



Definición de tablas

Modelo de datos con integridad referencial

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 de creación de una tabla utilizando DDL y definiendo campos, tipos de datos, nulidad, llaves primarias y foráneas de acuerdo a un modelo de datos existente para satisfacer un requerimiento.*

¿Para qué utilizamos la
restricción Foreign Key?



**/* Creando un modelo de datos
con integridad referencial */**

Recordatorio de integridad referencial

Integridad de datos = datos correctos + datos completos

- Problemas de diseño de una base de datos pueden causar problemas de integridad.
- En esta clase aprenderemos a evitar algunos de ellos.

Análisis de caso de integridad referencial

- Como creadores de bases de datos es posible que se nos escapen elementos claves como las restricciones en los campos de una tabla.
- Supongamos que tenemos una tabla clientes con un campo email. El email puede considerarse un dato clave dado que además de ser único debería ser not null.

¿Qué pasa si nuestra tabla clientes tiene registros con el email null? Veamos esto a continuación en un ejercicio.

Ejercicio guiado: “Modificación de campos null a not null”



Sigue los pasos...

- **Paso 1:** Creamos la tabla clientes create table clientes;
create table clientes (id integer unique not null, name varchar(25) not null, email
varchar(50));
- **Paso 2:** Insertamos al menos 3 registros, todos sin ingresar el campo email.
insert into clientes(id, name) values (1, 'Nombre 1');
insert into clientes(id, name) values (2, 'Nombre 2');
insert into clientes(id, name) values (3, 'Nombre 3');

```
descripciones=# select * from clientes;  
id | name | email  
---+-----+-----  
 1 | Nombre 1 |  
 2 | Nombre 2 |  
 3 | Nombre 3 |  
(3 rows)
```



Sigue los pasos...

- **Paso 3:** Modificamos el campo email a not null con alter table.

```
ALTER TABLE clientes ALTER COLUMN email SET NOT NULL;
```

Con esto modificamos el campo email, sin embargo nuestra tabla clientes ya tiene registros y al correr el comando veremos el siguiente error.

```
ERROR: column "email" of relation "clientes" contains null values
```

Veamos entonces cómo solucionar este inconveniente muy común en las bases de datos.



Sigue los pasos...

- **Paso 4:** Debemos actualizar la tabla con el comando UPDATE, seguidamente debemos indicar la columna a modificar y por último con WHERE definir la condición cuando el campo sea nulo.

```
descripciones=# UPDATE clientes SET email = '' WHERE email IS NULL;  
UPDATE 3
```



Sigue los pasos...

- **Paso 5:** Corremos nuevamente el comando de alteración de la columna

```
ALTER TABLE clientes ALTER COLUMN email SET NOT NULL;
```

```
descripciones=# alter table clientes alter column email set not null;  
ALTER TABLE
```

De datos nulos a predeterminados

- En SQL también tenemos la opción de asignar datos predeterminados para evitar la presencia de datos nulos.
- Esto se logra con la función COALESCE.
- Esta función permite seleccionar campos que sean nulos en una tabla y modificarlos con un valor determinado.

Sigamos trabajando en nuestra tabla de clientes.

Sigue los pasos...

- **Paso 6:** Supongamos que ahora deseamos incorporar a la tabla clientes un campo de fecha.

```
ALTER TABLE clientes ADD COLUMN fecha date;
```

```
descripciones=# \d clientes;
```

Table "public.clientes"				
Column	Type	Collation	Nullable	Default
id	integer		not null	
name	character varying(25)		not null	
email	character varying(50)		not null	
fecha	date			

Sigue los pasos...

- **Paso 7:** Agregamos un nuevo campo a la tabla de clientes pero los registros que ya existen no tienen dicha fecha incorporada. Asignemos una fecha a esos registros con COALESCE.

```
UPDATE clientes SET fecha = COALESCE(fecha, '2024-01-01');
```

```
descripcion=# select * from clientes;
 id |  name  | email |  fecha
----+-----+-----+-----
  1 | Nombre 1 |      | 2024-01-01
  2 | Nombre 2 |      | 2024-01-01
  3 | Nombre 3 |      | 2024-01-01
(3 rows)
```

Ejercicio propuesto: "Aplica lo aprendido"



Aplica lo aprendido

En la tabla clientes del ejercicio anterior, el campo fecha no tiene definida la restricción de nulidad. Modifica la tabla y asigna la restricción, recuerda que debes realizar primero el UPDATE y luego alterar la tabla.



Ejercicio propuesto: "Datos nulos en tabla de productos"



Datos nulos en tabla de productos

Un cliente nos hace entrega de un dataset de productos pero tiene inconvenientes dado que el campo SKU posee valores nulos. El data set se encuentra en la plataforma con el nombre “Material de apoyo - Base de datos productos”.

Este cliente nos solicita que modifiquemos la tabla y para los valores nulos del SKU le asignemos el valor de “Dato no ingresado”.

id	nombre	precio	sku
1	Televisor 50 pulgadas	1500.00	dato no ingresado
2	Celular de alta gama	1000.00	dato no ingresado
3	Laptop de última generación	2000.00	dato no ingresado
4	Nevera con congelador	500.00	dato no ingresado
5	Lavadora automática	300.00	dato no ingresado

Datos nulos en tabla de productos

Por último, el cliente nos solicita que no quiere que sigan ingresando datos con el SKU en nulo. En este sentido nos da la libertad de modificar la base de datos para impedir dicho inconveniente.



/*Borrar tablas referenciadas */

Tablas relacionadas

- Otro caso típico en la administración de una base de datos es el borrado de tablas que estén referenciadas.
- En SQL cuando existen tablas relacionadas la restricción de clave foránea nos alertará y dará error.
- Para evitar esto, podemos detener la restricción de clave foránea de manera temporal.

Veamos esto en un ejercicio

Ejercicio guiado: “Borrar tablas relacionadas”



Borrar tablas relacionadas

Ignorar la restricción de clave foránea

- **Paso 1:** Creamos las tablas libros y autores con el siguiente código.

```
CREATE TABLE autores (  
  id INT NOT NULL,  
  nombre VARCHAR(255) NOT NULL,  
  PRIMARY KEY (id)  
);
```

Tabla padre

```
CREATE TABLE libros (  
  id INT NOT NULL,  
  titulo VARCHAR(255) NOT NULL,  
  autor_id INT NOT NULL,  
  PRIMARY KEY (id),  
  FOREIGN KEY (autor_id) REFERENCES autores (id)  
);
```

Tabla hija



Borrar tablas relacionadas

Ignorar la restricción de clave foránea

- **Paso 2:** Insertamos algunos datos en ambas tablas

```
INSERT INTO autores (id, nombre)
VALUES
(1, 'Juan Pérez'),
(2, 'María García'),
(3, 'Pedro Rodríguez');
```

```
INSERT INTO libros (id, titulo, autor_id)
VALUES
(1, 'El Quijote', 1),
(2, 'La Divina Comedia', 2),
(3, 'Hamlet', 3);
```



Borrar tablas relacionadas

Ignorar la restricción de clave foránea

- **Paso 3:** Consultamos los datos de ambas tablas

```
SELECT libros.titulo, autores.nombre  
FROM libros  
INNER JOIN autores  
ON libros.autor_id = autores.id;
```

titulo	nombre
-----	-----
El Quijote	Juan Pérez
La Divina Comedia	María García
Hamlet	Pedro Rodríguez

Resultado de la consulta



Borrar tablas relacionadas

Ignorar la restricción de clave foránea

- **Paso 4:** Comprobación de la restricción de clave foránea.

```
delete from autores;
```

ERROR: update or delete on table "autores" violates foreign key constraint

"libros_autor_id_fkey" on table "libros"

DETAIL: Key (id)=(1) is still referenced from table "libros".



El error se debe a que los registros de autores no pueden ser eliminados porque están referenciados con los registros de la tabla libros y estos no pueden quedar sin su referencia.



¿Qué pasa si borramos la tabla hija?

En el ejercicio borramos la tabla hija libros. **¿Qué pasaría si eliminamos la tabla hija?**

```
delete from libros;  
DELETE 3
```

En este caso, sí es posible eliminar los registros de libros dado que la restricción se aplica del lado de la referencia (la tabla padre).

De los ejercicios ¿Qué
funcionalidad fue más
complicada de implementar?





Próxima sesión...

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

{desafío}
latam_

*Academia de
talentos digitales*

