

Ejercicio 1

La siguiente estructura fue creada en el esquema “inicial” y tiene datos cargados:

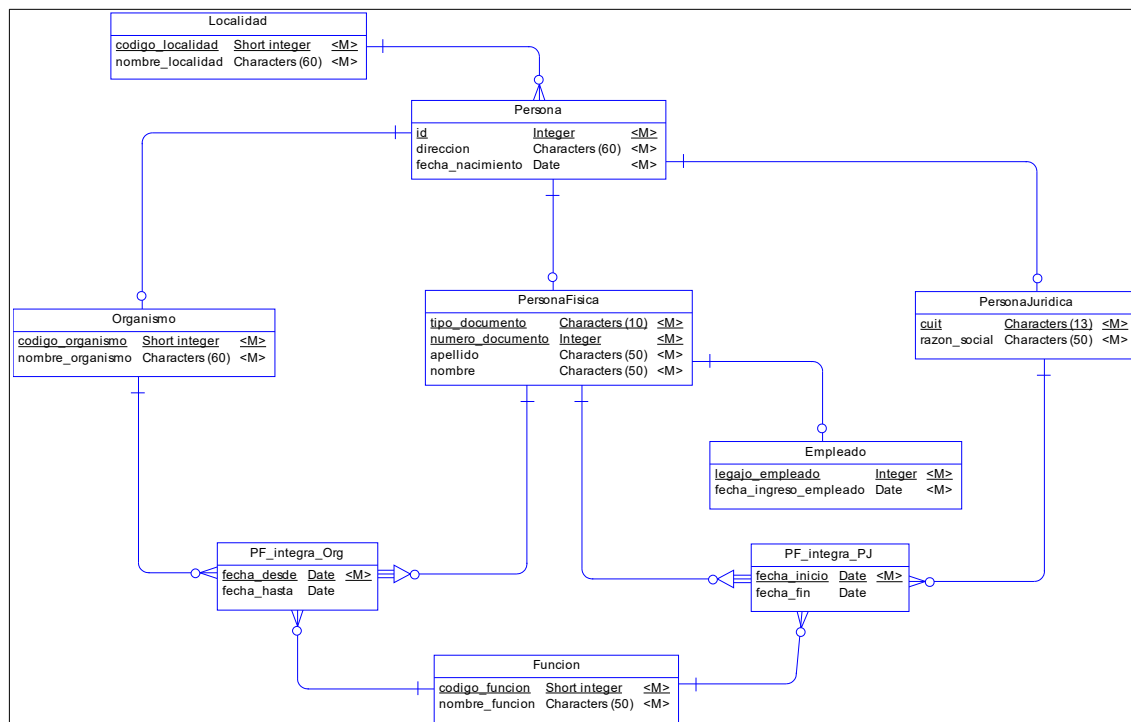
```
-- Creación del esquema "inicial"
create schema inicial;

-- Creación de la tabla funcion dentro del esquema "inicial"
create table inicial.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)
);

-- Creación de la tabla localidad dentro del esquema "inicial"
create table inicial.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)
);

-- Creación de la tabla persona dentro del esquema "inicial"
create table inicial.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,
    codigo_organismo_referenciado int2 null, -- Referencia a otra persona de tipo ORGANISMO
    cuit_referenciado varchar(13) null, -- Referencia a otra persona de tipo PERSONA_JURIDICA
    id_funcion int4 null, -- Referencia a la tabla funcion
    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 inicial.localidad
(id),
    constraint fk_persona_organismo foreign key (codigo_organismo_referenciado) references
inicial.persona (codigo_organismo),
    constraint fk_persona_juridica foreign key (cuit_referenciado) references inicial.persona
(cuit),
    constraint fk_persona_funcion foreign key (id_funcion) references inicial.funcion (id)
);
```

La tabla persona tiene todos los tipos de personas que existen indicándose su tipo en el atributo “tipo_persona”. Los atributos “codigo_organismo_referenciado” y “cuit_referenciado”, permiten establecer una referencia desde una persona física hacia un organismo o hacia una persona jurídica. La columna *id_funcion* indica la función que desempeña la persona física en el organismo o en la persona jurídica. Se desea modificar totalmente la estructura incluida la migración de los datos almacenados, según el siguiente modelo conceptual:



Para la generación del modelo físico se requiere utilizar claves de negocios para los identificativos. La estructura y los datos 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 el esquema **inicial**.

Para cargar las fechas_desde/fecha_inicio en las tablas correspondientes a las entidades “PF_integra_org” y “PF_integra_PJ” utilizar el valor fijo “01/01/2024”.

Una vez concluido el trabajo, se requiere eliminar la estructura inicial.

SOLUCIÓN

Ejercicio 1:

Creación de tablas

```
-- Esquema final
-- Crea estructura nueva
create schema final;

CREATE TABLE final.funcion (
    codigo_funcion int2 NOT NULL,
    nombre_funcion varchar(50) NOT NULL,
    CONSTRAINT pk_funcion PRIMARY KEY (codigo_funcion)
);

CREATE TABLE final.localidad (
    codigo_localidad int2 NOT NULL,
    nombre_localidad varchar(60) NOT NULL,
    CONSTRAINT pk_localidad PRIMARY KEY (codigo_localidad)
);

CREATE TABLE final.persona (
    codigo_persona int4 NOT NULL,
    fecha_nacimiento date NOT NULL,
    direccion varchar(60) NOT NULL,
    codigo_localidad int2 NOT NULL,
    CONSTRAINT pk_persona PRIMARY KEY (codigo_persona),
    CONSTRAINT fk_persona_localidad FOREIGN KEY (codigo_localidad) REFERENCES
final.localidad(codigo_localidad)
);

CREATE TABLE final.persona_fisica (
    tipo_documento varchar(10) NOT NULL,
    numero_documento int4 NOT NULL,
    apellido varchar(50) NOT NULL,
    nombre varchar(50) NOT NULL,
    codigo_persona int4 NOT NULL,
    CONSTRAINT pk_persona_fisica PRIMARY KEY (tipo_documento, numero_documento),
    CONSTRAINT fk_persona_fisica_persona FOREIGN KEY (codigo_persona) REFERENCES
final.persona(codigo_persona)
);

CREATE TABLE final.persona_juridica (
    cuit varchar(13) NOT NULL,
    razon_social varchar(50) NOT NULL,
    codigo_persona int4 NOT NULL,
    CONSTRAINT pk_persona_juridica PRIMARY KEY (cuit),
    CONSTRAINT fk_persona_juridica_persona FOREIGN KEY (codigo_persona) REFERENCES
final.persona(codigo_persona)
);

CREATE TABLE final.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_fisica FOREIGN KEY (tipo_documento, numero_documento)
REFERENCES final.persona_fisica(tipo_documento, numero_documento)
);

CREATE TABLE final.organismo (
    codigo_organismo int2 NOT NULL,
    nombre_organismo varchar(50) NOT NULL,
    codigo_persona int4 NOT NULL,
    CONSTRAINT pk_organismo PRIMARY KEY (codigo_organismo),
    CONSTRAINT fk_organismo_persona FOREIGN KEY (codigo_persona) REFERENCES
final.persona(codigo_persona)
);

CREATE TABLE final.persona_integra_organismo (
    tipo_documento varchar(10) NOT NULL,
    numero_documento int4 NOT NULL,
    codigo_organismo int2 NOT NULL,
    codigo_funcion int2 NOT NULL,
```

```

        fecha_desde date NOT NULL,
        fecha_fin date NULL,
        CONSTRAINT pk_persona_integra_org PRIMARY KEY (tipo_documento, numero_documento,
fecha_desde),
        CONSTRAINT fk_org_funcion FOREIGN KEY (codigo_funcion) REFERENCES
final.funcion(codigo_funcion),
        CONSTRAINT fk_org_organismo FOREIGN KEY (codigo_organismo) REFERENCES
final.organismo(codigo_organismo),
        CONSTRAINT fk_org_persona_fisica FOREIGN KEY (tipo_documento,numero_documento) REFERENCES
final.persona_fisica(tipo_documento,numero_documento)
);

CREATE TABLE final.persona_integra_personajuridica (
        tipo_documento varchar(10) NOT NULL,
        numero_documento int4 NOT NULL,
        cuit varchar(13) NOT NULL,
        codigo_funcion int2 NOT NULL,
        fecha_desde date NOT NULL,
        fecha_fin date NULL,
        CONSTRAINT pk_persona_integra_pj PRIMARY KEY (tipo_documento, numero_documento,
fecha_desde),
        CONSTRAINT fk_pj_funcion FOREIGN KEY (codigo_funcion) REFERENCES
final.funcion(codigo_funcion),
        CONSTRAINT fk_pj_persona_fisica FOREIGN KEY (tipo_documento,numero_documento) REFERENCES
final.persona_fisica(tipo_documento,numero_documento),
        CONSTRAINT fk_pj_persona_juridica FOREIGN KEY (cuit) REFERENCES final.persona_juridica(cuit)
);

--- UNIQUES
ALTER TABLE final.persona_fisica add CONSTRAINT uk_pf_persona UNIQUE (codigo_persona);
ALTER TABLE final.persona_juridica add CONSTRAINT uk_pj_persona UNIQUE (codigo_persona);
ALTER TABLE final.organismo add CONSTRAINT uk_org_persona UNIQUE (codigo_persona);
ALTER TABLE final.empleado add CONSTRAINT uk_emp_persona UNIQUE (codigo_persona);

```

Migración de datos

```
insert into final.localidad (codigo_localidad, nombre_localidad)
-- Migrar datos de inicial.localidad a final.localidad
select codigo_localidad, nombre_localidad
from inicial.localidad;

-- Migrar datos de inicial.funcion a final.funcion
insert into final.funcion (codigo_funcion, nombre_funcion)
select codigo_funcion, nombre_funcion
from inicial.funcion;

-- Migrar datos de inicial.persona a final.persona
-- Migrar datos de inicial.persona a final.persona con el mapeo correcto de codigo_localidad
insert into final.persona (codigo_persona , fecha_nacimiento, direccion, codigo_localidad)
select
    p.id,
    p.fecha_nacimiento,
    p.direccion,
    l.codigo_localidad -- Tomamos el código de la tabla localidad
from
    inicial.persona p
join
    inicial.localidad l
on
    p.id_localidad = l.id ; -- Relacionamos persona con localidad mediante codigo_postal

-- Migrar datos de inicial.persona a final.persona_fisica para personas físicas
insert into final.persona_fisica (tipo_documento, numero_documento, apellido, nombre,
codigo_persona)
select tipo_documento , numero_documento ,apellido , nombre , id
from inicial.persona
where tipo_persona = 'PERSONA_FISICA';

-- Migrar datos de inicial.persona a final.persona_juridica para personas jurídicas
insert into final.persona_juridica (cuit, razon_social, codigo_persona)
select cuit, razon_social, id
from inicial.persona
where tipo_persona = 'PERSONA_JURIDICA';

-- Migrar datos de inicial.persona a final.organismo para organismos
insert into final.organismo (codigo_organismo, nombre_organismo,codigo_persona)
select codigo_organismo, nombre_organismo,id
from inicial.persona
where tipo_persona = 'ORGANISMO';
select * from inicial.persona where tipo_persona='ORGANISMO'
update inicial.persona set nombre_organismo = 'Organismo ab' where tipo_persona='ORGANISMO'

-- Migrar datos de inicial.persona a final.empleado para personas físicas con legajo
insert into final.empleado (numero_legajo, tipo_documento, numero_documento, fecha_ingreso)
select legajo_empleado, tipo_documento, numero_documento, fecha_ingreso_empleado
from inicial.persona
where tipo_persona = 'PERSONA_FISICA' and legajo_empleado is not null;

-- Migrar datos a final.persona_integra_personajuridica
insert into final.persona_integra_personajuridica (tipo_documento, numero_documento, cuit,
codigo_funcion, fecha_desde)
select tipo_documento, numero_documento, cuit_referenciado, codigo_funcion, '2024-01-01'
from inicial.persona p
join inicial.funcion f on p.id_funcion = f.id -- Suponiendo que id_funcion existe
where tipo_persona = 'PERSONA_FISICA' and cuit_referenciado is not null;

--
-- Migrar datos a final.persona_integra_organismo
insert into final.persona_integra_organismo (tipo_documento, numero_documento, codigo_organismo,
codigo_funcion, fecha_desde)
select tipo_documento, numero_documento, codigo_organismo_referenciado, id_funcion, '2024-01-01'
from inicial.persona p
--join inicial.funcion f on p.id_funcion = f.id -- Suponiendo que id_funcion existe
where tipo_persona = 'PERSONA_FISICA' and codigo_organismo_referenciado is not null;
```

Ejercicio 1: Eliminación esquema inicial

```
drop table localidad  
drop table funcion  
drop table persona  
drop schema "inicial"
```