

```
CREATE OR REPLACE VIEW PazientiSettoreMedicoNimesulide AS
SELECT E.Paziente, P.SettoreMedico
FROM Esordio E NATURAL JOIN Terapia T
     INNER JOIN Patologia P ON E.Patologia = P.Nome
     INNER JOIN Farmaco F ON T.Farmaco = F.NomeCommerciale
WHERE F.PrincipioAttivo = 'Nimesulide'
GROUP BY E.Paziente, P.SettoreMedico
HAVING COUNT(DISTINCT T.Farmaco) = (SELECT COUNT(*)
                                   FROM Farmaco
                                   WHERE PrincipioAttivo = 'Nimesulide');
```

```

SELECT P.Nome, P.Cognome
FROM PazientiSettoreMedicoNimesulide PSMN INNER JOIN Paziente P
    ON PSMN.Paziente = P.CodFiscale
WHERE PSMN.SettoreMedico = 'Ortopedia'
    AND NOT EXISTS (SELECT *
                     FROM PazientiSettoreMedicoNimesulide PSMN2
                     WHERE PSMN2.Paziente = PSMN.Paziente
                        AND PSMN2.SettoreMedico <> PSMN.SettoreMedico);

```

A.A. precedente:

Scrivere una query che restituisca nome e cognome dei pazienti che hanno assunto, man mano nella vita, tutti i farmaci a base di nimesulide.

```

SELECT Nome, Cognome
FROM Paziente
WHERE CodFiscale IN(
    SELECT E.Paziente
    FROM Esordio E NATURAL JOIN Terapia T
        INNER JOIN Patologia P ON E.Patologia = P.Nome
        INNER JOIN Farmaco F ON T.Farmaco = F.NomeCommerciale
    WHERE F.PrincipioAttivo = 'Nimesulide'
    GROUP BY E.Paziente
    HAVING COUNT(DISTINCT T.Farmaco) = (SELECT COUNT(*)
                                       FROM Farmaco
                                       WHERE PrincipioAttivo = 'Nimesulide')
);

```

Esercizio 3

Scrivere una function che, ricevuto in ingresso il codice fiscale di un paziente, restituisca il suo stato attuale di salute *SS* ottenuto mediante l'espressione

$$SS = n \left(\sum_{i=1}^n w_i g_i \right)^{-1},$$

dove n è il numero di esordi attualmente in corso, g_i è la gravità con cui la patologia è stata contratta nello esordio i -esimo, e w_i è un coefficiente di penalizzazione pari a: 1 se l'esordio i -esimo non ha terapie fallite; 1.5 se l'esordio i -esimo ha da 1 a 2 terapie fallite; 2.5 se l'esordio i -esimo ha più di 3 terapie fallite.

```

DROP FUNCTION IF EXISTS stato_salute;

DELIMITER $$

CREATE FUNCTION stato_salute(codiceFiscale CHAR(16))
RETURNS DOUBLE NOT DETERMINISTIC
BEGIN
    -- risultato della function
    DECLARE statoSalute DOUBLE DEFAULT 0;

    -- parametro 'n'
    DECLARE esordiInCorso INTEGER DEFAULT 0;

    -- parametri caratterizzanti l'esordio 'i'
    DECLARE gravita INTEGER DEFAULT 0;
    DECLARE terapieFallite INTEGER DEFAULT 0;
    DECLARE w DOUBLE DEFAULT 0;
    DECLARE finito INTEGER DEFAULT 0;

```

```

DECLARE cursoreTerapie CURSOR FOR
  SELECT D2.Paziente, D2.Gravita,
         IFNULL(TerapieFallite, 0) AS TherapieFallite
  FROM(
    SELECT E.Paziente, E.Patologia, E.DataEsordio, E.Gravita,
           COUNT(*) AS TherapieFallite
    FROM Esordio E NATURAL JOIN Terapia T
    WHERE E.DataGuarigione IS NULL
           AND T.DataFineTerapia IS NOT NULL
    GROUP BY E.Paziente, E.Patologia, E.DataEsordio, E.Gravita
  ) AS D1
  NATURAL RIGHT OUTER JOIN (SELECT *
                             FROM Esordio E
                             WHERE E.DataGuarigione IS NULL) AS D2;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET finito = 1;

OPEN cursoreTerapie;

scan: LOOP

  FETCH cursoreTerapie INTO gravita, terapieFallite;

  IF finito = 1 THEN
    LEAVE scan;
  END IF;

  CASE
    WHEN terapieFallite = 0 THEN SET w = 1;
    WHEN terapieFallite BETWEEN 1 AND 2 THEN SET w = 1.5;
    ELSE SET w = 2.5;
  END CASE;

  SET statoSalute = statoSalute + (w*gravita);
  SET esordiInCorso = esordiInCorso + 1;

END LOOP scan;

CLOSE cursoreTerapie;

-- poiché l'espressione per il calcolo di SS è definita solamente
-- nel caso di esordi in corso, si sceglie di restituire il valore -1
-- se il paziente, parametro della function, è sano.

IF statoSalute <> 0 THEN
  SET statoSalute = esordiInCorso*(IFNULL(1/statoSalute, 0));
ELSE
  SET statoSalute = -1;
END IF;

RETURN statoSalute;
END $$

DELIMITER ;

```

A.A. precedente:

Scrivere una query che restituisca l'anno (o gli anni) in cui si sono verificati, complessivamente fra i pazienti di Pisa e Firenze, più del 30% degli esordi di rinite nel trimestre Marzo-Maggio rispetto al totale degli esordi della stessa patologia nello stesso trimestre dello stesso anno.

```
CREATE OR REPLACE VIEW ParzialiEsordiCitta AS
SELECT YEAR(E.DataEsordio) AS Anno, P.Citta, COUNT(*) AS NumeroEsordi
FROM Esordio E INNER JOIN Patologia PA ON E.Patologia = PA.Nome
    INNER JOIN Paziente P ON E.Paziente = P.CodFiscale
WHERE MONTH(E.DataEsordio) BETWEEN 3 AND 5
    AND PA.Nome = 'Rinite'
    AND (P.Citta = 'Pisa' OR P.Citta = 'Firenze')
GROUP BY YEAR(E.DataEsordio), P.Citta;

SELECT PEC.Anno
FROM ParzialiEsordiCitta PEC
GROUP BY PEC.Anno
HAVING SUM(PEC.NumeroEsordi) >
    0.3*(SELECT COUNT(*)
        FROM Esordio E INNER JOIN Patologia PA ON E.Patologia = PA.Nome
        WHERE YEAR(E.DataEsordio) = PEC.Anno
            AND MONTH(E.DataEsordio) BETWEEN 3 AND 5
            AND PA.Nome = 'Rinite');
```

Esercizio 4

Scrivere una query che restituisca l'anno (o gli anni) in cui si sono verificati, complessivamente fra i pazienti di Pisa e Firenze, più del 30% degli esordi di rinite nel trimestre Marzo-Maggio rispetto al totale degli esordi della stessa patologia nello stesso trimestre dello stesso anno, con il picco più alto di esordi raggiunto dai pazienti di Pisa.

```
CREATE OR REPLACE VIEW ParzialiEsordiCitta AS
SELECT YEAR(E.DataEsordio) AS Anno, P.Citta, COUNT(*) AS NumeroEsordi
FROM Esordio E INNER JOIN Patologia PA ON E.Patologia = PA.Nome
    INNER JOIN Paziente P ON E.Paziente = P.CodFiscale
WHERE MONTH(E.DataEsordio) BETWEEN 3 AND 5
    AND PA.Nome = 'Rinite'
    AND (P.Citta = 'Pisa' OR P.Citta = 'Firenze')
GROUP BY YEAR(E.DataEsordio), P.Citta;

SELECT PI.Anno
FROM (SELECT Anno, NumeroEsordi
    FROM ParzialiEsordiCitta
    WHERE Citta = 'Pisa') AS PI
LEFT OUTER JOIN
    (SELECT Anno, NumeroEsordi
    FROM ParzialiEsordiCitta
    WHERE Citta = 'Firenze') AS FI
USING (Anno)
WHERE PI.NumeroEsordi + IFNULL(FI.NumeroEsordi, 0) >
    0.3*(SELECT COUNT(*)
        FROM Esordio E INNER JOIN Patologia PA ON E.Patologia = PA.Nome
        WHERE YEAR(E.DataEsordio) = PI.Anno
            AND MONTH(E.DataEsordio) BETWEEN 3 AND 5
            AND PA.Nome = 'Rinite')
    AND PI.NumeroEsordi > IFNULL(FI.NumeroEsordi, 0);
```

A.A. precedente:

Aggiungere un attributo ridondante *StatoSalute* alla tabella PAZIENTE che contenga, dato un paziente, il relativo attuale stato di salute espresso come

$$SS = n \left(\sum_{i=1}^n w_i g_i \right)^{-1},$$

dove n è il numero di esordi attualmente in corso, g_i è la gravità con cui la patologia è stata contratta nello esordio i -esimo, e w_i è un coefficiente di penalizzazione pari a: 1 se l'esordio i -esimo non ha terapie fallite; 1.5 se l'esordio i -esimo ha da 1 a 2 terapie fallite; 2.5 se l'esordio i -esimo ha più di 3 terapie fallite.

```
ALTER TABLE Paziente
ADD COLUMN StatoSalute DOUBLE NOT NULL DEFAULT -1 AFTER Reddito;

CREATE OR REPLACE VIEW TotaliEsordi AS
SELECT E.Paziente, COUNT(*) AS EsordiInCorso
FROM Esordio E
WHERE E.DataGuarigione IS NULL
GROUP BY E.Paziente;

CREATE OR REPLACE VIEW DettaglioIncompletoEsordi AS
SELECT E.Paziente, E.Patologia, E.DataEsordio, E.Gravita,
       (SELECT CASE
          WHEN COUNT(*) = 0 THEN 1
          WHEN COUNT(*) BETWEEN 1 AND 2 THEN 1.5
          ELSE 2.5
        END) AS w
FROM Esordio E NATURAL JOIN Terapia T
WHERE E.DataGuarigione IS NULL
      AND T.DataFineTerapia IS NOT NULL
GROUP BY E.Paziente, E.Patologia, E.DataEsordio, E.Gravita;

CREATE OR REPLACE VIEW DettaglioEsordi AS
SELECT E.Paziente, E.Gravita, IFNULL(w, 1) as w
FROM DettaglioIncompletoEsordi DIG
      NATURAL RIGHT OUTER JOIN (SELECT *
                                FROM Esordio E
                                WHERE E.DataGuarigione IS NULL);

CREATE OR REPLACE VIEW StatiSalute
SELECT Paziente, EsordiInCorso*(1/SUM(w*Gravita)) AS StatoDiSalute
FROM TotaliEsordi NATURAL JOIN DettaglioEsordi
GROUP BY Paziente, EsordiInCorso;

-- Si suppone che un paziente sano, cioè senza alcun esordio in corso,
-- mantenga il valore default dell'attributo StatoSalute.
-- Di fatto, la formula per il calcolo di SS è definita solamente
-- se il paziente ha almeno un esordio in corso, quindi n>0.
-- L'esercizio non richiede di scrivere il codice che gestisce
-- l'evoluzione dell'attributo StatoSalute nel tempo.

UPDATE Paziente
SET StatoSalute = (SELECT StatoDiSalute
                  FROM StatiSalute
                  WHERE Paziente = CodFiscale);
```