

SOLUZIONI TURNO 1, TURNO 2 E A.A. PRECEDENTE

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)
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 (uguale per entrambi i turni)

Indicare nome e cognome dei pazienti che hanno contratto tutte le patologie.

```
SELECT Nome, Cognome
FROM Paziente
WHERE CodFiscale IN (
    SELECT Paziente
    FROM Esordio E
    GROUP BY Paziente
    HAVING COUNT(DISTINCT Patologia) = (SELECT COUNT(*)
                                         FROM Patologia));
```

Esercizio 2 (turno 1 solo keyword in celeste; turno 2 aggiungere keyword in verde e cambiare il nome della vista col nome fra parentesi quadre)

Turno 1:

Indicare nome e cognome del paziente visitato più volte mentre era affetto da almeno una patologia. Se più pazienti rispettano la suddetta condizione, indicarli tutti.

Turno 2:

Indicare nome e cognome del paziente visitato più volte mentre non era affetto da patologie. Se più pazienti rispettano la suddetta condizione, indicarli tutti.

```
CREATE OR REPLACE VIEW VisiteDuranteMalattia[VisiteSano] AS
SELECT V.Paziente, COUNT(*) AS QuanteVisite
FROM Visita V
WHERE NOT EXISTS ( SELECT *
                   FROM Esordio E
                   WHERE E.Paziente = V.Paziente
```

```

        AND((V.Data > E.DataEsordio AND
              E.DataGuarigione IS NULL)
        OR
        (E.DataGuarigione IS NOT NULL AND
        (V.Data > E.DataEsordio AND
        V.Data < E.DataGuarigione))
        )
    )
GROUP BY V.Paziente;

SELECT Nome, Cognome
FROM Paziente
WHERE CodFiscale IN(
    SELECT VDM[VS].Paziente
    FROM VisiteDuranteMalattia[VisiteSano] VDM[VS]
    WHERE VDM[VS].QuanteVisite =
        (SELECT MAX(QuanteVisite)
        FROM VisiteDuranteMalattia[VisiteSano]));

```

Esercizio 3 (turno 1 keyword solo in celeste; turno 2 aggiungere la parte in verde;
a.a. precedente aggiungere la parte in verde e le condizioni fra parentesi quadre)

Turno 1:

Indicare il nome delle patologie contratte esclusivamente dopo il compimento del sessantesimo anno di età.

Turno 2:

Calcolare il numero di patologie contratte esclusivamente dopo il compimento del sessantesimo anno di età.

A.A. precedente:

Indicare il nome delle patologie contratte esclusivamente da pazienti di Pisa dopo il compimento del sessantesimo anno di età.

```

SELECT COUNT(DISTINCT Nome)
FROM Patologia PA
WHERE PA.Nome NOT IN(
    SELECT Patologia
    FROM Esordio E INNER JOIN Paziente P
    ON E.Paziente = P.CodFiscale
    WHERE NOT(
        (YEAR(E.DataEsordio)-YEAR(P.DataNascita) = 60
        [AND P.Citta = "Pisa"]
        AND( (MONTH(E.DataEsordio) = MONTH(P.DataNascita)
        AND DAY(E.DataEsordio) > DAY(P.DataNascita))
        OR
        (MONTH(E.DataEsordio) > MONTH(P.DataNascita))
        )
        )
    )
    OR
    (YEAR(E.DataEsordio)-YEAR(P.DataNascita) > 60
    [AND P.Citta = "Pisa"])
    )
);

```

Esercizio 4 (turno 2, sostituire le condizioni nei due WHERE con quelle fra parentesi quadre)

Turno 1:

Per ciascun settore medico, indicarne il nome e il costo totale dei farmaci oggetto di terapie effettuate nel triennio 2008-2010 per curare patologie contratte per la prima volta nello stesso periodo. Al costo dei farmaci sottrarre la percentuale di esenzione, ove prevista.

Turno 2:

Per ciascun settore medico, indicarne il nome e il costo totale dei farmaci oggetto di terapie effettuate cinque anni fa per curare patologie contratte per la prima volta nello stesso periodo. Al costo dei farmaci sottrarre la percentuale di esenzione, ove prevista.

```
SELECT PA.SettoreMedico,
       SUM(((DATEDIFF(T.DataFineTerapia, T.DataInizioTerapia)*
                    T.Posologia)/F.Pezzi)*(F.Costo - F.Costo*(
                    PA.PercEsenzione/100))) AS CostoTotale
FROM Terapia T INNER JOIN Patologia PA ON T.Patologia = PA.Nome
INNER JOIN Farmaco F ON T.Farmaco = F.NomeCommerciale
WHERE T.DataFineTerapia IS NOT NULL
      AND YEAR(T.DataEsordio) >= 2008           [=YEAR(CURRENT_DATE)-5]
      AND YEAR(T.DataInizioTerapia) >= 2008     [=YEAR(CURRENT_DATE)-5]
      AND YEAR(T.DataFineTerapia) <= 2010       [=YEAR(CURRENT_DATE)-5]
      AND NOT EXISTS(
        SELECT *
        FROM Esordio E
        WHERE E.Paziente = T.Paziente
              AND E.Patologia = T.Patologia
              AND YEAR(E.DataEsordio) < 2008    [=YEAR(CURRENT_DATE)-5])
GROUP BY PA.SettoreMedico;
```

Esercizio 5 (uguale per entrambi i turni; il trigger è escluso per l'anno accademico precedente)

Creare e popolare una tabella INTERVALLO(CodiceFiscale, Specializzazione, Giorni) contenente, per ogni paziente, il periodo di tempo medio fra una visita e la successiva della stessa specializzazione, espresso in giorni. Scrivere poi un trigger che mantenga aggiornata tale tabella.

```
CREATE TABLE Intervallo (
  CodiceFiscale CHAR(100) NOT NULL,
  Specializzazione CHAR(100) NOT NULL,
  Giorni INT(11) NOT NULL,
  PRIMARY KEY (CodiceFiscale, Specializzazione)
) ENGINE=InnoDB;

INSERT INTO Intervallo
SELECT D.Specializzazione, D.Paziente, AVG(Giorni)
FROM (
  SELECT M1.Specializzazione, VPrima.Paziente,
         DATEDIFF(VDopo.Data, VPrima.Data) AS Giorni
  FROM (Visita VPrima INNER JOIN Medico M1
        ON VPrima.Medico = M1.Matricola)
  INNER JOIN
    (Visita VDopo INNER JOIN Medico M2
     ON VDopo.Medico = M2.Matricola)
  ON M1.Specializzazione = M2.Specializzazione
  AND VPrima.Data < VDopo.Data
```

```

        AND VPrima.Paziente = VDopo.Paziente
WHERE NOT EXISTS(
    SELECT *
    FROM Visita VMezzo INNER JOIN Medico M3
        ON VMezzo.Medico = M3.Matricola
    WHERE M3.Specializzazione = M1.Specializzazione
        AND VMezzo.Data > VPrima.Data
        AND VMezzo.Data < VDopo.Data
        AND VMezzo.Paziente = VPrima.Paziente)
) AS D
GROUP BY D.Specializzazione, D.Paziente;

DELIMITER $$
CREATE TRIGGER AggiornaIntervalloVisite
AFTER INSERT INTO Visita FOR EACH ROW
BEGIN
    SET @specializzazioneVisita = (SELECT Specializzazione
                                    FROM Medico
                                    WHERE Matricola = NEW.Medico);

    UPDATE Intervallo
    SET Giorni = (
        SELECT AVG(Giorni)
        FROM(
            SELECT DATEDIFF(VDopo.Data, VPrima.Data) AS Giorni
            FROM (Visita VPrima INNER JOIN Medico M1
                ON VPrima.Medico = M1.Matricola)
                INNER JOIN
                (Visita VDopo INNER JOIN Medico M2
                ON VDopo.Medico = M2.Matricola)
            ON M1.Specializzazione = @specializzazioneVisita
                AND M2.Specializzazione = @specializzazioneVisita
                AND VPrima.Data < VDopo.Data
                AND VPrima.Paziente = NEW.Paziente
                AND VDopo.Paziente = NEW.Paziente
            WHERE NOT EXISTS(
                SELECT *
                FROM Visita VMezzo INNER JOIN Medico M3
                    ON VMezzo.Medico = M3.Matricola
                WHERE M3.Specializzazione = @specializzazioneVisita
                    AND VMezzo.Data > VPrima.Data
                    AND VMezzo.Data < VDopo.Data
                    AND VMezzo.Paziente = VPrima.Paziente
            )
        ) AS D
    WHERE CodiceFiscale = NEW.Paziente
        AND Specializzazione = NEW.Specializzazione);
END $$
DELIMITER ;

```