



Definición de tablas

DDL y creación de una tabla

Implementar estructuras de dato relacionales utilizando lenguaje de definición de datos DDL a partir de un modelo de datos para la creación y mantención de las definiciones de los objetos de una base de datos

- Unidad 1:
Bases de datos relacionales
- Unidad 2:
Manipulación de datos y transaccionalidad en las operaciones
- Unidad 3:
Definición de tablas
- Unidad 4:
Modelos Entidad-Relación y Relacional



Te encuentras aquí



¿Qué aprenderás en esta sesión?

- *Reconoce la sintaxis básica para la creación de expresiones DDL que resuelven un requerimiento de definición de datos.*

¿Qué es DML?



¿Qué lenguaje DML
hemos utilizado para
manipular datos?



**/* El lenguaje de definición
de datos DDL */**

¿Qué es DDL?

- Lenguaje de definición de datos (Data Definition Language).
- Permite definir las estructuras que almacenarán los datos así como de los procedimientos o funciones que permitan consultarlos.
- SQL es también un lenguaje de definición de datos.

SQL

Funcionalidades al ser un DDL

Las principales funcionalidades son

- Creación.
- Modificación.
- Borrado.
- Entre otras operaciones sobre tablas y registros de una base de datos.



Ejercicio guiado

"Repaso de operaciones
sobre una base de datos"



Repaso de operaciones sobre una base de datos

A continuación, realizaremos un repaso de algunas operaciones básicas en una base de datos utilizando SQL.

- **Paso 1:** Crear una base de datos

```
create database nombre_base_de_datos;
```

- **Paso 2:** Mostrar todas las tablas de una base de datos.

```
\dt;
```



Repaso de operaciones sobre una base de datos

- **Paso 3:** Mostrar una tabla en específica.

```
\dt nombre_tabla;
```

- **Paso 4:** Eliminar una tabla de la base de datos.

```
drop table nombre_tabla;
```

- **Paso 5:** Modificar un campo de una tabla.

```
ALTER TABLE nombre_tabla ADD column_name datatype;
```



/* Creación de una tabla */

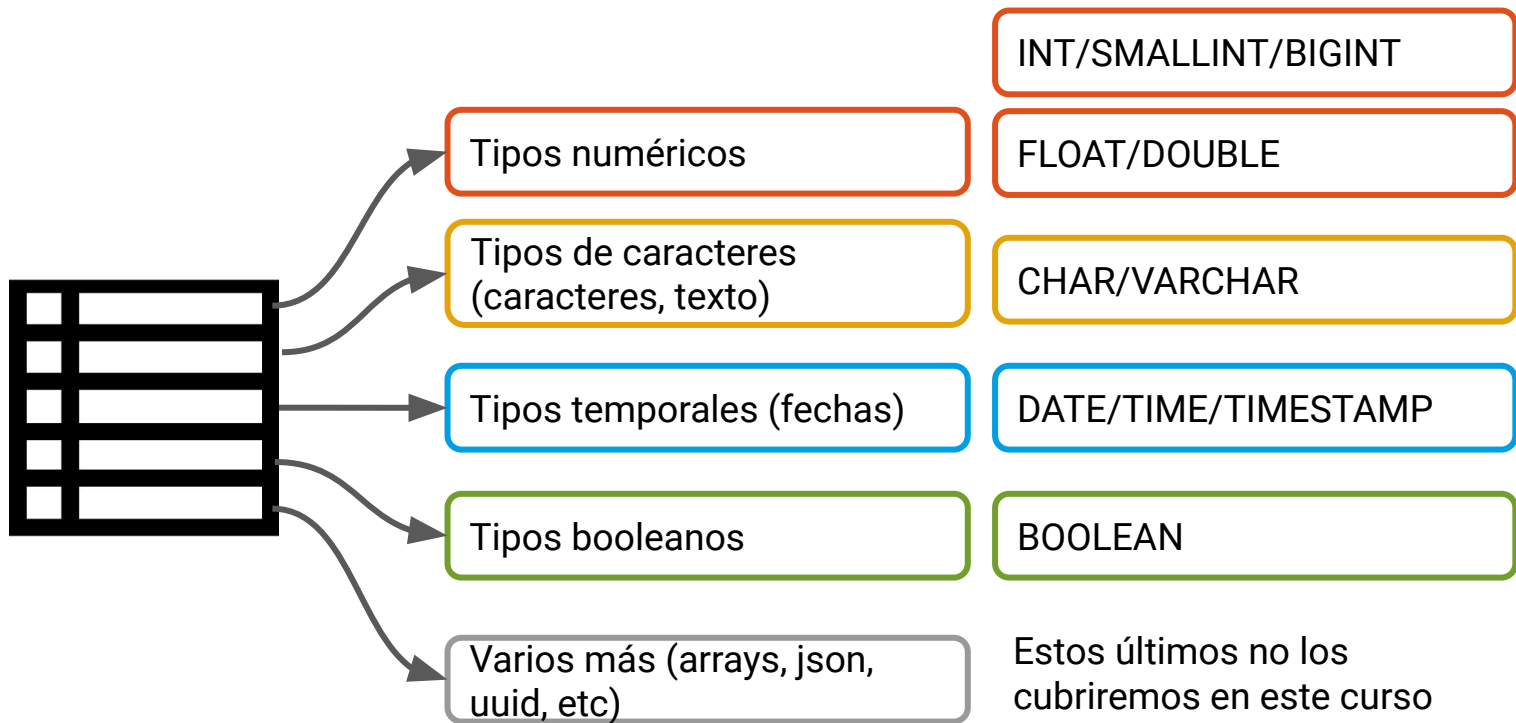
Definición de campos

- Como ya vimos, el comando para crear una tabla es `create table`.
- Analicemos en detalle la estructura:

```
create table clientes (id int unique not null primary key, nombre varchar not null, rut varchar unique not null);
```

1. Primero creamos y definimos el nombre.
2. Entre paréntesis indicamos los campos que estarán presentes en la tabla.
3. Para cada campo indicamos el tipo de dato y sus restricciones.

Tipos de datos



La restricción de nulidad

- Las restricciones en SQL se definen como Constraint.
- Estas, nos garantizan que los datos cumplan con ciertos parámetros.
- Estos parámetros básicamente van en función de las necesidades y la naturaleza de los campos que se vayan definiendo.



Tipos de restricciones

Check

Condición de restricción que se define a una columna.

- Al implementarse check la restricción se aplicará siempre que la evaluación sea verdadera.
- Su evaluación es un proceso booleano.
- Veamos el código de ejemplo.

```
create table productos (  
    product_no int,  
    name text,  
    price number CHECK (price > 0)  
);
```

Para el campo precio se ingresarán valores que cumplan la condición definida en paréntesis.

Tipos de restricciones

Not null

Una columna no puede recibir un valor nulo, es decir, obliga a ingresar algún dato.

- Se considera como una restricción de verificación de datos.
- Retornará un error en caso de que no se ingresen datos en los campos donde se haya definido la restricción.

```
CREATE TABLE products (  
    product_no integer NOT NULL,  
    name text NOT NULL,  
    price numeric  
);
```

Nótese la restricción not null en los campos product_no y name.

Tipos de restricciones

Unique

Restricción de seguridad que asegura la información de una columna.

- Previene la duplicación de datos.
- Todos los datos de una columna estarán restringidos y serán únicos.

```
CREATE TABLE products (  
    product_no integer UNIQUE,  
    name text,  
    price numeric  
);
```

Nótese la restricción unique en product_no, esto hará que este campo no tenga datos duplicados.

Tipos de restricciones

Primary key

Restricción de seguridad que asegura la información de una columna y además le otorga una identificación.

- Incorpora por defecto las restricciones de not null y unique.

```
CREATE TABLE products (  
    product_no integer PRIMARY KEY,  
    name text,  
    price numeric  
);
```

Con primary key podemos omitir la declaración de not null y unique dado que se asume por defecto.

Tipos de restricciones

Foreign key

Restricción de seguridad que en la tabla que se defina los datos deben coincidir con los datos de otra tabla.

- Para implementarlo deben existir al menos dos tablas.
- En una de ellas la clave primaria.
- En la siguiente la clave foránea para validar las coincidencias.

```
CREATE TABLE products (  
    product_no integer PRIMARY KEY,  
    name text,  
    price numeric  
);
```

```
CREATE TABLE orders (  
    order_id integer PRIMARY KEY,  
    product_no integer REFERENCES products  
    (product_no),  
    quantity integer  
);
```

Para más restricciones visita la [documentación oficial](#).

Descripción de tablas

- Con el comando `\d nombre_tabla` podemos ver la descripción de una tabla.
- Para realizarlo, deberás estar conectado a la base de datos `\c nombre_base_datos`
- Con la descripción de una tabla podemos observar en detalle los campos que se han generado, el tipo de dato y sus restricciones.

```
descripciones=# \d products;
               Table "public.products"
  Column      | Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
 product_no   | integer   |           | not null |
 name         | text      |           |          |
 price        | numeric   |           |          |
Indexes:
    "products_pkey" PRIMARY KEY, btree (product_no)
```

¿Qué son las
restricciones?



¿Qué restricción
utilizamos para evitar
que una fila contenga un
dato vacío?



¿Qué restricción
garantiza la integridad
referencial entre dos
tablas?



Ejercicio propuesto

"Aplicando lo aprendido"

{desafío}
latam_



Contexto

A continuación, deberás crear una base de datos que almacene información de clientes bancarios.

- Crear una tabla para clientes con los campos:
 - a. ID
 - b. Nombre
 - c. Apellido
 - d. Rut
 - e. Teléfono.
 - f. Email.



Contexto

- Crear una tabla para cuentas donde existan los campos:
 - a. ID
 - b. Número de cuenta.
 - c. Fecha de creación.
 - d. Balance.
- La clave foránea deberás definirla en el campo correspondiente para que se genere la relación con clientes.

Nota: Para los campos ID considera utilizar SERIAL para la generación autoincremental de estos valores.



¿Qué concepto costó
más comprender?





Próxima sesión...

- *Guía de ejercicios*

{desafío}
latam_

*Academia de
talentos digitales*

