



Definición de tablas

Modificación, eliminación y truncado 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?

- *Construye sentencias utilizando DDL para la modificación de los atributos de una tabla de acuerdo a los requerimientos planteados.*

¿Qué instrucción SQL
utilizamos para
modificar los campos de
una tabla?



/* Modificación de una tabla */

Modificar un campo en una tabla

- Como desarrolladores nos vamos a enfrentar a situaciones donde la definición de los datos, sus tipos e incluso los campos de una tabla no estén correctamente definidos.
- SQL, nos ofrece la opción de modificar o alterar una tabla que ya ha sido definida.
- Esto recordemos que se realiza con la instrucción ALTER TABLE.

Ejercicio guiado

"Aplicando modificaciones
en una tabla"



Aplicando modificaciones en una tabla

Un cliente que realiza ventas de teléfono nos solicita una base de datos que permita guardar la información de los dispositivos. Una de las informaciones más importantes es la dirección MAC. Esta dirección consiste en:

- Código de 12 dígitos agrupados de dos en dos y separados generalmente por dos puntos.
- Ejemplo de dirección MAC: 00:2A:66:22:9A:C7

Además de la dirección MAC, deberá estar presente el modelo y año del dispositivo, esto definido de manera inicial, en el camino se pueden agregar más campos.



Aplicando modificaciones en una tabla

- **Paso 1:** Creamos la base de datos con el nombre Mawashi Phones.

```
create database mawashi_phones;
```

- **Paso 2:** Nos conectamos a la base de datos.

```
\c mawashi_phones;
```



Aplicando modificaciones en una tabla

- **Paso 3:** Creamos la tabla con los campos que nos solicita inicialmente el cliente.

```
create table phones (id int primary key, modelo varchar(50), mac_address int unique, fecha_fabricacion varchar(50));
```

- **Paso 4:** Insertamos un registro para probar la tabla.

```
insert into phones (id, modelo, mac_address, fecha_fabricacion) values (1, 'Iphone 14', '1B:2A:3C:4D:5F:6G', '2022-09-09');
```



Análisis de errores

- Al realizar el insert en la tabla se observa el error:

```
ERROR: invalid input syntax for type  
integer: "1B:2A:3C:4D:5F:6G"
```

- Este error corresponde a que no se está cumpliendo la sintaxis de tipado que definimos.
- Ante ese escenario, debemos modificar el campo mac_adress.

Aplicando modificaciones en una tabla

- **Paso 5:** Modificamos el tipo de dato definido en el campo mac_address.

```
alter table phones alter column mac_address type varchar(50);
```

1. Seleccionamos la tabla a modificar.
2. Seleccionamos la columna.
3. Definimos el nuevo tipo de dato que tendrá dicha columna.



Aplicando modificaciones en una tabla

- **Paso 6:** Comprobamos que se hayan aplicado los cambios.

```
\d phones;
```

Table "public.phones"				
Column	Type	Collation	Nullable	Default
id	integer		not null	
modelo	character varying(50)			
mac_address	character varying(50)			
fecha_fabricacion	character varying(50)			
Indexes:				
"phones_pkey" PRIMARY KEY, btree (id)				
"phones_mac_address_key" UNIQUE CONSTRAINT, btree (mac_address)				

Aplicando modificaciones en una tabla

- **Paso 7:** Insertamos nuevamente el registro.

```
insert into phones (id, modelo, mac_address, fecha_fabricacion) values (1, 'Iphone 14', '1B:2A:3C:4D:5F:6G', '2022-09-09');
```

Resultado:

```
INSERT 0 1
```



Modificar una condición de nulidad

- Recordemos que la condición de nulidad es un tipo de restricción.
- Esta restricción hará que los datos de una determinada columna no puedan estar vacíos.
- Durante la definición de una tabla esta restricción ayudará a que la integridad de la información se fortalezca.

En nuestra base de datos de mawashi phones no tenemos la condición de nulidad en campos que son importantes como el mac_address.

Aplicando modificaciones en una tabla

- **Paso 8:** Con el alter table también podemos agregar nuevas restricciones sin reescribir las ya existentes.

```
alter table phones alter column mac_address set not null;
```

Table "public.phones"				
Column	Type	Collation	Nullable	Default
id	integer		not null	
modelo	character varying(50)			
mac_address	character varying(50)		not null	
fecha_fabricacion	character varying(50)			
Indexes:				
"phones_pkey" PRIMARY KEY, btree (id)				
"phones_mac_address_key" UNIQUE CONSTRAINT, btree (mac_address)				

Eliminación de una tabla

- Para eliminar una tabla ocupamos la instrucción `DROP TABLE`.
- Esta instrucción eliminará una tabla y esta acción solo puede ser ejecutada por un super usuario.
- Un super usuario es un rol que tiene permisos especiales, uno de ellos es eliminar una tabla que pueda afectar la base de datos.
- Este comando debe ser ejecutado con mucha precaución dado que elimina todo lo que se encuentre en la tabla.
- Sintaxis:

```
DROP TABLE nombre_tabla;
```

Truncado de una tabla

- Truncate tiene un comportamiento similar al delete.
- Ambos borran todos los registros de una tabla.
- Pero no afectan el *schema* de la base de datos.
- Schema, se refiere a toda la configuración lógica que exista en una base de datos.

Características de truncate

- Reinicio de identidades: Truncate automáticamente reinicia las secuencias de identidad en la tabla.
- Cascada: Truncate automáticamente modificará todas las tablas que estén relacionadas mediante una clave foránea.
- Restricciones: Utilizar con cautela en caso de existir múltiples tablas relacionadas.

Para más características visitar la [documentación oficial](#).

Veamos cómo se aplica truncate a la tabla de phones.

Aplicando modificaciones en una tabla

- **Paso 9:** Insertamos un nuevo registro a la tabla phones.

```
insert into phones (id, modelo, mac_address, fecha_fabricacion) values (2, 'Iphone 13', '5B:1A:2C:7D:8F:7h', '2021-09-24');
```

id	modelo	mac_address	fecha_fabricacion
1	Iphone 14	1B:2A:3C:4D:5F:6G	2022-09-09
2	Iphone 13	5B:1A:2C:7D:8F:7h	2021-09-24
(2 rows)			

Aplicando modificaciones en una tabla

- **Paso 9:** Aplicamos truncate a la tabla y vemos su resultado.

```
truncate phones;
```

```
id | modelo | mac_address | fecha_fabricacion  
---+-----+-----+-----  
(0 rows)
```

Se eliminan los registros pero la estructura de la tabla no se ve afectada y por ende nuestra base de datos conserva su lógica.

¿Por qué debemos
tener precaución con
DROP TABLE?



Si queremos eliminar
registros de una tabla
¿Qué es lo más
recomendable?



Ejercicio propuesto

"Agregar campos a phones"



Agregar campos a phones

A continuación, aplicarás lo aprendido en esta sesión incorporando los siguientes campos a la tabla phones:

- Memoria interna.
- Memoria ram.
- Peso.
- Dimensiones.

Para estos campos considera que son datos que mezclan datos numéricos y de texto. Recuerda asignar el tipo de dato a cada campo la restricción de nulidad.



Agregar campos a phones

Además, agrega al campo ID la instrucción SERIAL para que al momento de generarse un nuevo registro esta columna sea llenada de manera automática e incremental.

Tips:

1. Para agregar nuevas columnas: `alter table nombre_tabla add column.`
2. Para modificar una columna como el ID: `alter table nombre_tabla alter column.`

Para este ejercicio puedes utilizar la [documentación oficial](#) de PostgreSQL.



¿Qué concepto te costó
más comprender?





Próxima sesión...

- *Guía de ejercicios*

{desafío}
latam_

*Academia de
talentos digitales*

