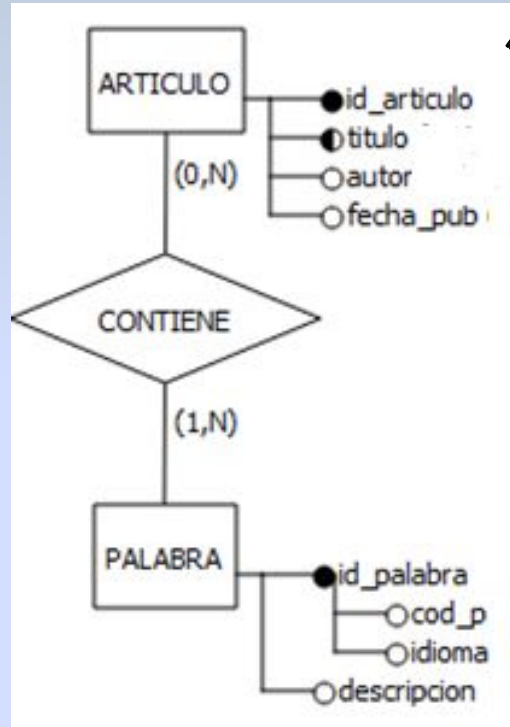


Creación de esquemas



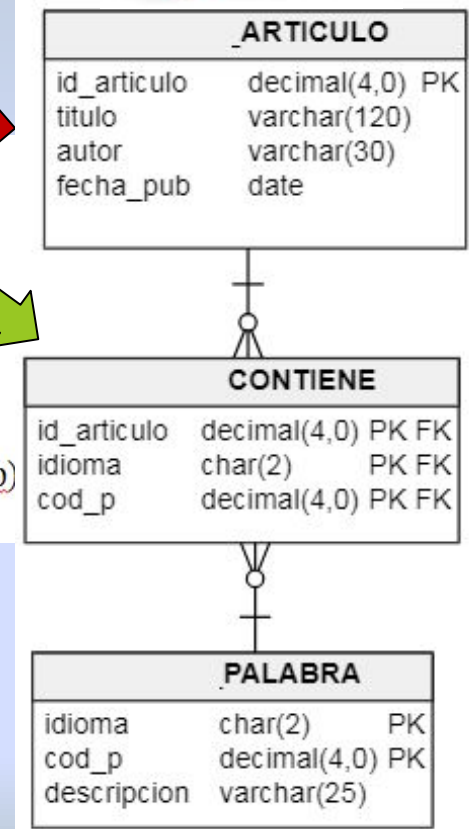
Ciclo de vida en el modelado de datos



DERExt

ARTICULO (id_articulo, titulo, autor, fecha_pub);
 PALABRA (idioma, cod_p, descripcion);
 CONTIENE (id_articulo, idioma, cod_p);
 RIR
 CONTIENE(id_articulo) << ARTICULO(id_articulo)
 CONTIENE(idioma, cod_p) << PALABRA(idioma, cod_p)

*Esquema Lógico Estándar -
Relacional*



*Esquema Lógico Tardío – Esquema
de BD Post-Relacional*

Reglas de Transformación DERExt

Recordar los algoritmos de conversión de un *Diagrama de Entidades y Relaciones Extendido (DERExt)* a *un Esquema Lógico Estándar (ELE)* basado en el Modelo Relacional.

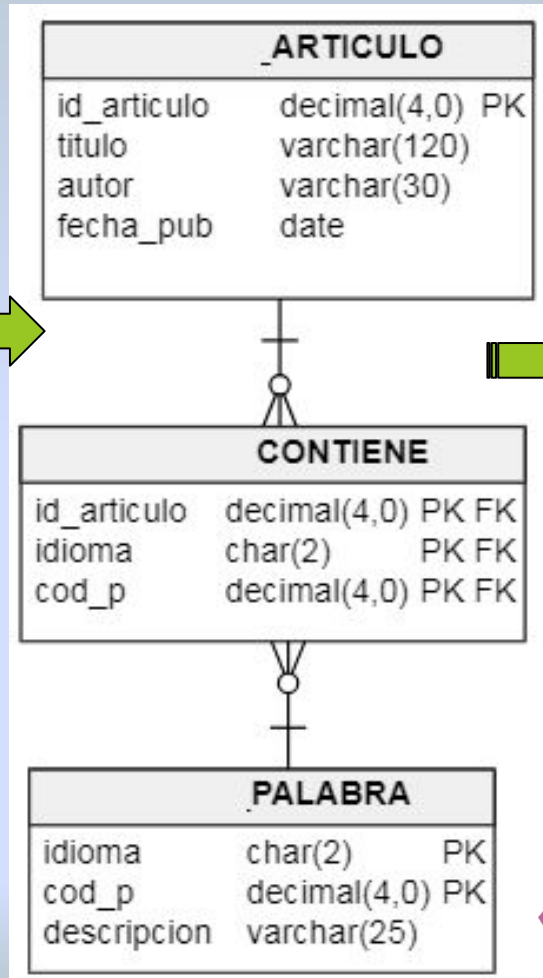
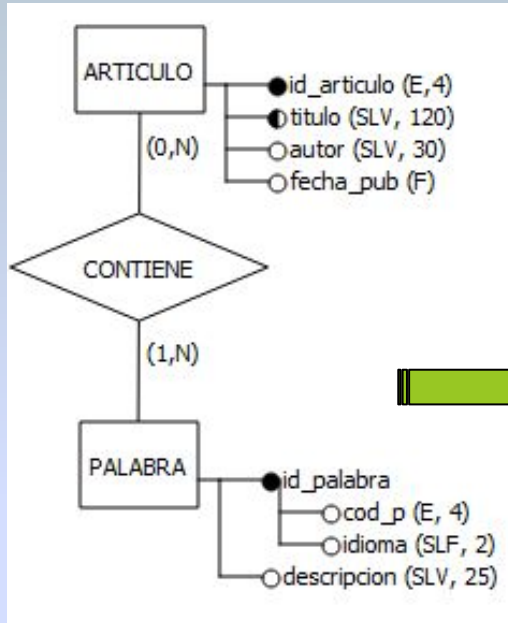
- Reglas de transformación de entidades. (Fuertes y Débiles)
- Reglas de transformación de atributos. (identificadores principales y alternativos, simples o compuestos, univaluados, mulivaluados, opcionales u obligatorios)
- Reglas de transformación de relaciones. (Unarias, binarias y ternarias, con sus diferentes cardinalidades)
- Reglas de transformación adicionales. (Agregaciones y jerarquías)

Transformar el DERExt en un Modelo Binario y obtener así el *Esquema Lógico Tardío (ELT)* o Esquema de Base de Datos (Revisar apunte de la cátedra de Estructuras de Almacenamiento de Datos)

Diseño de Bases de Datos

- En contextos de desarrollo de proyectos en general es necesario utilizar **herramientas** para el diseño de bases de datos
- La diversidad de enfoques hace difícil encontrar herramientas computacionales que permitan representar todas las reglas del modelo de datos
- Están basadas en el **Modelo Binario** (sólo soportan relaciones binarias) entonces hay que realizar las transformaciones de, atributos multivaluados, relaciones ternarias o agregaciones, atributos compuestos para lograr una transformación sin pérdida de semántica
- Algunos DBMS soportan características post-relacionales, pero a veces las herramientas no!!!

Esquema de Base de Datos



```

1  -- Created by Vertabelo (http://vertabelo.com)
2  -- Last modification date: 2018-08-09 17:11:11
3
4  -- tables
5  -- Table: TP1_E1_ARTICULO
6  CREATE TABLE TP1_E1_ARTICULO (
7      id_articulo decimal(4,0) NOT NULL,
8      titulo varchar(120) NOT NULL,
9      autor varchar(30) NOT NULL,
10     fecha_pub date NOT NULL,
11     CONSTRAINT PK_TP1_E1_ARTICULO
12     PRIMARY KEY (id_articulo)
13 );
14
15 -- Table: TP1_E1_CONTIENE
16 CREATE TABLE TP1_E1_CONTIENE (
17     id_articulo decimal(4,0) NOT NULL,
18     idioma char(2) NOT NULL,
19     cod_p decimal(4,0) NOT NULL,
20     CONSTRAINT PK_TP1_E1_CONTIENE
21     PRIMARY KEY (id_articulo, idioma, cod_p)
22 );

```

DERExt de la
cátedra de EAdD



Herramienta de Modelado de
Bases de Datos

Esquema de Base de
Datos SQL

Esquema de Base de Datos

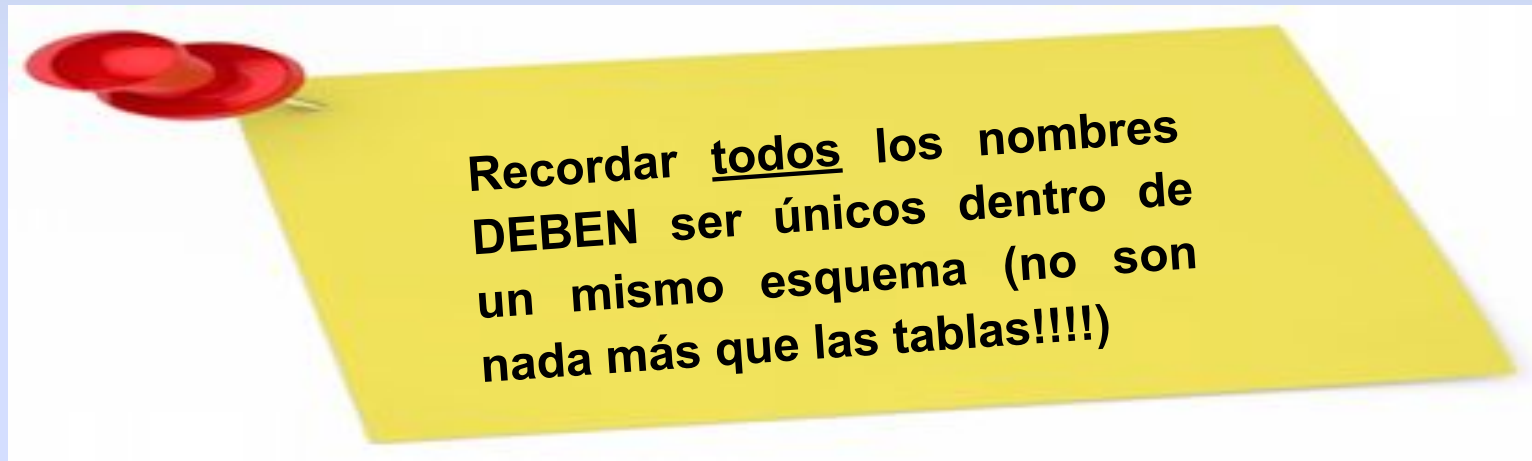
- Una base de datos relacional consiste en un **conjunto de tablas**, a cada una de las cuales se le asigna un nombre exclusivo dentro del **esquema de cada usuario**
- El conjunto de objetos (tablas, vistas, procedimientos) conforman el esquema de un usuario
- Cada **fila** de la tabla representa una relación entre un conjunto de valores. Dado que cada tabla es un conjunto de dichas relaciones, hay una fuerte correspondencia entre el concepto de tabla y el concepto matemático de relación, del que toma su nombre el modelo de datos relacional

Esquema de Base de Datos

- Cada tabla posee un conjunto de columnas cabeceras (**atributos**) cuyo nombre debe ser único dentro de la tabla
- Para cada atributo hay un conjunto de valores permitidos, llamado **dominio** de ese atributo
- Las columnas **pueden ser de distintos tipos**:
numéricos (edad, cantidad de hijos), alfanuméricos (nombre, dirección), fechas (fecha de nacimiento, fecha de ingreso a la compañía), booleano (posee auto propio, cumplió el servicio militar)

Esquema de Base de Datos

- Todos los datos registrados **en una columna deben ser del mismo tipo**



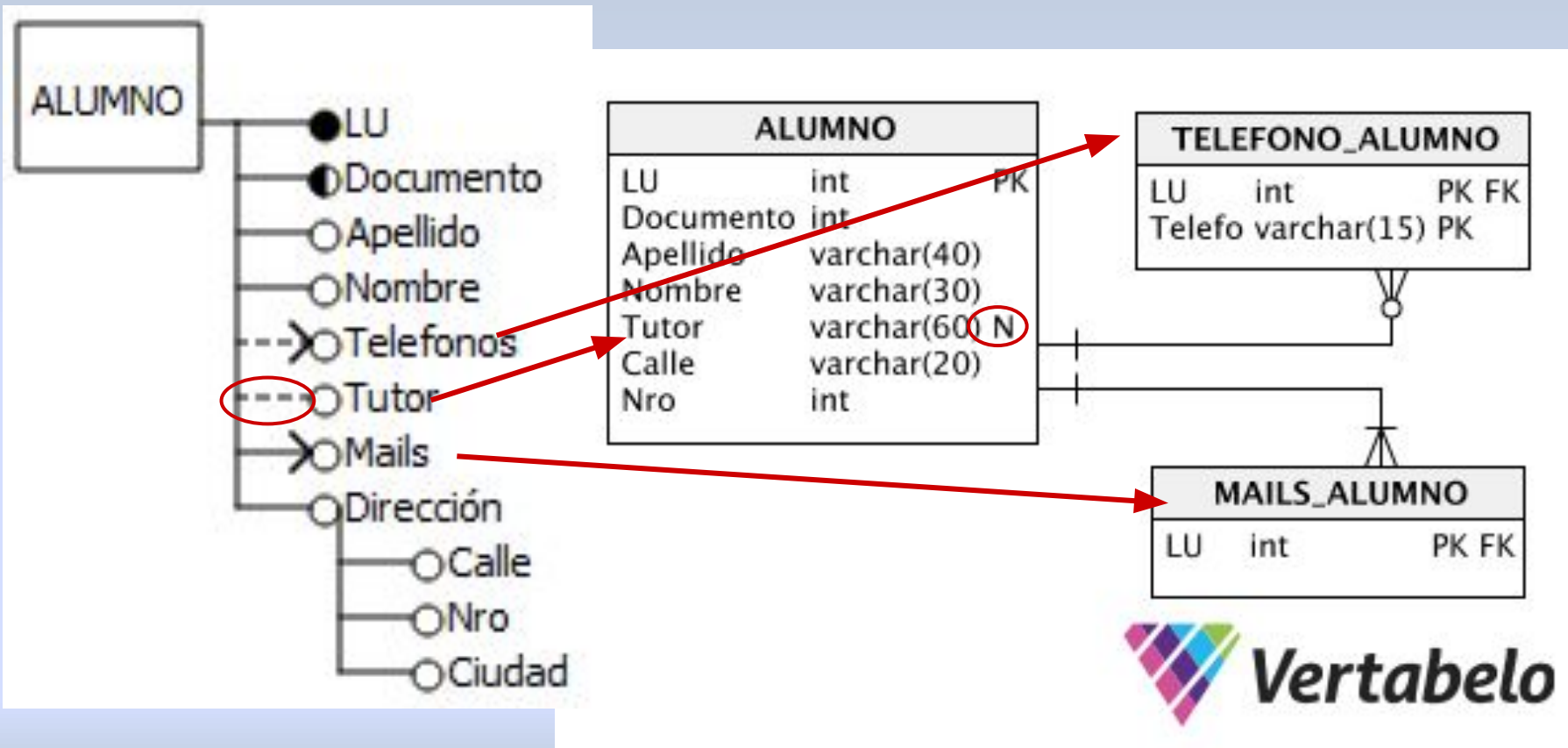
Lenguaje SQL

La definición de los datos se realiza a través de **sentencias** de DDL

- Sus **sentencias** permiten definir la semántica del esquema relacional: que tablas o relaciones se establecen, sus posibles valores (dominios), asociaciones, restricciones y demás objetos.
- Los datos o información de dichas tablas las guarda el SGBD en tablas propias, denominadas tablas de **metadatos**
- La tabla de una BD relacional está basada en la relaciones del modelo relacional (Visto en EAdD)

Creación de tablas

Aplicando las reglas de derivación



Creación de tablas

Cada columna debe tener un determinado tipo de dato

El tipo de dato limita el conjunto de valores posibles que se pueden asignar a una columna

ALUMNO		
LU	int	PK
Document	int	
Apellido	varchar(40)	
Nombre	varchar(30)	
Tutor	varchar(60)	N
Calle	varchar(20)	
Nro	int	
Ciudad	varchar(30)	



```
CREATE TABLE ALUMNO(  
    LU            integer      NOT NULL,  
    Documento    integer      NOT NULL,  
    Apellido      varchar(40)  NOT  
NULL,  
    Nombre       varchar(30)   NOT  
NULL,  
    Tutor        varchar(60),  
    Calle        varchar(20)   NOT NULL,  
    Nro          integer       NOT NULL,  
    Ciudad       varchar(30)   NOT NULL,  
    CONSTRAINT PK_ALUMNO PRIMARY  
KEY (LU));
```

Puede colocarse la definición de la clave primaria en sentencia aparte

```
ALTER TABLE ALUMNO  
ADD CONSTRAINT PK_ALUMNO  
PRIMARY KEY (LU);
```

Tipos de Datos PostgreSQL



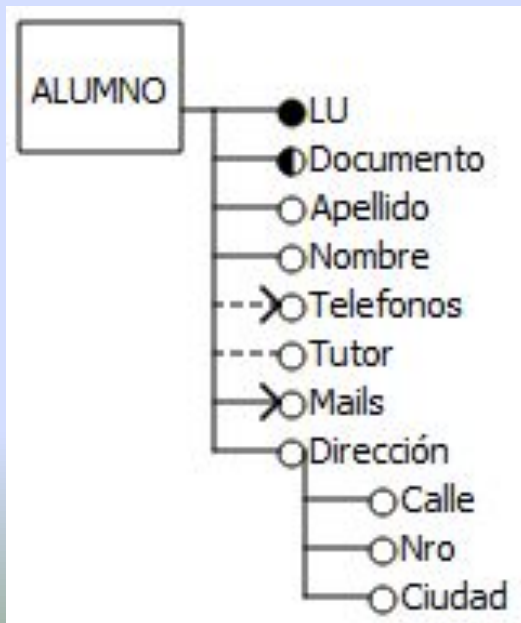
- PostgreSQL incluye un conjunto **considerable** de *tipos de datos básicos* que se adaptan a muchas aplicaciones
- Algunos de los tipos de datos más utilizados son *integer* para números enteros, *numeric* para los números fraccionarios, *char* y *varchar* para cadenas de caracteres de longitud fija o variable, *date* para fechas, *time* para horas del día, y *timestamp* para los valores que contienen fecha y hora, entre otros

En <https://www.postgresql.org/docs/<version>/static/ddl.html> se encuentra una lista completa de todos los tipos de datos soportados por PostgreSQL

Restricciones de Clave Primaria y Clave Alternativa

En algunos casos la clave primaria está compuesta por más de un atributo, pero la cláusula de PRIMARY KEY sigue siendo **UNA SOLA**, no olvidarlo!!

Las claves alternativas se especifican con la cláusula UNIQUE



```
CREATE TABLE ALUMNO(  
    LU integer NOT NULL,  
    Documento integer UNIQUE  
    Apellido varchar(40) NOT NULL,  
    Nombre varchar(30) NOT NULL,  
    .....  
);
```


Restricciones de Integridad Referencial (RIRs)

Regla (Restricción) de Integridad Referencial (RIR): *“Si en una tabla hay una clave extranjera, sus valores deben coincidir con alguno de los valores de la clave primaria de la tabla a la que hace referencia, o bien deben ser nulos”*

En SQL existen dos maneras de colocar las claves extranjeras:

Sentencia de creación de la tabla (CREATE TABLE)

Sentencia de actualización de la tabla (ALTER TABLE). Por ejemplo si la tabla ya está creada

SQL: Cláusula FOREIGN KEY

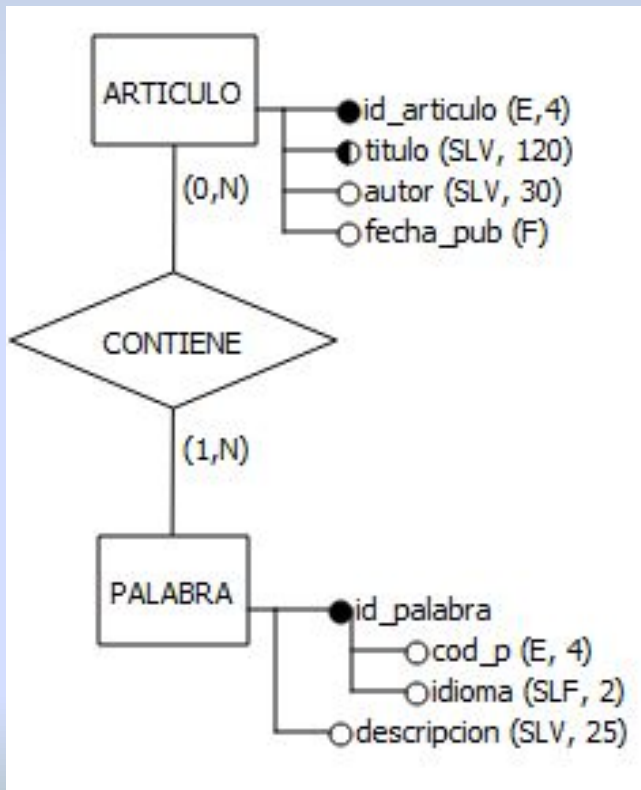
CREATE TABLE NombreTabla (Tabla Referenciante)

.....
, FOREIGN KEY (ListaColumnasReferenciantes)
REFERENCES NombreTablaReferenciada
[(ListaColumnasReferenciadas)]

- ✓ La **ListaColumnasReferenciantes** es una lista de atributos que se corresponden con **ListaColumnasReferenciada** que es la lista de atributos de la clave primaria o alternativa de la tabla **Referenciada**
- ✓ No se requiere igualdad de nombres, pero sí compatibilidad de tipos e datos
- ✓ Si la clave extranjera es de una única columna, puede especificarse directamente en la definición de la columna con la cláusula **REFERENCES NombreTablaReferenciada**

Un ejemplo...

Partimos de un DEREExt y debemos recordar las [reglas de derivación](#) para poder construir el esquema relacional bajo el **modelo relacional**



Claves primarias subrayadas y alternativas doble subrayado

ARTICULO (id_articulo, titulo, autor, fecha_pub, nacional)

PALABRA (cod_p, idioma, descrip)

CONTIENE (id_articulo, cod_p, idioma)

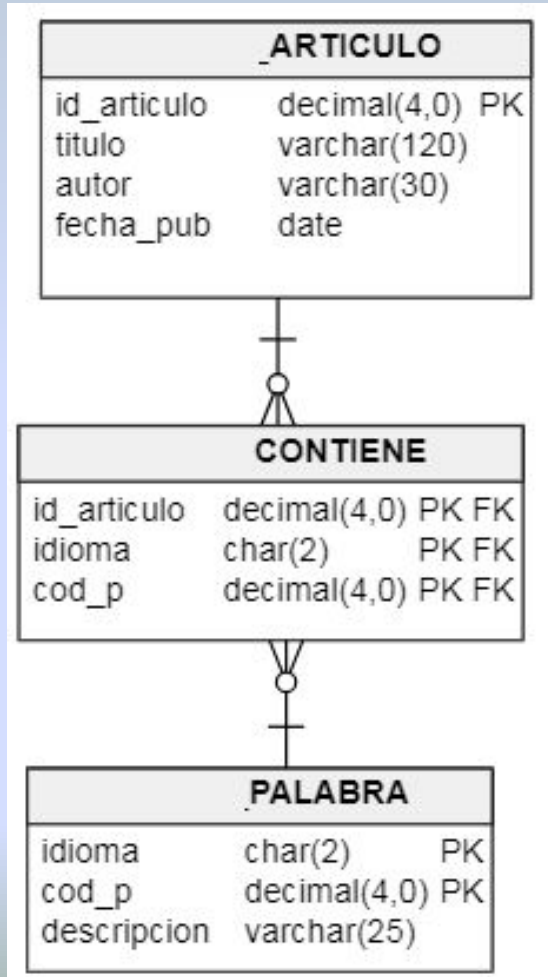
RIRs o claves alternativas o foreign key

CONTIENE(cod_p, idioma) << PALABRA (cod_p, idioma)

CONTIENE(id_articulo) << ARTICULO (id_articulo)

Un ejemplo (cont)

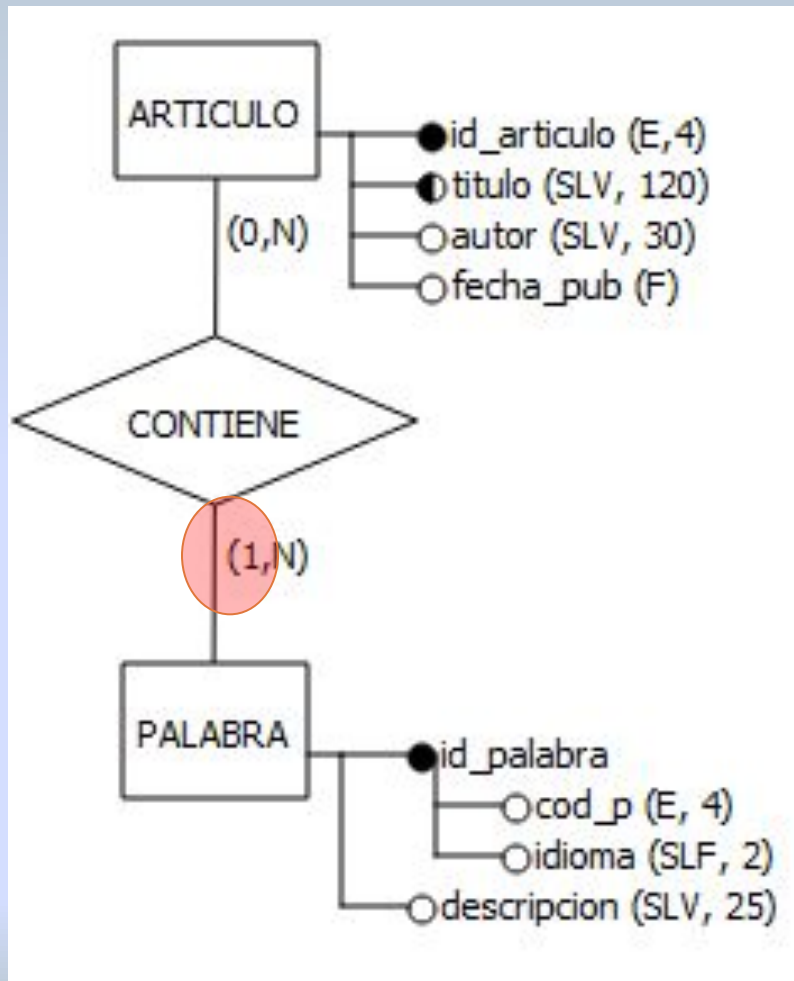
El esquema de base de datos se completa con la definición de los dominios (*por ahora sólo los tipos de datos*) de cada uno de los atributos.



```
CREATE TABLE ARTICULO(  
    id_articulo integer NOT NULL,  
    titulo varchar(150) NOT NULL UNIQUE,  
    autor varchar(60) NOT NULL,  
    nacionalidad varchar(15) NOT NULL,  
    fecha_pub date NOT NULL,  
    CONSTRAINT PK_ARTICULO PRIMARY KEY  
    (id_articulo)  
);
```

.....

Qué me quedó por representar?



ARTICULO (id_articulo, *titulo*, autor, fecha_pub, nacional)

PALABRA (cod_p, idioma, descrip)

CONTIENE (id_articulo, cod_p, idioma)

No debo olvidarme que en el DERExt representé que “***todo artículo debe contener al menos una palabra***”

Generar un script de creación de esquemas

- Un script de creación de esquemas es un archivo de texto, por lo general extensión .sql, que contiene un conjunto de sentencias SQL.
- Cada uno de esas sentencias SQL **NO** debe tener errores de sintaxis, caso contrario el script **aborta**.
- Cada desarrollador que los confecciona **debe testearlos completos** en una única ejecución completa.



Generar un script de creación de esquemas

```
-- SCRIPT DE CREACIÓN DE ESQUEMA EJERCICIO 1 TP 1
-- CURSADA 2017
-- VIVIANA FERRAGGINE
```

```
-- BORRADO DE TABLAS
```

```
--
DROP TABLE ARTICULO CASCADE;
DROP TABLE CONTIENE CASCADE;
DROP TABLE PALABRA CASCADE;
```

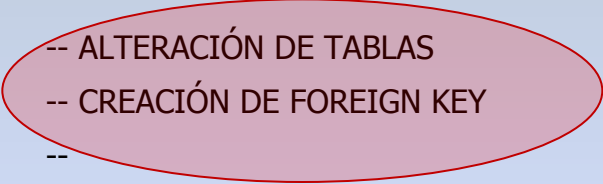
```
-- CREACIÓN DE TABLAS
```

```
--
CREATE TABLE ARTICULO(
    id_articulo integer NOT NULL,
    titulo      varchar(150) NOT NULL UNIQUE,
    autor       varchar(60)  NOT NULL,
    nacionalidad varchar(15) NOT NULL,
    fecha_pub   date        NOT NULL,
    CONSTRAINT PK_ARTICULO PRIMARY KEY (id_articulo));
```

```
CREATE TABLE CONTIENE(
    id_articulo integer NOT NULL,
    cod_p       integer NOT NULL,
    idioma      char(2)  NOT NULL,
    CONSTRAINT PK_CONTIENE PRIMARY KEY (id_articulo,
    cod_p, idioma)
);
```

```
CREATE TABLE PALABRA(
    cod_p integer NOT NULL,
    idioma char(2) NOT NULL,
    descripcion varchar(60) NOT NULL,
    CONSTRAINT PK_PALABRA PRIMARY KEY (cod_p,
    idioma)
);
```

Generar un script de creación de esquemas (cont)



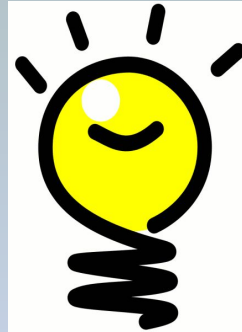
```
-- ALTERACIÓN DE TABLAS  
-- CREACIÓN DE FOREIGN KEY
```

```
--
```

```
ALTER TABLE CONTIENE  
    ADD CONSTRAINT FK_CONTIENE_ARTICULO  
    FOREIGN KEY (id_articulo)  
    REFERENCES ARTICULO(id_articulo);
```

```
ALTER TABLE CONTIENE  
    ADD CONSTRAINT FK_CONTIENE_PALABRA  
    FOREIGN KEY (cod_p, idioma)  
    REFERENCES PALABRA(cod_p, idioma);
```

Tips útiles



- Al resolver los ejercicios prácticos en tu cuenta de PostgreSQL (en tu esquema), es **útil** agregar un prefijo a las tablas que indique a que práctico/ejercicio corresponden ... te ayudará a mantener ordenadas las resoluciones de los ejercicios.

Por ejemplo

CREATE TABLE TP1_EJ1_ARTICULO.... Porque es probable que te encuentres con otra tabla ARTICULO (diferente) en otro práctico



Un ejemplo (cont)

Script de creación de esquema

[..\Prácticos\script creacion esq ejemplo.sql](#)

Un ejemplo con errores para “levantar”

[..\Prácticos\script creacion esq ejemplo con errores.sql](#)

Borrado de tablas

DROP TABLE nombre_tabla [CASCADE | RESTRICT]

→ *Se elimina la definición de la tabla y todas las filas que contiene*

- Si es RESTRICT, se rechaza si hay objetos definidos a partir de la tabla (es la opción por defecto)
- Si es CASCADE, se eliminan todos los objetos dependientes de la tabla (también los objetos que dependan a su vez de ellos)

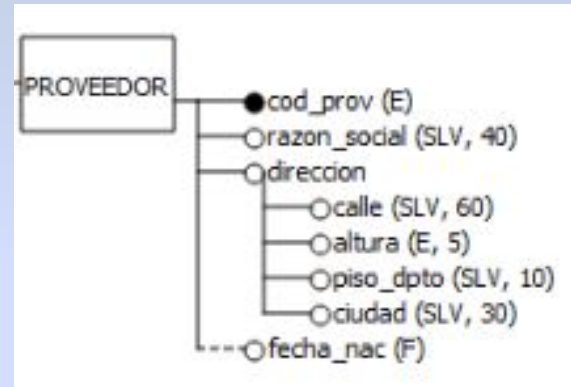
→ *tener precaución en su uso*



Características post-relacionales

```
CREATE TYPE tipo_direccion as (  
  calle varchar(60),  
  altura numeric(5,0),  
  piso_dpto varchar(10),  
  ciudad varchar(30)  
);
```

```
CREATE TABLE PROVEEDOR(  
  cod_prov integer NOT NULL,  
  razon_social varchar(40) NOT NULL UNIQUE,  
  direccion tipo_direccion not null,  
  fecha_nac date,  
  fecha_pub date NOT NULL,  
  CONSTRAINT PK_PROVEEDOR  
    PRIMARY KEY (cod_prov)  
);
```





Vertabelo

<https://www.vertabelo.com/>

Vertabelo - Design Your Database

<https://www.vertabelo.com/>

Incognito (2)

100 Shares

Twitter LinkedIn Reddit YouTube Facebook Google+

DESIGN YOUR DATABASE ONLINE

Easy way for clean database design

Try it now for free

Watch it in action

Created with Vertabelo

Our website uses cookies. By using this website, you agree to their use in accordance with the browser settings. You can modify your browser settings or learn more about our cookie policy. [Privacy Policy](#)

Esperando a www.vertabelo.com...

Escribe aquí para buscar

15:58 13/08/2019

The screenshot displays the Vertabelo website's homepage. At the top, there's a navigation bar with links for FEATURES, PRICING, SUPPORT, LEARN SQL, and BLOG, along with 'Log in' and 'Sign up' buttons. The main content area features a large heading 'DESIGN YOUR DATABASE ONLINE' and a subheading 'Easy way for clean database design'. Below this is a 'Try it now for free' button and a 'Watch it in action' button with a play icon. To the right, a database design diagram is shown, featuring tables like 'product', 'product_type', 'shipment_details', and 'shipment_type' with their respective fields and relationships. A sidebar on the left shows social media sharing options. At the bottom, there's a cookie consent banner and a Windows taskbar with various application icons.



Vertabelo

<https://www.vertabelo.com/>

Vertabelo - Design Your Database

https://www.vertabelo.com/

100 Shares

Twitter LinkedIn Reddit YouTube Facebook Google+

DESIGN YOUR DATABASE ONLINE

Easy way for clean database design

Try it now for free

Watch it in action

Log in Sign up

CLICK HERE

Created with Vertabelo

Our website uses cookies. By using this website, you agree to their use in accordance with the browser settings. You can modify your browser settings or learn more about our cookie policy. For more information see our Privacy Policy

Esperando a www.vertabelo.com...

Escribe aquí para buscar

15:58 13/08/2019



Vertabelo

Configuración x Reciente - Google Drive x BDI_2019_02_CreacionEsquema x Creación Cursada 2018 - Bases x Vertabelo x

my.vertabelo.com/sign-up/create-account


Vertabelo Help SIGN UP Log in


1. Create account

Create an account

Create your Vertabelo Account in less than 1 minute.

Already have an account? [Log in here.](#)

 **CLICK HERE**

 Student or lecturer? [Register here »](#)

Name:




Email:

Password:

☐ I agree to [Terms of service](#) and [Privacy policy](#)

[CREATE ACCOUNT](#) [Cancel](#)

Request a feature Report a problem Ask a question About us Terms of service Privacy policy Documentation

Follow us on:   

Vertabelo v. 19.2.2 All rights reserved

Powered by e-point SA internet software house

ES 10:08 21/08/2019



Vertabelo

Configuración x Reciente - Google Drive x BDI_2019_02_CreacionEsquema x Creación Cursada 2018 - Bases x Vertabelo x

my.vertabelo.com/sign-up/create-academic

Create an Academic Account

Academic Accounts are intended for educational purposes only. They are granted to students and lecturers learning or teaching database design.

Please, keep in mind that even if you're a faculty member you're not allowed to use this account for any other purpose than teaching database modeling. Also, this account isn't meant for non-profit organizations or start-ups.

• **Name:**

Status: ☐ Lecturer ☒ Student

• **Academic institution:**

• **Official website:**

• **Course title:**

• **choose month:**

• **Email:**

• **Set password:**

☐ • I agree to [Terms of service](#) and [Privacy policy](#)



Vertabelo



<https://my.vertabelo.com/drive>

Documents | My account | Help ▾

Viviana Ferraggine

2

My Vertabelo

Name ▾	Owners	Last modified
 MySQL demo database model	Viviana Ferraggine	2018-08-09 11:17
 Sample SQL script	Viviana Ferraggine	2018-08-09 11:17

Documents area

This is where all your diagrams, connections are shown.

● ● ● ● ● ● ● ●

[Skip](#) [← Back](#) [Next →](#)



Vertabelo



Es seguro | <https://my.vertabelo.com/drive>



Documents

My account

Help ▾



3

My Vertabelo

Shared

Recent

Trash

Navigation tree

See all your folders within Vertabelo Documents.



Skip

← Back

Next →



Sample SQL script




Owners

Viviana

Viviana



Vertabelo

 Viviana Ferraggine   Log out

4

Properties panel

When you select a document, its properties are displayed in this panel. Follow activities related to a document or find its details here.

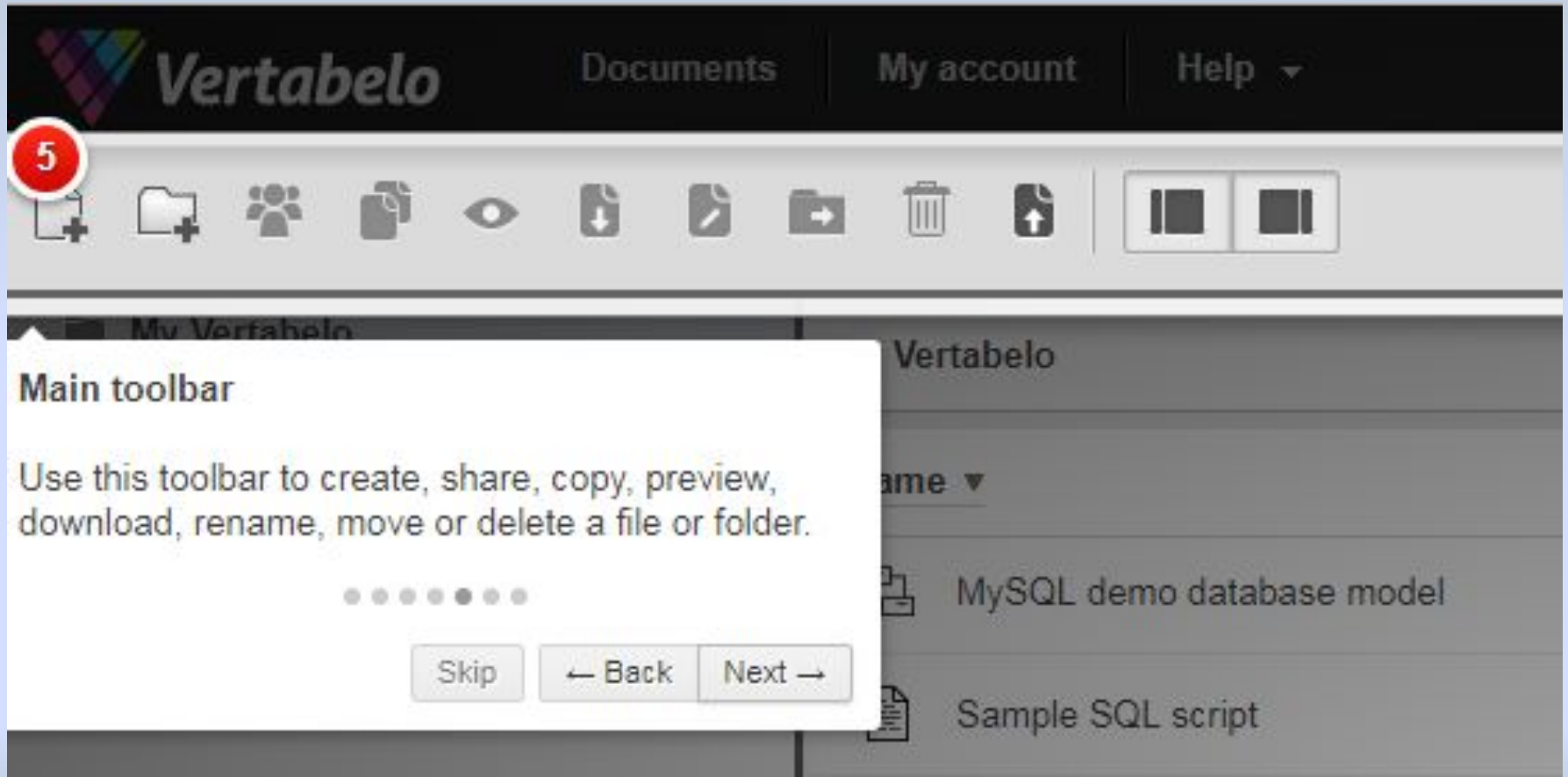
● ● ● ● ● ● ● ●

Skip ← Back Next →

My Vertabelo

Activity

Details



The screenshot shows the Vertabelo web interface. At the top is a dark navigation bar with the Vertabelo logo, "Documents", "My account", and "Help" with a dropdown arrow. Below this is a light gray toolbar containing icons for creating files/folders, sharing, copying, previewing, downloading, renaming, moving, and deleting. To the right of the toolbar are two window management icons. A red circle with the number "5" is positioned over the first icon (create file/folder). A white tutorial overlay is in the foreground, titled "Main toolbar", explaining the toolbar's functions and providing navigation buttons: "Skip", "← Back", and "Next →". The background shows a file list with items like "MySQL demo database model" and "Sample SQL script".

5

Vertabelo Documents My account Help ▾

Main toolbar

Use this toolbar to create, share, copy, preview, download, rename, move or delete a file or folder.

● ● ● ● ● ● ●

Skip ← Back Next →

My Vertabelo

Vertabelo

name ▾

MySQL demo database model

Sample SQL script



Vertabelo

Vertabelo my.vertabelo.com/drive Sebastian Villar 2 Log out

Documents My account Help

My Vertabelo

- Labs
- Shared
- Recent
- Trash

Name	Owners	Last modified
Labs	Sebastian Villar	2019-04-22 15:15
MySQL demo database model	Sebastian Villar	2019-03-22 14:57
Sample SQL script	Sebastian Villar	2019-03-22 14:57
test	Sebastian Villar	2019-07-15 20:40
TPE	Gustavo A. Correa Reina, Viviana	2019-08-13 14:43
Unc Peliculas	Viviana Ferraggine, Sebastian	2019-08-12 17:01

My Vertabelo

Activity Details

You edited [Unc Peliculas](#).
2019-08-12 16:58

Viviana Ferraggine changed Sebastian Villar's permissions to [Unc Peliculas](#) to owner.
2019-08-12 16:54

Viviana Ferraggine shared [Unc Peliculas](#) to Sebastian Villar.
2019-08-12 16:52

Viviana Ferraggine edited [Unc Peliculas](#).
2019-08-12 16:51

Viviana Ferraggine edited [Unc Peliculas](#).
2019-08-12 16:45



Vertabelo

Create new model

Choose your database engine and click Start modeling button

★ Model name:

★ Database engine:



PostgreSQL 9.x



IBM DB2 9.7



Oracle Database 11g/12c



Microsoft SQL Server 2012 &
2014 & 2016



MySQL 5.x



HSQLDB 2.3.x



SQLite 3.x

★ Initial model:

Start working with an empty diagram.

START MODELING

[★] Obligatory fields

6



7

Main toolbar

The toolbar contains icons for file operations, copy/paste, undo, redo, and generation, among others.

Toolbox

Use toolbox buttons to select and add elements in the diagram.

4

Model

- + Tables
- + References
- + Sequences
- + Text notes
- + Views

Navigation tree

Browse all objects in the model. Click the element name to select the element. Double-click the element name to find the element in the diagram.



Skip

← Back

Next →



Vertabelo

Vertabelo

my.vertabelo.com/model/RnDb2I3Vwty4RPyJHpNKpmEEUXp8Tlxf

Incógnito (2)

Sebastian Villar

Log out

Ejemplo (Edit mode)

MODEL STRUCTURE

Model

- Tables
 - Table_1
- References
- Sequences
- Text notes
- Views

TABLE PROPERTIES

SQL preview

General

Name: Table_1

Comment:

Columns + Add column

Primary key

Alternate (unique) keys

Indexes

Checks

Additional SQL scripts

Additional properties

Format

TABLE PROBLEMS

Errors (1)

Table must have at least one column.

Warnings (1)

You should change default table name.

PROBLEMS (2)

Bibliografía

La documentación de PostgreSQL Versión 11 se encuentra en:

<https://www.postgresql.org/docs/11/index.html>

Dentro del capítulo II ***The SQL Language*** en el ítem 5. **Data Definition** se encuentra detalles para la creación y actualización de la definición de las tablas como:

- 5.1. Table Basics - <https://www.postgresql.org/docs/11/ddl-basics.html>
- 5.2. Default Values - <https://www.postgresql.org/docs/11/ddl-default.html>
- 5.3. Constraints - <https://www.postgresql.org/docs/11/ddl-constraints.html>
- 5.3.2. Not-Null Constraints - <https://www.postgresql.org/docs/11/ddl-constraints.html#id-1.5.4.5.6>
- 5.3.3. Unique Constraints - <https://www.postgresql.org/docs/11/ddl-constraints.html#DDL-CONSTRAINTS-UNIQUE-CONSTRAINTS>
- 5.3.4. Primary Keys - <https://www.postgresql.org/docs/11/ddl-constraints.html#DDL-CONSTRAINTS-PRIMARY-KEYS>
- 5.3.5. Foreign Keys - <https://www.postgresql.org/docs/11/ddl-constraints.html#DDL-CONSTRAINTS-FK>
- 5.5. Modifying Tables - <https://www.postgresql.org/docs/11/ddl-alter.html>

Bibliografía

Dentro del capítulo II ***The SQL Language*** en el ítem 8. **Data Types** se encuentra detalles para los diferentes tipos de datos soportados por Postgresql en:

<https://www.postgresql.org/docs/11/datatype.html>

Más detalles de las sentencias anteriores pueden encontrarse en <https://www.postgresql.org/docs/11/sql-commands.html>

La herramienta de diseño de bases de datos Vertabelo se encuentra en: **www.vertabelo.com**

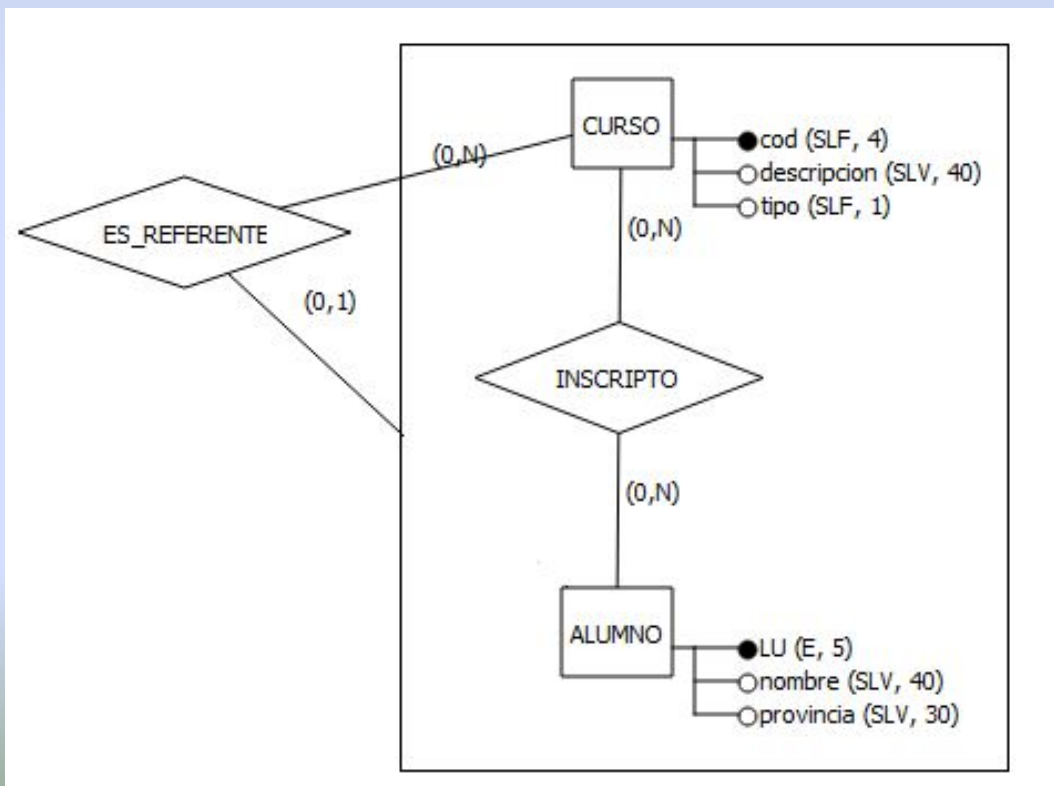
Bibliografía

- Diseño de Bases de Datos Relacionales. Adoración de Miguel Castaño, Mario Piattini Velthuis, Esperanza Marcos Martínez. Editorial Ra-Ma. 2000. Capítulo 2.
- Fundamentos de Bases de Datos. Abraham Silberschatz, Henry F. Korth, S. Sudarshan. Cuarta edición. MCGRAW-HILL/INTERAMERICANA DE ESPAÑA, S. A. U.. 2002. Capítulo 2.
- Database Management Systems. Raghu Ramakrishnan, Johannes Gehrke. MCGRAW-HILL. Second Ed. Capítulo 1.



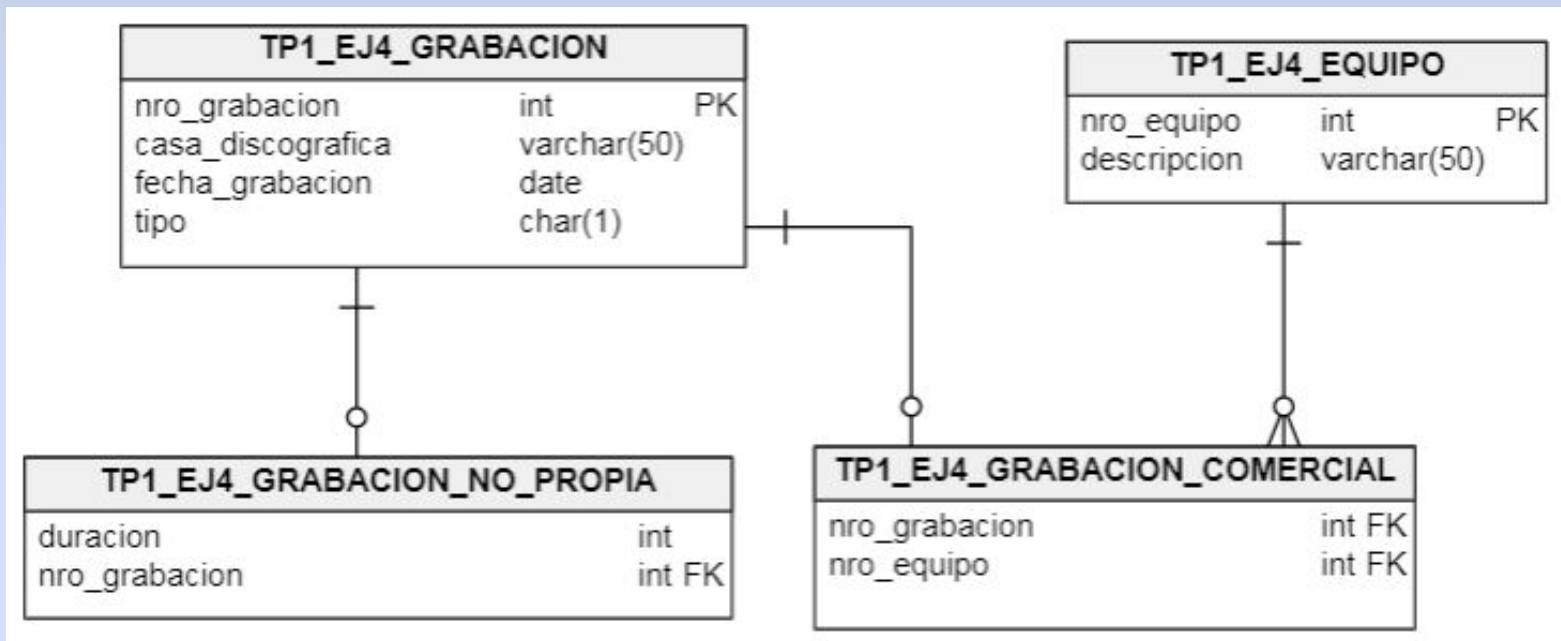
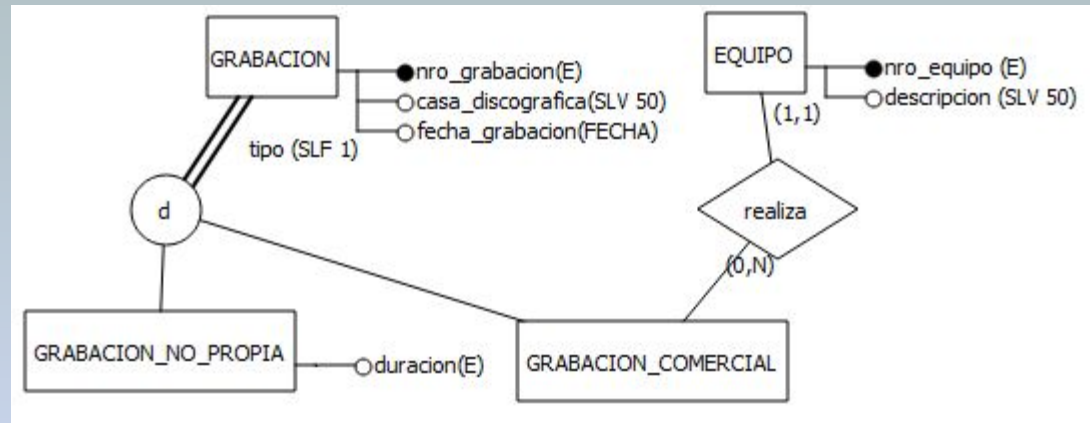
Vertabelo

Tratemos de llevar a Vertabelo el DERExt del Ejercicio 2 del TP 1 y generar el script SQL de creación del esquema. Nota: primero derivá las tablas al modelo relacional (o modelo binario)



Repaso

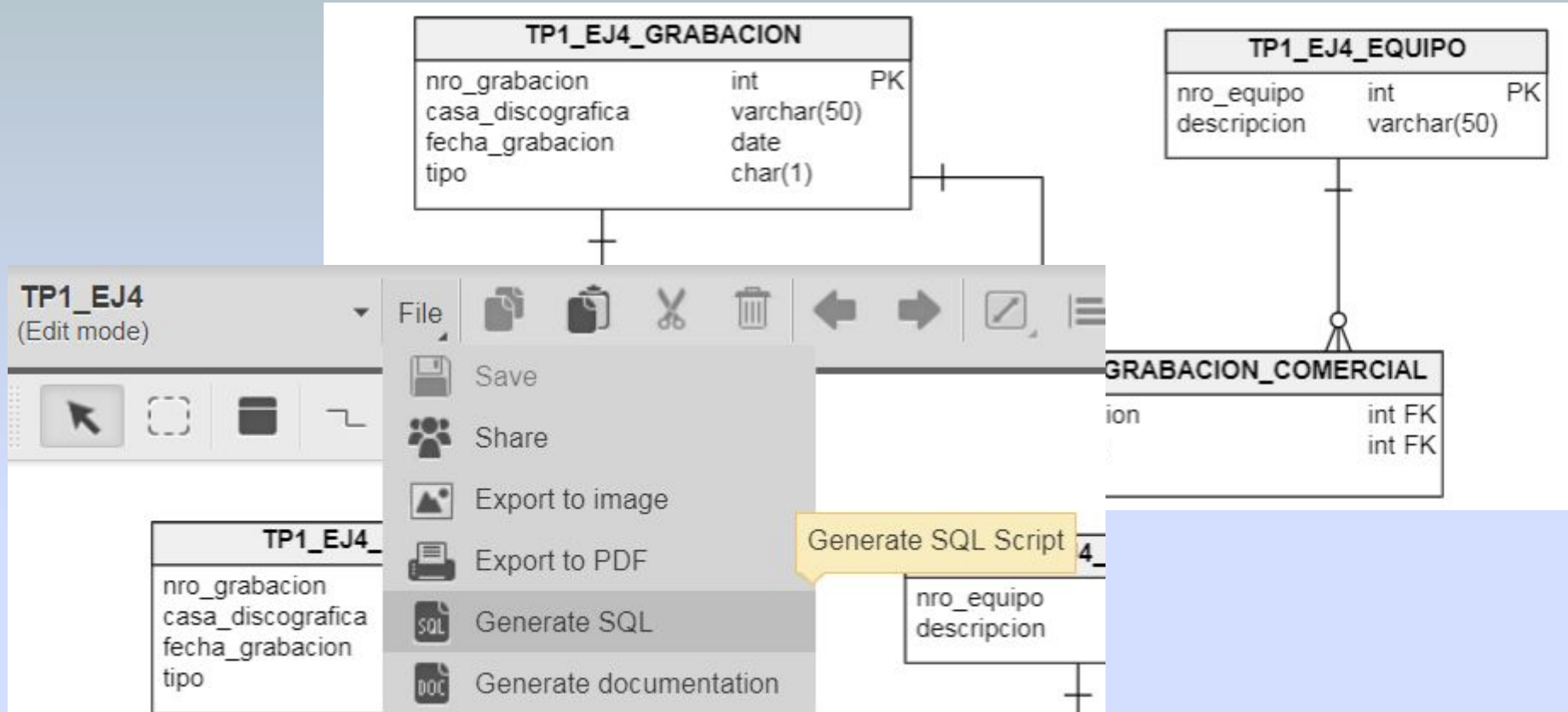
DERExt Ej 4 TP 4



<https://my.vertabelo.com/login>
New Physical Data Model en
Vertabelo para PostgreSQL



Repaso



Generar (Generate) y Salvar (Save) el script sql con las sentencia de creación y alteración de tablas del esquema en cuestión.

Repaso

```
TP1_EJ4_create.sql
1  -- Created by Vertabelo (http://vertabelo.com)
2  -- Last modification date: 2019-08-23 20:47:23.375
3
4  -- tables
5  -- Table: TP1_EJ4_EQUIPO
6  CREATE TABLE TP1_EJ4_EQUIPO (
7      nro_equipo int NOT NULL,
8      descripcion varchar(50) NOT NULL,
9      CONSTRAINT TP1_EJ4_EQUIPO PRIMARY KEY (nro_equipo)
10 );
11
12 -- Table: TP1_EJ4_GRABACION
13 CREATE TABLE TP1_EJ4_GRABACION (
14     nro_grabacion int NOT NULL,
15     casa_discografica varchar(50) NOT NULL,
16     fecha_grabacion date NOT NULL,
17     tipo char(1) NOT NULL,
18     CONSTRAINT PK_TP1_EJ4_GRABACION PRIMARY KEY (nro_grabacion)
19 );
```

Seleccionar el texto del script y ejecutarlo en
<http://dbases.exa.unicen.edu.ar/phppgadmin/>

Repaso

En <http://dbases.exa.unicen.edu.ar/phpPgAdmin/> ingresar con usuario **unc_nro_libreta** y ejecutar código SQL paracrear las tablas en la BD **Postgresql**

phpPgAdmin - SQL - Google Chrome

No es seguro | dbases.exa.unicen.edu.ar/phpPgAdmin/sqledit.php?action=sql&server=127.0.0.1:54...

SQL? Buscar

Servidor?: PostgreSQL (127.0.0.1:5432:allow) ▼ Base de Datos?: cursada ▼

Ruta de la búsqueda en los esquemas?:
unc_

```
-- Created by Vertabelo (http://vertabelo.com)
-- Last modification date: 2019-08-23 20:47:23.375

-- tables
-- Table: TP1_EJ4_EQUIPO
CREATE TABLE TP1_EJ4_EQUIPO (
    nro_equipo int NOT NULL,
    descripcion varchar(50) NOT NULL,
    CONSTRAINT TP1_EJ4_EQUIPO PRIMARY KEY (nro_equipo)
);
```

o subir un script SQL: Seleccionar archivo Ningún archivo seleccionado

☐ Paginar resultados

Ejecutar Reestablecer

subir al inicio ▼