

# Integridad y restricciones

Sitio:

[Agencia de Aprendizaje a lo largo de la Vida](#)

Curso:

Administración de Base de Datos 1° G

Libro:

Integridad y restricciones

Imprimido por:

Eduardo Manuel Moreno

Día:

lunes, 2 de septiembre de 2024, 11:39

# Tabla de contenidos

- 1. Introducción
- 2. Integridad de atributo
- 3. Integridad de claves

## Integridad y restricciones



### ¿Qué es la integridad de datos?

Se entiende por integridad de datos a las restricciones, controles y validaciones que diseñamos para proteger la información almacenada en la base de datos y que la misma quede libre de incoherencias según nuestro criterio y la lógica de negocio de los datos que se modelan.

Es importante tener en cuenta que una base de datos suele ser accedida tanto para leer cómo modificar su información, por diferentes programas/aplicaciones desarrolladas por terceros.



### ¿Qué queremos decir cuando mencionamos que los datos que se almacenan deben ser coherentes?

Queremos decir que la información que se va a almacenar en la base de datos debe respetar las reglas del negocio.

SQL permite establecer diferentes propiedades para asegurar la integridad de los datos. Estas propiedades incluyen:

- Tipos de dato, definiciones NULL y NOT NULL.
- Valores por omisión para campos con definiciones DEFAULT
- Propiedades IDENTITY
- Reglas de validación con la propiedad CHECK
- Desencadenadores (triggers)
- Índices
- Claves primarias/foráneas

Las restricciones definen reglas indicando los valores permitidos en las columnas y son el mecanismo estándar para asegurar integridad.

## Integridad de atributo



### ¿Qué es la integridad de atributo?

La integridad de atributo, también llamada de dominio, es la validación de las entradas en una determinada columna.

Se puede asegurar la integridad de dominio restringiendo el tipo (a través de tipos de dato), el formato (a través de las restricciones **CHECK** y de las reglas de validación) o el rango de valores posibles (restricciones **CHECK**, definiciones **DEFAULT**, definiciones **NULL** y **NOT NULL**).

Cuando **creamos o modificamos** la estructura de una tabla podemos indicar si un atributo **acepta o no valores nulos**. Básicamente, es donde especificamos si cada atributo es de carga obligatoria con información al momento de insertar un registro en la tabla. Los atributos que aceptan valores nulos no son de carga obligatoria.



Atención: un valor **NULL** no es lo mismo que un cero, que un blanco o una cadena de caracteres de longitud cero.

Los valores por omisión indican que valor será guardado en una columna si no se especifica un valor para la columna cuando se inserta una fila. Cada columna en una tabla puede contener una sola definición **DEFAULT**. Va asociado a la columna y no al tipo de dato.

**SQL** proporciona campos **auto incrementables (o autonuméricos)**. Cada tabla puede tener sólo una columna auto incrementable, la que contendrá una secuencia de valores generados por el sistema que unívocamente identifican a cada fila de la tabla. Esta columna puede ser solo una del total de atributos y como es lógico no soporta el **NULL**.

La propiedad **CHECK** es:

- una restricción que permite definir el modo en que **SQL** fuerza la integridad de la base de datos de un campo.
- es una regla de validación para el campo, solo restringen valores para una columna.
- asegura la integridad de dominio al limitar los valores que son aceptados para una columna.
- se puede aplicar una misma restricción **CHECK** a múltiples columnas creando la restricción a nivel de tabla.
- se puede modificar o eliminar una restricción **CHECK** una vez que ha sido creada

## Integridad de claves



### ¿Qué es una clave primaria?

Una **clave primaria** a un atributo o a una combinación de atributos que identifica de forma única a cada fila de una tabla. Es decir, no puede haber dos filas en una tabla que tengan la misma clave primaria.

La clave primaria puede especificarse directamente en el momento de la creación de la tabla. En el estándar **SQL**, las claves primarias pueden estar compuestas por una o más columnas (campos de una tabla) y cada columna que forme parte de la clave primaria queda implícitamente definida como **NOT NULL**.

En el estándar **SQL**, las claves primarias pueden estar compuestas por una o más columnas (atributos de una tabla) y cada columna que forme parte de la clave primaria queda implícitamente definida como **NOT NULL**.



### ¿Qué es una clave foránea?

Una **clave foránea** es aquella columna que existiendo como dependiente en una tabla, es a su vez clave primaria en otra tabla. Con la restricción "foreign key" se define un campo (o varios) cuyos valores coinciden con la clave primaria de otra tabla, es decir, se define una referencia entre dos tablas, donde se relacionan clave primaria de una tabla con clave foránea de la otra.

Una tabla puede tener varias restricciones **foreign key**. Una restricción **foreign key no puede modificarse**, debe eliminarse y volverse a crear.

No se puede eliminar una tabla referenciada en una restricción **foreign key**, aparece un mensaje de error.

La integridad referencial se activa en cuanto creamos una clave foránea y a partir de ese momento se comprueba cada vez que se modifiquen datos que puedan alterarla.