

Instituto Tecnológico de Costa Rica
Centro Académico de Alajuela
Bases de Datos II

Grupo 8:
Marco Rodríguez Vargas – Carné: 2022149445
Kevin Carranza Jiménez – Carné: 2015100260
Jorge Esteban Benavides Castro – Carné: 2022230697

Profesor:
Alberto Shum Chan

Fecha de entrega:
26/8/2023 a las 11:59pm

II Semestre, 2023

Tema: Procedimientos almacenados (SP) en Oracle

Objetivo: Poner en práctica el conocimiento adquirido en clase sobre procedimientos almacenados y desarrollar un conjunto de procedimientos y funciones para importar datos de un archivo plano a un modelo relacional.

Entrega en el TecDigital:

Deben presentar un archivo .PDF con el código anexo.

Se debe incluir un archivo de texto con el código.

Forma de trabajo: grupos de 3 personas.

Ejercicio:

1. Cree el siguiente modelo relacional en la base de datos.

- Utilice secuencias para manejar los id de las tablas donde corresponda.
- Defina las llaves primarias y foráneas.
- Todos los campos deben ser no nulos.

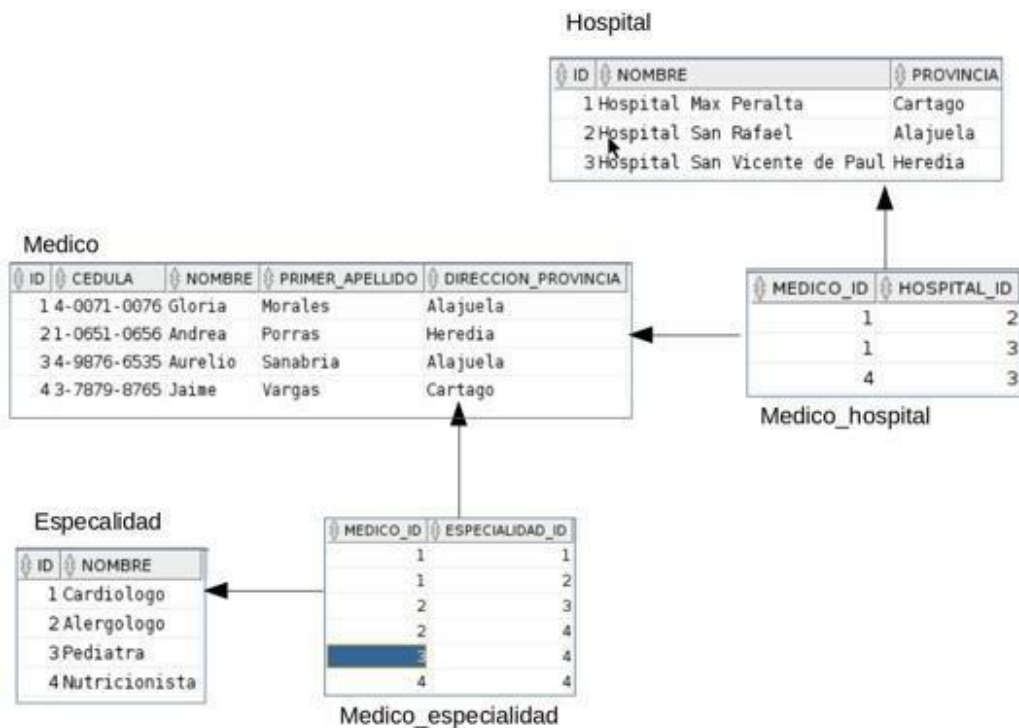


Figura 1. Modelo de datos simplificado de un sistema de manejo de información de hospitales, médicos y especialidades. Las flechas entre dos relaciones representan la restricción llave foránea.

2. Inserte los registros que muestra la figura 1 en todas las relaciones o tablas.

3. Cree una función que reciba la cédula de un médico y devuelva, en un campo de texto, la lista de especialidades asociadas a este, separadas por coma y en orden alfabético ascendente. Nombre de la función lista_especialidades.

4. Normalización utilizando procedimientos almacenados

4.1. Utilizando el diagrama de la figura 1, escriba un procedimiento almacenado que permita importar los siguientes datos e incluirlos en cada una de las relaciones (es decir debe normalizarlos por medio de procedimientos y funciones almacenadas en la base de datos). Utilice el lenguaje de programación de Oracle, PLSQL.

Los datos están almacenados en una relación llamada **Temporal**, un ejemplo de estos datos se muestra a continuación:

Medico-cedula	Medico-nombre	Medico-apellido	Medico-provincia	Especialidades	Hospitales en los que brinda servicios
3-0098-8768	Marta	Morales	Cartago	Alergólogo, Pediatra, Nutricionista, Odontólogo	Hospital Max Peralta
2-0876-4527	Flor	Flores	Heredia	Nutricionista, Cardióloga, Médico general	Hospital San Vicente de Paúl
1-9976-0442	Kevin	Moraga	Alajuela	Cardiólogo, Pediatra, Hepatólogo	Hospital San Rafael

Para las especialidades y médicos se debe verificar si estos existen en la base de datos antes de insertarlos, es decir **solo se deben incluir si no existen**.

Los hospitales ya deben existir en la base de datos.

Tenga en cuenta que existen las siguientes funciones en Oracle:

Substr: devuelve una sub-hilera.

Replace: permite reemplazar un texto por otro dentro de una hilera.

Length: devuelve la longitud de la hilera.

Instr: devuelve la posición de un carácter en una hilera.

1.

-- Eliminar las tablas si existen --

```
DROP TABLE t1_MedicoXEspecialidad;
```

```
DROP TABLE t1_MedicoXHospital;
```

```
DROP TABLE t1_Especialidad;
```

```
DROP TABLE t1_Medico;
```

```
DROP TABLE t1_Hospital;
```

-- Creacion de las tablas junto con sus llaves primarias y foraneas. --

```
CREATE TABLE t1_Hospital (  
    id NUMBER(8) NOT NULL,  
    nombre VARCHAR2(30) NOT NULL,  
    provincia VARCHAR2(30) NOT NULL  
);
```

```
CREATE TABLE t1_Medico (  
    id NUMBER(8) NOT NULL,  
    cedula VARCHAR2(15) NOT NULL,  
    nombre VARCHAR2(30) NOT NULL,  
    primer_apellido VARCHAR2(30) NOT NULL,  
    direccion_provincia VARCHAR2(30) NOT NULL  
);
```

```
CREATE TABLE t1_Especialidad (  
    id NUMBER(8) NOT NULL,  
    nombre VARCHAR2(30) NOT NULL  
);
```

```
CREATE TABLE t1_MedicoXHospital (  
    medico_id NUMBER(8) NOT NULL,  
    hospital_id NUMBER(8) NOT NULL  
);
```

```
CREATE TABLE t1_MedicoXEspecialidad (  
    medico_id NUMBER(8) NOT NULL,  
    especialidad_id NUMBER(8) NOT NULL  
);
```

-- Creacion de las llaves primarias. --

```
ALTER TABLE t1_Hospital
ADD ( CONSTRAINT t1_hospital_id_pk
      PRIMARY KEY (id)
    );
```

```
ALTER TABLE t1_Medico
ADD ( CONSTRAINT t1_medico_id_pk
      PRIMARY KEY (id)
    );
```

```
ALTER TABLE t1_Especialidad
ADD ( CONSTRAINT t1_especialidad_id_pk
      PRIMARY KEY (id)
    );
```

```
ALTER TABLE t1_MedicoXHospital
ADD ( CONSTRAINT t1_medicoxhospital_id_pk
      PRIMARY KEY (medico_id, hospital_id)
    );
```

```
ALTER TABLE t1_MedicoXEspecialidad
ADD ( CONSTRAINT t1_medicoxespecialidad_id_pk
      PRIMARY KEY (medico_id, especialidad_id)
    );
```

-- Creacion de las llaves foraneas. --

```
ALTER TABLE t1_MedicoXHospital
ADD ( CONSTRAINT t1_medicoxhospital_t1_medico_fk
      FOREIGN KEY (medico_id)
      REFERENCES t1_Medico(id)
    );
```

```
ALTER TABLE t1_MedicoXHospital
```

```
ADD ( CONSTRAINT t1_medicoxhospital_t1_hospital_fk
      FOREIGN KEY (hospital_id)
      REFERENCES t1_Hospital(id)
    );
```

```
ALTER TABLE t1_MedicoxEspecialidad
ADD ( CONSTRAINT t1_medicoxespecialidad_t1_medico_fk
      FOREIGN KEY (medico_id)
      REFERENCES t1_Medico(id)
    );
```

```
ALTER TABLE t1_MedicoxEspecialidad
ADD ( CONSTRAINT t1_medicoxespecialidad_t1_Especialidad_fk
      FOREIGN KEY (especialidad_id)
      REFERENCES t1_Especialidad(id)
    );
```

-- Secuencias --

-- Eliminar las secuencias --

```
DROP SEQUENCE T1_hospital_seq;
DROP SEQUENCE T1_medico_seq;
DROP SEQUENCE T1_especialidad_seq;
```

```
CREATE SEQUENCE T1_hospital_seq
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE;
```

```
CREATE SEQUENCE T1_medico_seq
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE;
```

```
CREATE SEQUENCE T1_especialidad_seq
START WITH 1
```

INCREMENT BY 1

NOCACHE

NOCYCLE;

2.

-- Inserts con secuencias--

```
INSERT INTO t1_Hospital (id, nombre, provincia) VALUES (T1_hospital_seq.NEXTVAL, 'Hospital Max Peralta', 'Cartago');
```

```
INSERT INTO t1_Hospital (id, nombre, provincia) VALUES (T1_hospital_seq.NEXTVAL, 'Hospital San Rafael', 'Alajuela');
```

```
INSERT INTO t1_Hospital (id, nombre, provincia) VALUES (T1_hospital_seq.NEXTVAL, 'Hospital San Vicente de Paul', 'Heredia');
```

```
INSERT INTO t1_Medico (id, cedula, nombre, primer_apellido, direccion_provincia) VALUES (T1_medico_seq.NEXTVAL, '4-0071-0076', 'Gloria', 'Morales', 'Alajuela');
```

```
INSERT INTO t1_Medico (id, cedula, nombre, primer_apellido, direccion_provincia) VALUES (T1_medico_seq.NEXTVAL, '1-0651-0656', 'Andrea', 'Porras', 'Heredia');
```

```
INSERT INTO t1_Medico (id, cedula, nombre, primer_apellido, direccion_provincia) VALUES (T1_medico_seq.NEXTVAL, '4-9876-6535', 'Aurelio', 'Sanabria', 'Alajuela');
```

```
INSERT INTO t1_Medico (id, cedula, nombre, primer_apellido, direccion_provincia) VALUES (T1_medico_seq.NEXTVAL, '3-7879-8765', 'Jaime', 'Vargas', 'Cartago');
```

```
INSERT INTO t1_Especialidad (id, nombre) VALUES (T1_especialidad_seq.NEXTVAL, 'Cardiologo');
```

```
INSERT INTO t1_Especialidad (id, nombre) VALUES (T1_especialidad_seq.NEXTVAL, 'Alergologo');
```

```
INSERT INTO t1_Especialidad (id, nombre) VALUES (T1_especialidad_seq.NEXTVAL, 'Pediatria');
```

```
INSERT INTO t1_Especialidad (id, nombre) VALUES (T1_especialidad_seq.NEXTVAL, 'Nutricionista');
```

-- Inserts normales --

```
INSERT INTO t1_MedicoXHospital (medico_id, hospital_id) VALUES (1, 2);
```

```
INSERT INTO t1_MedicoXHospital (medico_id, hospital_id) VALUES (1, 3);
```

```
INSERT INTO t1_MedicoXHospital (medico_id, hospital_id) VALUES (4, 3);
```

```
INSERT INTO t1_MedicoXEspecialidad (medico_id, especialidad_id) VALUES (1, 1);
```

```
INSERT INTO t1_MedicoXEspecialidad (medico_id, especialidad_id) VALUES (1, 2);
```

```
INSERT INTO t1_MedicoXEspecialidad (medico_id, especialidad_id) VALUES (2, 3);
```

```
INSERT INTO t1_MedicoXEspecialidad (medico_id, especialidad_id) VALUES (2, 4);
```

```
INSERT INTO t1_MedicoXEspecialidad (medico_id, especialidad_id) VALUES (3, 4);
```

```
INSERT INTO t1_MedicoXEspecialidad (medico_id, especialidad_id) VALUES (4, 4);
```

3.

```
CREATE OR REPLACE FUNCTION lista_especialidades (p_cedula VARCHAR2)
RETURN VARCHAR2 IS
especialidades VARCHAR2(100) := 'Lista de especialidades asociadas al médico: ' ;
CURSOR c1 IS
    SELECT m.cedula, e.nombre
    FROM T1_Medico m
    JOIN T1_MedicoXEspecialidad me
    ON m.id = me.medico_id
    JOIN T1_Especialidad e
    ON me.especialidad_id = e.id
    WHERE m.cedula = p_cedula
    ORDER BY e.nombre ASC;
BEGIN
    FOR especialidad IN c1 LOOP
        especialidades := especialidades || especialidad.nombre || ', ';
    END LOOP;
    RETURN especialidades;
END;
/
-- Habilitar la salida del servidor DBMS --
SET SERVEROUTPUT ON;

DECLARE
    especialidades_resultado VARCHAR2(100);
BEGIN
    especialidades_resultado := lista_especialidades('1-0651-0656');

    -- Eliminar la coma final si existe --
    IF SUBSTR(especialidades_resultado, - 2) = ', ' THEN
        especialidades_resultado := SUBSTR(especialidades_resultado, 1,
LENGTH(especialidades_resultado) - 2);
    END IF;

    DBMS_OUTPUT.PUT_LINE(especialidades_resultado);
END;
/
```


4.

--Tabla temporal

drop table Temporal;

create table Temporal (

cedula varchar2(12),

nombre varchar2(30),

apellido varchar2(30),

provincia varchar2(10),

especialidad varchar2(100),

hospitales varchar2(50));

insert into Temporal (cedula, nombre, apellido, provincia, especialidad, hospitales) values

('3-0098-8768', 'Marta', 'Morales', 'Cartago', 'Alérgologo, Pediatra, Nutricionista, Odontólogo',

'Hospital Max Peralta');

insert into Temporal (cedula, nombre, apellido, provincia, especialidad, hospitales) values

('2-0876-4527', 'Flor', 'Flores', 'Heredia', 'Nutricionista, Cardióloga, Médico general', 'Hospital San

Vicente de Paúl');

insert into Temporal (cedula, nombre, apellido, provincia, especialidad, hospitales) values

('1-9976-0442', 'Kevin', 'Moraga', 'Alajuela', 'Cardiólogo, Pediatra, Hepatólogo', 'Hospital San Rafael');

-- select * from Temporal; --

/

--Normalización proceso.

CREATE OR REPLACE PROCEDURE normalizar_proc AS

m_nombre VARCHAR (30);

m_especialidades VARCHAR2 (200);

m_hospital_id NUMBER(10);

var_1 NUMBER;

next_id NUMBER(10);

next_id_1 NUMBER(10);

CURSOR hospital_c (h_nombre VARCHAR2) IS (SELECT t1_Hospital.id

FROM t1_Hospital

WHERE t1_Hospital.nombre = h_nombre);

BEGIN

FOR fila IN (SELECT * FROM Temporal) LOOP

SELECT COUNT(*) INTO var_1 FROM t1_Medico WHERE t1_Medico.cedula = fila.cedula;

IF var_1 = 0 THEN

```
FOR sepa IN hospital_c(fila.hospitales) LOOP
```

```
    m_hospital_id := sepa.id;
```

```
    next_id := T1_medico_seq.NEXTVAL;
```

```
    INSERT INTO t1_Medico(id, cedula, nombre, primer_apellido, direccion_provincia)
        VALUES (next_id, fila.cedula, fila.nombre, fila.apellido, fila.provincia);
```

```
    m_especialidades := TRIM(fila.especialidad);
```

```
    INSERT INTO t1_MedicoXHospital VALUES (next_id, m_hospital_id);
```

```
    IF LENGTH (m_especialidades) > 0 THEN
```

```
        LOOP
```

```
            IF INSTR(m_especialidades, ',') > 0 THEN
```

```
                m_nombre := SUBSTR(m_especialidades, 1, INSTR(m_especialidades, ',')-1);
```

```
                m_especialidades := TRIM(SUBSTR(m_especialidades, INSTR(m_especialidades, ',')+1));
```

```
            ELSE
```

```
                m_nombre := m_especialidades;
```

```
                m_especialidades := '';
```

```
            END IF;
```

```
            SELECT COUNT(*) INTO next_id_1 FROM t1_Especialidad WHERE t1_Especialidad.nombre
= m_nombre;
```

```
            IF next_id_1 = 0 THEN
```

```
                next_id_1 := T1_especialidad_seq.NEXTVAL;
```

```
                INSERT INTO t1_Especialidad VALUES (next_id_1, m_nombre);
```

```
            ELSE
```

```
                SELECT t1_Especialidad.id INTO next_id_1 FROM t1_Especialidad WHERE
t1_Especialidad.nombre = m_nombre;
```

```
            END IF;
```

```
            INSERT INTO t1_MedicoXEspecialidad VALUES (next_id, next_id_1);
```

```
            EXIT WHEN m_especialidades IS NULL;
```

```
        END LOOP;
```

```
    END IF;
```

```
END LOOP;
```

```
END IF;
```

```
END LOOP;  
END;  
/
```

```
EXECUTE normalizar_proc();
```