

# Preventing Data Inconsistencies in MySQL: Strategies for Avoiding Lost Updates



yoonsin91 · Follow

Published in Level Up Coding

3 min read · Mar 30

[Listen](#)[Share](#)

A few weeks ago, I encountered a situation where our inventory tracking system reported inconsistencies. After researching the issue, I discovered that concurrency problems in our MySQL database may have been the root cause. Further investigation revealed a “lost update” concurrency issue. To prevent similar issues from arising in the future, I have been exploring various strategies for optimizing our database. Before finding a solution, it is necessary to understand what concurrency is and why transaction isolation is important.

## Concurrency Issues in MySQL

Concurrency situations in MySQL can occur when multiple transactions attempt to access or modify the same data simultaneously, which can potentially result in data inconsistencies or errors. These situations include lost updates, dirty reads, and

[Open in app](#)[Sign up](#)[Sign In](#)

transaction to be lost. To prevent lost updates, a transaction isolation level of at least REPEATABLE READ is required, which uses shared locks to prevent other transactions from modifying the data.

- Dirty reads occur when a transaction reads uncommitted changes made by another transaction, which may later be rolled back. A transaction isolation level of READ COMMITTED or higher can prevent dirty reads.

- Phantom reads occur when a transaction sees a set of rows that satisfies a certain condition, but another transaction inserts or deletes rows that also satisfy the same condition, causing the first transaction to see a different set of rows. To prevent phantom reads, a transaction isolation level of at least REPEATABLE READ is required, which uses gap locks to prevent other transactions from inserting or deleting rows that would affect the first transaction's results.

## Transaction Isolation Levels in MySQL

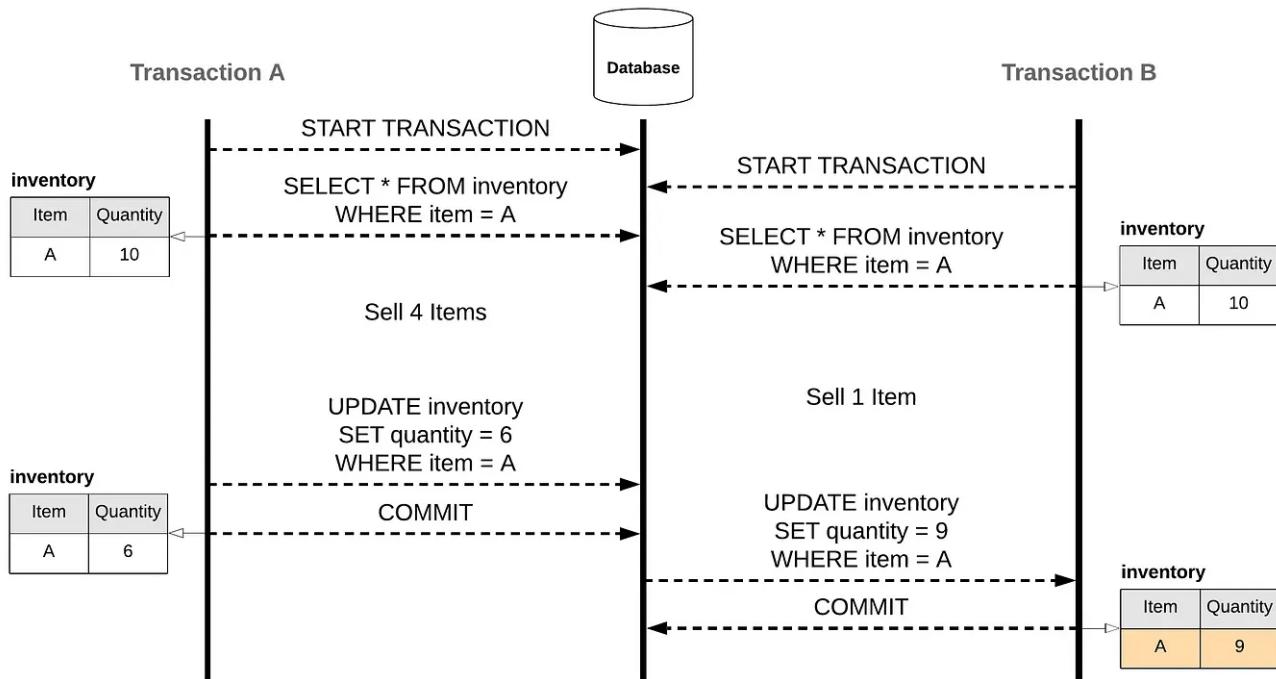
Isolation is critical in databases as it ensures transactional consistency and prevents data corruption or inconsistencies caused by concurrent updates. It enables transactions to occur in isolation, allowing them to commit or roll back independently of other transactions, ensuring that the data remains consistent and accurate.

MySQL provides four transaction isolation levels that determine how transactions interact with one another and how changes are visible to other transactions.

1. **READ UNCOMMITTED:** Allows dirty reads, meaning a transaction can read uncommitted changes made by another transaction.
2. **READ COMMITTED:** Prevents dirty reads by only allowing a transaction to read changes that have been committed by other transactions.
3. **REPEATABLE READ:** Ensures that a transaction can read the same set of rows consistently, even if other transactions are updating the same data simultaneously. This is achieved by placing shared locks on all the rows accessed by the transaction.
4. **SERIALIZABLE:** Provides the highest level of isolation by ensuring that transactions are executed in a serialized manner, preventing any concurrent modifications to the data. This is achieved by placing range locks on all the rows that the transaction accesses.

## How do lost updates happen?

In my case, my database is using repeatable read isolation. Here is a situation where a lost update can occur:



Credit to: <https://medium.com/@chester.yw.chu/對於-mysql-repeatable-read-isolation-常見的三個誤解-7a9afbac65af>

A lost update in inventory occurs when two transactions try to update the same item simultaneously, but only one succeeds, causing the other transaction's changes to be lost, leading to inventory inconsistencies.

### How to avoid lost update?

- Use Atomic operations

```
UPDATE inventory SET quantity = quantity - 4 WHERE item = A;
```

- use lock

```
SELECT * FROM inventory FOR UPDATE;
UPDATE inventory SET quantity = 6 WHERE item = A; COMMIT;
```

## Summary

Ensuring transactional consistency is crucial for preventing data corruption or inconsistencies caused by concurrent updates. Concurrency issues in MySQL can

occur when multiple transactions access or modify the same data simultaneously. To prevent lost updates, dirty reads, and phantom reads, MySQL provides several transaction isolation levels. In cases where lost updates may occur, atomic operations or locks can be used to prevent inventory inconsistencies. By understanding these concepts and implementing appropriate strategies, databases can be optimized to prevent concurrency issues and ensure transactional consistency.

**Credit:****對於 MySQL Repeatable Read Isolation 常見的三個誤解**

Phantom, Lost Update and Range Lock in MySQL Repeatable Read Isolation Level

medium.com

**複習資料庫的Isolation Level 與常見的五個Race Conditions 圖解**

Dirty Read, Non-repeatable Read, Lost Update, Phantom, Write Skew

medium.com

MySQL

Database

Query Optimization

Data Consistency



Follow

**Written by yoonsin91**

26 Followers · Writer for Level Up Coding

## More from yoonsin91 and Level Up Coding



 yoonsin91

### The Repositories Pattern in Lumen

The repositories pattern is widely used pattern whose main point is that the application does not have to know data is implemented.

3 min read · Oct 29, 2017

 155

 3



```

commit ffcf2c01b7ef612893529cef188cc1961ed64521 (HEAD -> master, origin/master, origin/bors/staging, origin/HEAD)
Merge: fc991bf81 5159211da
Author: iohk-bors[bot] <43231472+iohk-bors[bot]@users.noreply.github.com>
Date: Tue Nov 8 17:44:34 2022 +0000

Merge #4563

4563: New p2p topology file format r=coot a=coot

Fixes #4559.

Co-authored-by: Marcin Szamotulski <coot@coot.me>
Co-authored-by: olgahryniuk <67585499+olgahryniuk@users.noreply.github.com>

commit fc991bf814891a9349f22cf278632d39b04d4628
Merge: 5633d1c05 5cd94d372
Author: iohk-bors[bot] <43231472+iohk-bors[bot]@users.noreply.github.com>
Date: Tue Nov 8 13:07:58 2022 +0000

Merge #4613

4613: Update building-the-node-using-nix.md r=CarlosLopezDeLara a=CarlosLopezDeLara

Build the cardano-node executable. No default configuration.

Co-authored-by: CarlosLopezDeLara <carlos.lopezdelara@iohk.io>

commit 5159211da7a644686a973e4fb316b64ebb1aa34c
Author: olgahryniuk <67585499+olgahryniuk@users.noreply.github.com>
Date: Tue Nov 8 13:25:10 2022 +0200

```

 Jacob Bennett in Level Up Coding

## Use Git like a senior engineer

Git is a powerful tool that feels great to use when you know how to use it.

◆ · 4 min read · Nov 15, 2022

 6.8K  67 



 Arslan Mirza in Level Up Coding

## The Dark Side of Software Engineering

## Why I Regret My Career Choice

◆ · 10 min read · Jun 13

👏 691

💬 14



# Lumen + AWS Elastic Beanstalk

yoonsin91

## How to setup Lumen + AWS Elastic Beanstalk [Basic]

it's just simple guide as how to setup Lumen with Elastic Beanstalk.

4 min read · Sep 4, 2017

👏 43

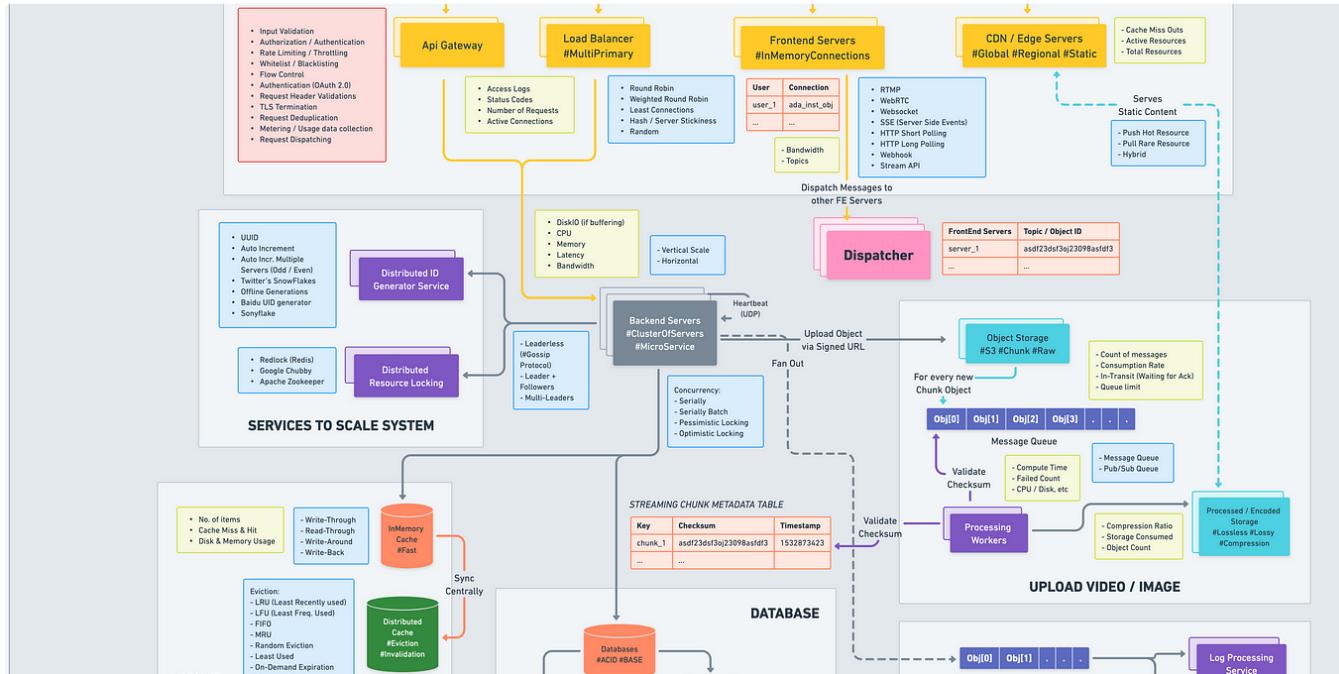
💬 1



See all from yoonsin91

See all from Level Up Coding

## Recommended from Medium



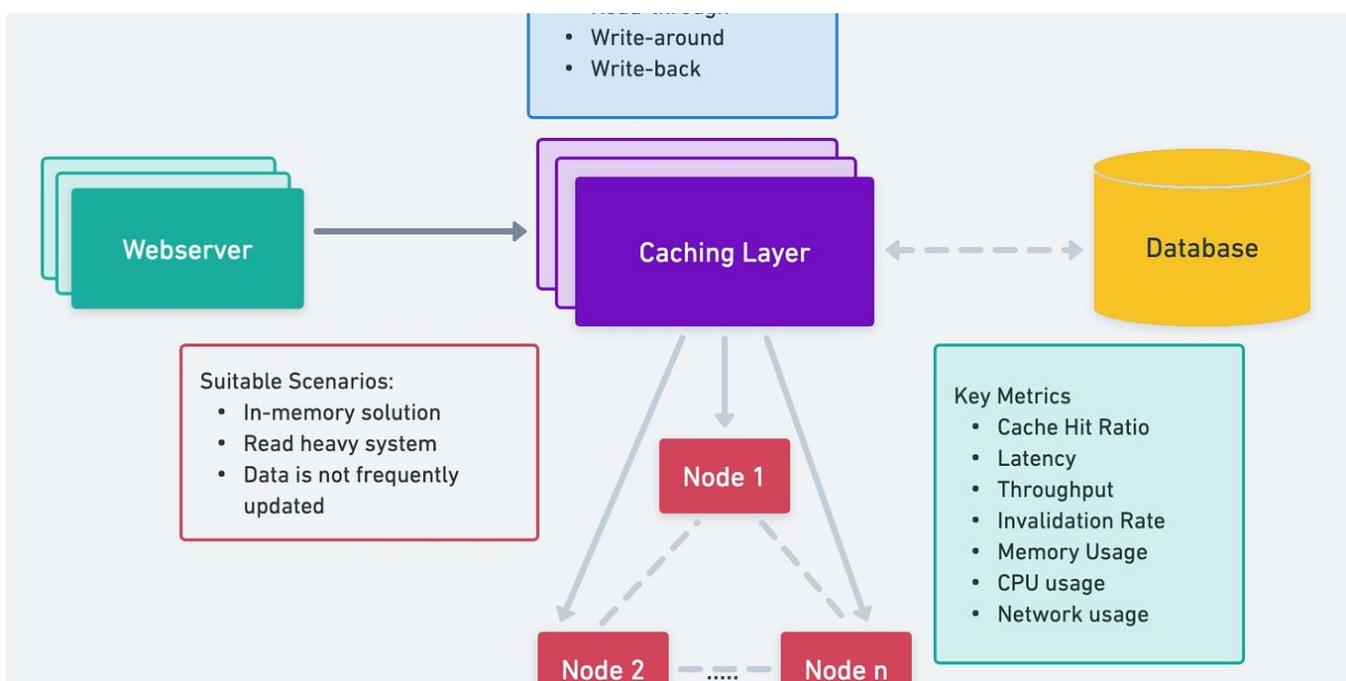
Love Sharma in Dev Genius

## System Design Blueprint: The Ultimate Guide

Developing a robust, scalable, and efficient system can be daunting. However, understanding the key concepts and components can make the...

◆ · 9 min read · Apr 20

4.7K 32



Love Sharma in Dev Genius

## A Comprehensive Guide to Distributed Caching

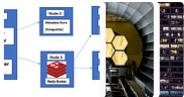
An essential website requires a web server to receive requests and a database to write or read data. However, this simple setup will only...

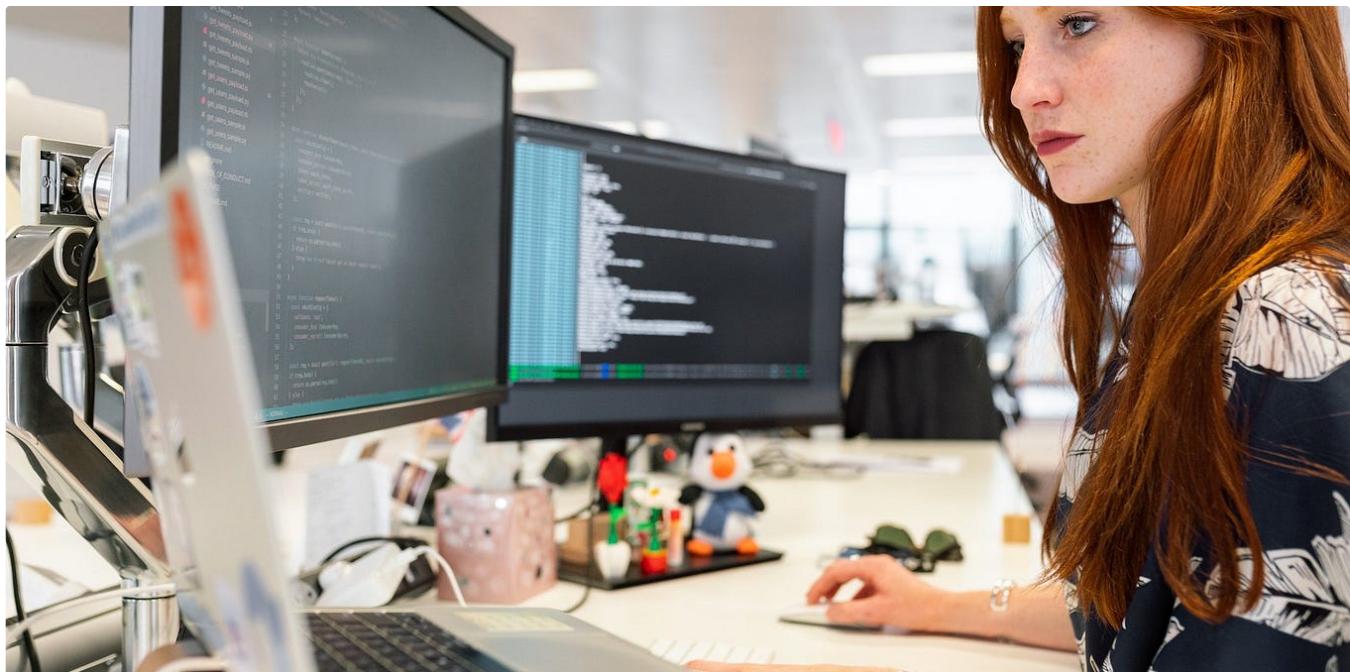
◆ · 13 min read · Feb 9

👏 503 💬 4



## Lists

 **New\_Reading\_List**  
173 stories · 7 saves



The Coding Diaries in The Coding Diaries

## Why Experienced Programmers Fail Coding Interviews

A friend of mine recently joined a FAANG company as an engineering manager, and found themselves in the position of recruiting for...

◆ · 5 min read · Nov 2, 2022

👏 4.5K 💬 101



 JP Brown

## What Really Happens to a Human Body at Titanic Depths

A Millisecond-by-Millisecond Explanation

◆ · 4 min read · Jun 23

 27K 323 Raphael De Lio

## Understanding Transactions in Redis (How to

This content is also available as a YouTube video. You can watch it here.

◆ · 6 min read · Jan 16

👏 6

+



 Park Sehun

## Why you should migrate from NoSQL to SQL database

While many talk about migrating SQL DB to NoSQL, today, I would talk about the opposite scenario.

◆ · 3 min read · 6 days ago

👏 1

+

See more recommendations