indicazione $I(F, P_i)$ del farmaco F , oggetto dell'eliminazione richiesta.

```

CREATE OR REPLACE VIEW FarmaciGastroPiuPatologie AS
SELECT I1.Farmaco
FROM Indicazione I1 INNER JOIN Patologia P ON I1.Patologia = P.Nome
WHERE EXISTS (SELECT *
              FROM Indicazione I2 INNER JOIN Patologia P2
              ON I2.Patologia = P2.Nome
              WHERE I2.Farmaco = I1.Farmaco
              AND P2.SettoreMedico = 'Gastroenterologia')
GROUP BY I1.Farmaco
HAVING COUNT(DISTINCT I1.Patologia) > 1;

```

```

CREATE OR REPLACE VIEW TotaliTeraPieFarmacoPatologia AS
SELECT T.Farmaco, T.Patologia, COUNT(*) AS NumeroTeraPieFarmacoPatologia
FROM Terapia T NATURAL JOIN FarmaciGastroPiuPatologie
WHERE YEAR(T.DataInizioTerapia) > YEAR(CURRENT_DATE)-10
GROUP BY T.Farmaco, T.Patologia;
CREATE OR REPLACE VIEW TotaliTeraPieFarmaco AS
SELECT TTFP.Farmaco,
       SUM(TTFP.NumeroTeraPieFarmacoPatologia) AS TotaleTeraPie
FROM TotaliTeraPieFarmacoPatologia TTFP
GROUP BY TTFP.Farmaco;

```

```

CREATE OR REPLACE VIEW TotaliSubtotali AS
SELECT TTFP.*, TTF.TotaleTeraPie
FROM TotaliTeraPieFarmacoPatologia TTFP
LEFT OUTER JOIN
TotaliTeraPieFarmaco TTF
ON (TTFP.Farmaco = TTF.Farmaco AND
    TTFP.NumeroTeraPieFarmacoPatologia > 0.8*TTF.TotaleTeraPie);

```

```

DELETE FROM Indicazione
WHERE (Farmaco, Patologia) IN(
    SELECT TS.Farmaco, TS.Patologia
    FROM TotaliSubtotali TS
    WHERE 1 = (SELECT COUNT(*)
              FROM TotaliSubtotali TS2
              WHERE TS2.Farmaco = TS.Farmaco
              AND TS2.TotaleTeraPie IS NOT NULL)
    AND TS.TotaleTeraPie IS NULL); /*subtotali al di sotto del target*/

```