

Bases de datos transaccionales

Entre los diferentes tipos de bases de datos que existen, encontramos las **bases de datos transaccionales**; estas bases de datos se han convertido en una de las piezas clave para cualquier tipo de empresa u organización. En esta entrada explicaremos **qué son las bases de datos transaccionales** y veremos sus principales características.

¿Qué son las bases de datos transaccionales? Definición

Las **bases de datos transaccionales** son bases de datos que tiene como fin el envío y recepción de datos a gran velocidad. Están destinadas generalmente al entorno de análisis de calidad, datos de producción e industrial, y su objetivo principal es asegurar las transacciones dentro de una base de datos relacional o, en caso de que no se puedan asegurar, revertirlas, de manera que evitan que las transacciones queden incompletas, es decir, o se realiza la transacción o no pasa nada (vuelve al estado original).

Características y funciones a tener en cuenta

La principal característica de este tipo de [bases de datos](#) es que permiten llevar a cabo un gran número de transacciones cortas en línea, permitiendo así el procesamiento de consultas rápido a una información muy actual y detallada. Además, mantienen la integridad de los datos en entornos de acceso múltiple, lo que garantiza altos niveles de efectividad.

Otras **características de las bases de datos transaccionales** que podemos señalar son las siguientes:

- Manejan datos operativos que provienen de sistemas OLTP (on-line transactional processing), cuyo objetivo es controlar y ejecutar tareas fundamentales para la empresa, puesto que permiten obtener una fotografía de los procesos de negocio en curso.
- Aunque las bases de datos transaccionales recogen los datos necesarios para la aplicación a la hora de ejecutar sus operaciones, actualmente también capturan datos sobre el contexto histórico de la transacción, con el fin de poder utilizarlos en análisis posteriores.
- Este tipo de bases de datos están optimizada para añadir actualizaciones cortas y rápidas en tiempo real por los usuarios finales.
- Se pueden realizar consultas cuando estas están estandarizadas y son simples.
- Tienen una gran velocidad de procesamiento, lo que permite realizar consultas y obtener resultados de manera muy rápida.

- Ocupan poco espacio.
- Tienen un diseño normalizado basado en tablas.

Ventajas y desventajas

Veamos qué tipo de **ventajas y desventajas tienen las bases de datos transaccionales** a la hora de implementarlas en la empresa.

Ventajas

Una de las principales ventajas de las bases de datos transaccionales es que permiten asegurar la integridad de los datos (puesto que están diseñadas con propiedades ACID), además, se puede modificar la información sin poner en riesgo dicha integridad.

Son rápidas y operan con muy baja latencia, permitiendo replicar datos o recuperarlos de los almacenes en muy poco tiempo, lo que facilita llevar a cabo operaciones de monitorio en tiempo real. Además, reducen al máximo el riesgo de pérdida de datos por fallos en el sistema.

Dado que actualmente recogen datos históricos de las transacciones, se pueden emplear para facilitar los análisis posteriores.

Desventajas

La principal desventaja de las bases de datos transaccionales es la limitación que tienen para generar informes, puesto que, como ya hemos mencionado, permiten realizar consultas simples para obtener una radiografía de los procesos del negocio, especialmente de los de producción. Por lo tanto, necesitan trabajar de manera asociada a una [base de datos relacional](#) o a bases de datos multidimensionales para poder sacarles mayor partido (por **definición, las bases de datos multidimensionales** están creadas usando entradas de las bases de datos relacionales existentes).

Además, el historial de datos que facilitan a través de su consulta es limitado a datos actuales o recientes.

¿Cómo funcionan y para qué sirven?

Las bases de datos transaccionales utilizan lenguaje SQL, que permite el acceso y la modificación de los datos dentro de la bases de datos.

Por las características que hemos visto más arriba, las bases de datos transaccionales resultan útiles cuando la integridad de los datos es importante, puesto que no permiten que la transacción se complete si uno de los pasos de la misma falla.

Así, una transacción generará un proceso atómico que puede conllevar bien operaciones de inserción, modificación o borrado de datos. Este proceso debe ser validado con un «commit» o invalidado con «rollback»; con el primero se ejecuta la operación y con el segundo no se produce, volviendo al estado original.

Sirven especialmente para evitar que la integridad de la base de datos se vea afectada por posibles fallos en el sistema cuando la transacción se está produciendo. Y para obtener información de manera rápida y en tiempo real.

Uso y aplicaciones

Las bases de datos transaccionales son una de las primeras bases de datos que se implementan en los sistemas de las empresas u organizaciones, puesto que dan apoyo a las tareas operativas de las mismas.

Así, entre las diferentes **aplicaciones de las bases de datos transaccionales**, encontramos las siguientes:

- Las bases de datos transaccionales nos permiten obtener, como ya hemos visto, datos almacenados de manera muy rápida, que además ofrecen una imagen actual de los procesos de la empresa. Por lo tanto, una de las aplicaciones que se le dan a estas bases de datos es la obtener información para su posterior análisis y toma de decisiones tácticas.
- A través de las bases de datos transaccionales se pueden organizar los almacenes de datos de las empresas, puesto que permiten dotarles de un esquema común y optimizarlos para el procesamiento de consultas complejas.
- Permiten contextualizar las transacciones llevadas a cabo por aplicaciones operativas, lo que otorga una visión más completa al análisis de la información.
- Cuando se integran con bases de datos analíticas en una sola plataforma, se consigue una mayor consistencia del procesamiento de transacciones y, por tanto, la obtención de información útil (como pueden ser informes sobre tendencias de ventas o el comportamiento de los clientes).

Base de datos relacional vs transaccional

Lo cierto es que más que hablar de **base datos relacional vs transaccional**, debemos hablar de **bases de datos transaccional y relacional**, puesto que ambas son dos aspectos de la gestión de datos, que actualmente van de la mano, puesto que, como ya dijimos más arriba, la base de datos transaccionales funciona de manera asociada a una base de datos relacional.

Es decir, la función de la base de datos transaccionales es asegurar que las transacciones dentro de la base de datos relacional se cumplen de manera completa o, en caso de haber un error o fallo en el proceso, se reviertan y por tanto no se completan (un sistema de «todo o nada»).

En cierto sentido, la base de datos relacional nos indica cómo se almacenan los datos y la base de datos transaccional nos dice cómo se modifican esos datos.

Ejemplos

Vamos a ver algunos **ejemplos de bases de datos transaccionales** para ilustrar mejor esta entrada.

Así, el primer ejemplo lo encontramos en las transferencias bancarias; las bases de datos transaccionales se emplean aquí para validar o deshacer la transferencia. Puesto que las transferencias bancarias se desarrollan en dos operaciones distintas, primero se descuenta el dinero en la cuenta origen y luego se suma en la cuenta de destino, si la segunda operación falla, el dinero se devuelve a la cuenta de origen y la transferencia no tiene lugar y, por tanto, no se registra en la base de datos. El sistema gestor de la base de datos transaccional garantiza que las dos operaciones se cumplen o ninguna de ellas lo haga.

Otro ejemplo lo encontramos en los sistemas de venta online, puesto, como en el caso anterior, registra una transacción, una venta y un ingreso de dinero para que esta se produzca. Si algún error durante el proceso (por ejemplo, en la pasarela de pago), la transacción no se completa, es decir, no se produce la venta ni el descuento del dinero, y por tanto, no se registra en la base de datos.