

## 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 il nome commerciale di ciascun farmaco utilizzato da almeno un paziente per curare tutte le patologie per le quali è indicato.

```
SELECT DISTINCT T.Farmaco
FROM Terapia T
GROUP BY T.Paziente, T.Farmaco
HAVING COUNT(DISTINCT T.Patologia) = (SELECT COUNT(*)
                                       FROM Indicazione I
                                       WHERE I.Farmaco = T.Farmaco);
```

### Esercizio 2 (7 punti)

Scrivere una query che restituisca il numero di pazienti visitati solo da medici specializzati in cardiologia o neurologia, almeno due volte per ciascuna delle due specializzazioni. Si scriva la query senza usare viste.

```
SELECT FLOOR(COUNT(*)/2) AS NumeroPazienti
FROM (
    SELECT Specializzazione, Paziente
    FROM Visita V INNER JOIN Medico M ON V.Medico = M.Matricola
    WHERE V.Paziente NOT IN (
        SELECT V2.Paziente
        FROM Visita V2 INNER JOIN Medico M2
            ON V2.Medico = M2.Matricola
        WHERE M2.Specializzazione <> 'Cardiologia'
            AND M2.Specializzazione <> 'Neurologia')
    GROUP BY M.Specializzazione, V.Paziente
    HAVING COUNT(*) >= 2
) AS D
GROUP BY Paziente
HAVING COUNT(*) = 2;
```

*A.A. precedente:*

Scrivere una query che restituisca il numero di pazienti visitati solo da medici specializzati in cardiologia, almeno due volte.

```
SELECT COUNT(*)
FROM (
    SELECT V.Paziente
    FROM Visita V INNER JOIN Medico M ON V.Medico = M.Matricola
    WHERE V.Paziente NOT IN (
        SELECT V2.Paziente
        FROM Visita V2 INNER JOIN Medico M2 ON V2.Medico = M2.Matricola
        WHERE M2.Specializzazione <> 'Cardiologia')
    GROUP BY V.Paziente
    HAVING COUNT(*) >= 2
) AS D;
```

### Esercizio 3 (9 punti)

Creare una business rule che permetta di inserire un nuovo farmaco *F* e le relative indicazioni, qualora non vi siano già più di due farmaci di cui almeno uno basato sullo stesso principio attivo, aventi, ciascuno, un'indicazione per una stessa patologia per la quale *F* è indicato. Supporre che per prima cosa sia inserito il farmaco, dopodiché siano inserite le varie indicazioni.

/\*

Nella soluzione proposta si suppone che il farmaco *F* sia incondizionatamente inserito nella tabella Farmaco. Dopodiché, all'atto dell'inserimento di un'indicazione che viola il vincolo alla base della business rule, il trigger rimuove sia il farmaco che tutte le sue indicazioni dal database.

\*/

```
DELIMITER $$
CREATE TRIGGER RuleInserimentoFarmaco
BEFORE INSERT ON Indicazione FOR EACH ROW
BEGIN
    DECLARE farmaciStessaPatologia INTEGER DEFAULT 0;
    DECLARE farmaciStessoPrincipio INTEGER DEFAULT 0;

    SELECT COUNT(*) INTO farmaciStessaIndicazione
    FROM Indicazione I
    WHERE I.Patologia = NEW.Patologia;

    SELECT COUNT(*) INTO farmaciStessoPrincipio
    FROM Farmaco F INNER JOIN Indicazione I
    ON F.NomeCommerciale = I.Farmaco
    WHERE F.NomeCommerciale <> NEW.Farmaco
    AND I.Patologia = NEW.Patologia
    AND F.PrincipioAttivo = (SELECT F2.PrincipioAttivo
                             FROM Farmaco F2
                             WHERE F2.NomeCommerciale = NEW.Farmaco)
    ;

    IF farmaciStessaIndicazione > 2 AND farmaciStessoPrincipio >= 1 THEN
    BEGIN
        DELETE FROM Indicazione
        WHERE Farmaco = NEW.Farmaco;
```

```

DELETE FROM Farmaco
WHERE NomeCommerciale = NEW.Farmaco;

SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Farmaco e indicazioni rimosse.';

END;
END; $$
DELIMITER ;

```

#### Esercizio 4 (10 punti)

Un paziente effettua una visita di accertamento quando, dopo essere stato visitato inizialmente da un medico, desidera avere anche il parere di un altro medico della stessa specializzazione, dal quale si fa visitare senza iniziare, nel frattempo, alcuna terapia per la cura di patologie inerenti tale specializzazione. In generale, dopo una visita iniziale, un paziente può effettuare più visite di accertamento, posticipando ulteriormente l'inizio della terapia. Creare una materialized view ACCERTAMENTO contenente codice fiscale, nome e cognome dei pazienti che, nell'ultimo trimestre, relativamente ad almeno una visita iniziale, hanno effettuato una o più visite di accertamento, quante ne hanno effettuate per ogni visita iniziale, e il cognome del medico che ha effettuato tale visita iniziale. Gestire la materialized view mediante deferred refresh con cadenza trimestrale.

```

CREATE OR REPLACE VIEW VisitaTerapiaProssima AS
SELECT V.Paziente, V.Medico, V.Data, T.Patologia, T.DataEsordio, T.Farmaco,
       T.DataInizioTerapia
FROM Visita V INNER JOIN Terapia T USING(Paziente)
      INNER JOIN Patologia PA ON PA.Nome = T.Patologia
      INNER JOIN Medico M ON M.Matricola = V.Medico
WHERE V.Data BETWEEN MONTH(CURRENT_DATE) AND MONTH(CURRENT_DATE) - 3
      AND PA.SettoreMedico = M.Specializzazione
      AND DATEDIFF(T.DataInizioTerapia, V.Data) =
      (SELECT MIN(DATEDIFF(T2.DataInizioTerapia, V2.Data))
       FROM Visita V2 INNER JOIN Terapia T2 USING(Paziente)
              INNER JOIN Patologia PA2 ON PA2.Nome = T2.Patologia
              INNER JOIN Medico M2 ON M2.Matricola = V2.Medico
       WHERE V2.Paziente = V.Paziente
              AND V2.Medico = V.Medico
              AND V2.Data = V.Data
              AND T2.DataInizioTerapia > V2.Data
              AND PA2.SettoreMedico = M2.Specializzazione
      );

```

```

CREATE OR REPLACE VIEW TerapiaAccertamenti AS
SELECT Paziente, Patologia, DataEsordio, Farmaco, DataInizioTerapia
       COUNT(*) AS TotaleAccertamenti, MIN(Data) AS DataVisitaIniziale
FROM VisitaTerapiaProssima
GROUP BY Paziente, Patologia, DataEsordio, Farmaco, DataInizioTerapia
HAVING COUNT(DISTINCT Medico) = COUNT(*);

```

```

CREATE TABLE Accertamento
(
  IDAccertamento INT(11) NOT NULL AUTO_INCREMENT,
  CodFiscale CHAR(16) NOT NULL,
  Nome CHAR(100) NOT NULL,
  Cognome CHAR(100) NOT NULL,

```

```

TotaleAccertamenti INT(11) NOT NULL,
MedicoVisitaIniziale CHAR(100) NOT NULL,
PRIMARY KEY (IDAccertamento)
)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

DROP PROCEDURE IF EXISTS build_accertamento;

DELIMITER $$
CREATE PROCEDURE build_accertamento(OUT esito)
BEGIN
    DECLARE esito INTEGER DEFAULT 0;
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        SET esito = 1;
        SELECT "Errore: materialized view non aggiornata.";
    END;

    TRUNCATE TABLE Accertamento;

    INSERT INTO Accertamento(CodFiscale, Nome, Cognome,
                             TotaleAccertamenti,MedicoVisitaIniziale)
    SELECT P.CodFiscale, P.Nome, P.Cognome, TA.TotaleAccertamenti,
           M.Cognome AS CognomeMedicoVisitaIniziale
    FROM TerapiaAccertamenti TA NATURAL JOIN VisitaTerapiaProssima VTP
    INNER JOIN Paziente P ON TA.Paziente = P.CodFiscale
    INNER JOIN Medico M ON TA.Medico = M.Matricola
    WHERE VTP.Data = TA.DataVisitaIniziale;

END; $$

DROP EVENT IF EXISTS refresh_accertamento;
CREATE EVENT refresh_accertamento
ON SCHEDULE EVERY 3 MONTH
DO
BEGIN
    SET @esito = 0;
    CALL build_accertamento(@esito);
    IF @esito = 1 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT "Errore refresh Accertamento da temporal trigger.";
    END IF;
END; $$

DELIMITER ;

```

#### *A.A. precedente:*

Un paziente effettua una visita di accertamento quando, dopo essere stato visitato inizialmente da un medico, desidera avere anche il parere di un altro medico della stessa specializzazione, dal quale si fa visitare senza iniziare, nel frattempo, alcuna terapia per la cura di patologie inerenti tale specializzazione. In generale, dopo una visita iniziale, un paziente può effettuare più visite di accertamento, posticipando ulteriormente l'inizio della terapia. Creare e popolare una tabella ACCERTAMENTO contenente codice fiscale, nome e cognome dei pazienti che, nell'ultimo trimestre, relativamente ad almeno una visita iniziale, hanno effettuato una o più visite di accertamento.

```

CREATE OR REPLACE VIEW VisitaTerapiaProssima AS
SELECT V.Paziente, V.Medico, V.Data, T.Patologia, T.DataEsordio, T.Farmaco,
       T.DataInizioTerapia
FROM Visita V INNER JOIN Terapia T USING(Paziente)
       INNER JOIN Patologia PA ON PA.Nome = T.Patologia
       INNER JOIN Medico M ON M.Matricola = V.Medico
WHERE V.Data BETWEEN MONTH(CURRENT_DATE) AND MONTH(CURRENT_DATE) - 3
       AND PA.SettoreMedico = M.Specializzazione
       AND DATEDIFF(T.DataInizioTerapia, V.Data) =
       (SELECT MIN(DATEDIFF(T2.DataInizioTerapia, V2.Data))
        FROM Visita V2 INNER JOIN Terapia T2 USING(Paziente)
                INNER JOIN Patologia PA2 ON PA2.Nome = T2.Patologia
                INNER JOIN Medico M2 ON M2.Matricola = V2.Medico
        WHERE V2.Paziente = V.Paziente
              AND V2.Medico = V.Medico
              AND V2.Data = V.Data
              AND T2.DataInizioTerapia > V2.Data
              AND PA2.SettoreMedico = M2.Specializzazione
        );

```

```

CREATE OR REPLACE VIEW PazientiConAccertamenti AS
SELECT DISTINCT Paziente
FROM VisitaTerapiaProssima
GROUP BY Paziente, Patologia, DataEsordio, Farmaco, DataInizioTerapia
HAVING COUNT(DISTINCT Medico) = COUNT(*);

```

```

CREATE TABLE Accertamento
(
  CodFiscale CHAR(16) NOT NULL,
  Nome CHAR(100) NOT NULL,
  Cognome CHAR(100) NOT NULL,
  PRIMARY KEY (CodFiscale)
)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

INSERT INTO Accertamento
SELECT P.CodFiscale, P.Nome, P.Cognome
FROM PazientiConAccertamenti PCA NATURAL JOIN VisitaTerapiaProssima VTP
       INNER JOIN Paziente P ON PCA.Paziente = P.CodFiscale;

```