

INTEGRIDAD DE LOS DATOS

AUTOR: MAX JOSÉ BERMÚDEZ LEÓN

DICIEMBRE: 2020



San Marcos

Introducción

Imagine lo siguiente: Una empresa farmacéutica pregona información sobre la seguridad de su nuevo fármaco milagroso. Ahora bien, cuando la FDA inspecciona su centro de producción extraterritorial, se detiene el trabajo de inmediato; faltan datos de control de calidad de gran importancia. Por desgracia, son frecuentes ejemplos reales como este de integridad de datos comprometida. Existen problemas de precisión y coherencia de datos en todos los sectores que pueden provocar desde pequeños fastidios a graves problemas para la empresa.

En esta época de los big data, en la que se trata y almacena más información que nunca, cada vez es más importante poner en práctica medidas que conserven la integridad de los datos recabados. El primer paso para conservar la seguridad de los datos es entender los conceptos básicos de la integridad y saber cómo funciona. Siga leyendo para conocer qué es la integridad de datos, la razón de su importancia y qué opciones tiene a su disposición para conservar la integridad de sus datos.



Tabla de contenido

Introducción.....	1
Integridad de datos	3
Que es la integridad de los datos	3
Integridad en bases de datos	3
Tipos de Integridad de datos.....	5
Integridad física	5
Integridad lógica	6
Integridad referencial.....	7
Integridad de dominio	9
Integridad definida por el usuario	10
Que no es integridad de datos	11
La integridad de datos no es seguridad de datos	11
La integridad de datos no es calidad de datos	12
Valores por defecto.....	12
Valores inválidos	15
Restricciones en tablas	16
Conclusiones y recomendaciones.....	17
Referencias bibliográficas.....	18

La integridad de datos se refiere a la exactitud, fiabilidad, precisión, integralidad y coherencia general de los datos. La integridad de datos se mantiene gracias a un conjunto de procesos, reglas y normas que se ponen en práctica durante la fase de diseño.

Integridad de datos

Que es la integridad de los datos

La integridad de datos se refiere a la exactitud, fiabilidad, precisión, integralidad y coherencia general de los datos. La integridad de datos se mantiene gracias a un conjunto de procesos, reglas y normas que se ponen en práctica durante la fase de diseño.

Cuando la integridad de los datos es segura, la información almacenada en una base de datos seguirá siendo completa, precisa y fiable por mucho tiempo que pase almacenada o por muchas veces que acceda uno a ella. La integridad de los datos también garantiza que sus datos estarán a salvo de fuerzas externas.

Los problemas con la integridad de los datos pueden comenzar con una fuente humana, las personas que entran en los registros pueden cometer errores, lo que lleva a variaciones entre los datos originales y los datos almacenados en un sistema.

Del mismo modo, las personas pueden cometer errores durante la transferencia o la copia electrónica de datos, haciendo disparidad entre las diferentes versiones o referencias a un archivo.

Integridad en bases de datos

Al crear bases de datos, se debe prestar atención a la integridad de los datos y a cómo mantenerlos. Una buena base de datos hará cumplir la integridad de los datos siempre que sea posible.

Por ejemplo, un usuario podría accidentalmente intentar ingresar un número de teléfono en un campo de fecha.

Si el sistema aplica integridad a los datos, evitará que el usuario cometa estos errores. Mantener la integridad de los datos significa asegurarse de que los datos permanezcan intactos y sin cambios a lo largo de todo su ciclo de vida.

Esto incluye la captura de los datos, el almacenamiento, las actualizaciones, las transferencias, las copias de seguridad, etc. Cada vez que se procesan los datos existe el riesgo de que se corrompan (accidental o maliciosamente).

Para que la integridad de los datos se mantenga, es necesario que no haya habido cambios o alteraciones en los datos.

A medida que los datos son introducidos, almacenados, accedidos, movidos y actualizados, los puntos débiles en un sistema pueden comprometer los datos.

Fallas en una computadora pueden llevar a sobrescribir parcialmente los datos o errores de datos. Los virus pueden ser creados para atacar la integridad de los datos, algunos trabajan en silencio para dañar los datos, sin traicionar su presencia.

Las interrupciones en las diversas operaciones pueden dar lugar a problemas, como daños mecánicos, como la exposición a imanes o daños físicos causados por cortes de energía u otros eventos.

Los diseñadores de la arquitectura de datos, deben tener en cuenta la integridad de datos, desde bases de datos del gobierno hasta sistemas de archivos en los ordenadores personales, cuando se trabaja en estos sistemas.

Ellos piensan en la forma en que se utiliza el sistema, identifican las posibles amenazas evidentes, y desarrollan métodos para asegurar el sistema de protección de los datos.

El no pensar en el futuro puede dar lugar a compromisos catastróficos y la posible emisión accidental de datos seguros o sensibles, una preocupación especial con bases de datos que contienen información personal o identificable acerca de individuos e instituciones.

Las personas que trabajan con bases de datos pueden proporcionar capacitación en la integridad de datos, incluyendo recordatorios para revisar los datos a medida que se introducen, para guardar con regularidad los datos y para informar de inmediato de cualquier sospecha de compromisos o actividades cuestionables.

Cuanto más rápidamente se reconoce un problema, más fácil será reconocido. El personal de apoyo, como el personal de TI, también trabaja en la protección de bases de datos de ataques externos con herramientas como cortafuegos, antivirus, y análisis periódicos de código malicioso.

Del mismo modo, los sistemas también pueden ser diseñados para bloquear los datos en formato de sólo lectura para protegerlos de manipulaciones o interferencias que puedan poner en peligro su integridad.

Tipos de Integridad de datos

Existen dos tipos de integridad de datos: la integridad física y la lógica. Ambas son conjuntos de procesos y métodos que hacen cumplir esa integridad en bases de datos tanto jerárquicas como relacionales.

Integridad física

La integridad física es la protección de la integridad y la precisión de los datos tal y como están almacenados y como son extraídos. Cuando tiene lugar una catástrofe natural, se produce un apagón o unos hackers alteran las funciones de una base de datos, se dice que se ha comprometido la integridad física. El error humano, la erosión del almacenamiento y toda una retahíla de incidencias también pueden imposibilitar que los gestores de tratamiento



de datos, programadores de sistemas o aplicaciones y los auditores internos obtengan datos veraces.

Integridad lógica

La integridad lógica conserva los datos sin ningún cambio, puesto que se emplean de forma distinta en una base de datos relacional. La integridad lógica protege a los datos del error humano y también de los hackers, pero de una forma muy distinta a la integridad física.

Existen cuatro tipos de integridad lógica:

Integridad de la entidad

La integridad de la entidad se basa en la creación de unas claves primarias, o valores únicos, que identifican datos para asegurar que no aparezcan enumerados más de una vez y que no haya ningún campo de una tabla considerado nulo. Es una prestación de los sistemas relacionales que almacenan datos en tablas que pueden enlazarse y emplearse de formas muy distintas.

La integridad de entidad define una fila como entidad única para una tabla determinada. La integridad de entidad exige la integridad de las columnas de los identificadores o la clave principal de una tabla, mediante índices y restricciones UNIQUE, o restricciones PRIMARY KEY.

Ejemplos:

- Tenemos la siguiente relación:

DESPACHOS		
<i>Edificio</i>	<i>número</i>	<i>superficie</i>
Marina	120	10
Marina	122	15
Marina	230	20
Diagonal	120	10

En esta relación, puesto que la clave primaria está formada por edificio y número, no hay ningún despacho que tenga un valor nulo para edificio, ni tampoco para número.

Integridad referencial

La integridad referencial es una serie de procesos que aseguran que los datos se almacenen y se utilicen uniformemente. Las reglas integradas en la estructura de la base de datos sobre cómo se utilizan claves foráneas para garantizar que tan solo se produzcan cambios, incorporaciones o supresiones de datos adecuados. Las reglas pueden incluir restricciones que eliminen la entrada de datos duplicados, aseguren que los datos son veraces y/o impidan la entrada de datos no pertinentes.

Existen cuatro actualizaciones de la base de datos que pueden corromper la integridad referencial:

- La inserción de una fila hijo se produce cuando no coincide la llave foránea con la llave primaria del padre.
- La actualización en la llave foránea de la fila hijo, donde se produce una actualización en la clave ajena de la fila hijo con una sentencia UPDATE y la misma no coincide con ninguna llave primaria.
- La supresión de una fila padre, con la que, si una fila padre -que tiene uno o más hijos- se suprime, las filas hijos quedarán huérfanas.

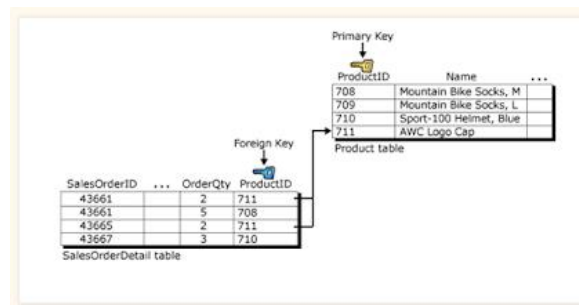
- La actualización de la clave primaria de una fila padre, donde si en una fila padre, que tiene uno o más hijos se actualiza su llave primaria, las filas hijos quedarán huérfanas.

Así mismo, la integridad referencial protege las relaciones definidas entre las tablas cuando se crean o se eliminan filas. En SQL Server la integridad referencial se basa en las relaciones entre claves externas y claves principales o entre claves externas y claves exclusivas, mediante restricciones FOREIGN KEY y CHECK. La integridad referencial garantiza que los valores de clave sean coherentes en las distintas tablas. Para conseguir esa coherencia, es preciso que no haya referencias a valores inexistentes y que, si cambia el valor de una clave, todas las referencias a ella se cambien en consecuencia en toda la base de datos.

Cuando se exige la integridad referencial, SQL Server impide a los usuarios:

- Agregar o cambiar filas en una tabla relacionada si no hay ninguna fila asociada en la tabla principal.
- Cambiar valores en una tabla principal que crea filas huérfanas en una tabla relacionada.
- Eliminar filas de una tabla principal cuando hay filas relacionadas coincidentes.

Ejemplos



- En las tablas Sales.SalesOrderDetail y Production.Product de la base de datos AdventureWorks, la integridad referencial se basa en la relación entre la clave externa

(ProductID) de la tabla Sales.SalesOrderDetail y la clave principal (ProductID) de la tabla Production.Product. Esta relación garantiza que un pedido de ventas no pueda nunca hacer referencia a un producto que no existe en la tabla Production.Product.

- Supongamos una base de datos con las entidades Persona y Factura. Toda factura corresponde a una persona y solamente una. Implica que en todo momento dichos datos sean correctos, sin repeticiones innecesarias, datos perdidos y relaciones mal resueltas.
- Supongamos que una persona se identifica por su atributo CEDULA. También tendrá otros atributos como el nombre y la dirección. La entidad Factura debe tener un atributo CEDULA_cliente que identifique a quién pertenece la factura.

Por sentido común es evidente que todo valor de CEDULA_cliente debe corresponder con algún valor existente del atributo DNI de la entidad Persona. Esta es la idea intuitiva de la integridad referencial.

Integridad de dominio

La integridad de dominio es el conjunto de procesos que garantizan la veracidad de cada dato de un dominio. En este contexto, un dominio es un conjunto de valores aceptables que una columna puede contener.

Puede incorporar restricciones y otras medidas que limiten el formato, tipo y cantidad de datos introducidos.

- Datos Requeridos: establece que una columna tenga un valor no NULL. Se define efectuando la declaración de una columna es NOT NULL cuando la tabla que contiene las columnas se crea por primera vez, como parte de la sentencia CREATE TABLE.
- Chequeo de Validez: cuando se crea una tabla cada columna tiene un tipo de datos y el DBMS asegura que solamente los datos del tipo especificado sean ingresados en

la tabla.

La integridad de dominio viene dada por la validez de las entradas para una columna determinada. Puede exigir la integridad de dominio para restringir el tipo mediante tipos de datos, el formato mediante reglas y restricciones CHECK, o el intervalo de valores posibles mediante restricciones FOREIGN KEY, restricciones CHECK, definiciones DEFAULT, definiciones NOT NULL y reglas.

Ejemplos:

- Si en la relación EMPLEADOS (CEDULA, nombre, apellido, edademp) hemos declarado que dominio (CEDULA) es el dominio predefinido de los enteros, entonces no podremos insertar, por ejemplo, ningún empleado que tenga por DNI el valor “Luis”, que no es un entero.
- Si en la relación EMPLEADOS (CEDULA, nombre, apellido, edademp) se ha declarado que dominio (CEDULA) es el dominio predefinido de los enteros, entonces no se permitirá consultar todos aquellos empleados cuyo CEDULA sea igual a ‘Elena’ (CEDULA = ‘Elena’). El motivo es que no tiene sentido que el operador de comparación = se aplique entre un CEDULA que tiene por dominio los enteros, y el valor ‘Elena’, que es una serie de caracteres.

Integridad definida por el usuario

La integridad definida por el usuario comprende las reglas y restricciones creadas por el usuario para adaptarse a sus necesidades particulares. En ocasiones con la integridad de entidad, referencial y de dominio no basta para salvaguardar los datos. A menudo, deben tenerse en cuenta determinadas reglas corporativas concretas e incorporarse en las medidas referentes a la integridad de los datos.

Todas las categorías de integridad admiten la integridad definida por el usuario. Esto incluye todas las restricciones de nivel de columna y nivel de tabla en CREATE TABLE, procedimientos almacenados y desencadenadores.

Que no es integridad de datos

Cuando se habla tanto de la integridad de los datos, resulta fácil hacerse un lío y confundir el verdadero significado del término. Muchas veces la seguridad y la calidad de los datos son sustituidas erróneamente por la integridad, cuando cada uno de estos términos hace referencia a una realidad específica.

La integridad de datos no es seguridad de datos

La seguridad de datos es el conjunto de medidas adoptadas para evitar la corrupción de los datos. Incorpora el uso de sistemas, procesos y procedimientos que mantienen los datos inaccesibles para los demás, que podrían usarlos de forma perjudicial o distinta a la prevista. Las infracciones en materia de seguridad pueden ser pequeñas y fáciles de contener o grandes y provocar daños considerables.

Si bien la integridad de los datos se ocupa de mantener la información intacta y veraz en el transcurso de su existencia, el objetivo de la seguridad es el de proteger la información de ataques externos. La seguridad de datos tan solo es una de muchas facetas de la integridad de los datos. La seguridad de los datos no es lo suficientemente amplia para que incluya los múltiples procesos necesarios para que los datos no se alteren con el tiempo.



La integridad de datos no es calidad de datos

¿Los datos de su base de datos cumplen los estándares definidos por su empresa y las necesidades de su negocio? La calidad de datos da respuesta a esas preguntas con todo un conjunto de procesos que miden la antigüedad, relevancia, precisión, integralidad y fiabilidad de sus datos.

Al igual que sucede con la seguridad, la calidad de los datos tan solo es una parte de la integridad de los datos, aunque crucial. La integridad de datos engloba todos los aspectos de la calidad de datos y va más allá al implantar un conjunto de reglas y procesos que rigen la forma en la que se introducen los datos, se almacenan, se transfieren, etc.

Valores por defecto

Hemos visto que, si al insertar registros no se especifica un valor para un campo que admite valores nulos, se ingresa automáticamente "null" y si el campo está declarado "identity", se inserta el siguiente de la secuencia. A estos valores se les denomina valores por defecto o predeterminados.

Un valor por defecto se inserta cuando no está presente al ingresar un registro y en algunos casos en que el dato ingresado es inválido.

Para campos de cualquier tipo no declarados "not null", es decir, que admiten valores nulos, el valor por defecto es "null". Para campos declarados "not null", no existe valor por defecto, a menos que se declare explícitamente con la cláusula "default".

Para todos los tipos, excepto los declarados "identity", se pueden explicitar valores por defecto con la cláusula "default".

Podemos establecer valores por defecto para los campos cuando creamos la tabla. Para ello utilizamos "default" al definir el campo. Por ejemplo, queremos que el valor por defecto del campo "autor" de la tabla "libros" sea "Desconocido" y el valor por defecto del campo "cantidad" sea "0":

```
CREATE TABLE libros(
    codigo int identity,
    titulo varchar(40),
    autor varchar(30) NOT NULL DEFAULT 'Desconocido',
    editorial varchar(20),
    precio decimal(5,2),
    cantidad tinyint DEFAULT 0
);
```

Si al ingresar un nuevo registro omitimos los valores para el campo "autor" y "cantidad", Sql Server insertará los valores por defecto; el siguiente valor de la secuencia en "codigo", en "autor" colocará "Desconocido" y en cantidad "0".

Entonces, si al definir el campo explicitamos un valor mediante la cláusula "default", ése será el valor por defecto.

Ahora, al visualizar la estructura de la tabla con "sp_columns" podemos entender lo que informa la columna "COLUMN_DEF", muestra el valor por defecto del campo.

También se puede utilizar "default" para dar el valor por defecto a los campos en sentencias "insert", por ejemplo:

```
INSERT INTO libros (titulo,autor,precio,cantidad)
VALUES ('El gato con botas',DEFAULT,DEFAULT,100);
```

Si todos los campos de una tabla tienen valores predeterminados (ya sea por ser "identity", permitir valores nulos o tener un valor por defecto), se puede ingresar un registro de la siguiente manera:

```
INSERT INTO libros DEFAULT VALUES;
```

La sentencia anterior almacenará un registro con los valores predeterminados para cada uno



de sus campos.

Entonces, la cláusula "default" permite especificar el valor por defecto de un campo. Si no se explicita, el valor por defecto es "null", siempre que el campo no haya sido declarado "not null".

Los campos para los cuales no se ingresan valores en un "insert" tomarán los valores por defecto:

- Si tiene el atributo "identity": el valor de inicio de la secuencia si es el primero o el siguiente valor de la secuencia, no admite cláusula "default"
- Si permite valores nulos y no tiene cláusula "default", almacenará "null";
- Si está declarado explícitamente "not null", no tiene valor "default" y no tiene el atributo "identity", no hay valor por defecto, así que causará un error y el "insert" no se ejecutará.
- Si tiene cláusula "default" (admita o no valores nulos), el valor definido como predeterminado
- Para campos de tipo fecha y hora, si omitimos la parte de la fecha, el valor predeterminado para la fecha es "1900-01-01" y si omitimos la parte de la hora, "00:00:00".
- Un campo sólo puede tener un valor por defecto.

Una tabla puede tener todos sus campos con valores por defecto. Que un campo tenga valor por defecto no significa que no admita valores nulos, puede o no admitirlos.

Valores inválidos

Un valor NULL en las bases de datos es un valor del sistema que ocupa un byte de almacenamiento e indica que un valor no está presente a diferencia de un espacio o cero o cualquier otro valor predeterminado. El campo en una base de datos que contiene el valor NULL significa que el contenido de esta celda es desconocido en el momento de mirarlo. Una columna que permite valores NULL también permite insertar filas sin ningún valor en esa columna. Existen varios pros y contras de usar valores NULL en contraposición a los valores predeterminados:

- Pros
 - El valor NULL no tiene el tipo de datos, por lo que se puede insertar en cualquier estructura de datos y en cualquier columna de la base de datos. Los valores predeterminados, por otro lado, deben tener su tipo de datos especificado y un valor predeterminado en una columna puede verse igual en otra columna, pero puede ser de un tipo diferente.
 - NULL se usa a menudo en esquemas donde un valor es opcional. Es un método conveniente para omitir la entrada de datos para campos desconocidos sin tener que implementar reglas adicionales, como almacenar valores negativos en un campo entero para representar datos omitidos.
 - Dado que el valor NULL ocupa solo 1 bit de espacio de memoria, pueden ser útiles al optimizar la base de datos. Usar esos valores es mucho más eficiente que los valores predeterminados, p. Ej. caracteres de 8 bits y enteros de 16 bits.
 - Si bien los requisitos de su sistema pueden cambiar con el tiempo y los tipos de valores predeterminados con ellos, el valor NULL siempre es NULL, por lo

que no es necesario actualizar el tipo de datos.

- La asignación de No Nulo a los esquemas de tabla también puede ayudar con la validación de la tabla, en el sentido de que la columna con criterios No Nulo requerirá que se inserte un valor. Los valores predeterminados no tienen estas capacidades.
- Contraste
 - Los valores NULL se confunden fácilmente con cadenas de caracteres vacías, que devuelven un valor en blanco al usuario cuando se seleccionan. En este sentido, los valores predeterminados son menos confusos y son la opción más segura, a menos que el valor predeterminado se establezca en la cadena vacía.
 - Si se permiten valores NULL en la base de datos, pueden causarle al diseñador algo de tiempo y trabajo adicional, ya que pueden hacer que la lógica de la base de datos sea más complicada, especialmente cuando hay muchas comparaciones con valores nulos.

Restricciones en tablas

Las CONSTRAINTS son restricciones que se utilizan para limitar el tipo de dato que puede recibir una columna de una tabla.

Las restricciones se pueden definir cuando creamos la tabla (CREATE TABLE) o posteriormente con la sentencia ALTER TABLE.

Las posibles restricciones son:

- NOT NULL

- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK
- DEFAULT

Conclusiones y recomendaciones

Defender la integridad de sus datos comerciales críticos utilizando métodos convencionales puede parecer una tarea difícil. Sin embargo, las herramientas modernas de integración de datos proporcionan una alternativa eficiente, que ofrece detección y depuración de errores en tiempo real.

Así mismo debemos de sacar en claro es que más vale prevenir que curar, resulta realmente fácil y barato ser precavido para evitar el desastre y tremendamente caro en tiempo y dinero recuperar los datos de un dispositivo que ha sido dañado por no tener ningún cuidado con el mismo.

Hemos hablado de integridad de datos porque la información no sólo se pierde, sino que también se corrompe, siendo igualmente perjudicial en ambos casos ya que el resultado es el mismo, se nos impide el acceso a la información, por ello debemos tener especial cuidado en ambos aspectos. La prevención es la mejor arma y desde luego no es algo que resulte difícil, además de que no debemos olvidarnos de las copias de seguridad que son la mejor forma para mantener nuestros datos a salvo independientemente del sistema que utilicemos, por ello siempre deberemos ser precavidos en ese aspecto.



Referencias bibliográficas

Esta lectura constituyen un apoyo en el proceso de aprendizaje, así mismo para apoyarse aun mas en el proceso, se debe utilizar los libros,

- Bertone, N. (2017). *Introducción a las bases de datos: fundamentos y diseño*. Pearson Educación
- Silberschatz, A. & Sudarshan, S. (2018). *Fundamentos de bases de datos* (6a. ed.). McGraw-Hill Interamericana.
- Bernal, Nieto. (2017). *Diseño de base de datos*. Universidad del Norte
- Pulido Romero, E. Escobar Domínguez, Ó. y Núñez Pérez, J. Á. (2019). *Base de datos*. Grupo Editorial Patria.

Tecnologias-Informacion. (2020). Integridad de Datos. Sitio de descarga de la información: <https://www.tecnologias-informacion.com/integridaddatos.html>

Wikipedia. (2020). Integridad de Datos. Sitio de descarga de la información: https://es.wikipedia.org/wiki/Integridad_de_datos

Talend. (2020). ¿En qué consiste la calidad de datos? . Sitio de descarga de la información: <https://www.talend.com/es/resources/what-is-data-quality/>



www.usanmarcos.ac.cr

San José, Costa Rica