

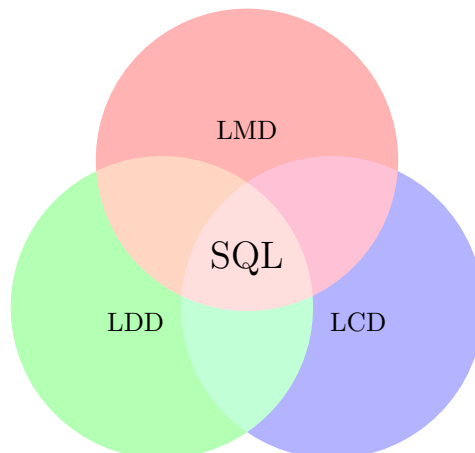
Laboratorio de Base de Datos Prelaboratorio Nro. 7, Server Programming

Prof. Solazver Solé
Preps. Victor Albornoz, Yenifer Ramirez
Semestre B-2018

1. SQL

SQL por sus siglas en inglés (**Structured Query Language**)¹ es un lenguaje de dominio específico, usado en la programación y diseñado para administrar los datos persistentes en los sistemas de gestión de bases de datos relacionales (SGBDR).

Originalmente se basó en el **álgebra relacional** y en **cálculo relacional de tuplas**, SQL es tanto un lenguaje de manipulación de datos (**LMD**), como un lenguaje de definición de datos (**LDD**) y un lenguaje de control de datos (**LCD**).



¹Lenguaje estructurado de consultas

2. Procedimientos almacenados de bases de datos y funciones

Aunque es adecuado para muchas aplicaciones tratar al programa de aplicación de bases de datos como si este se encontrara en un computador cliente, en ocasiones es útil crear módulos de programa de bases de datos (procedimientos o funciones) que el DBMS almacena y ejecuta en el servidor de bases de datos. Es lo que históricamente se conoce como procedimientos almacenados de bases de datos, aunque pueden ser funciones o procedimientos. Los procedimientos almacenados, son módulos de programa que el DBMS almacena en el servidor de bases de datos. Los procedimientos y funciones almacenados deben ser escritos en un lenguaje de programación de propósito general. De forma alternativa, un procedimiento almacenado puede estar compuesto por comandos SQL sencillos, como recuperaciones y actualizaciones.

Si el procedimiento (o función) se escribe en un lenguaje de programación de propósito general, es típico especificar el lenguaje dentro del código.

Los procedimientos almacenados son útiles en las siguientes circunstancias:

- Si varias aplicaciones necesitan un mismo programa de bases de datos, este último se puede almacenar en el servidor e invocarlo desde esas aplicaciones. Esto reduce la duplicidad del esfuerzo y mejora la modularidad del software.
- La ejecución de un programa en el servidor puede reducir el coste derivado de la transferencia y la comunicación de datos entre el cliente y el servidor en ciertas situaciones.
- Estos procedimientos pueden mejorar la potencia de modelado de las vistas al permitir que los usuarios de bases de datos cuenten con tipos más complejos de datos derivados. Además, se pueden utilizar esos tipos para comprobar restricciones más complejas que quedan fuera de la especificación de aserciones y triggers. En general, muchos DBMSs comerciales permiten escribir procedimientos y funciones almacenados en un lenguaje de programación de propósito general. De forma alternativa, un procedimiento almacenado puede estar compuesto por comandos SQL sencillos, como recuperaciones y actualizaciones.
- Los procedimientos almacenados van de la mano con los disparadores "Triggers", ofrecen estructuras de control, permiten realizar cálculos complejos desde el servidor y son fáciles de implementar.

3. Aserciones y Disparadores

En SQL, podemos especificar restricciones generales mediante aserciones declarativas, utilizando la sentencia `CREATE ASSERTION` del LDD. Una aserción es un predicado incluido en un programa que se asume siempre se cumple en ese punto del flujo de programa. En caso de ser violada la condición especificada en el predicado, se aborta por completo la operación que causo la violación de dicho predicado.

Otra sentencia relacionada con `CREATE ASSERTION` es `CREATE TRIGGER`, pero los triggers se utilizan de modo diferente. En muchos casos es conveniente especificar el tipo de acción que se llevará a cabo cuando se producen ciertos eventos y cuando se satisfacen determinadas condiciones. En lugar de ofrecer al usuario la opción de abortar una operación que provoca una violación (como con `CREATE ASSERTION`) el DBMS debe contar con otras opciones. Por ejemplo, puede ser útil especificar una condición que, en caso de ser violada, informe al usuario de alguna forma. Un director puede estar interesado en ser informado de si los gastos de viaje de un empleado exceden un cierto límite, recibiendo un mensaje cuando esto sucede. La acción que el DBMS debe tomar en este caso es enviar un mensaje a ese usuario. La condición se utiliza entonces para monitorizar la base de datos. Es posible especificar otras acciones, como la ejecución de un procedimiento almacenado específico o la activación de otras actualizaciones. En SQL se utiliza la sentencia `CREATE TRIGGER` para implementar dichas acciones. Un trigger especifica un evento (por ejemplo, una operación de actualización de la base de datos), una condición y una acción. La acción se ejecuta automáticamente si se satisface la condición cuando se produce el evento.

4. Transacciones

Una transacción es un programa en ejecución que constituye una unidad lógica del procesamiento de una base de datos. Una transacción incluye una o más operaciones de acceso a la base de datos (operaciones de inserción, eliminación, modificación o recuperación). Las operaciones con bases de datos que forman una transacción pueden estar incrustadas dentro de una aplicación o pueden especificarse interactivamente mediante un lenguaje de consulta de alto nivel como SQL. Una forma de definir los límites de una transacción es especificando explícitamente las sentencias `begin transaction` y `end transaction` en un programa de aplicación; en este caso, todas las operaciones de acceso a la base de datos que se encuentran entre las dos sentencias son consideradas como una transacción.

El concepto de transacción proporciona un mecanismo para definir las unidades lógicas del procesamiento de una base de datos. Los sistemas de procesamiento de transacciones son sistemas con grandes bases de datos y cientos de usuarios concurrentes ejecutando transacciones de bases de datos. Entre estos sistemas podemos citar los de reservas en aerolíneas, banca, procesamiento de tarjetas de crédito, mercado de acciones, etcétera. Estos sistemas requieren una alta disponibilidad y una respuesta rápida para cientos de usuarios simultáneos. Podemos definir una transacción de una base de datos, como un conjunto de instrucciones que deben ejecutarse de forma atómica, es decir estas instrucciones deben completarse en su totalidad como si se tratase de una sola (todas se ejecutan o ninguna se ejecuta).

Las transacciones se utilizan para representar una unidad lógica de procesamiento de base de datos que debe completarse en su totalidad para garantizar su exactitud. Es necesario emplear transacciones para lidiar con el problema de controlar la concurrencia, que surge cuando varias transacciones enviadas por varios usuarios interfieren entre sí de un modo que produce unos resultados incorrectos. Además una base de datos debe garantizar una correcta recuperación de errores.

¿Por qué cree usted que el control de la concurrencia y la recuperación son necesarios en un sistema de bases de datos?

5. Propiedades de una transacción (propiedades ACID)

Las transacciones deben poseer varias propiedades, a menudo denominadas propiedades ACID, que deben ser implementadas por el control de la concurrencia y los métodos de recuperación del DBMS. Las propiedades ACID son las siguientes:

- Atomicidad. Una transacción es una unidad atómica de procesamiento; o se ejecuta en su totalidad o no se ejecuta en absoluto. Si una transacción falla y no se completa por alguna razón (por ejemplo, por una caída del sistema en medio de su ejecución), la técnica de recuperación debe deshacer sus efectos en la base de datos.
- Conservación de la consistencia. Una transacción está conservando la consistencia si su ejecución completa lleva a la base de datos de un estado consistente a otro.
- Aislamiento. Una transacción debe aparecer como si estuviera ejecutándose de forma aislada a las demás. Es decir, la ejecución de una transacción no debe interferir con la ejecución de ninguna otra transacción simultánea.
- Durabilidad. Los cambios aplicados a la base de datos por una transacción confirmada deben persistir en la base de datos. Estos cambios no deben perderse por culpa de un fallo.