

1º PARCIAL DE BASE DE DATOS – 29/09/2022

PRÁCTICA

Dada la siguiente estructura preexistente, construir el script para modificarla de tal manera que las claves primarias sean **id** manteniendo la integridad. Se aclara que la primary key actuales, corresponden a claves de negocio únicas.

```
/* Estructura preexistente */

CREATE TABLE provincia (
    codigo_provincia INT2 NOT NULL,
    nombre_provincia VARCHAR(60) NOT NULL,
    CONSTRAINT pk_provincia PRIMARY KEY (codigo_provincia));

CREATE TABLE departamento (
    codigo_provincia INT2 NOT NULL,
    item INT2 NOT NULL,
    nombre_departamento VARCHAR(60) NOT NULL,
    CONSTRAINT pk_departamento PRIMARY KEY (codigo_provincia, item),
    CONSTRAINT fk_departamento_provincia FOREIGN KEY (codigo_provincia)
        REFERENCES provincia (codigo_provincia));

CREATE TABLE localidad (
    codigo_localidad INT2 NOT NULL,
    codigo_provincia INT2 NOT NULL,
    item INT2 NOT NULL,
    codigo_postal INT2 NOT NULL,
    nombre_localidad VARCHAR(60) NOT NULL,
    CONSTRAINT pk_localidad PRIMARY KEY (codigo_localidad),
    CONSTRAINT fk_localidad_departamento FOREIGN KEY (codigo_provincia, item)
        REFERENCES departamento (codigo_provincia, item));

CREATE INDEX idx_localidad_departamento ON localidad (codigo_provincia,item);

CREATE TABLE persona (
    tipo_documento VARCHAR(10) NOT NULL,
    numero_documento INT4 NOT NULL,
    apellido VARCHAR(60) NOT NULL,
    nombre VARCHAR(60) NOT NULL,
    fecha_nacimiento DATE NOT NULL,
    domicilio VARCHAR(60) NULL,
    codigo_localidad INT2 NULL,
    tipo_documento_padre VARCHAR(10) NULL,
    numero_documento_padre INT4 NULL,
    tipo_documento_madre VARCHAR(10) NULL,
    numero_documento_madre INT4 NULL,
    CONSTRAINT pk_persona PRIMARY KEY (tipo_documento, numero_documento),
    CONSTRAINT chk_tipo_documento
        CHECK (tipo_documento IN ('DNI', 'LE', 'LC', 'PASAPORTE')),
    CONSTRAINT fk_persona_localidad FOREIGN KEY (codigo_localidad)
        REFERENCES localidad (codigo_localidad),
    CONSTRAINT fk_persona_padre
        FOREIGN KEY (tipo_documento_padre, numero_documento_padre)
        REFERENCES persona (tipo_documento, numero_documento),
    CONSTRAINT fk_persona_madre
        FOREIGN KEY (tipo_documento_madre, numero_documento_madre)
        REFERENCES persona (tipo_documento, numero_documento));

CREATE INDEX idx_persona_localidad ON persona (codigo_localidad);
CREATE INDEX idx_persona_padre ON persona (tipo_documento_padre,numero_documento_padre);
CREATE INDEX idx_persona_madre ON persona (tipo_documento_madre,numero_documento_madre);

CREATE TABLE empleado (
    tipo_documento VARCHAR(10) NOT NULL,
    numero_documento INT4 NOT NULL,
    numero_legajo INT4 NOT NULL,
    fecha_ingreso DATE NOT NULL,
    CONSTRAINT pk_empleado PRIMARY KEY (tipo_documento, numero_documento),
    CONSTRAINT unq_numero_legajo UNIQUE (numero_legajo),
    CONSTRAINT fk_empleado_persona FOREIGN KEY (tipo_documento, numero_documento)
        REFERENCES persona (tipo_documento, numero_documento));
```

```

CREATE TABLE oficina (
    codigo_oficina      INT2                NOT NULL,
    nombre_oficina      VARCHAR(60)         NOT NULL,
    CONSTRAINT pk_oficina PRIMARY KEY (codigo_oficina));

CREATE TABLE oficina_depends (
    codigo_oficina      INT2                NOT NULL,
    fecha_inicio        DATE                NOT NULL,
    codigo_oficina_padre INT2                NOT NULL,
    fecha_fin           DATE                NULL,
    CONSTRAINT pk_oficina_depends PRIMARY KEY (codigo_oficina, fecha_inicio),
    CONSTRAINT fk_oficina_depends_oficina FOREIGN KEY (codigo_oficina)
        REFERENCES oficina (codigo_oficina),
    CONSTRAINT fk_oficina_depends_oficina_padre FOREIGN KEY (codigo_oficina_padre)
        REFERENCES oficina (codigo_oficina));

CREATE INDEX idx_oficinadepends_oficinapadre ON oficina_depends (codigo_oficina_padre);

CREATE TABLE historial (
    tipo_documento      VARCHAR(10)         NOT NULL,
    numero_documento    INT4                NOT NULL,
    fecha_desde         DATE                NOT NULL,
    codigo_oficina      INT2                NOT NULL,
    fecha_hasta         DATE                NULL,
    CONSTRAINT pk_historial PRIMARY KEY (tipo_documento, numero_documento, fecha_desde),
    CONSTRAINT fk_historial_empleado FOREIGN KEY (tipo_documento, numero_documento)
        REFERENCES empleado (tipo_documento, numero_documento),
    CONSTRAINT fk_historia_fk_histor_oficina FOREIGN KEY (codigo_oficina)
        REFERENCES oficina (codigo_oficina)
);

CREATE INDEX idx_historial_oficina ON historial (codigo_oficina);

```

Entregar la respuesta a este ejercicio en un archivo denominado “Parcial1_Apellido_Nombre.sql” (por ejemplo, “Parcial1_Perez_Jose.sql”).

TEORÍA

1. ¿Qué es el diccionario de datos de una base de datos? Indique cómo se puede manipular éste para insertar nuevos datos, modificarlos o eliminarlos.
2. ¿Qué es un DBMS?
3. Suponga que posee una tabla que tiene 120 columnas y 3.000.000 de filas, siendo la primary key las tres primeras. Se desea hacer una partición vertical en 3 tablas dejando en la primera de ellas las columnas más utilizadas, pero las tablas deben tener la misma cantidad de columnas.
 - a. Describa las operaciones que debería realizar (no comandos sino las operaciones requeridas)
 - b. ¿Qué cantidad de filas tiene cada tabla?
 - c. ¿Qué cantidad de columnas tiene cada tabla?
4. Suponga que posee una tabla que tiene 100 columnas y 3.000.000 de filas, siendo la primary key las dos primeras. Se desea hacer una partición horizontal en 3 tablas con igual cantidad de filas.
 - a. Describa las operaciones que debería realizar (idem anterior).
 - b. ¿Qué cantidad de filas tiene cada tabla?
 - c. ¿Qué cantidad de columnas tiene cada tabla?

Agregar las respuestas a las preguntas de teoría, al final del archivo .sql de la parte de práctica.