

## BASES DE DATOS – PARCIAL 1 – 13/10/2020

Alumno: ALVEZ, RODRIGO ESTEBAN

Correo electrónico: rodrigoesteban.alvez@gmail.com

Duración del examen: 90 minutos

### PRÁCTICA

#### Ejercicio 1

Dada la siguiente estructura preexistente construir el modelo que la modifique de tal manera que las claves primarias sean **ID INT4** indicando además los elementos que serán únicos.

```
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 INDEX idx_departamento_provincia ON departamento (codigo_provincia);

CREATE TABLE localidad (
    codigo_provincia INT2 NOT NULL,
    item INT2 NOT NULL,
    codigo_localidad INT2 NOT NULL,
    codigo_postal INT2 NOT NULL,
    nombre_localidad VARCHAR(60) NOT NULL,
    CONSTRAINT pk_localidad PRIMARY KEY (codigo_provincia, item, 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_provincia INT2 NULL,
    item INT2 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_provincia, item, codigo_localidad) REFERENCES localidad (codigo_provincia, item, 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_padre ON persona (tipo_documento_padre, numero_documento_padre);
CREATE INDEX idx_persona_madre ON persona (tipo_documento_madre, numero_documento_madre);
CREATE INDEX idx_persona_localidad ON persona (codigo_provincia, item, codigo_localidad);

CREATE TABLE empleado (
    numero_legajo INT4 NOT NULL,
    tipo_documento VARCHAR(10) NOT NULL,
    numero_documento INT4 NOT NULL,
    fecha_ingreso DATE NOT NULL,
    CONSTRAINT pk_empleado PRIMARY KEY (numero_legajo),
    CONSTRAINT fk_empleado_persona FOREIGN KEY (tipo_documento, numero_documento) REFERENCES persona (tipo_documento, numero_documento),
    CONSTRAINT unq_empleado_persona UNIQUE (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_depends_oficina ON oficina_depends (codigo_oficina);
CREATE INDEX idx_depends_oficina_padre ON oficina_depends (codigo_oficina_padre);

CREATE TABLE historial (
    numero_legajo INT4 NOT NULL,
    fecha_desde DATE NOT NULL,
    codigo_oficina INT2 NOT NULL,
    fecha_hasta DATE NULL,
    CONSTRAINT pk_historial PRIMARY KEY (numero_legajo, fecha_desde),
    CONSTRAINT fk_historial_empleado FOREIGN KEY (numero_legajo) REFERENCES empleado (numero_legajo),
    CONSTRAINT fk_historial_oficina FOREIGN KEY (codigo_oficina) REFERENCES oficina (codigo_oficina));
CREATE INDEX idx_historial_oficina ON historial (codigo_oficina);
```

#### Ejercicio 2

Escriba todos los comandos que deben ejecutarse para modificar la tabla **persona** suponiendo que ya fue modificada la tabla **localidad**. Escriba luego los comandos necesarios para modificar la tabla **historial** suponiendo que ya fue modificada la tabla **oficina** y **empleado**. Recuerde incluir los índices correspondientes.

## **TEORÍA**

1. ¿Cómo se manipulan los datos del diccionario de datos de una base de datos relacional?
2. ¿Cómo se denomina al conjunto de comandos SQL que se encargan de gestionar la estructura de datos en un SGBDR? Mencione los mismos e indique para que sirve cada uno.
3. ¿Cómo se denomina a la regla o principio de las bases de datos relacionales, que permite que las aplicaciones no deban modificarse aún cuando el DBA deba mover los archivos o espacios donde se guardan las bases de datos de un disco a otro (o incluso de un servidor a otro)?