

PRIMARY KEY	Garantiza que los valores sean únicos y no nulos; identifica cada fila de forma única.	<code>EstudianteID INT PRIMARY KEY</code>
FOREIGN KEY	Establece una relación entre tablas; los valores deben coincidir con una clave primaria en otra tabla.	<code>FOREIGN KEY (EstudianteID) REFERENCES Estudiantes(EstudianteID)</code>
NOT NULL	Asegura que una columna no permita valores nulos.	<code>Nombre VARCHAR(50) NOT NULL</code>
UNIQUE	Asegura que todos los valores en una columna sean únicos (permite un solo valor nulo).	<code>Correo VARCHAR(100) UNIQUE</code>
CHECK	Restringe los valores en una columna para que cumplan una condición específica.	<code>Edad INT CHECK (Edad >= 18)</code>
DEFAULT	Define un valor por defecto para una columna si no se especifica uno.	<code>Estado VARCHAR(20) DEFAULT 'Activo'</code>
AUTO_INCREMENT	Genera automáticamente valores únicos para una columna (generalmente usada con claves primarias).	<code>EstudianteID INT AUTO_INCREMENT PRIMARY KEY</code>
INDEX	Mejora el rendimiento de las consultas creando un índice en una columna o combinación de columnas.	<code>CREATE INDEX idx_nombre ON Estudiantes(Nombre);</code>

Explicación Breve de las Restricciones

- **PRIMARY KEY:** Identificador único para cada fila.
- **FOREIGN KEY:** Relaciona datos entre tablas para garantizar integridad referencial.
- **NOT NULL:** Impide valores vacíos.
- **UNIQUE:** Evita duplicados en columnas específicas.
- **CHECK:** Define reglas que los datos deben cumplir.

- **DEFAULT:** Proporciona valores automáticos.
- **AUTO_INCREMENT:** Simplifica la generación de identificadores.
- **INDEX:** Acelera las búsquedas, aunque no modifica directamente la integridad de los datos.

Principales Restricciones

1. PRIMARY KEY

- Garantiza que la columna (o conjunto de columnas) tenga valores únicos y no nulos.
- Cada tabla puede tener solo una clave primaria.

Ejemplo:

sql

CopiarEditar

```
EstudianteID INT PRIMARY KEY
```

○

2. FOREIGN KEY

- Garantiza que los valores de una columna coincidan con los valores de una clave primaria en otra tabla.
- Se usa para mantener integridad referencial entre tablas.

Ejemplo:

sql

CopiarEditar

```
FOREIGN KEY (EstudianteID) REFERENCES Estudiantes(EstudianteID)
```

○

3. NOT NULL

- Impide que una columna acepte valores nulos.

Ejemplo:

sql

CopiarEditar

```
Nombre VARCHAR(50) NOT NULL
```

○

4. UNIQUE

- Garantiza que todos los valores en una columna (o combinación de columnas) sean únicos, pero permite valores nulos.

Ejemplo:

sql

CopiarEditar

```
Correo VARCHAR(100) UNIQUE
```

○

5. CHECK

- Garantiza que los valores de una columna cumplan una condición específica.
- No está completamente soportado en MySQL hasta versiones recientes.

Ejemplo:

sql

CopiarEditar

```
Edad INT CHECK (Edad >= 18)
```

○

6. DEFAULT

- Establece un valor predeterminado para una columna si no se proporciona uno.

Ejemplo:

sql

CopiarEditar

```
Estado VARCHAR(20) DEFAULT 'Activo'
```

○

7. AUTO_INCREMENT

- Genera automáticamente valores únicos en una columna, generalmente usada con claves primarias.

Ejemplo:

sql

CopiarEditar

```
EstudianteID INT AUTO_INCREMENT PRIMARY KEY
```

○

8. INDEX

- Mejora el rendimiento de las consultas al crear un índice en una columna.
- No es una restricción estricta, pero influye en cómo se acceden a los datos.

Ejemplo:

sql

CopiarEditar

```
CREATE INDEX idx_nombre ON Estudiantes(Nombre);
```

○

Restricciones Combinadas

Puedes combinar varias restricciones en una sola columna. Por ejemplo:

sql

CopiarEditar

```
CREATE TABLE Usuarios (  
    UsuarioID INT PRIMARY KEY AUTO_INCREMENT,  
    Nombre VARCHAR(50) NOT NULL,  
    Correo VARCHAR(100) UNIQUE,  
    FechaRegistro DATE DEFAULT CURRENT_DATE  
);
```

Importancia de las Restricciones

- **Integridad de los datos:** Aseguran que los datos sean consistentes y correctos.
- **Evitan errores:** Reducen la probabilidad de datos inválidos o duplicados.
- **Mejoran la relación entre tablas:** Las claves foráneas y primarias facilitan la organización y conexión entre datos relacionados.