■ Apache Iceberg

Explained Like You're 5

Time Travel & Partitioning — Made Simple

■ Part 1: Time Travel — The Toy Box Story

Forget computers for a second. Imagine you have a **toy box** with your favourite toys all arranged in a certain way. One day you decide to rearrange everything differently.

But your mum is sneaky — she **takes a photo** of your toy box *before* you rearrange it. Then you rearrange everything. Mum takes **another photo**.

■ Photo 1	■ Photo 2
Toys the **old** way	Toys the **new** way

The toys in the box are now arranged the new way. But if you look at Photo 1, you can remember exactly how everything was before — and put it all back if you wanted.

> ■ ***That's Iceberg Time Travel.*** *Every time you change your data, Iceberg secretly takes a photo (called a* ***snapshot****). You can always look at an old photo and go back to exactly how things were.*

In Trino you can do this with a simple command:

```
-- Go back to a specific point in time SELECT * FROM my_table FOR TIMESTAMP AS OF
TIMESTAMP '2024-01-15 10:00:00'; -- Or fully roll the table back CALL
my_catalog.system.rollback_to_snapshot('my_schema', 'my_table', 1234567890);
```

■ Part 2: Repartitioning — The Bookshelf Story

Now imagine you have a **big bookshelf** with 1,000 books thrown in randomly with no order. Every time someone asks *"find me all the scary books"*, you have to check **every single book** one by one. That takes forever.

So one day you reorganise the whole bookshelf:

BEFORE	DURING	AFTER
`random random random... (1,000 books, no order)`	Old shelf still has everything. New shelf being built next to it. ■ **2 shelves temporarily!**	New shelf: scary | funny | sad Old shelf thrown away ■

Now when someone asks for scary books you go **directly to the scary section** instead of checking all 1,000 books. That's the entire point!

> ■ ***That's Partitioning.*** *You're just reorganising where data lives in your storage bucket so Trino can find it instantly — without scanning every single file. The double storage is a* ***temporary*** *safety net. Once you're happy, you run cleanup and it goes back to normal.*

■ Part 3: The Storage Lifecycle

Step	What happens	Storage
1. Before	Only old partition files exist	100 GB
2. During	Old + new files both exist (safety net!)	~200 GB ■■
3a. Apply + Cleanup	Old files deleted, new files stay	~100 GB ■
3b. Rollback	New files deleted, old files stay	~100 GB ■

Once you're confident, run these two cleanup commands:

```
-- Step 1: remove old snapshots from metadata CALL
my_catalog.system.expire_snapshots('my_schema', 'my_table'); -- Step 2: physically
delete files no longer referenced CALL
my_catalog.system.remove_orphan_files('my_schema', 'my_table');
```

■ Key Takeaways

■ Time Travel	Iceberg takes a "photo" (snapshot) of your data on every change. You can go back to any photo at any time.
■ Partitioning	Reorganises files in your bucket so queries only read the files they actually need — like a sorted bookshelf.
■ Temporary Doubling	Storage doubles during repartition because Iceberg never overwrites old files. It's a safety net, not a bug.
■ Cleanup	Run expire_snapshots + remove_orphan_files when you're happy. Storage goes back to normal.
■ Rollback	Changed your mind? Just rollback to the old snapshot. No data was ever destroyed.