Instituto Tecnológico de Costa Rica

Escuela de Computación

Bases de Datos II

Grupo 20

Tarea 5

Profesor (a):
Alberto Shum Chan

Estudiante (s):

José Adrián Amador Ávila - 2016101574
Pablo Jesús Mora Barrantes - 2019205110
Jose Andrés Vargas Serrano - 2019211290

Alajuela, I Semestre 2023

1. Creación del modelo relacional

```
CREATE TABLE hospital(
    id INTEGER NOT NULL PRIMARY KEY,
    nombre VARCHAR(30) NOT NULL,
    provincia VARCHAR(30) NOT NULL);

CREATE TABLE medico(
    id INTEGER NOT NULL PRIMARY KEY,
    cedula VARCHAR(30) NOT NULL,
    nombre VARCHAR(30) NOT NULL,
    primer_apellido VARCHAR(30) NOT NULL,
    direccion_provincia VARCHAR(30) NOT NULL);

CREATE TABLE medico_hospital(
    medico_id INTEGER NOT NULL,
    hospital_id INTEGER NOT NULL,
    FOREIGN KEY(medico_id) REFERENCES medico(id),
    FOREIGN KEY(hospital_id) REFERENCES hospital(id));

CREATE TABLE medico_especialidad(
    medico_id INTEGER NOT NULL,
    especialidad_id INTEGER NOT NULL,
    FOREIGN KEY(medico_id) REFERENCES medico(id),
    FOREIGN KEY(especialidad_id) REFERENCES especialidad(id));

CREATE TABLE especialidad(
    id INTEGER NOT NULL PRIMARY KEY,
    nombre VARCHAR(30) NOT NULL);

CREATE SEQUENCE s_hospital
START 1
INCREMENT 1
OWNED BY hospital.id;

CREATE SEQUENCE s_medico
START 1
INCREMENT 1
OWNED BY medico.id;
```

```
CREATE SEQUENCE s_especialida
START 1
INCREMENT 1
OWNED BY especialidad.id;
```

2. Inserción de los datos

```
INSERT INTO hospital(id, nombre, provincia)
VALUES
    (NEXTVAL('s_hospital'), 'Hospital Max Peralta','Cartago'),
    (NEXTVAL('s_hospital'), 'Hospital San Rafael','Alajuela'),
    (NEXTVAL('s_hospital'),'Hospital San Vicente de Paul',
'Heredia');

INSERT INTO medico_hospital(medico_id, hospital_id)
VALUES
    (1,2);
    (1,3);
    (4,3);

INSERT INTO medico(id, cedula, nombre,
primer_apellido,direccion_provincia)
VALUES
    (NEXTVAL('s_medico'), '4-0071-0076', 'Gloria', 'Morales',
'Alajuela'),
    (NEXTVAL('s_medico'), 'Andrea', 'Porras', 'Heredia'),
    (NEXTVAL('s_medico'), 'Aurelio', 'Sanabria', 'Alajuela'),
    (NEXTVAL('s_medico'), 'Jaime', 'Vargas', 'Cartago');

INSERT INTO medico_especialidad(medico_id, especialidad_id)
VALUES
    (1,1),
    (1,2),
    (2,3),
    (2,4),
    (3,4),
```

```
    (4,4);

INSERT INTO especialidad(id, nombre)
VALUES
    (NEXTVAL('s_especialidad'),'Cardiologo'),
    (NEXTVAL('s_especialidad'),'Alergologo'),
    (NEXTVAL('s_especialidad'),'Pediatra'),
    (NEXTVAL('s_especialidad'),'Nutricionista');
```

3. Función lista_especialidades

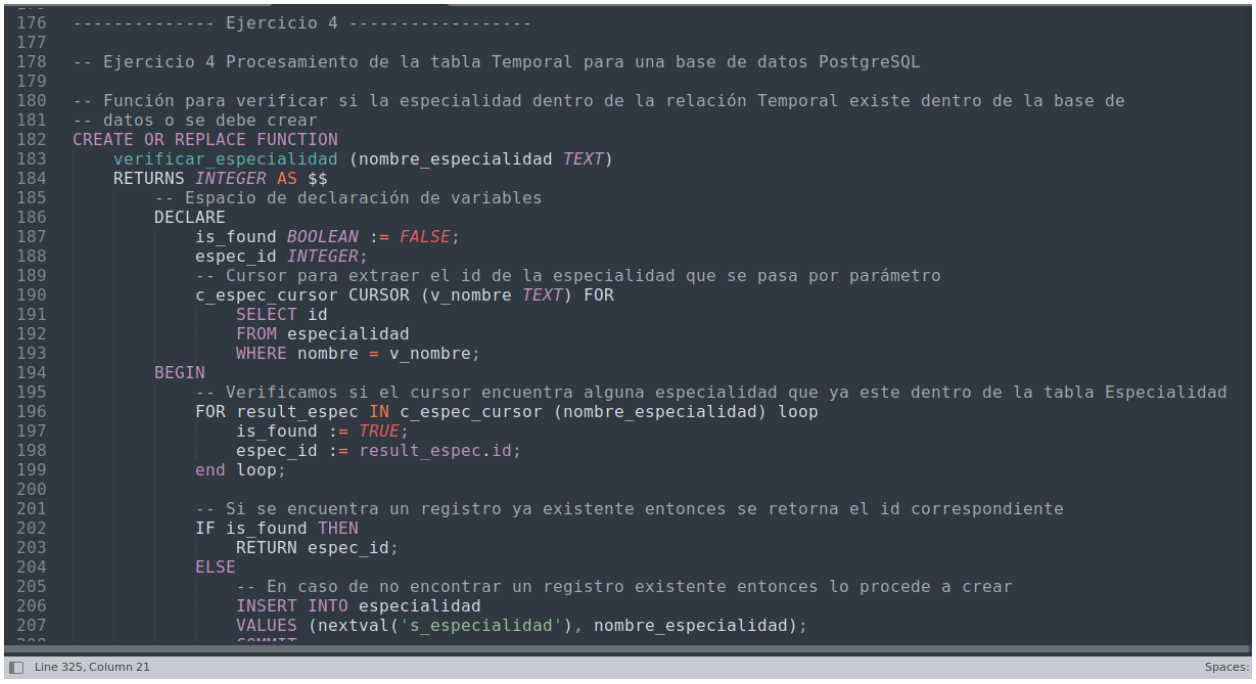
```
------------------función----------------------

CREATE OR REPLACE FUNCTION lista_especialidades (v_cedula TEXT)
    RETURNS VARCHAR AS $$
    DECLARE
    arow record;
    v_id INTEGER;
    v_nombres TEXT;
BEGIN
    SELECT id INTO v_id
    FROM medico
    WHERE cedula = v_cedula;  --acá ya tengo el id del médico
    for arow in(SELECT E.nombre
    into v_nombres
        FROM ESPECIALIDAD E
        INNER JOIN MEDICO_ESPECIALIDAD ME
        ON E.ID = ME.ESPECIALIDAD_ID
        WHERE MEDICO_ID = v_id
        ORDER BY E.NOMBRE ASC)
        loop
        v_nombres:=v_nombres||arow.nombre||',';
        end loop;
        v_nombres:=rtrim(v_nombres, ',');
       -- dbms_output.put_line (v_nombres);--
    return v_nombres;
END;
$$ LANGUAGE plpgsql;

-----------------pruebaFunción------------------

select lista_especialidades('4-0071-0076');
```

4. Normalización de la tabla Temporal



```
176  ------------- Ejercicio 4 -----------------
177
178  -- Ejercicio 4 Procesamiento de la tabla Temporal para una base de datos PostgreSQL
179
180  -- Función para verificar si la especialidad dentro de la relación Temporal existe dentro de la base de
181  -- datos o se debe crear
182  CREATE OR REPLACE FUNCTION
183      verificar_especialidad (nombre_especialidad TEXT)
184      RETURNS INTEGER AS $$
185          -- Espacio de declaración de variables
186          DECLARE
187              is_found BOOLEAN := FALSE;
188              espec_id INTEGER;
189              -- Cursor para extraer el id de la especialidad que se pasa por parámetro
190              c_espec_cursor CURSOR (v_nombre TEXT) FOR
191                  SELECT id
192                  FROM especialidad
193                  WHERE nombre = v_nombre;
194          BEGIN
195              -- Verificamos si el cursor encuentra alguna especialidad que ya este dentro de la tabla Especialidad
196              FOR result_espec IN c_espec_cursor (nombre_especialidad) loop
197                  is_found := TRUE;
198                  espec_id := result_espec.id;
199              end loop;
200
201              -- Si se encuentra un registro ya existente entonces se retorna el id correspondiente
202              IF is_found THEN
203                  RETURN espec_id;
204              ELSE
205                  -- En caso de no encontrar un registro existente entonces lo procede a crear
206                  INSERT INTO especialidad
207                  VALUES (nextval('s_especialidad'), nombre_especialidad);
```

Line 325, Column 21 Spaces:

```
205                  -- En caso de no encontrar un registro existente entonces lo procede a crear
206                  INSERT INTO especialidad
207                  VALUES (nextval('s_especialidad'), nombre_especialidad);
208                  COMMIT;
209                  -- Obtenemos el id generado automáticamente usando el cursor y luego lo retornamos
210                  OPEN c_espec_cursor (nombre_especialidad);
211                  FETCH c_espec_cursor INTO espec_id;
212                  CLOSE c_espec_cursor;
213                  RETURN espec_id;
214              END IF;
215          END;
216      $$ LENGUAGE plpgsql;
217
218  -- Procedimiento almacenado para el procesamiento de la relación Temporal
219  CREATE OR REPLACE PROCEDURE procesa_medico
220      LENGUAGE plpgsql
221      AS $$
222          DECLARE
223              -- Espacio para declarar variables
224              m_especialidades TEXT;
225              m_espec TEXT;
226              m_hospitales TEXT;
227              m_hosp TEXT;
228              v_medico_id INTEGER;
229              v_espec_id INTEGER;
230              v_hospital_id INTEGER;
231              -- Variable cursor para extraer todos los datos de la tabla Temporal
232              c_temporal_cursor FOR SELECT * FROM temporal;
233
234          BEGIN
235              -- Recorremos todos los registros dentro de la relación Temporal
236              FOR temporalRecord IN c_temporal_cursor LOOP
```

```
235     -- Recorremos todos los registros dentro de la relación Temporal
236     FOR temporalRecord IN c_temporal_cursor LOOP
237         -- Para cada medico en la relación, lo insertamos en la tabla Medico
238         INSERT INTO medico
239         VALUES (nextval('s_medico'), cur_medico.medico_cedula,
240                 cur_medico.medico_nombre,
241                 cur_medico.medico_apellido, cur_medico.medico_provincia);
242         COMMIT;
243         -- Ahora obtenemos el id autogenerado del médico que acaba de ser insertado
244         SELECT id INTO v_medico_id FROM medico WHERE cedula = cur_medico.medico_cedula;
245
246         -- Tomamos todas las especialidades relacionadas al médico
247         m_especialidades := trim(cur_medico.especialidades);
248
249         -- Realizamos un loop de todas las especialidades dentro de la tabla para generar las relaciones necesarias
250         IF length(m_especialidades) > 0 THEN
251             LOOP
252                 IF POSITION(',' IN m_especialidades) > 0 THEN
253                     -- seleccionar la primer especialidad
254                     m_espec := SUBSTRING(m_especialidades, 1, POSITION(',' IN m_especialidades) - 1);
255                     -- remover la primer especialidad
256                     m_especialidades := trim(SUBSTRING(m_especialidades, POSITION(',' IN m_especialidades) + 1));
257                 ELSE
258                     m_espec := m_especialidades;
259                     m_especialidades := '';
260                 END IF;
261                 -- verificar si la especialidad existe en la BD, si no la inserta y devuelve el valor del id generad
262                 v_espec_id := verificar_especialidad(m_espec);
263
264                 -- Crear la relación entre el medico y la especialidad
265                 INSERT INTO medico_especialidad (medico_id, especialidad_id)
266                 VALUES (v_medico_id, v_espec_id);
267                 COMMIT;
268
269             EXIT WHEN m_especialidades IS NULL;
270             END LOOP;
271         END IF;
272
273         m_hospitales := trim(cur_medico.hospitales);
274         --Realizamos un loop para todos los hospitales dentro de la tabla
275         IF length(m_hospitales) > 0 THEN
276             LOOP
277                 IF POSITION(',' IN m_hospitales) > 0 THEN
278                     --seleccionar el primer hospital
279                     m_hosp := SUBSTRING(m_hospitales, 1, POSITION(',' IN m_hospitales) - 1);
280                     --quitamos el primer hospital
281                     m_hosp := trim(SUBSTRING(m_hospitales, POSITION(',' IN m_hospitales) + 1));
282                 ELSE
283                     m_hosp := m_hospitales;
284                     m_hospitales := '';
285                 END IF;
286
287                 -- Tomamos el id del hospital al que el medico trabaja
288                 SELECT id INTO v_hospital_id FROM hospital WHERE nombre = m_hosp;
289
290                 -- creamos la relación entre el médico y el hospital
291                 INSERT INTO medico_hospital (medico_id, hospital_id)
292                 VALUES (v_medico_id, v_hospital_id);
293                 COMMIT;
294
295             EXIT WHEN m_hospitales IS NULL;
296             END LOOP;
297         END IF;
298     END LOOP;
```

```sql

-- creación de la relación Temporal con datos de prueba para la función y el procedimiento creado

CREATE TABLE temporal (
    medico_cedula TEXT,
    medico_nombre TEXT,
    medico_apellido TEXT,
    medico_provincia TEXT,
    especialidades TEXT,
    hospitales TEXT
);

INSERT INTO temporal VALUES('3-0098-8768', 'Marta', 'Morales', 'Cartago',
                    'Alergologo, Pediatra, Nutricionista, Odontologo',
                    'Hospital Max Peralta');
INSERT INTO temporal VALUES('2-0876-4527', 'Flor', 'Flores', 'Heredia',
                    'Nutricionista, Cardiologo, Medico General',
                    'Hospital San Vicente de Paul');
INSERT INTO temporal VALUES('1-9976-0442', 'Kevin', 'Moraga', 'Alajuela',
                    'Cardiologo, Pediatra, Hepatologo',
                    'Hospital San Rafael');


-- Ejecución del procedimiento almacenado
CALL procesa_medico;
```