República Bolivariana de Venezuela

Ministerio del Poder popular para la Educación Universitaria

Universidad Politécnica Territorial del Estado Portuguesa “Juan de Jesús Montilla”

Programa Nacional de Formación en Informática

Acarigua – Edo. Portuguesa

BASE DE DATOS

Docente: Estudiantes:

Silva H. José B 28503559 Jesus Rivero

Unidad curricular: PNF:

Administración de Base de Datos Informática Sección 831

Acarigua, Septiembre 2023

Contenido

[**Introducción:** 3](#_Toc145010307)

[**Database backup** 4](#_Toc145010308)

[**Ejemplo:** 5](#_Toc145010309)

[**Tablespace** 6](#_Toc145010310)

[**Ejemplo:** 7](#_Toc145010311)

[**Backup set** 8](#_Toc145010312)

[**Ejemplo:** 9](#_Toc145010313)

[**Introducción a la recuperación en base de datos** 10](#_Toc145010314)

[**Clasificación** **de algoritmos de recuperación.** 12](#_Toc145010315)

[**Técnicas de Recuperación: basadas en la actualización diferida** 14](#_Toc145010316)

[**Técnicas de Recuperación: Basadas en la actualización inmediata** 16](#_Toc145010317)

[Comparación: 19](#_Toc145010318)

[**Conclusión:** 20](#_Toc145010319)

# **Introducción:**

En el mundo actual, la gestión de datos desempeña un papel central en prácticamente todas las esferas de la sociedad y los negocios. La correcta administración de bases de datos es esencial para garantizar la integridad, disponibilidad y seguridad de los datos críticos que impulsan la toma de decisiones y las operaciones de empresas e instituciones. Uno de los aspectos más cruciales de la gestión de bases de datos es la recuperación de datos en situaciones adversas, como fallos de hardware, errores de software, errores humanos y otros eventos imprevistos que pueden poner en riesgo la integridad de los datos almacenados. En este contexto, este trabajo se sumerge en el fascinante mundo de la recuperación de datos en bases de datos. Exploraremos las técnicas y estrategias utilizadas para garantizar que, incluso en los momentos más críticos, los datos puedan ser restaurados a un estado coherente y operativo. Examinaremos dos enfoques fundamentales en las técnicas de recuperación: la actualización inmediata y la actualización diferida, y discutiremos sus ventajas y desventajas en el contexto de la gestión de bases de datos. Además, exploraremos otras estrategias, como la confirmación retrasada, el registro de transacciones y el registro de cambios, que desempeñan un papel vital en el proceso de recuperación. A medida que avanzamos en esta exploración, comprenderemos la importancia crítica de la recuperación de datos en bases de datos, ya que se encuentra en la intersección de la tecnología, la seguridad de la información y la gestión empresarial. La habilidad para garantizar la disponibilidad y la integridad de los datos en momentos de crisis se ha convertido en una competencia clave en el mundo moderno, y este trabajo proporcionará una visión profunda de los conceptos y enfoques detrás de esta tarea esencial.

# **Database backup**

Una copia de seguridad de una base de datos Database backup es una copia de los datos almacenados en una base de datos en un momento específico en el tiempo. Esta copia de seguridad se crea con el propósito de preservar los datos en caso de pérdida, daño, corrupción o eliminación accidental de la información original. Los backups de bases de datos son una parte fundamental de la gestión de datos y de la seguridad de la información en sistemas de bases de datos.

Las copias de seguridad de bases de datos se utilizan en varios escenarios, incluidos los siguientes:

Recuperación ante fallos: Si una base de datos sufre un fallo, ya sea debido a un error de hardware, un error de software, un ataque cibernético u otros problemas, las copias de seguridad pueden utilizarse para restaurar la base de datos a un estado previo al fallo y recuperar los datos perdidos.

Cumplimiento normativo: En muchas industrias y organizaciones, se requiere mantener copias de seguridad regulares de los datos para cumplir con normativas y regulaciones específicas.

Restauración de datos eliminados o corruptos: Si se eliminan datos importantes de una base de datos o se corrompen debido a errores humanos, las copias de seguridad pueden ser útiles para recuperar esos datos.

Migración y clonación de bases de datos: Las copias de seguridad se pueden utilizar para crear copias de una base de datos en otro servidor o entorno, lo que es útil para pruebas, desarrollo o migraciones.

# **Ejemplo:**

Crear una copia de seguridad completa de la base de datos "mi\_tienda"

El resultado se guardará en un archivo SQL

-Primero, establecemos la ubicación donde se guardará la copia de seguridad

SET @backup\_path = '/ruta/del/archivo/backup\_mi\_tienda.sql';

A continuación, ejecutamos el comando para crear la copia de seguridad

Utilizamos el comando "mysqldump" para exportar todos los datos de la base de datos y guardarlo en un archivo SQL. Los parámetros -u y -p se utilizan para especificar el nombre de usuario y la contraseña de la base de datos

En este ejemplo, el usuario es "usuario\_db" y la contraseña es "contraseña\_db"

También se especifica el nombre de la base de datos "mi\_tienda"

El resultado se redirige al archivo de copia de seguridad

SET @command = CONCAT(

'mysqldump -u usuario\_db -pcontraseña\_db mi\_tienda > "', @backup\_path, '"'

);

-- Ejecutar el comando

PREPARE backup\_statement FROM @command;

EXECUTE backup\_statement;

DEALLOCATE PREPARE backup\_statement;

-- Una vez ejecutado, la copia de seguridad se almacenará en el archivo especificado en @backup\_path

# **Tablespace**

Un "backup de tablespace" se refiere a una copia de seguridad de una o más áreas lógicas de almacenamiento en una base de datos que están organizadas en tablespaces. Los tablespaces son estructuras lógicas utilizadas para administrar y organizar el almacenamiento de tablas, índices y otros objetos en una base de datos. Realizar un backup de tablespace implica crear una copia de seguridad de estos espacios lógicos para proteger los datos almacenados en ellos.

Aquí hay algunos puntos clave relacionados con los backups de tablespaces:

Segmentos de almacenamiento: Dentro de un tablespace, los datos se almacenan en segmentos de almacenamiento, que incluyen segmentos de datos, segmentos de índices, segmentos temporales y otros. Cada uno de estos segmentos puede respaldarse de forma independiente.

Tipos de tablespaces: En una base de datos, puede haber varios tipos de tablespaces, como tablespaces de datos, tablespaces de índices, tablespaces temporales, entre otros. Cada tipo de tablespace puede contener datos específicos y, por lo tanto, puede ser respaldado individualmente según las necesidades.

Selección de tablespaces: Al realizar un backup de tablespace, los administradores de bases de datos pueden elegir respaldar tablespaces específicos según los requisitos de recuperación y la importancia de los datos. Esto permite una recuperación selectiva en caso de fallos.

Métodos de respaldo: Los backups de tablespaces se pueden realizar utilizando herramientas y comandos específicos proporcionados por el sistema de gestión de bases de datos. Estos métodos pueden variar según el sistema utilizado (por ejemplo, Oracle Database, PostgreSQL, MySQL) y pueden incluir copias de seguridad físicas o lógicas.

Restauración y recuperación: En caso de un fallo en la base de datos o la pérdida de datos en un tablespace específico, los backups de tablespaces se utilizan para restaurar esos tablespaces a un estado anterior. Esto puede implicar la restauración de archivos físicos o la carga de datos desde una copia lógica.

Programación y retención: Los backups de tablespaces suelen programarse de forma regular como parte de la estrategia general de copia de seguridad de una base de datos. También se establece una política de retención para determinar cuánto tiempo se deben conservar las copias de seguridad de tablespaces.

# **Ejemplo:**

Supongamos que tienes un tablespace llamado "mi\_tablespace" y deseas realizar un backup de él:

Iniciar un backup del tablespace "mi\_tablespace"

ALTER TABLESPACE mi\_tablespace BEGIN BACKUP;

Esto copiará los archivos de datos del tablespace a una ubicación de respaldo. Puedes especificar la ubicación de respaldo adecuada

En este ejemplo, utilizamos '/ruta/del/respaldo/'

Copiar archivos de datos del tablespace

En sistemas Unix/Linux, puedes utilizar el comando 'cp':

cp /ruta/del/tablespace/mi\_tablespace.dbf /ruta/del/respaldo/

En sistemas Windows, puedes utilizar el comando 'copy' o 'xcopy':

copy "C:\ruta\del\tablespace\mi\_tablespace.dbf" "C:\ruta\del\respaldo\"

Finalizar el backup del tablespace "mi\_tablespace"

ALTER TABLESPACE mi\_tablespace END BACKUP;

En este ejemplo:

Utilizamos el comando ALTER TABLESPACE con la opción BEGIN BACKUP para iniciar el backup del tablespace "mi\_tablespace". Esto coloca el tablespace en modo de copia de seguridad y permite que se realice una copia física de los archivos de datos.

Luego, copiamos los archivos de datos del tablespace a una ubicación de respaldo adecuada utilizando los comandos apropiados para tu sistema operativo. En el ejemplo, hemos proporcionado comentarios con ejemplos de comandos para sistemas Unix/Linux y Windows.

Finalmente, utilizamos el comando ALTER TABLESPACE con la opción END BACKUP para finalizar el backup del tablespace. Esto saca el tablespace del modo de copia de seguridad.

# **Backup set**

Un "backup set" es una colección de archivos de copia de seguridad que se agrupan juntos como una unidad lógica. Estos archivos de copia de seguridad contienen datos respaldados de una base de datos u otro tipo de información que se almacena en un sistema de almacenamiento externo, como cintas magnéticas, discos duros externos o servidores de copia de seguridad en la nube. Los backup sets se utilizan comúnmente en el contexto de la administración de copias de seguridad y recuperación de datos.

Agrupación lógica: Los backup sets agrupan archivos de copia de seguridad relacionados en una sola unidad lógica para facilitar su gestión y recuperación. Estos archivos pueden incluir copias de seguridad completas, copias de seguridad incrementales o diferencias, archivos de registro de transacciones y otros datos respaldados.

Identificación y etiquetado: Cada backup set suele estar etiquetado de manera única para que los administradores de copias de seguridad puedan identificar fácilmente su contenido y fecha de creación. Esto facilita el seguimiento de las copias de seguridad y su recuperación posterior.

Restauración selectiva: Los backup sets permiten a los administradores de bases de datos y sistemas restaurar selectivamente datos o componentes específicos de una copia de seguridad, en lugar de tener que restaurar todo el conjunto de datos de una vez. Esto es útil para recuperar datos individuales o recuperar un sistema a un punto en el tiempo específico.

Eficiencia de almacenamiento: Al agrupar varios archivos de copia de seguridad en un conjunto, se puede lograr una mejor eficiencia de almacenamiento al eliminar duplicados y reducir el espacio necesario para mantener múltiples copias de seguridad.

Política de retención: Los backup sets suelen estar sujetos a políticas de retención que determinan cuánto tiempo se mantienen las copias de seguridad en el almacenamiento antes de su eliminación o reemplazo. Esto es importante para la gestión del espacio de almacenamiento.

Seguridad y encriptación: Los backup sets pueden estar protegidos mediante medidas de seguridad, como la encriptación, para garantizar la confidencialidad de los datos respaldados.

La creación y gestión de backup sets son prácticas esenciales en la administración de copias de seguridad de datos críticos, ya que proporcionan una forma organizada y eficiente de mantener y recuperar datos en caso de pérdida o corrupción. La forma en que se crean y gestionan los backup sets puede variar según el software y la tecnología utilizados en la infraestructura de copia de seguridad y recuperación.

# **Ejemplo:**

Creación de un Backup Set: Supongamos que deseas crear un backup set llamado "BackupSet2023" que incluye copias de seguridad de las tablas de clientes y productos de tu base de datos.

Nombre del Backup Set: BackupSet2023

Contenido: Tablas de clientes y productos

Proceso de Copia de Seguridad:

El software de respaldo inicia un proceso de copia de seguridad y se conecta a la base de datos.

Se seleccionan las tablas de clientes y productos para respaldar.

Las copias de seguridad de estas tablas se comprimen y cifran para proteger los datos.

Los archivos de copia de seguridad resultantes se almacenan en un dispositivo de almacenamiento de respaldo, como una unidad de disco o un servidor de copia de seguridad en la nube.

Etiquetado y Organización:

Cada archivo de copia de seguridad dentro del conjunto "BackupSet2023" se etiqueta con información relevante, como la fecha y la hora de creación.

Archivo de Copia de Seguridad 1: Backup\_Clientes\_2023-09-07.bkp

Archivo de Copia de Seguridad 2: Backup\_Productos\_2023-09-07.bkp

Recuperación Selectiva:

Si en el futuro necesitas restaurar datos específicos, puedes utilizar el conjunto "BackupSet2023" para recuperar las copias de seguridad de las tablas de clientes o productos sin necesidad de restaurar todo el sistema.

Política de Retención:

De acuerdo con la política de retención de tu empresa, se mantendrán estas copias de seguridad durante un período específico. Por ejemplo, podrían retenerse durante 30 días antes de ser reemplazadas por nuevas copias de seguridad.

# **Introducción a la recuperación en base de datos**

La recuperación en bases de datos es un aspecto fundamental de la gestión de datos y se refiere al proceso de restauración de una base de datos a un estado consistente y operativo después de un fallo, un error humano, una corrupción de datos o cualquier otro evento que haya afectado su integridad. La recuperación es esencial para garantizar la disponibilidad y la fiabilidad de los sistemas de bases de datos, especialmente en entornos críticos como sistemas bancarios, sistemas de gestión de inventarios y sistemas de registros médicos, entre otros.

Conceptos clave relacionados con la recuperación en bases de datos:

Fallas en la base de datos: Las fallas pueden ser de varios tipos, como fallas de hardware (disco duro defectuoso), fallas de software (error de programación), fallas del sistema (corte de energía), ataques cibernéticos (virus, ransomware) y errores humanos (eliminación accidental de datos).

Puntos de control (checkpoints): Los sistemas de gestión de bases de datos (DBMS) suelen utilizar puntos de control periódicos para registrar el estado actual de la base de datos. Esto permite realizar una recuperación más eficiente a partir de un punto conocido en el tiempo si ocurre una falla.

Registro de transacciones: Los DBMS mantienen un registro de todas las transacciones que afectan la base de datos. Este registro permite reconstruir los cambios realizados en la base de datos en caso de una recuperación.

Recuperación de respaldo (backup): La base de datos se respalda periódicamente para permitir la restauración en caso de pérdida de datos. Los backups pueden ser completos (de toda la base de datos) o parciales (de partes específicas).

Recuperación a nivel de transacción: Los sistemas de bases de datos a menudo permiten la recuperación a nivel de transacción, lo que significa que puedes deshacer cambios específicos de una transacción sin afectar otras transacciones.

Recuperación ante fallas: Cuando se produce una falla, el DBMS utiliza la información de los puntos de control, el registro de transacciones y los backups para determinar el punto de recuperación adecuado y restaurar la base de datos a ese estado.

Recuperación y redundancia: En entornos críticos, se utilizan técnicas de redundancia como replicación de datos, clústeres de alta disponibilidad y almacenamiento redundante para garantizar una recuperación más rápida y una menor pérdida de datos.

Pruebas de recuperación: Es esencial realizar pruebas regulares de recuperación para garantizar que los procedimientos de recuperación sean efectivos y que los datos se puedan restaurar con éxito en caso de una falla real.

# **Clasificación** **de algoritmos de recuperación.**

Los algoritmos de recuperación en bases de datos se clasifican en función de su enfoque y su nivel de granularidad. Aquí te presento una clasificación general de los algoritmos de recuperación:

Recuperación a nivel de sistema:

Reinicio de base de datos (Database Restart): En caso de una falla, se reinicia toda la base de datos desde un punto de control o un backup completo. Este enfoque es simple pero puede implicar una pérdida significativa de datos y tiempo de inactividad.

Copia de seguridad y restauración (Backup and Restore): Se utilizan backups regulares para restaurar la base de datos a un estado previo a la falla. Este método puede ser más rápido que el reinicio de la base de datos completa, pero la pérdida de datos generalmente sigue siendo posible.

Recuperación a nivel de transacción:

Recuperación de registro de transacciones (Transaction Log Recovery): Este enfoque se centra en la recuperación de transacciones individuales. El sistema de gestión de bases de datos (DBMS) utiliza el registro de transacciones para deshacer transacciones no confirmadas (rollback) y aplicar transacciones confirmadas (commit).

Puntos de verificación (Checkpointing): Los puntos de control se utilizan para registrar el estado de la base de datos en momentos específicos. Durante la recuperación, el DBMS puede retroceder a un punto de control conocido y aplicar las transacciones desde ese punto en adelante.

Recuperación a nivel de página o bloque:

Recuperación de página (Page Recovery): En este nivel de granularidad, se recuperan páginas individuales o bloques de datos dañados o corruptos. Esto permite una recuperación más fina y evita la pérdida de datos innecesaria.

Journaling (Diario): Algunos sistemas de bases de datos utilizan registros de cambios (journals) para registrar las modificaciones a nivel de página o bloque. Estos registros se pueden utilizar para aplicar los cambios correctos durante la recuperación.

Recuperación a nivel de registro:

Recuperación de registro (Log Recovery): Este enfoque se centra en la recuperación de registros de transacciones individuales. Los registros de transacciones se utilizan para deshacer o reflejar cambios en la base de datos en caso de una falla.

Recuperación a nivel de objeto o tabla:

Recuperación de objetos individuales (Object Recovery): En algunos sistemas, es posible realizar la recuperación solo de objetos de base de datos específicos, como tablas o índices, en lugar de toda la base de datos.

Recuperación en tiempo real:

Replicación en tiempo real (Real-Time Replication): En entornos críticos, los datos se replican en tiempo real a servidores de respaldo para garantizar una recuperación rápida y sin pérdida de datos en caso de una falla.

Recuperación basada en registros de cambios (Change Data Capture - CDC):

CDC captura los cambios de datos (Data Change Tracking): Se registran los cambios de datos en una base de datos en tiempo real o periódicamente. Esto permite recuperar cambios específicos realizados en los datos en caso de una falla.

# **Técnicas de Recuperación: basadas en la actualización diferida**

Las técnicas de recuperación basadas en la actualización diferida son estrategias utilizadas en sistemas de gestión de bases de datos (DBMS) para gestionar la recuperación de datos en caso de fallos. Estas técnicas se centran en retrasar la escritura de datos actualizados en el almacenamiento persistente hasta que se haya confirmado que la transacción se completó con éxito (confirmación). Esto ayuda a evitar que los datos se escriban en el almacenamiento permanente si la transacción falla, lo que reduce la posibilidad de corrupción de datos en caso de una falla del sistema o un error.

Técnicas comunes de recuperación basadas en la actualización diferida:

Protocolo de confirmación retrasada (Deferred Commit Protocol):

En este enfoque, las transacciones realizan actualizaciones en la memoria de la base de datos, pero no se escriben inmediatamente en el almacenamiento permanente. En su lugar, se mantienen en un área temporal llamada "memoria intermedia" o "memoria intermedia de actualización". La confirmación de una transacción solo ocurre después de que se ha completado con éxito. Si la transacción falla, sus actualizaciones en la memoria intermedia no se escriben en el almacenamiento permanente.

**Ventajas:**

Evita la escritura de datos no confirmados en el almacenamiento permanente.

Reduce la sobrecarga de E/S en el almacenamiento permanente durante la ejecución de transacciones.

Facilita la recuperación y el deshacer de transacciones fallidas.

**Desventajas:**

Puede requerir espacio adicional en la memoria intermedia para almacenar actualizaciones pendientes.

Puede aumentar el riesgo de pérdida de datos si se produce una falla antes de la confirmación.

Registro de transacciones (Transaction Logging):

En esta técnica, todas las actualizaciones de transacciones se registran en un archivo de registro de transacciones antes de que se apliquen a la base de datos real. El registro de transacciones contiene una secuencia de registros de cambios que describen las operaciones realizadas por las transacciones. Los registros de transacciones se escriben en el almacenamiento permanente antes de que las actualizaciones se apliquen a la base de datos.

**Ventajas:**

Proporciona un registro detallado de todas las transacciones, lo que facilita la recuperación en caso de fallos.

Permite la recuperación a nivel de registro, lo que significa que se pueden deshacer o reflejar transacciones individuales en caso de problemas.

**Desventajas:**

Puede aumentar la sobrecarga de E/S debido a la escritura constante de registros de transacciones.

Requiere espacio adicional en disco para almacenar los registros de transacciones.

Estas técnicas de recuperación basadas en la actualización diferida son cruciales para garantizar la integridad de los datos y la recuperación adecuada en sistemas de bases de datos. La elección de la técnica adecuada depende de los requisitos de la aplicación, la tolerancia a la pérdida de datos y otros factores específicos del entorno de la base de datos.

# **Técnicas de Recuperación: Basadas en la actualización inmediata**

Las técnicas de recuperación basadas en la actualización inmediata son estrategias utilizadas en sistemas de gestión de bases de datos (DBMS) para gestionar la recuperación de datos en caso de fallos. A diferencia de las técnicas de actualización diferida, en las cuales se retrasa la escritura de datos actualizados en el almacenamiento permanente hasta que se haya confirmado que la transacción se completó con éxito, las técnicas de actualización inmediata implican escribir datos en el almacenamiento permanente tan pronto como se realice una actualización. A continuación, se describen dos técnicas comunes de recuperación basadas en la actualización inmediata:

**Confirmación inmediata (Immediate Commit):**

En este enfoque, una vez que una transacción ha realizado una actualización en la base de datos, los cambios se escriben de inmediato en el almacenamiento permanente, incluso antes de que la transacción se haya completado con éxito y se haya confirmado. Esto significa que los cambios son visibles para otras transacciones de inmediato. Si una transacción falla después de realizar cambios, es necesario realizar una reversión (rollback) para deshacer los cambios realizados por esa transacción.

Ventajas:

Las actualizaciones son visibles para otras transacciones de inmediato, lo que puede ser útil en situaciones en las que se requiera concurrencia.

Reduce la necesidad de espacio de memoria intermedia para almacenar cambios no confirmados.

Desventajas:

Puede aumentar la probabilidad de que ocurran problemas de integridad de datos si una transacción falla después de realizar cambios.

Puede requerir más esfuerzo de recuperación y reversión en caso de fallos.

**Registro de cambios (Change Logging):**

En esta técnica, en lugar de escribir directamente en la base de datos, las actualizaciones de una transacción se registran en un archivo de registro de cambios (log). El registro de cambios contiene información detallada sobre las operaciones realizadas por las transacciones, incluidas las actualizaciones. Las actualizaciones se aplican a la base de datos solo después de que la transacción se haya completado con éxito y se haya confirmado. Si una transacción falla, los cambios registrados en el registro de cambios se pueden deshacer sin afectar la base de datos real.

Ventajas:

Proporciona un registro detallado de todas las operaciones realizadas por las transacciones, lo que facilita la recuperación en caso de fallos.

Permite un mayor control sobre las operaciones de reversión en caso de fallas.

Desventajas:

Aumenta la sobrecarga de E/S debido a la escritura constante en el registro de cambios y requiere espacio adicional en disco para almacenar el registro de cambios.

La elección entre técnicas de actualización inmediata y actualización diferida depende de los requisitos específicos de la aplicación y del equilibrio entre concurrencia y recuperación. Las técnicas de actualización inmediata tienden a ofrecer una mayor visibilidad de los cambios y una menor necesidad de almacenamiento en memoria intermedia, pero pueden ser más propensas a problemas de integridad de datos en caso de fallos. Por otro lado, las técnicas de actualización diferida pueden reducir el riesgo de problemas de integridad de datos, pero pueden requerir más recursos de almacenamiento y más esfuerzo de recuperación.

Para ilustrar las diferencias entre las técnicas de actualización inmediata y actualización diferida en la recuperación de bases de datos, consideremos un escenario hipotético con dos transacciones: una transacción de depósito y una transacción de retiro en una cuenta bancaria.

**Escenario:**

**Técnica de Actualización Inmediata:**

En este enfoque, las actualizaciones se escriben de inmediato en el almacenamiento permanente.

Transacción de Depósito (Actualización Inmediata):

Un cliente realiza un depósito de $100 en su cuenta.

El saldo se actualiza inmediatamente en la base de datos.

Si la transacción se confirma con éxito, los cambios son permanentes.

Si la transacción falla (por ejemplo, debido a un error en la transacción), los $100 ya se han agregado al saldo y deben deshacerse.

Transacción de Retiro (Actualización Inmediata):

Un cliente intenta retirar $200 de su cuenta.

El saldo actual es $100.

La base de datos intenta aplicar la actualización de retiro de inmediato.

La transacción falla porque no hay suficiente saldo.

Los cambios ya se han aplicado a la base de datos y deben deshacerse para mantener la integridad de la cuenta.

Técnica de Actualización Diferida:

En este enfoque, las actualizaciones se mantienen en la memoria temporal hasta que se confirme la transacción.

**Transacción de Depósito (Actualización Diferida):**

Un cliente realiza un depósito de $100 en su cuenta.

El saldo se actualiza en la memoria temporal, pero no en el almacenamiento permanente.

Si la transacción se confirma con éxito (por ejemplo, después de verificar fondos suficientes), los cambios se escriben en el almacenamiento permanente.

Si la transacción falla, los cambios en la memoria temporal se descartan sin afectar el saldo en la base de datos.

Transacción de Retiro (Actualización Diferida):

Un cliente intenta retirar $200 de su cuenta.

El saldo actual es $100.

La base de datos registra la solicitud de retiro en la memoria temporal.

La transacción falla debido a fondos insuficientes.

Como la actualización no se escribió en el almacenamiento permanente, no se han afectado los fondos en la cuenta.

# Comparación:

En la técnica de actualización inmediata, los cambios se aplican de inmediato y pueden causar problemas de integridad si la transacción falla después de la actualización mientras que en la técnica de actualización diferida, las actualizaciones se retienen en la memoria temporal hasta que se confirma la transacción, lo que evita problemas de integridad en caso de fallas.

La elección entre estas técnicas depende de los requisitos de la aplicación y las consideraciones de recuperación. La actualización inmediata puede ser más eficiente en términos de concurrencia, pero la actualización diferida proporciona un mayor control sobre la reversión de cambios en caso de fallas.

# **Conclusión:**

En el ámbito de la gestión de bases de datos, la recuperación de datos emerge como una preocupación crítica y una disciplina fundamental. La correcta administración de datos es esencial para salvaguardar la integridad y disponibilidad de la información vital en un mundo donde la dependencia de la tecnología y la toma de decisiones basada en datos es omnipresente. Este trabajo ha explorado a fondo las técnicas y estrategias de recuperación en bases de datos, arrojando luz sobre dos enfoques principales: la actualización inmediata y la actualización diferida. Hemos comprendido que la elección entre estas dos técnicas no es trivial, sino que depende de una serie de factores, incluida la necesidad de concurrencia, la tolerancia a la pérdida de datos y la complejidad del sistema en cuestión. La actualización inmediata, que registra cambios de inmediato en el almacenamiento permanente, ofrece ventajas en términos de visibilidad y eficiencia, pero conlleva un mayor riesgo de corrupción de datos en caso de falla. Por otro lado, la actualización diferida, que retiene los cambios hasta que se confirma la transacción, minimiza el riesgo de problemas de integridad, aunque a expensas de la concurrencia y el consumo de recursos de almacenamiento.