

BASES DE DATOS – RECUPERATORIO PARCIAL 1 – 27/10/2020

Alumno: ALVEZ, RODRIGO ESTEBAN

Correo electrónico: rodrigoesteban.alvez@gmail.com

Grupo: 1

Duración del examen: 105 minutos

Sea la siguiente estructura preexistente la cual ya contiene datos cargados y que reside en un esquema denominado **inicial**:

```
create table funcion (
    id int4 not null,
    codigo_funcion int2 not null,
    nombre_funcion varchar(50) not null,
    constraint pk_funcion primary key (id),
    constraint unq_funcion unique (codigo_funcion));

create table localidad (
    id int4 not null,
    codigo_localidad int2 not null,
    nombre_localidad varchar(60) not null,
    constraint pk_localidad primary key (id),
    constraint unq_localidad unique (codigo_localidad));

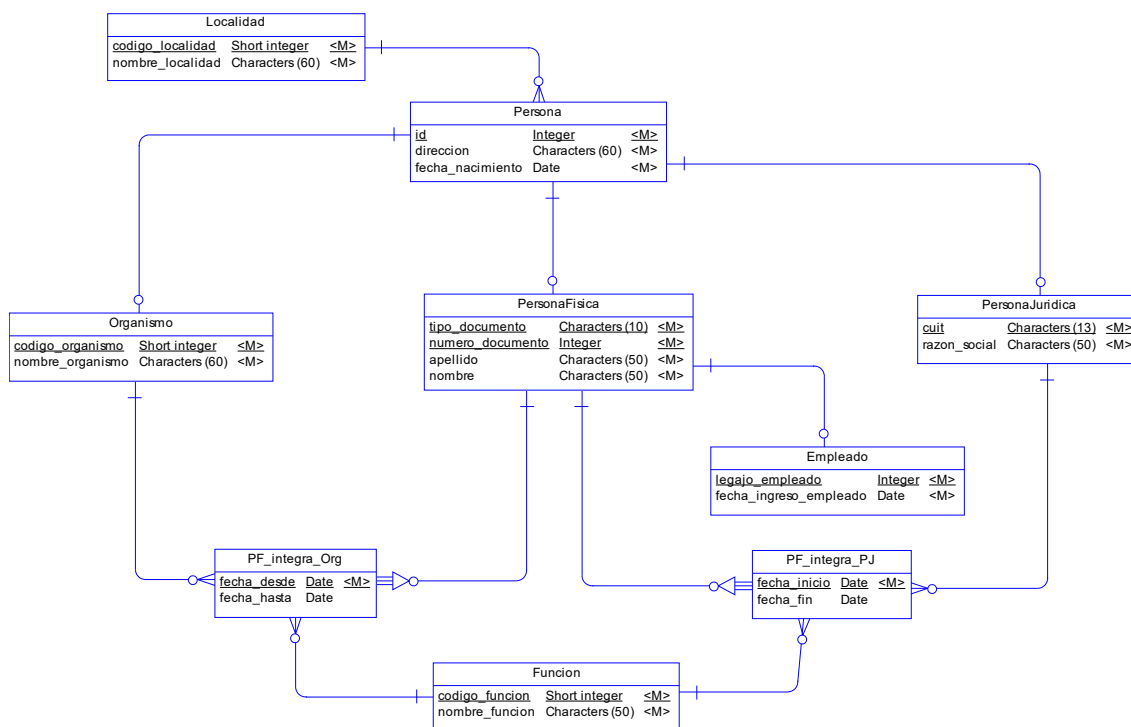
create table persona (
    id int4 not null,
    tipo_persona varchar(20) not null,
    fecha_nacimiento date not null,
    codigo_organismo int2 null,
    nombre_organismo varchar(50) null,
    cuit varchar(13) null,
    razon_social varchar(50) null,
    tipo_documento varchar(10) null,
    numero_documento int4 null,
    apellido varchar(50) null,
    nombre varchar(50) null,
    legajo_empleado int4 null,
    fecha_ingreso_empleado date null,
    direccion varchar(60) not null,
    id_localidad int4 not null,
    constraint pk_persona primary key (id),
    constraint chk_tipo_persona
        check (tipo_persona in ('PERSONA_FISICA', 'PERSONA_JURIDICA', 'ORGANISMO')),
    constraint chk_tipo_documento check (tipo_documento in ('DNI', 'LE', 'LC', 'PASAPORTE')),
    constraint unq_organismo unique (codigo_organismo),
    constraint unq_persona_juridica unique (cuit),
    constraint unq_persona_fisica unique (tipo_documento, numero_documento),
    constraint fk_persona_localidad foreign key (id_localidad) references localidad (id));

create table pf_integra (
    id int4 not null,
    id_pf int4 not null,
    id_pj int4 null,
    id_org int4 null,
    id_funcion int4 not null,
    fecha_inicio date not null,
    fecha_fin date null,
    constraint pk_pf_integra primary key (id),
    constraint fk_pf_integra_pf foreign key (id_pf) references persona (id),
    constraint fk_pf_integra_pj foreign key (id_pj) references persona (id),
    constraint fk_pf_integra_org foreign key (id_org) references persona (id),
    constraint fk_pf_integra_funcion foreign key (id_funcion) references funcion (id),
    constraint unq_pf_integra_pj unique (id_pf, id_pj, fecha_inicio),
    constraint unq_pf_integra_org unique (id_pf, id_org, fecha_inicio));
```

La tabla **persona** tiene todos los tipos de personas que existen indicándose su tipo en el atributo **tipo_persona**. La tabla **pf_integra**, está conformada por todas la personas físicas que están vinculadas a una persona jurídica o bien a un organismo y con una función determinada. En la misma tabla se mantiene esa información; siempre existe una persona física y además con quién está vinculada que será una persona jurídica **ó** un organismo (uno u otro – jamás pueden ser nulos ambos).

Se desea modificar totalmente la estructura **incluida la migración de los datos almacenados**, según el siguiente modelo conceptual y utilizando para la generación del modelo físico **claves de negocios** para los identificativos de cada fila y serán ubicados en un esquema (a crear) denominado **final**. No se pide la confección del modelo físico pero sí el

script de creación de las tablas dispuestas en el orden que corresponde más la migración de los datos. Para la migración de los datos se debe tener en cuenta que los originales están en un esquema distinto del destino.



Una vez concluido el trabajo, se requiere eliminar cada una de las tablas y relaciones de la estructura inicial.

Observación:

Dada la tabla **alumno** y **localidad** con los siguientes datos, donde **alumno** tiene dos FK con **localidad** (una por la localidad donde vive y otra por la localidad en la que nació)

ALUMNO			
codigo	nombre	vive_en	nacio_en
1	Cacho	3000	3000
2	Federico	2000	3000
3	Damián	3000	2000
4	Maria	3000	5000

LOCALIDAD	
cod_postal	nomloca
3000	Santa Fe
2000	Rosario
5000	Córdoba

Si se quiere obtener el nombre de cada alumno y las localidades en las que nació y en la que vive, la consulta será:

```

select a.nombre alumno, lv.nomloca vive, ln.nomloca nacio
  from alumno a, localidad lv, localidad ln
 where a.vive_en = lv.cod_postal and a.nacio_en = ln.localidad;
    
```

alumno	vive	nacio
Cacho	Santa Fe	Santa Fe
Federico	Rosario	Santa Fe
Damián	Santa Fe	Rosario
Maria	Santa Fe	Córdoba

Puntaje:

- Estructura: 50 puntos
- Pasaje de datos: 40 puntos
- Eliminación elementos estructura inicial: 10