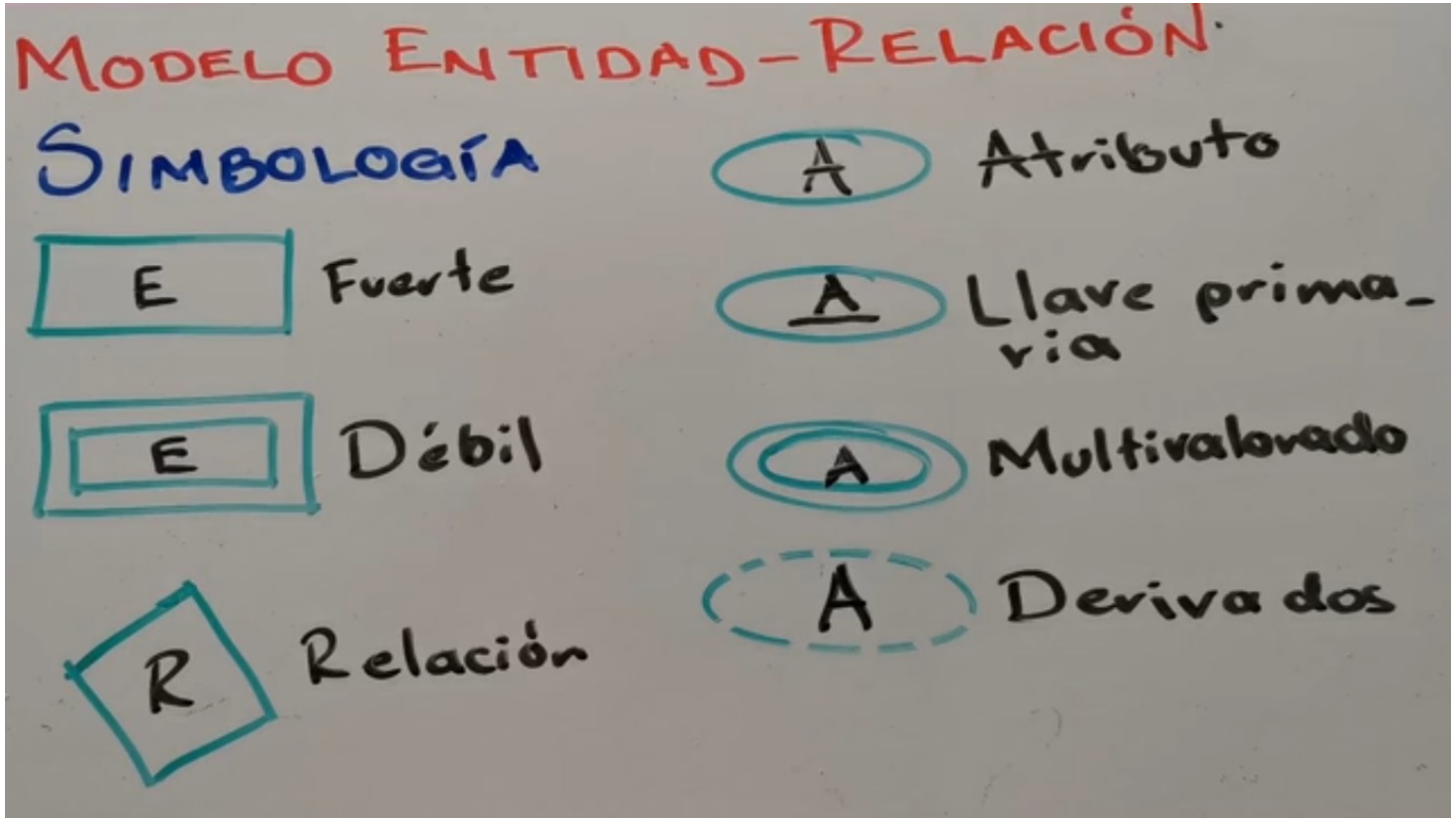
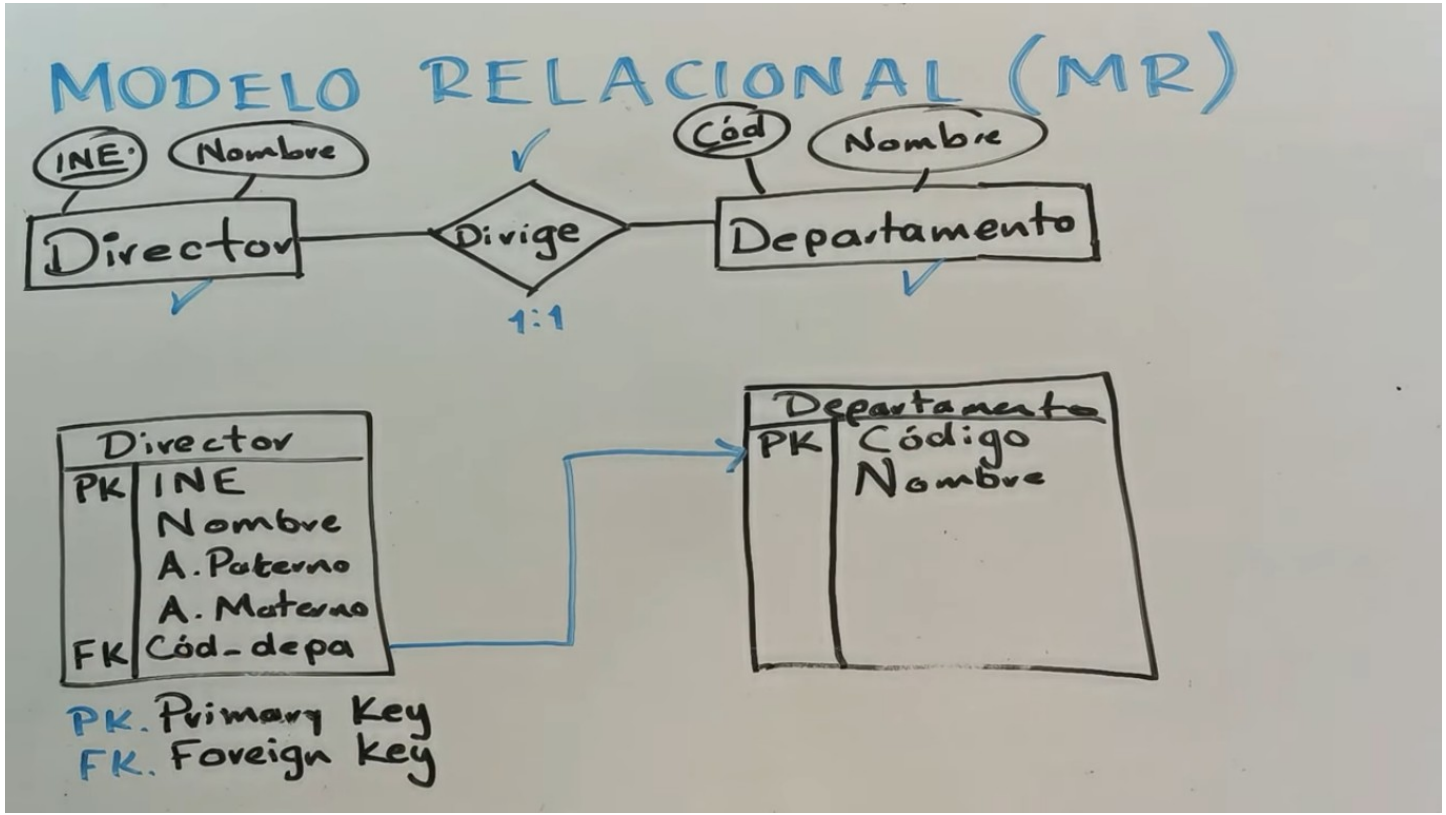
Bases de datos
Clase 7: De Entidad Relación a SQL
Parte I

Leonardo Bravo Illanes
Escuela de Informática y Telecomunicaciones
Universidad Diego Portales

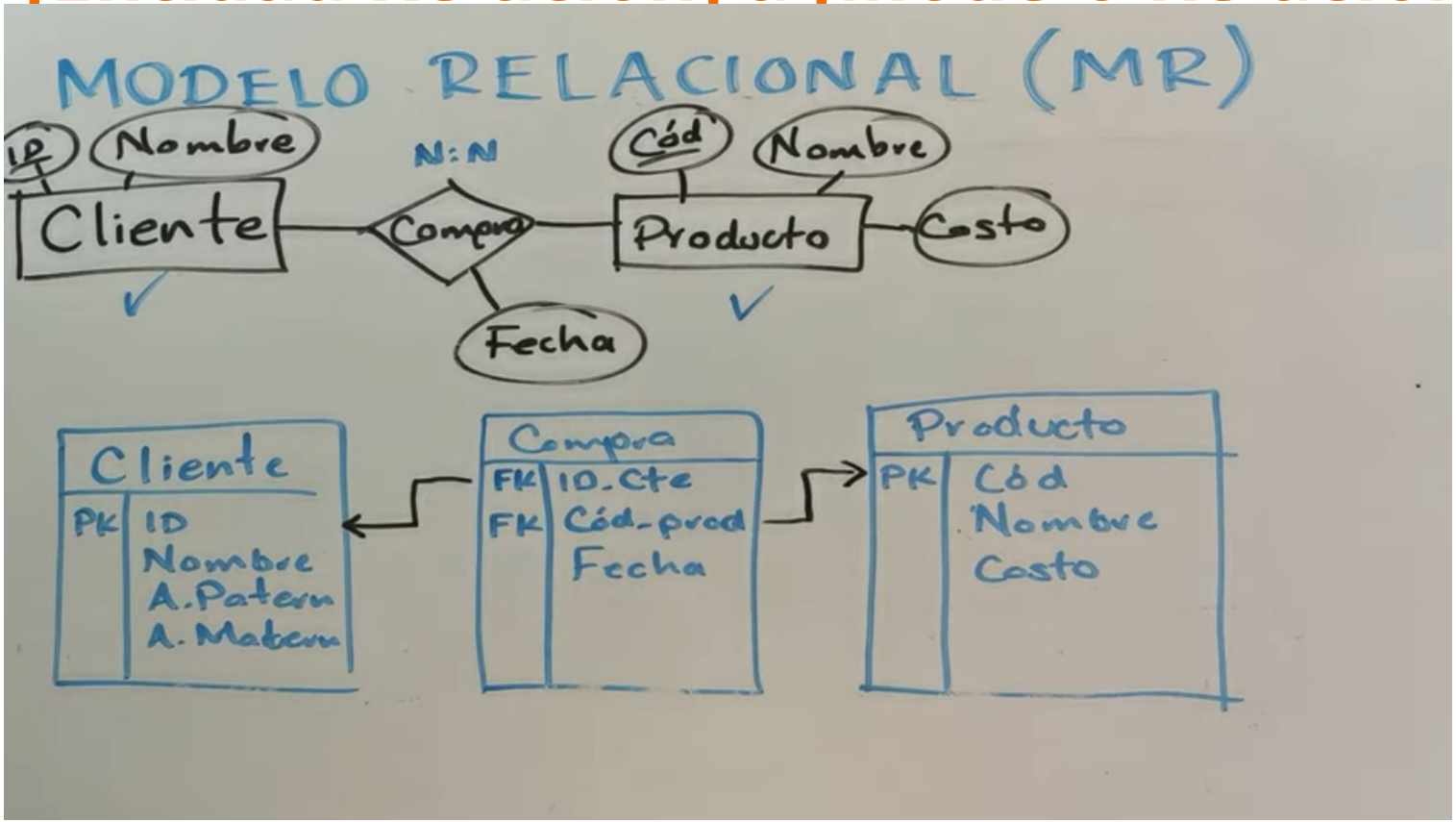
De [Entidad Relación] full Simbologia



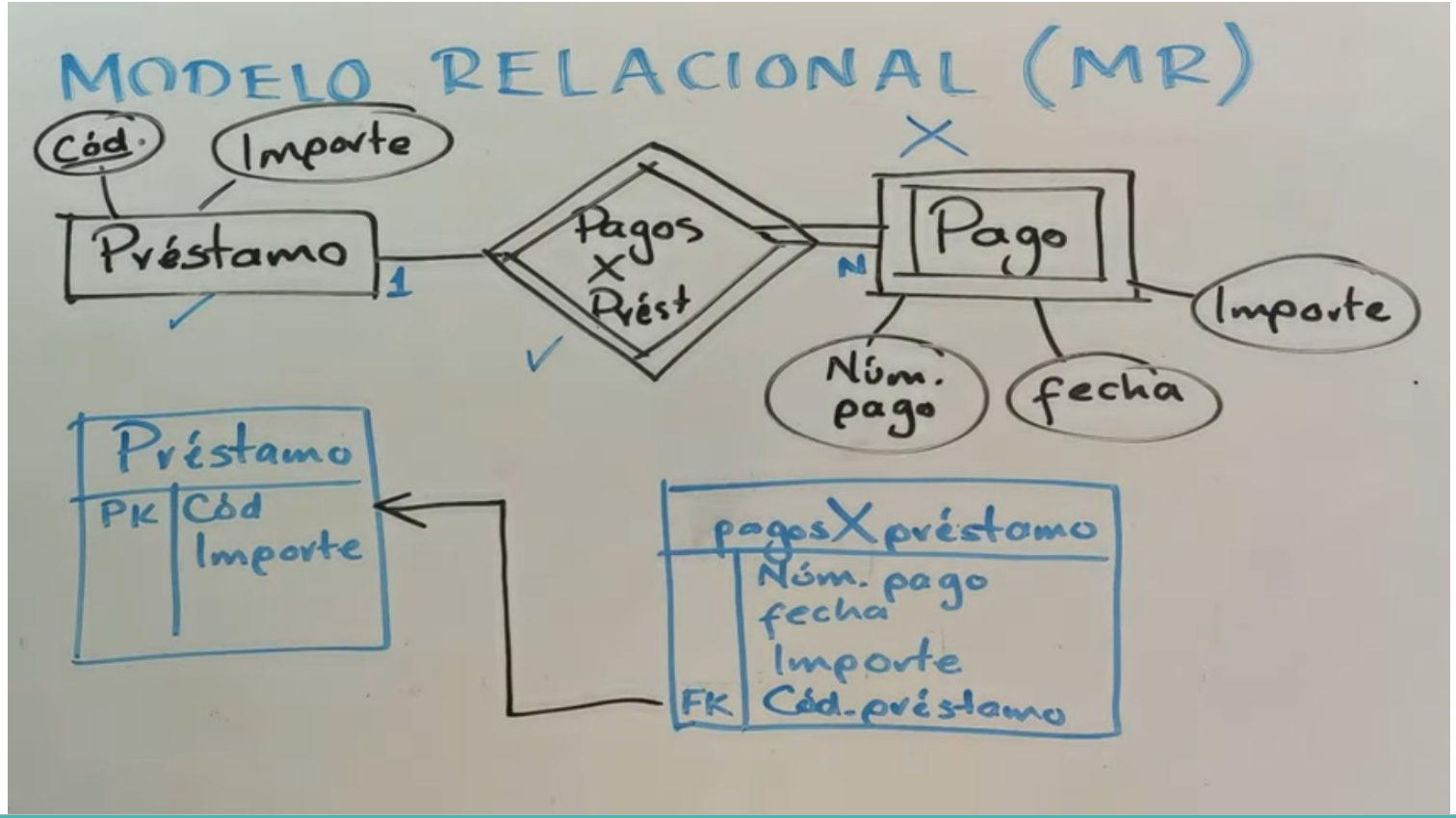
De [Entidad Relación] a [Modelo Relacional]



De [Entidad Relación] a [Modelo Relacional]

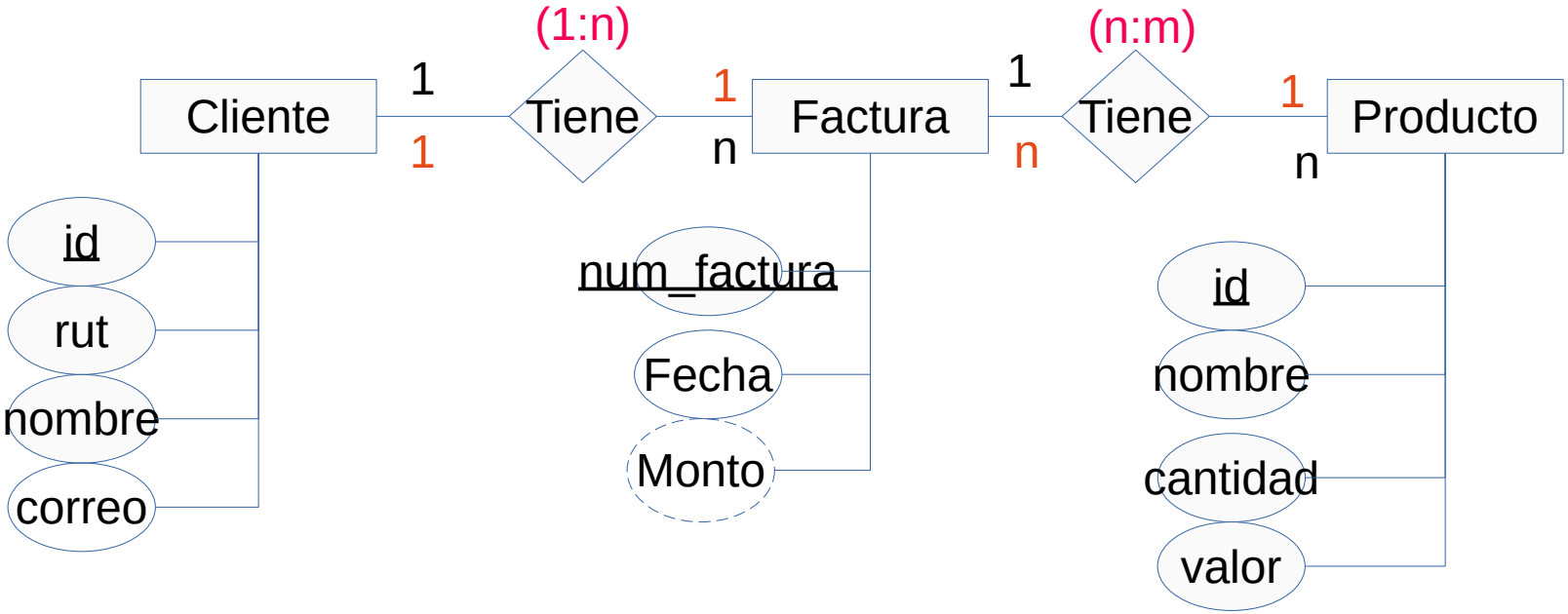


De [Entidad Relación] a [Modelo Relacional]



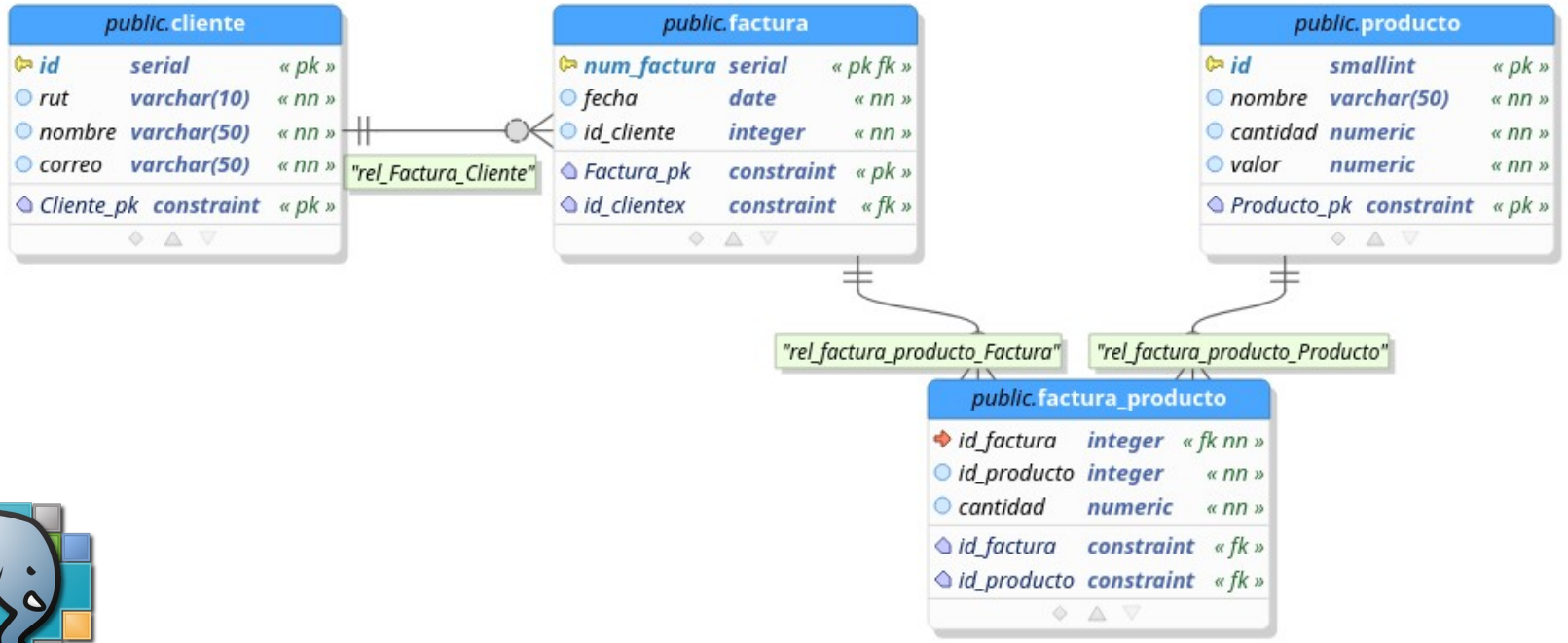
De [Entidad Relación] a [Modelo Relacional]

Ejercicio:



De [Entidad Relación] a [Modelo Relacional]

Ejercicio Solución:



De [Entidad Relación] a [Modelo Relacional]

Ejercicio Solución:

¿Y como se puede materializar este Modelo Relacional en una DB?

De [Entidad Relación] a [Modelo Relacional]

Ejercicio Solución:

¿Y como se puede materializar este Modelo Relacional en una DB?

Por medio de SQL

[Modelo Relacional] traducido a [SQL]

Ejercicio Solución:

```
CREATE DATABASE facturador3
    OWNER = postgres;
-- ddl-end --


-- object: public.cliente | type: TABLE --
-- DROP TABLE IF EXISTS public.cliente CASCADE;
CREATE TABLE public.cliente (
    id serial NOT NULL,
    rut varchar(10) NOT NULL,
    nombre varchar(50) NOT NULL,
    correo varchar(50) NOT NULL,
    CONSTRAINT "Cliente_pk" PRIMARY KEY (id)
);
```

[Modelo Relacional] traducido a [SQL]

Ejercicio Solución:

```
CREATE TABLE public.factura (
    num_factura serial NOT NULL,
    fecha date NOT NULL,
    id_cliente integer NOT NULL,
    CONSTRAINT "Factura_pk" PRIMARY KEY (num_factura)
);
```

[Modelo Relacional] traducido a [SQL]

Ejercicio Solución:

```
CREATE TABLE public.producto (
    id smallint NOT NULL,
    nombre varchar(50) NOT NULL,
    cantidad numeric NOT NULL,
    valor numeric NOT NULL,
    CONSTRAINT "Producto_pk" PRIMARY KEY (id)
);
```

[Modelo Relacional] traducido a [SQL]

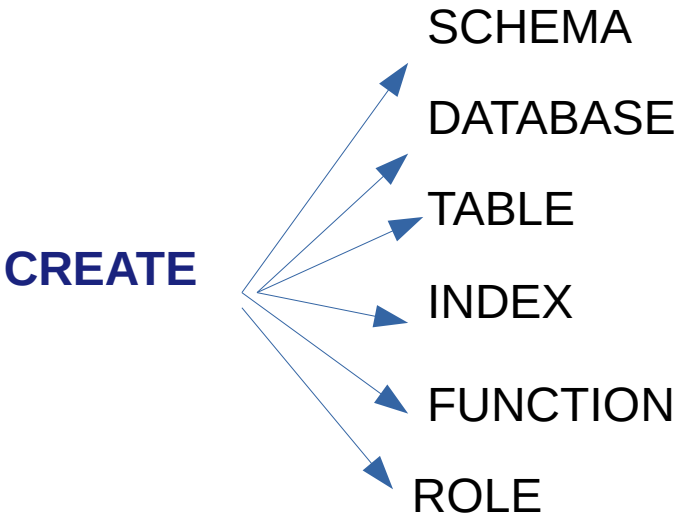
Ejercicio Solución:

```
-- object: id_clientex | type: CONSTRAINT --
-- ALTER TABLE public.factura DROP CONSTRAINT IF EXISTS id_clientex CASCADE;
ALTER TABLE public.factura ADD CONSTRAINT id_clientex FOREIGN KEY (num_factura)
REFERENCES public.cliente (id) MATCH SIMPLE
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --

-- object: id_factura | type: CONSTRAINT --
-- ALTER TABLE public.factura_producto DROP CONSTRAINT IF EXISTS id_factura CASCADE;
ALTER TABLE public.factura_producto ADD CONSTRAINT id_factura FOREIGN KEY (id_factura)
REFERENCES public.factura (num_factura) MATCH SIMPLE
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --

-- object: id_producto | type: CONSTRAINT --
-- ALTER TABLE public.factura_producto DROP CONSTRAINT IF EXISTS id_producto CASCADE;
ALTER TABLE public.factura_producto ADD CONSTRAINT id_producto FOREIGN KEY (id_factura)
REFERENCES public.producto (id) MATCH SIMPLE
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --
```

SQL -CREATE




```
[sudo] contraseña para leo:
psql (12.9 (Ubuntu 12.9-0ubuntu0.20.04.1))
Type "help" for help.

facturador3=# CREATE
ACCESS METHOD        DATABASE             FOREIGN TABLE        MATERIALIZED VIEW    ROLE                 STATISTICS           TEMPORARY            UNIQUE
AGGREGATE            DOMAIN               FUNCTION             OPERATOR             RULE                 SUBSCRIPTION         TEXT SEARCH          UNLOGGED
CAST                 EVENT TRIGGER        GROUP                POLICY               SCHEMA               TABLE                TRANSFORM            USER
COLLATION            EXTENSION            INDEX                PROCEDURE            SEQUENCE             TABLESPACE           TRIGGER              USER MAPPING FOR
CONVERSION           FOREIGN DATA WRAPPER LANGUAGE             PUBLICATION          SERVER               TEMP                 TYPE                 VIEW
```

SQL -CREATE TABLE

CREATE DATABASE nombre_db

SQL -CREATE TABLE


```
CREATE TABLE nombre_tabla (
nombre_atributo1 tipo_atributo1,
nombre_atributo2 tipo_atributo2,
...
);
```

SQL -CREATE TABLE


```
CREATE TABLE nombre_tabla (
nombre_atributo1 tipo_atributo1,
nombre_atributo2 tipo_atributo2,
...
);
```

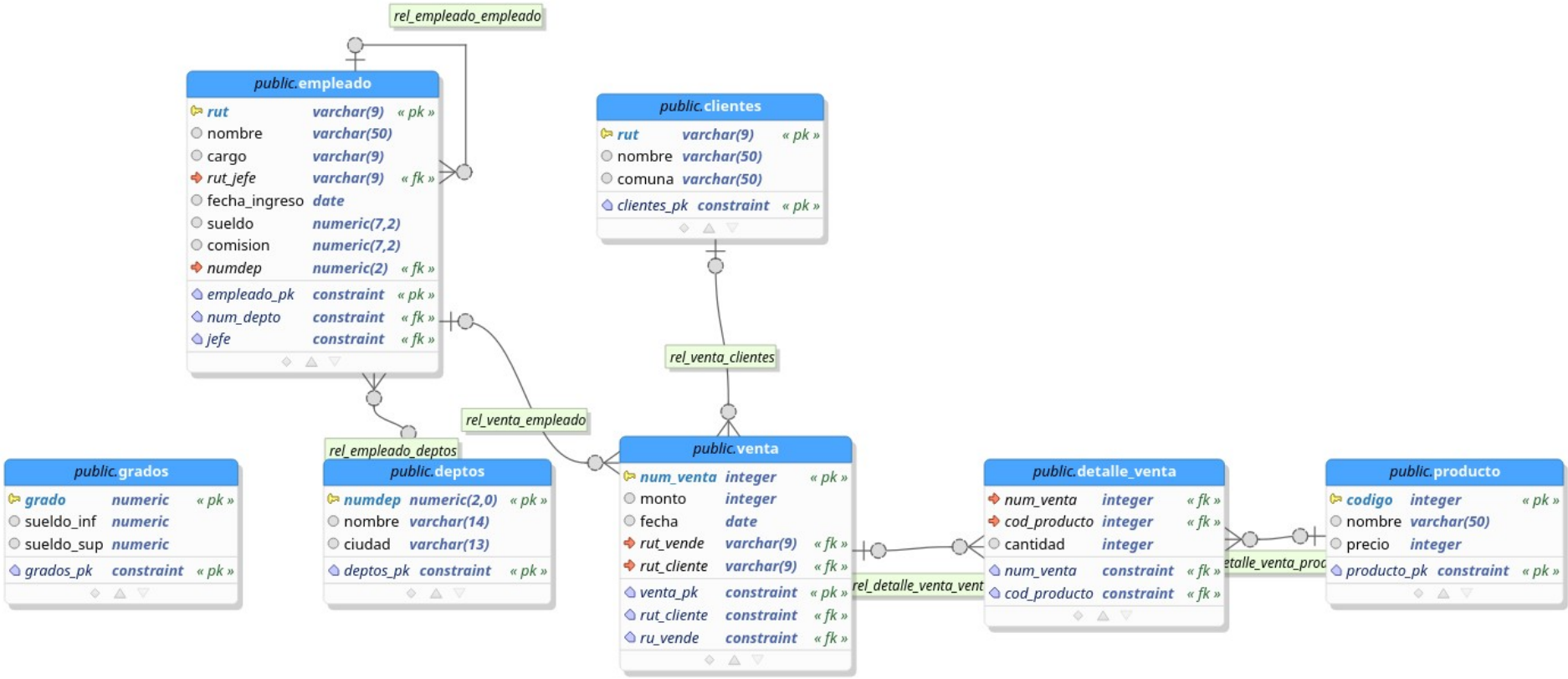
Ejercicio - Tienda

EMPLEADOS: rut, nombre, cargo, rut_jefe, sueldo, comision, numdep
DEPTOS: numdep, nombre, ciudad
GRADOS: grado, sueldo_inf, sueldo_sup
CLIENTES: rut, nombre, comuna
PRODUCTOS: codigo, nombre, precio
VENTAS: num_venta, monto, fecha, rut_vende, rut_cliente
VENTAS_DETALLE: num_venta, cod_producto, cantidad

---

1) Cree Modelo Entidad – Relación
2) Cree Modelo de Datos (Modelo Relacional)
3) Cree sentencias SQLs para crear Schema

Ejercicio - Tienda



Muchas gracias!

Correo:

leonardo.bravo@mail.udp.cl

Material Disponible en:
https://gitlab.com/l30bravo/db_udp