

## Carga Masiva

```
DROP TABLE IF EXISTS categoria CASCADE;
DROP TABLE IF EXISTS actor CASCADE;
DROP TABLE IF EXISTS traduccion CASCADE;
DROP TABLE IF EXISTS pais CASCADE;
DROP TABLE IF EXISTS ciudad CASCADE;
DROP TABLE IF EXISTS direccion CASCADE;
DROP TABLE IF EXISTS tienda CASCADE;
DROP TABLE IF EXISTS cliente CASCADE;
DROP TABLE IF EXISTS pelicula CASCADE;
DROP TABLE IF EXISTS empleado CASCADE;
DROP TABLE IF EXISTS renta CASCADE;
DROP TABLE IF EXISTS detalle_categoria CASCADE;
DROP TABLE IF EXISTS detalle_traduccion CASCADE;
DROP TABLE IF EXISTS detalle_actor CASCADE;

CREATE TABLE categoria (
    id_categoria SERIAL PRIMARY KEY,
    nombre VARCHAR NOT NULL
);
CREATE TABLE actor (
    id_actor SERIAL PRIMARY KEY,
    nombre VARCHAR NOT NULL,
    apellido VARCHAR NOT NULL
);
CREATE TABLE traduccion (
    id_traduccion SERIAL PRIMARY KEY,
    nombre VARCHAR NOT NULL
);
CREATE TABLE pais (
    id_pais SERIAL PRIMARY KEY,
    nombre VARCHAR NOT NULL
);
CREATE TABLE ciudad (
    id_ciudad SERIAL PRIMARY KEY,
    id_pais INT NOT NULL REFERENCES pais(id_pais),
    nombre VARCHAR NOT NULL
);
CREATE TABLE direccion (
    id_direccion SERIAL PRIMARY KEY,
    id_ciudad INT NOT NULL REFERENCES ciudad(id_ciudad),
    distrito VARCHAR NOT NULL,
    codigo VARCHAR NOT NULL
);
```

```
CREATE TABLE tienda (  
    id_tienda SERIAL PRIMARY KEY,  
    id_direccion INT REFERENCES direccion(id_direccion),  
    nombre VARCHAR NOT NULL  
);  
  
CREATE TABLE cliente (  
    id_cliente SERIAL PRIMARY KEY,  
    id_tienda INT NOT NULL REFERENCES tienda(id_tienda),  
    id_direccion INT NOT NULL REFERENCES direccion(id_direccion),  
    nombre VARCHAR NOT NULL,  
    apellido VARCHAR NOT NULL,  
    correo VARCHAR NOT NULL,  
    fecha_registro DATE NOT NULL,  
    activo VARCHAR NOT NULL  
);  
  
CREATE TABLE pelicula (  
    id_pelicula SERIAL PRIMARY KEY,  
    titulo VARCHAR NOT NULL,  
    descripcion VARCHAR NOT NULL,  
    ano INT NOT NULL,  
    duracion INT NOT NULL,  
    dias INT NOT NULL,  
    costo DECIMAL NOT NULL,  
    clasificacion VARCHAR NOT NULL  
);  
  
CREATE TABLE empleado (  
    id_empleado SERIAL PRIMARY KEY,  
    id_tienda INT NOT NULL REFERENCES tienda(id_tienda),  
    id_direccion INT NOT NULL REFERENCES direccion(id_direccion),  
    nombre VARCHAR NOT NULL,  
    apellido VARCHAR NOT NULL,  
    correo VARCHAR NOT NULL,  
    activo VARCHAR NOT NULL,  
    usuario VARCHAR NOT NULL,  
    contrasena VARCHAR NOT NULL  
);
```

```

CREATE TABLE renta (
    id_renta SERIAL PRIMARY KEY,
    id_cliente INT NOT NULL REFERENCES cliente(id_cliente),
    id_empleado INT NOT NULL REFERENCES empleado(id_empleado),
    id_pelicula INT NOT NULL REFERENCES pelicula(id_pelicula),
    monto DECIMAL NOT NULL,
    fecha_renta TIMESTAMP NOT NULL,
    fecha_retorno TIMESTAMP NOT NULL,
    fecha_pago TIMESTAMP
);

CREATE TABLE detalle_categoria (
    id_detalle_categoria SERIAL PRIMARY KEY,
    id_pelicula INT NOT NULL REFERENCES pelicula(id_pelicula),
    id_categoria INT NOT NULL REFERENCES categoria(id_categoria)
);

CREATE TABLE detalle_actor (
    id_detalle_actor SERIAL PRIMARY KEY,
    id_pelicula INT NOT NULL REFERENCES pelicula(id_pelicula),
    id_actor INT NOT NULL REFERENCES actor(id_actor)
);

CREATE TABLE detalle_traduccion (
    id_detalle_traduccion SERIAL PRIMARY KEY,
    id_pelicula INT NOT NULL REFERENCES pelicula(id_pelicula),
    id_traduccion INT NOT NULL REFERENCES traduccion(id_traduccion)
);

ALTER TABLE tienda ADD COLUMN id_empleado INT;
ALTER TABLE tienda
    ADD CONSTRAINT id_empleado FOREIGN KEY (id_empleado) REFERENCES empleado(id_empleado);

INSERT INTO categoria(nombre)
SELECT lower(t.categoria_pelicula)
FROM temporal AS t
WHERE t.categoria_pelicula != '-'
GROUP BY lower(t.categoria_pelicula);

INSERT INTO actor(nombre,apellido)
SELECT split_part(initcap(t.actor_pelicula), ' ', 1) AS nombre
    , split_part(initcap(t.actor_pelicula), ' ', 2) AS apellido
FROM temporal AS t
WHERE t.actor_pelicula != '-'
GROUP BY t.actor_pelicula;

```

```

INSERT INTO traduccion(nombre)
SELECT lower(t.lenguaje_pelicula)
FROM temporal AS t
WHERE t.lenguaje_pelicula != '-'
GROUP BY lower(t.lenguaje_pelicula);

INSERT INTO pelicula(titulo,descripcion,ano,duracion,dias, costo,clasificacion)
SELECT lower(nombre_pelicula),
       descripcion_pelicula,
       ano_lanzamiento::INT,
       duracion::INT,
       dias_renta::INT,
       costo_renta::DECIMAL,
       lower(clasificacion)
FROM temporal
WHERE nombre_pelicula != '-' AND
       descripcion_pelicula != '-' AND
       ano_lanzamiento != '-' AND
       duracion != '-' AND
       dias_renta != '-' AND
       costo_renta != '-' AND
       clasificacion != '-'
GROUP BY lower(nombre_pelicula),
       descripcion_pelicula,
       ano_lanzamiento,
       duracion,
       dias_renta,
       costo_renta,
       lower(clasificacion);

INSERT INTO pais(nombre)
SELECT lower(t.pais_cliente)
FROM temporal AS t
WHERE t.pais_cliente != '-'
GROUP BY lower(t.pais_cliente)
UNION
SELECT lower(t.pais_empleado)
FROM temporal AS t
WHERE t.pais_empleado != '-'
GROUP BY lower(t.pais_empleado)
UNION
SELECT lower(t.pais_tienda)
FROM temporal AS t
WHERE t.pais_tienda != '-'

```

```

GROUP BY lower(t.pais_tienda);

INSERT INTO detalle_categoria(id_pelicula,id_categoria)
SELECT p.id_pelicula, g.id_categoria
FROM temporal AS t
inner join pelicula AS p
ON p.titulo = lower(t.nombre_pelicula)
inner join categoria g
ON g.nombre = lower(t.categoria_pelicula)
group by p.id_pelicula, g.id_categoria ;

INSERT INTO detalle_actor(id_pelicula, id_actor)
SELECT p.id_pelicula, a.id_actor
FROM temporal AS t
inner join pelicula AS p
ON p.titulo = lower(t.nombre_pelicula)
inner join actor a
ON (a.nombre = split_part(initcap(t.actor_pelicula), ' ', 1) AND
    a.apellido = split_part(initcap(t.actor_pelicula), ' ', 2))
group by p.id_pelicula, a.id_actor ;

INSERT INTO detalle_traduccion(id_pelicula,id_traduccion)
SELECT p.id_pelicula, tr.id_traduccion
FROM temporal AS t
inner join pelicula AS p
ON p.titulo = lower(t.nombre_pelicula)
inner join traduccion tr
ON tr.nombre = lower(t.lenguaje_pelicula)
group by p.id_pelicula, tr.id_traduccion ;

INSERT INTO ciudad(id_pais,nombre)
SELECT p.id_pais, t.ciudad_cliente
FROM temporal AS t
inner join pais AS p
ON p.nombre = lower(t.pais_cliente)
WHERE t.ciudad_cliente != '-'
group by p.id_pais, t.ciudad_cliente;

```

```
INSERT INTO direccion (id_ciudad,distrito,codigo)
SELECT c.id_ciudad, t.direccion_cliente, t.codigo_postal_cliente
FROM temporal AS t
inner join ciudad AS c
ON c.nombre = t.ciudad_cliente
WHERE t.direccion_cliente != '-'
group by c.id_ciudad, t.direccion_cliente, t.codigo_postal_cliente
UNION
SELECT c.id_ciudad, t.direccion_tienda, t.codigo_postal_tienda
FROM temporal AS t
inner join ciudad AS c
ON c.nombre = t.ciudad_tienda
WHERE t.direccion_tienda != '-'
group by c.id_ciudad, t.direccion_tienda, t.codigo_postal_tienda
UNION
SELECT c.id_ciudad, t.direccion_empleado, t.codigo_postal_empleado
FROM temporal AS t
inner join ciudad AS c
ON c.nombre = t.ciudad_empleado
WHERE t.direccion_empleado != '-'
group by c.id_ciudad, t.direccion_empleado, t.codigo_postal_empleado;

INSERT INTO tienda(id_direccion, nombre)
SELECT d.id_direccion,t.nombre_tienda
FROM temporal AS t
inner join direccion AS d
ON d.distrito = t.direccion_tienda
WHERE t.nombre_tienda != '-'
group by d.id_direccion,t.nombre_tienda;
```

```

INSERT INTO cliente(id_tienda,id_direccion,nombre,apellido,correo,fecha_registro,
activo)
SELECT ti.id_tienda,d.id_direccion,
split_part(initcap(t.nombre_cliente), ' ', 1) AS nombre,
split_part(initcap(t.nombre_cliente), ' ', 2) AS apellido,
t.correo_cliente,
to_date(t.fecha_creacion, 'DD-MM-YYYY'),
t.cliente_activo
FROM temporal AS t
inner join direccion AS d
ON d.districto = t.direccion_cliente
inner join tienda AS ti
ON ti.nombre = t.tienda_preferida
WHERE t.nombre_cliente != '-' AND
t.correo_cliente != '-' AND
t.fecha_creacion != '-' AND
t.cliente_activo != '-'
group by d.id_direccion,ti.id_tienda,t.nombre_cliente, t.correo_cliente, t.fecha_
creacion, t.cliente_activo

```

```

INSERT INTO empleado(id_tienda,id_direccion,nombre,apellido,correo,activo,usuario
,contrasena)
SELECT ti.id_tienda,d.id_direccion,
split_part(initcap(t.nombre_empleado), ' ', 1) AS nombre,
split_part(initcap(t.nombre_empleado), ' ', 2) AS apellido,
t.correo_empleado,
t.empleado_activo,
t.usuario_empleado,
t.contrasena_empleado
FROM temporal AS t
inner join direccion AS d
ON d.districto = t.direccion_empleado
inner join tienda AS ti
ON ti.nombre = t.tienda_empleado
WHERE t.nombre_empleado != '-' AND
t.correo_empleado != '-' AND
t.fecha_creacion != '-' AND
t.empleado_activo != '-' AND
t.usuario_empleado != '-' AND
t.contrasena_empleado != '-'
group by d.id_direccion,ti.id_tienda,t.nombre_empleado, t.correo_empleado, t.empl
eado_activo,t.usuario_empleado,t.contrasena_empleado

```

```

INSERT INTO renta(id_cliente,id_empleado,id_pelicula,monto,fecha_renta,fecha_reto
rno,fecha_pago)
SELECT id_cliente,e.id_empleado,p.id_pelicula,t.monto_a_pagar::DECIMAL,
TO_TIMESTAMP(t.fecha_renta,'DD-MM-YYYY HH24:MI'),
TO_TIMESTAMP(t.fecha_retorno,'DD-MM-YYYY HH24:MI'),
TO_TIMESTAMP(t.fecha_pago,'DD-MM-YYYY HH24:MI')
FROM temporal AS t
inner join empleado AS e
ON (e.nombre = split_part(initcap(t.nombre_empleado), ' ', 1) AND
    e.apellido = split_part(initcap(t.nombre_empleado), ' ', 2))
inner join cliente AS c
ON (c.nombre = split_part(initcap(t.nombre_cliente), ' ', 1) AND
    c.apellido = split_part(initcap(t.nombre_cliente), ' ', 2))
inner join pelicula AS p
ON p.titulo = lower(t.nombre_pelicula)
WHERE t.monto_a_pagar != '-' AND
    t.fecha_renta != '-' AND
    t.fecha_retorno != '-'
group by c.id_cliente,e.id_empleado,p.id_pelicula,t.monto_a_pagar,t.fecha_renta,t
.fecha_retorno,t.fecha_pago;

```