

BASI DI DATI – INGEGNERIA INFORMATICA
ESERCIZI D'ESAME SU TRIGGER ED EVENT

Esercizio 1

Modificare la tabella ESORDIO aggiungendo un attributo EsordiPrecedenti contenente il numero di esordi precedenti di patologie dello stesso paziente relative allo stesso settore medico, curati con successo. L'attributo EsordiPrecedenti deve essere aggiornato ogni cinque giorni, a partire dal 1° Luglio 2015 alle ore 6:00 del mattino.

```
ALTER TABLE Esordio
ADD COLUMN EsordiPrecedenti INTEGER DEFAULT NULL AFTER Cronica;
```

```
DELIMITER $$
```

```
DROP EVENT IF EXISTS aggiorna_esordi_precedenti $$
```

```
CREATE EVENT aggiorna_esordi_precedenti
ON SCHEDULE EVERY 5 DAY
STARTS '2015-07-01 06:00:00'
DO
BEGIN
    DECLARE paziente CHAR(100) DEFAULT '';
    DECLARE patologia CHAR(100) DEFAULT '';
    DECLARE data_esordio DATE DEFAULT NULL;
    DECLARE esordi_precedenti INTEGER DEFAULT 0;
    DECLARE finito INTEGER DEFAULT 0;

    DECLARE esordi CURSOR FOR
    SELECT
        E.Paziente,
        E.Patologia,
        E.DataEsordio,
        COUNT(*) AS EsordiPrecedenti
    FROM
        (
            Esordio E
            INNER JOIN
            Patologia P1 ON E.Patologia = P1.Nome
        )
        INNER JOIN
        (
            Esordio EPREC
            INNER JOIN
            Patologia P2 ON EPREC.Patologia = P2.Nome
        )
        USING(Paziente, SettoreMedico)
```

```

WHERE
    EPREC.DataEsordio < E.DataEsordio
    AND EPREC.DataGuarigione IS NOT NULL
GROUP BY
    E.Paziente,
    E.Patologia,
    E.DataEsordio;

DECLARE CONTINUE HANDLER FOR NOT FOUND
SET finito = 1;

OPEN esordi;

scan: LOOP

    FETCH esordi INTO paziente,
                patologia,
                data_esordio,
                esordi_precedenti;

    IF finito = 1 THEN
        LEAVE scan;
    END IF;

    UPDATE
        Esordio E
    SET
        E.EsordiPrecedenti = esordi_precedenti
    WHERE
        E.Paziente = paziente
        AND E.Patologia = patologia
        AND E.DataEsordio = data_esordio;

END LOOP scan;

END $$

DELIMITER ;

```

Esercizio 2

Scrivere un trigger che impedisca l'inserimento di due terapie consecutive per lo stesso paziente qualora siano caratterizzate dallo stesso farmaco e la più recente abbia una posologia superiore al doppio rispetto alla precedente.

```

DROP TRIGGER IF EXISTS vieta_terapie_consecutive;

```

```

DELIMITER $$

CREATE TRIGGER vieta_terapie_consecutive
BEFORE INSERT ON Terapia
FOR EACH ROW
BEGIN

    DECLARE ultimo_farmaco CHAR(100);
    DECLARE ultima_posologia INTEGER DEFAULT 0;

    SELECT T1.Farmaco,
           T1.Posologia
           INTO ultimo_farmaco, ultima_posologia
    FROM Terapia T1
    WHERE T1.Paziente = NEW.Paziente
          AND T1.DataFineTerapia =
            (
              SELECT MAX(T2.DataFineTerapia)
              FROM Terapia T2
              WHERE T2.Paziente = T1.Paziente
            );

    IF ultimo_farmaco = NEW.Farmaco
       AND NEW.Posologia > 2*ultima_posologia
    THEN
        SIGNAL SQLSTATE 45000
        SET MESSAGE_TEXT = 'Terapia non consentita';
    END IF;

END $$

DELIMITER ;

```

Esercizio 3

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.
*/

```

```

DROP TRIGGER IF EXISTS rule_inserimento_farmaco;

DELIMITER $$

CREATE TRIGGER rule_inserimento_farmaco
BEFORE INSERT ON Indicazione FOR EACH ROW
BEGIN
    DECLARE farmaci_stessa_patologia INTEGER DEFAULT 0;
    DECLARE farmaci_stesso_principio INTEGER DEFAULT 0;

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

    SELECT COUNT(*) INTO farmaci_stesso_principio
    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
            );

    -- almeno due farmaci indicati per la stessa patologia
    -- di cui almeno uno basato sullo stesso principio attivo
    IF farmaci_stessa_patologia > 2
        AND farmaci_stesso_principio >= 1
    THEN
        BEGIN
            -- cancello tutte le altre indicazioni eventualmente
            -- già inserite prima di questa (bloccata dal trigger)
            DELETE FROM Indicazione
            WHERE Farmaco = NEW.Farmaco;

            -- cancello il farmaco
            DELETE FROM Farmaco
            WHERE NomeCommerciale = NEW.Farmaco;

            -- sollevo un segnale sqlstate 45000 che fa sì che
            -- l'indicazione che si stava tentando di inserire
            -- venga bloccata (la insert su Indicazione che
            -- ha scatenato il trigger viene quindi abortita

```

```

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

```

```

DELIMITER ;

```

Esercizio 6

Scrivere un evento che impedisca alle terapie in corso di protrarsi oltre un mese qualora, durante la terapia, il paziente abbia contratto al massimo tre patologie di cui almeno una a carico della parte del corpo oggetto della terapia. Spostare tali terapie in una tabella TERAPIEINTERROTTE, avente lo schema (NomePaziente, CognomePaziente, Farmaco, DataInizio, DataInterruzione).

```

CREATE TABLE TerapieInterrotte(
    NomePaziente CHAR(100) NOT NULL,
    CognomePaziente CHAR(100) NOT NULL,
    Farmaco CHAR(100) NOT NULL,
    DataInizio DATE NOT NULL,
    DataFine DATE NOT NULL,
    PRIMARY KEY (NomePaziente, CognomePaziente, Farmaco, DataInizio)
);

```

```

DROP EVENT IF EXISTS interrompi_terapie;

```

```

DELIMITER $$

```

```

CREATE EVENT interrompi_terapie ON SCHEDULE EVERY 1 DAY
DO
BEGIN

```

```

    DECLARE paziente CHAR(100) DEFAULT '';
    DECLARE patologia CHAR(100) DEFAULT '';
    DECLARE data_esordio DATE DEFAULT NULL;
    DECLARE farmaco CHAR(100) DEFAULT '';
    DECLARE data_inizio_terapia DATE DEFAULT NULL;

```

```

    DECLARE finito INTEGER DEFAULT 0;

```

```

    DECLARE cognome_paziente CHAR(100) DEFAULT '';
    DECLARE nome_paziente CHAR(100) DEFAULT '';

```

```

    DECLARE terapie_da_interrompere CURSOR FOR
    SELECT T.Paziente,
           T.Patologia,
           T.DataEsordio,
           T.Farmaco,
           T.DataInizioTerapia

```

```

        INTO paziente, patologia, data_esordio,
            farmaco, data_inizio_terapia
FROM Terapia T
    INNER JOIN
        Patologia PA1 ON T.Patologia = PA1.Nome
WHERE T.DataInizioTerapia + INTERVAL 1 MONTH = CURRENT_DATE
    AND 3 >= (
        SELECT COUNT(*)
        FROM Esordio E
        WHERE E.Paziente = T.Paziente
            AND E.DataEsordio BETWEEN
                T.DataEsordio AND CURRENT_DATE
    )
AND EXISTS
(
    SELECT *
    FROM Esordio E
        INNER JOIN
            Patologia PA2 ON E.Patologia = PA2.Nome
    WHERE E.Paziente = T.Paziente
        AND E.DataEsordio BETWEEN
            T.DataEsordio AND CURRENT_DATE
        AND PA2.ParteCorpo = PA1.ParteCorpo);

DECLARE CONTINUE HANDLER FOR NOT FOUND
SET finito = 1;

OPEN terapie_da_interrompere;

scan: LOOP

    FETCH esordi INTO paziente, patologia, data_esordio,
        farmaco, data_inizio_terapia;

    IF finito = 1 THEN
        LEAVE scan;
    END IF;

    SELECT P.Nome, P.Cognome
        INTO nome_paziente, cognome_paziente
    FROM Paziente P
    WHERE P.CodFiscale = paziente;

    INSERT INTO TerapieInterrotte
    VALUES(nome_paziente, cognome_paziente, farmaco,
        data_inizio_terapia, CURRENT_DATE);

```

```
DELETE FROM Terapia T
WHERE T.Paziente = paziente
      AND T.Patologia = patologia
      AND T.DataEsordio = data_esordio
      AND T.Farmaco = farmaco
      AND T.DataInizioTerapia = data_inizio_terapia;
```

```
CLOSE terapie_da_interrompere;
```

```
END $$
```

```
DELIMITER ;
```