

## Conceptos que aborda la práctica:

- Creación de bases de datos, tablas y dominios definidos por el usuario.
- Modificación sobre tablas y dominios
- Restricciones de clave primaria, de clave candidata y de clave foránea.
- Integridad referencial
- Ingeniería hacia adelante y hacia atrás.

## Objetivos de los Ejercicios:

- Conocer las sentencias que conforman el lenguaje de creación de datos (DDL – Data Definition Language) de SQL, es decir, el subconjunto de sentencias que nos permitirán crear, modificar o borrar los objetos de una base de datos.
- Investigar sobre la instalación e instalar el software necesario para llevar a cabo las actividades de la asignatura, en pos de lograr las competencias de una persona analista de sistemas, cuando se trata de utilizar nuevas tecnologías para el desarrollo de aplicaciones informáticas.

## Descripción:

Dado un modelo relacional, utilizaremos ingeniería hacia adelante (Forward Engineer) para realizar el modelo físico de la base de datos. Y también lo haremos utilizando las sentencias de definición de datos del lenguaje SQL- DDL (por sus siglas en inglés Data Definition Language).

## Software a utilizar:

- DBMS PostgreSQL 12 o superior.
- PgAdmin 4. (IDE Gráfico de PostgreSQL) o DBeaver
- SQL Power Architect (opcional).

**RECUERDEN: Para consultar los tipos de datos disponibles para PostgreSQL utilizar la documentación oficial en el siguiente enlace:**

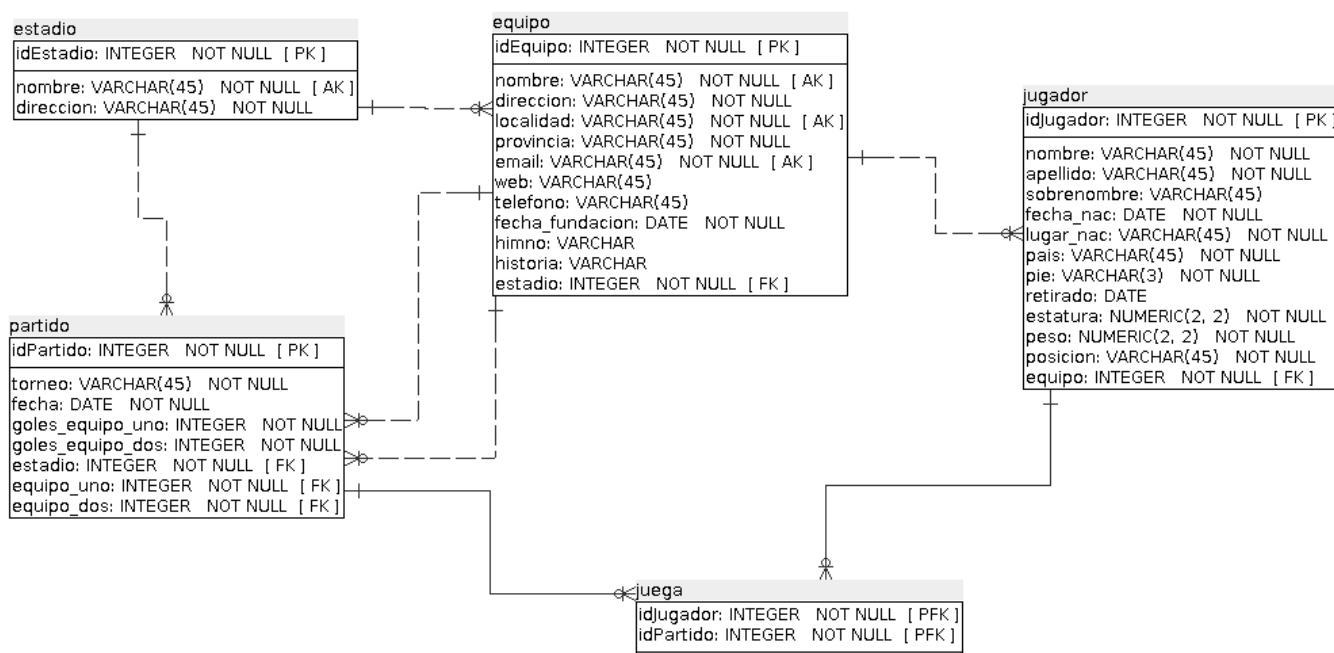
**<https://www.postgresql.org/docs/current/datatype.html>**

**Para consultar las distintas restricciones y sus formas de escritura, utilizar la documentación oficial en el siguiente enlace:**

**<https://www.postgresql.org/docs/current/ddl-constraints.html>**

## Ejercicios:

**Ejercicio 1:** Dado el siguiente modelo relacional en SQL Power Architect sobre los partidos de equipos de fútbol:



### Aclaraciones del modelo:

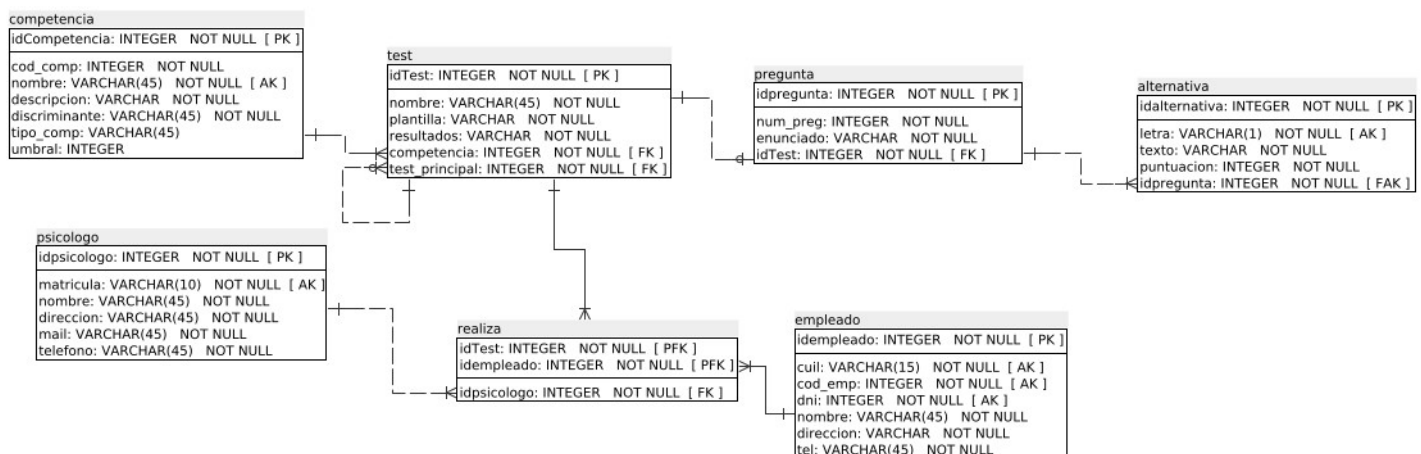
- Las claves primarias de las tablas son números enteros autoincrementales.
- Las acciones de integridad referencial para el UPDATE y DELETE son en todos los casos: NO ACTION.
- Los atributos himno e historia en la relación “equipo” permiten nulos.

- Los atributos sobrenombre y retirado en la relación “jugador” permiten nulos.
- El atributo pie de la relación “jugador” toma valores del conjunto {DER, IZQ}.

Ahora, utilizando PgAdmin4 de PostgreSQL debe:

- Crear la base de datos y las tablas mediante sentencias de SQL DDL.
- Establecer las restricciones de integridad, referenciales y de dominio que plantea el modelo.

**Ejercicio 2:** Dado el siguiente modelo relacional en SQL Power Architect sobre una serie de tests que se aplican a los empleados de una organización:



Aclaraciones del modelo:

- Las claves primarias de las tablas son números enteros autoincrementales.
- Las acciones de integridad referencial para el UPDATE y DELETE son en todos los casos: NO ACTION.
- En la Relación Competencia, los valores que toma el discriminante pertenecen al siguiente conjunto definido por extensión {intelectual, emocional}.
- El atributo de la Relación Competencia, tipo\_comp permite nulos (es el tipo de las competencias intelectuales)

- El atributo de la Relación Competencia, tipo\_comp toma valores del conjunto {matemática, lingüística, artística, musical, cultural}
- El atributo de la Relación Competencia, umbral permite nulos (el umbral sólo lo poseen las competencias emocionales)
- El atributo de la Relación Test, test\_principal permite nulos
- En este ejercicio, algunas tablas poseen claves candidatas:
  - Relación Pregunta: CK (idtest + num\_preg)
  - Relación Alternativa: CK (idpregunta + letra)

Ahora, utilizando PgAdmin4 de PostgreSQL debe:

- Crear la base de datos y las tablas mediante sentencias de SQL DDL.
- Establecer las restricciones de integridad, referenciales y de dominio que plantea el modelo.

## Seguimos con DDL: practicamos un poco más

**Ejercicio de ejemplo:** Con este ejercicio se pretende recorrer sentencias de creación, modificación y eliminación en una base de datos y en la estructura de sus tablas.

**a)** Crear una base de datos llamada Empresa:

```
CREATE DATABASE empresa;
```

**b)** En la base de datos empresa, crear las siguientes tablas, respetando las restricciones planteadas:

Empleado		
Columna	Dominio	Restricciones
id_empleado	Número natural autoincremental	Clave primaria
documento	Número natural	Clave candidata (obligatorio, único)
apellido	Cadena de caracteres, máximo 50	Obligatorio
nombre	Cadena de caracteres, máximo 50	Obligatorio
fecha_nacimiento	Fecha	Obligatorio, menor o igual a la fecha actual

Empleado		
direccion	Cadena de caracteres, máximo 50	Obligatorio

```
CREATE TABLE Empleado(
    id_empleado SERIAL PRIMARY KEY,
    documento INT NOT NULL UNIQUE CHECK(documento > 0 AND documento <
99999999),
    apellido VARCHAR(50) NOT NULL,
    nombre VARCHAR(50) NOT NULL,
    fecha_nac DATE NOT NULL CHECK(fecha_nac < CURRENT_DATE),
    direccion VARCHAR(50) NOT NULL
);
```

Ó, si escribimos las restricciones luego de la definición de las columnas:

```
CREATE TABLE Empleado(
    id_empleado SERIAL,
    documento INT NOT NULL UNIQUE,
    apellido VARCHAR(50) NOT NULL,
    nombre VARCHAR(50) NOT NULL,
    fecha_nac DATE NOT NULL CHECK(fecha_nac < CURRENT_DATE),
    direccion VARCHAR(50) NOT NULL,
CONSTRAINT pk_idempleado PRIMARY KEY (id_empleado),
CONSTRAINT documento_valido CHECK(documento > 0 AND documento < 99999999)
);
```

Departamento		
Columna	Dominio	Restricciones
id_departamento	Número natural autoincremental	Clave primaria
numero	Número natural	Clave candidata (obligatorio, único)
nombre	Cadena de caracteres, máximo 50	Obligatorio

```
CREATE TABLE Departamento(
    id_departamento SERIAL,
    numero SMALLINT NOT NULL UNIQUE CHECK(numero > 0),
    CONSTRAINT pk_id_departamento PRIMARY KEY (id_departamento),
);
```

Trabaja		
Columna	Dominio	Restricciones
director	Número natural	Clave primaria Debe ser un empleado válido

Trabaja		
departamento	Número natural	Clave primaria Debe ser un departamento válido
fecha_ingreso	Fecha	Clave primaria Obligatorio

```
CREATE TABLE Trabaja(
    director INT REFERENCES empleado(id_empleado),
    departamento INT REFERENCES departamento(id_departamento),
    fecha_ingreso DATE NOT NULL,
    PRIMARY KEY(director, departamento, fecha_ingreso)
);
```

Ó

```
CREATE TABLE Trabaja(
    director INT,
    departamento INT,
    fecha_ingreso DATE NOT NULL,
    CONSTRAINT fk_director FOREIGN KEY (director) REFERENCES empleado
(id_empleado),
    CONSTRAINT fk_departamento FOREIGN KEY (departamento) REFERENCES
departamento (id_departamento),
    CONSTRAINT pk_trabaja PRIMARY KEY (director, departamento,
fecha_ingreso)
);
```

Proyecto		
Columna	Dominio	Restricciones
id_proyecto	Número natural autoincremental	Clave primaria
numero	Número natural	Clave candidata
nombre	Cadena de caracteres, máximo 50	Obligatorio
departamento	Número natural	Debe ser un departamento válido

```
CREATE TABLE Proyecto(
    id_proyecto SERIAL,
    numero INT,
    nombre VARCHAR(50) NOT NULL,
    departamento INT,
    CONSTRAINT pk_proyecto PRIMARY KEY (id_proyecto),
```

```

    CONSTRAINT fk_departamento FOREIGN KEY (departamento) REFERENCES
departamento (id_departamento)
);

```

**Ejercicio 3:** De acuerdo a las siguientes especificaciones, utilizando DDL, crear, modificar o eliminar cuando sea necesario:

**a)** Crear la base de datos llamada PersonasDesaparecidas

**b)** En la base de datos creada en el ítem anterior, crear las siguientes tablas a partir de su definición:

Persona		
Columna	Dominio	Restricciones
id_persona	Número natural autoincremental	Clave primaria
documento	Número natural	Clave candidata (obligatorio, único)
apellido	Cadena de caracteres, máximo 45	Obligatorio
nombre	Cadena de caracteres, máximo 45	Obligatorio
fecha_nacimiento	Fecha	Opcional
ciudad	Cadena de caracteres, máximo 50	Opcional
provincia	Cadena de caracteres, máximo 50	Opcional
email	Cadena de caracteres, máximo 100	Opcional

PersonaTelefono		
Columna	Dominio	Restricciones
persona	Número natural	Clave primaria Persona válida
telefono	Cadena de caracteres, máximo 50	Clave primaria

Acontecimiento		
Columna	Dominio	Restricciones
id_acontecimiento	Número natural autoincremental	Clave primaria
numero	Número natural	Clave candidata
persona	Número natural	Persona válida
fecha	Fecha	Obligatorio

Acontecimiento		
hora	Hora	Obligatorio
ciudad	Cadena de caracteres, máximo 50	Obligatorio
provincia	Cadena de caracteres, máximo 50	Obligatorio
expediente	Cadena de caracteres, fijos 12	Obligatorio, único
detalles	Cadena de caracteres indefinido	Opcional
estado	Cadena de caracteres, máximo 255	Opcional

Familiar		
Columna	Dominio	Restricciones
acontecimiento	Número natural	Clave primaria Acontecimiento válido
persona	Número natural	Clave primaria Persona válida
parentesco	Cadena de caracteres, máximo 30	Opcional

**c)** En las tablas creadas en el ítem anterior, realizar las siguientes modificaciones:

- Realizar una modificación en la tabla Persona agregando las siguientes columnas:

Columna	Dominio	Restricciones
genero	Enumeración Masculino o Femenino	Obligatorio
direccion	Cadena de caracteres, máximo 100	Opcional

- Se decidió posteriormente que sólo se guardará un único número de teléfono por persona, por lo que deberá agregar una nueva columna en la tabla Persona para el teléfono y eliminar la tabla PersonaTelefono.
- Se decidió también que no es necesaria la columna de parentesco de la tabla Familiar, por lo que deberá eliminar la columna.
- Deberá también renombrar la tabla Familiar, por FamiliarPersona para mejorar la claridad de su nombre.



**Ejercicio 4:** De acuerdo a las siguientes especificaciones, utilizando DDL, crear, modificar o eliminar cuando sea necesario:

- a)** Crear la base de datos llamada Facturacion.
- b)** Crear los dominios ean (Entero, con  $0 < \text{valor} < 99999999999999$ ) y genero ('F' o 'M').
- c)** En la base de datos Facturacion, crear las siguientes tablas:

Cliente		
Columna	Dominio	Restricciones
id_cliente	Número natural autoincremental	Clave primaria
documento	Número Natural	Clave candidata (obligatorio, único) $0 < \text{valor} < 100.000.000$
apellido	Cadena de caracteres, máximo 50	Obligatorio valor $\neq$ ""
nombre	Cadena de caracteres, máximo 50	Obligatorio valor $\neq$ ""
genero	Género (dominio creado)	Obligatorio
domicilio	Cadena de caracteres, máximo 100	Opcional

Artículo		
Columna	Dominio	Restricciones
id_articulo	Número natural autoincremental	Clave primaria
codigo	Código EAN13	Clave candidata (obligatorio, único)
descripcion	Cadena de caracteres, máximo 100	Obligatorio valor $\neq$ ""
precio	Número real	Obligatorio valor $> 0$

Factura		
Columna	Dominio	Restricciones
id_factura	Número natural autoincremental	Clave primaria
numero	Número Natural	Clave candidata (obligatorio, único) valor > 0
fecha	Fecha	Obligatorio Por defecto hoy
hora	Hora	Obligatorio Por defecto ahora
cliente	Número Natural	Un cliente válido Política de actualización: Actualizar Política de eliminación: No permitir borrado

Renglón		
Columna	Dominio	Restricciones
factura	Número Natural	Clave primaria Factura válida Política de actualización: Actualizar Política de eliminación: Permitir borrado
numero	Número Natural	Clave primaria valor > 0
articulo	Código EAN13	Único para la misma factura Artículo válido Política de actualización: Actualizar Política de eliminación: No permitir borrado
cantidad	Número real	Obligatorio valor > 0