

MySQL: Restricciones y Modificaciones de Tablas

Restricciones de datos

Repasemos, las restricciones en bases de datos SQL son reglas aplicadas a las columnas de una tabla para mantener la integridad y consistencia de los datos. Son herramientas esenciales para garantizar la validez y coherencia de la información almacenada en la base de datos.

Clave Primaria (Primary Key): Puedes designar una columna como clave primaria que garantiza que los valores en esa columna sean únicos. Para hacerlo, puedes agregar la opción PRIMARY KEY después del tipo de dato.

Valores Predeterminados (Default Values): Puedes especificar un valor predeterminado para una columna en caso de que no se proporcione ningún valor durante la inserción de datos. Para hacerlo, puedes agregar DEFAULT valor después del tipo de dato.

```
CREATE TABLE pedidos (  
    pedido_id INT,  
    fecha_pedido DATE DEFAULT CURRENT_DATE,  
    estado VARCHAR(20) DEFAULT 'Pendiente'  
);
```

Valores no nulos (Not Null): Puedes requerir que una columna no pueda contener valores nulos (NULL) agregando NOT NULL después del tipo de dato.

```
CREATE TABLE productos (  
    producto_id INT PRIMARY KEY,  
    nombre VARCHAR(50) NOT NULL,  
    precio DECIMAL(10, 2) NOT NULL  
);
```

Clave Foránea (Foreign Key): Puedes establecer una relación entre dos tablas mediante una clave foránea. Esto se hace al agregar FOREIGN KEY después del tipo de dato, junto con la referencia a la tabla y columna a la que se hace referencia.

```
CREATE TABLE pedidos (  
    pedido_id INT PRIMARY KEY,  
    cliente_id INT,  
    fecha_pedido DATE,  
    FOREIGN KEY (cliente_id) REFERENCES clientes(cliente_id)  
);
```

UNIQUE: Esta restricción garantiza que los valores en una columna sean únicos en toda la tabla, excepto por los valores nulos.

```
CREATE TABLE estudiantes (  
    matricula INT UNIQUE,  
    nombre VARCHAR(50),  
    edad INT  
);
```

CHECK: La restricción CHECK permite establecer condiciones específicas para los valores de una columna.

```
CREATE TABLE empleados (  
    empleado_id INT,  
    salario DECIMAL(10, 2) CHECK (salario >= 0),  
    nombre VARCHAR(50)  
);
```

INDEX: Puedes crear índices en una columna para acelerar las consultas de búsqueda.

```
CREATE INDEX idx_empleado_id ON empleados (empleado_id);
```

AUTO_INCREMENT: Esta restricción se utiliza generalmente en columnas numéricas enteras, como claves primarias, para que los valores se generan automáticamente al insertar filas en la tabla.

```
CREATE TABLE pedidos (  
    pedido_id INT AUTO_INCREMENT PRIMARY KEY,  
    fecha_pedido DATE,  
    total DECIMAL(10, 2)  
);
```

ENUM: La restricción ENUM permite especificar un conjunto de valores permitidos para una columna. Los valores deben ser seleccionados de la lista de opciones definida.

```
CREATE TABLE colores (  
    id INT PRIMARY KEY,  
    nombre_color ENUM('Rojo', 'Verde', 'Azul')  
);
```

Comando ALTER TABLE

El comando ALTER TABLE en SQL se utiliza para realizar modificaciones en una tabla existente. Puedes utilizarlo para agregar, modificar o eliminar columnas, así como para establecer restricciones adicionales en la estructura de la tabla. Es una herramienta esencial para mantener y adaptar una base de datos a medida que evolucionan los requisitos del sistema.

Agregar una columna nueva:

```
ALTER TABLE nombre_tabla ADD nombre_columna tipo_de_dato  
restriccion_opcional;
```

Modificar una columna existente:

```
ALTER TABLE nombre_tabla MODIFY nombre_columna nuevo_tipo_de_dato  
nueva_restriccion_opcional;
```

Eliminar una columna existente:

```
ALTER TABLE nombre_tabla DROP nombre_columna;
```

Agregar una restricción como un CHECK o UNIQUE puedes usar “ADD CONSTRAINT”:

```
ALTER TABLE nombre_tabla ADD CONSTRAINT nombre_restriccion CHECK  
(nombre_columna IS NOT NULL);
```

! La diferencia con MODIFY es que aquí debes ponerle un nombre a la restricción.

Agregar una restricción de clave foránea (FOREIGN KEY) puedes usar “ADD FOREIGN KEY”:

```
ALTER TABLE nombre_tabla  
ADD FOREIGN KEY (nombre_columna) REFERENCES  
nombre_tabla_a_referenciar(nombre_campo_a_referenciar);
```

Eliminar una clave foránea puedes utilizar “DROP FOREIGN KEY”:

```
ALTER TABLE nombre_tabla DROP FOREIGN KEY nombre_de_la_clave_foranea;
```

Para saber el nombre de una clave foránea puedes consultarlo por medio de esta query:

```
SELECT
    constraint_name,
    table_name,
    column_name,
    referenced_table_name,
    referenced_column_name
FROM
    information_schema.key_column_usage
WHERE
    referenced_table_name IS NOT NULL
    AND table_name = "nombre_de_la_tabla";
```

CURRENT_DATE: Devuelve la fecha actual sin la hora.

NOW(): Devuelve la fecha y hora actuales.

📌 *CURRENT_DATE y NOW() deben estar entre paréntesis para que la query funcione:*

“... DEFAULT (CURRENT_DATE);”