



---

Universidad Nacional Autónoma de México  
Facultad de Ciencias

Fundamentos de Bases de Datos

## Práctica 9

### Bloqueo y Concurrency

Autor: Arroyo Martínez Erick Daniel

---



## 1. Introducción

En sistemas de bases de datos transaccionales, uno de los desafíos principales es permitir que múltiples usuarios accedan y modifiquen los datos de manera simultánea sin comprometer la consistencia e integridad de la información almacenada. Este aspecto, conocido como **concurrency**, se gestiona a través de mecanismos de bloqueo y control de transacciones que aseguran que las operaciones realizadas por distintas transacciones no interfieran de manera indeseada entre sí.

A continuación, se presentan los conceptos fundamentales relacionados con la concurrency y el bloqueo en el contexto de MySQL, los cuales constituyen los antecedentes necesarios para comprender el desarrollo de esta práctica:

1. **Niveles de aislamiento:** En un sistema de bases de datos, el nivel de aislamiento define el grado de visibilidad que las transacciones tienen sobre los cambios de otras transacciones concurrentes. MySQL implementa cuatro niveles de aislamiento principales:
  - **READ UNCOMMITTED:** Permite ver los cambios aún no confirmados por otras transacciones, lo cual puede derivar en lecturas de datos inconsistentes.
  - **READ COMMITTED:** Garantiza que solo se leerán los cambios confirmados, evitando lecturas de datos no comprometidos.
  - **REPEATABLE READ:** Asegura que, dentro de una misma transacción, el resultado de una lectura no cambiará, evitando lecturas inconsistentes durante la transacción.
  - **SERIALIZABLE:** Logra el aislamiento total mediante bloqueos, ejecutando transacciones de forma secuencial, lo cual elimina cualquier interferencia entre transacciones.
2. **Tipos de bloqueo:** MySQL utiliza varios tipos de bloqueos para controlar el acceso a los datos en ambientes concurrentes. Los bloqueos se clasifican principalmente en:
  - **Bloqueo compartido (S):** Permite que varias transacciones lean una fila sin modificarla.
  - **Bloqueo exclusivo (X):** Restringe el acceso a una fila a una sola transacción que tiene intención de modificarla, evitando que otras transacciones puedan leer o escribir en dicha fila.
3. **Escalado de bloqueos:** Para optimizar el rendimiento y minimizar los bloqueos innecesarios, MySQL implementa una estrategia de escalado de bloqueos. Esto permite que el sistema seleccione dinámicamente el nivel de granularidad más apropiado para aplicar bloqueos, ya sea en filas, páginas o tablas enteras, dependiendo de la operación que se realice.
4. **Bloqueos explícitos:** Además de los bloqueos automáticos aplicados por el motor de almacenamiento, MySQL permite que los usuarios establezcan bloqueos explícitos mediante los comandos

LOCK TABLES y UNLOCK TABLES. Estos bloqueos son útiles para operaciones críticas en las que es necesario evitar que otras transacciones accedan a las tablas bloqueadas durante el tiempo que se realicen las modificaciones.

5. **Deadlocks (interbloqueos):** Los deadlocks ocurren cuando dos o más transacciones intentan acceder a recursos que están bloqueados por otras transacciones, generando un ciclo de espera que impide la continuación de las transacciones implicadas. MySQL maneja los deadlocks automáticamente, terminando una de las transacciones en conflicto y permitiendo que la otra complete su operación.

## Objetivos

El propósito de esta práctica es que los estudiantes desarrollen habilidades en el manejo de concurrencia y bloqueo en bases de datos relacionales utilizando MySQL. A través de la resolución de una problemática específica, se espera que los estudiantes apliquen los conceptos de concurrencia aprendidos previamente y comprendan su importancia en la gestión de bases de datos con acceso concurrente.

Los objetivos específicos de la práctica son:

1. Aplicar los niveles de aislamiento de MySQL en un contexto real, asegurando la consistencia y fiabilidad de los datos al manejar transacciones concurrentes.
2. Desarrollar y aplicar estrategias de bloqueo y escalado de bloqueos en situaciones de acceso concurrente, eligiendo de manera adecuada entre bloqueos compartidos y exclusivos para optimizar el rendimiento y la seguridad de la base de datos.
3. Identificar y prevenir condiciones de interbloqueo (deadlock), implementando métodos de resolución de conflictos de acceso que permitan a la base de datos seguir funcionando sin interrupciones.
4. Diseñar y utilizar bloqueos explícitos en MySQL, evaluando su impacto en la eficiencia y seguridad de las operaciones de datos.
5. Desarrollar la capacidad de análisis crítico para identificar los puntos clave donde aplicar bloqueos o sincronización, justificando cada implementación con base en los requerimientos de concurrencia del problema dado.
6. Fortalecer la habilidad para justificar las decisiones de diseño en cuanto al manejo de concurrencia y bloqueos, explicando cómo cada implementación contribuye a la integridad y rendimiento en un entorno simulado de bases de datos.

## Especificaciones de Desarrollo

### Problemática

En la empresa (**esquema de práctica anterior**), cada pedido está relacionado con un único pago, y para garantizar la consistencia, es crucial que:

- El proceso de pago se verifique antes de crear la orden y actualizar el inventario.
- Si el pago es exitoso, se crea la orden, se vincula el pago, se generan los detalles de la orden y se actualiza el inventario.
- Si el pago falla, el sistema debe permitir el procesamiento de otros pagos pendientes sin afectar la consistencia de los datos del inventario.

Para ello, se debe asegurar que, cuando un pago está siendo procesado, ningún otro proceso pueda modificar el inventario hasta que el pago haya sido confirmado como exitoso o fallido.

Para el flujo de pago y creación de pedidos, se propone el siguiente control de concurrencia:

1. **Selección de Productos:** Múltiples usuarios pueden seleccionar productos sin interferencia en el inventario (desde la aplicación).
2. **Proceso de Pago:** Cuando un usuario procede al pago, se debe asegurar la consistencia mediante los siguientes pasos:

- Se bloquean los demás procesos de pago que puedan modificar el inventario, utilizando un bloqueo de nivel de tabla o transacciones con aislamiento en la tabla `products` para evitar que otras transacciones afecten el stock.
- Si el pago falla, se libera el bloqueo, permitiendo que el sistema continúe con el procesamiento de los pedidos pendientes.
- Si el pago es exitoso, se realizan las siguientes acciones:
  - a) Crear el pedido en la tabla `orders`, vinculándolo con el pago verificado.
  - b) Crear los detalles del pedido en la tabla `orderdetails`.
  - c) Actualizar el inventario, reduciendo la columna `quantityInStock` en la tabla `products` para cada producto del pedido.
- Una vez completada la actualización del inventario, se desbloquea la sección crítica, permitiendo el procesamiento de otros pagos.

3. **Control de Concurrencia para Actualización de Información de Cliente:**

- Los estudiantes deben implementar un mecanismo de concurrencia que permita que los clientes o sus representantes de ventas actualicen la información del cliente sin conflicto.
- Sugerencia: Los estudiantes pueden optar por una de las siguientes estrategias:
  - Usar bloqueos optimistas con detección de cambios mediante un campo `version` o `updated_at` en la tabla `customers`, permitiendo así la detección de cambios antes de proceder con la actualización.
  - Utilizar bloqueos explícitos en la tabla `customers` mediante transacciones con el nivel de aislamiento, asegurando que si un usuario o representante está realizando una actualización, el otro proceso esperará hasta que finalice la transacción.

4. Los alumnos deben analizar el estado final del esquema y exponer si existen áreas donde la concurrencia podría generar inconsistencias. Es decir, deben presentar una situación donde se podrían generar errores en el sistema de forma breve.

## Consideraciones

### Libertad de Implementación y Justificación de Elecciones

Dado que los alumnos tienen total libertad en la implementación, se les solicita que presenten una **breve justificación** para cada herramienta o técnica empleada. La justificación debe enfocarse en mostrar cómo cada elección contribuye a resolver los problemas de concurrencia y consistencia planteados, resaltando:

1. La relevancia de la herramienta o técnica en el contexto del problema específico (por ejemplo, cómo un trigger asegura que el inventario se actualice solo cuando el pago se confirma).

2. Cómo la implementación propuesta evita errores de concurrencia, como la pérdida de consistencia de datos o problemas de acceso concurrente no controlado.
3. La razón por la que se considera que cada elección no es trivial y contribuye a una solución robusta.

## Evaluación de la Propuesta

Es importante destacar que **la evaluación de la propuesta no se basará únicamente en la implementación funcional**, sino también en la calidad y relevancia de las justificaciones presentadas. Los siguientes puntos serán considerados en la evaluación:

- Si la solución propuesta realmente aborda y soluciona el problema de concurrencia y consistencia.
- La robustez de la implementación, en términos de cómo maneja situaciones de concurrencia sin comprometer la integridad de los datos.
- La justificación proporcionada para cada elección: soluciones triviales o que no atiendan los problemas planteados no serán evaluadas positivamente.

## Entregables

### Script de SQL para Resolución de la Problemática

Se debe entregar un script en formato `.sql` que incluya la implementación de la solución para ambas problemáticas de concurrencia planteadas:

- **Pagos y Órdenes:** El script debe incluir toda la lógica para el control de concurrencia en el proceso de pagos y creación de órdenes. Esto debe abarcar el bloqueo de los pagos, la creación de la orden solo después de la confirmación del pago, la actualización del inventario y la liberación de los recursos al finalizar la transacción.
- **Actualización de Información del Cliente:** Incluir en el script la implementación para el control de concurrencia en la actualización de datos de cliente, permitiendo que tanto el cliente como el representante de ventas puedan realizar cambios sin generar inconsistencias.

### Documento PDF con Identificación de Áreas de Mejora y Ejemplos de Ejecución

El segundo entregable será un documento en formato PDF que debe incluir los siguientes puntos:

- **Identificación de Áreas de Mejora:** Los estudiantes deben analizar su solución y señalar posibles áreas de mejora en el manejo de concurrencia. Esta sección debe incluir un análisis de cómo se podrían optimizar los bloqueos, reducir conflictos de concurrencia o manejar de forma más eficiente los recursos.
- **Argumentación y Elecciones de Implementación:** Este documento debe incluir una justificación clara de las herramientas y técnicas empleadas (bloqueos, transacciones, triggers, etc.), explicando cómo cada elección contribuye a resolver los problemas de concurrencia de manera robusta.
- **Ejemplos de Ejecución de la Solución:** Dado que simular alta concurrencia puede ser complejo para los alcances de este trabajo, se requiere que los estudiantes incluyan al menos un ejemplo concreto que ilustre cómo se ejecutaría la solución en un escenario de concurrencia. Este ejemplo debe mostrar cómo se protege la consistencia de datos y qué pasos específicos sigue la implementación.

## Rúbrica de Evaluación

Criterio	%	Descripción
Resolución Pagos y Órdenes	35 %	Incluye la correcta implementación del control de concurrencia en el proceso de pagos y creación de órdenes, con verificación de pagos y actualización de inventario.
Resolución de Información de Cliente)	20 %	Implementación efectiva del control de concurrencia en la actualización de información de cliente, considerando acceso concurrente desde el cliente y representante de ventas.
Justificación de la Implementación	35 %	Análisis y justificación de cada herramienta y técnica empleada, explicando cómo cada elección contribuye a resolver problemas de concurrencia y asegurar la consistencia de los datos.
Documentación	5 %	Claridad y organización del script y el documento PDF, incluyendo comentarios y explicaciones detalladas en el código SQL.
Identificación de Áreas de Mejora	5 %	Propuestas de optimización y análisis crítico de posibles mejoras en la solución planteada.

Cuadro 1: Rúbrica de Evaluación

## Material relacionado

- UNLOCK TABLES Statement
- MySQL isolation levels and how they work
- Optimistic Locking: Concurrency Control with a Version Column

## Recursos

- Live SQL
- SQL Language Reference
- SQL Tutorial
- mockaroo
- MySQL Documentation