Managing Data Files in Apache Iceberg

Russell Spitzer, Apple

This is not a contribution

Distributed Systems for Life

- Working in OSS Distributed Systems for 10+ Years
- Contributed to Datastax Spark-Cassandra Connector
- Committed to Apache Spark, Apache Cassandra, Apache Iceberg...
- Currently a Iceberg PMC member

How do we get good OLAP Performance?

Reading files is slow
Opening files is slow
The less files we touch the better

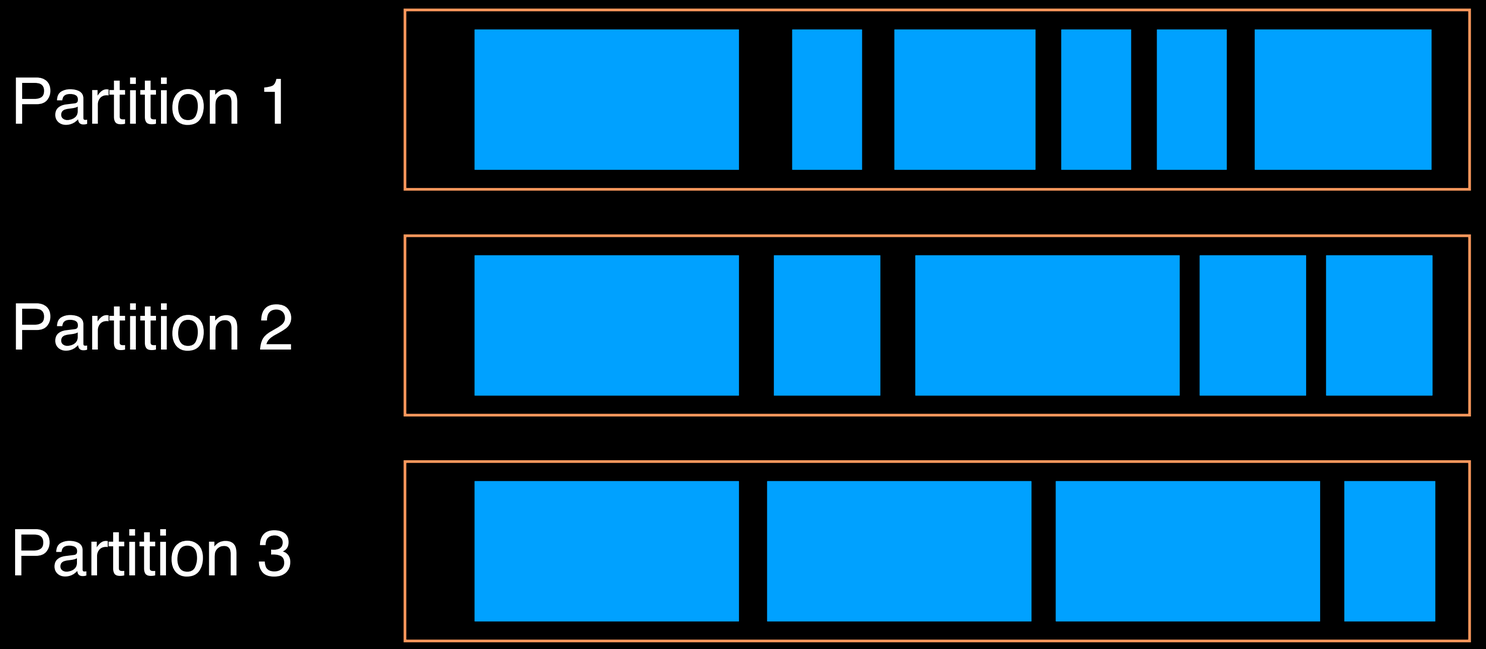
Apache Iceberg Provides RewriteDataFiles

Don't Keep Data Files Indefinitely

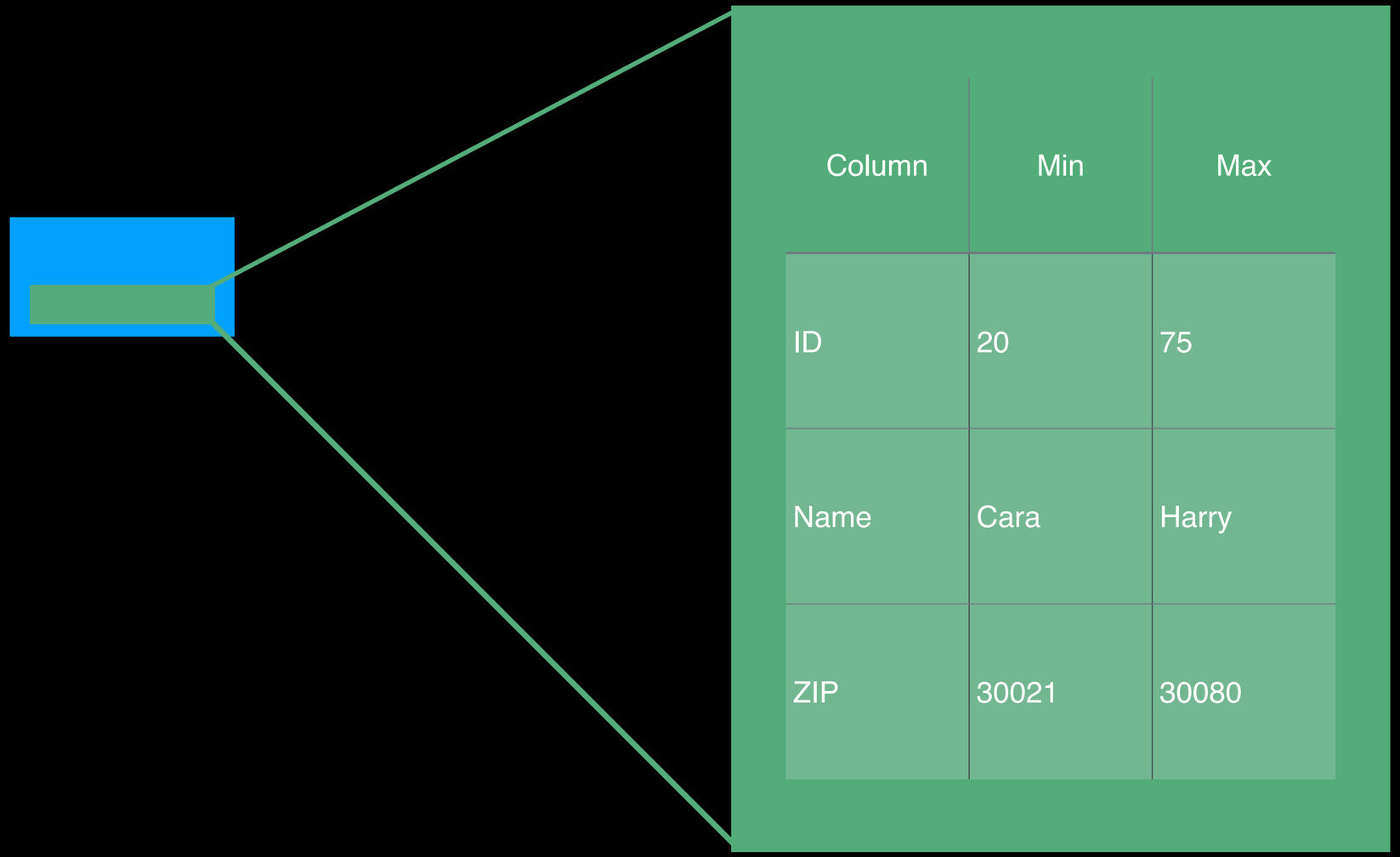
Apache Iceberg Provides ExpireSnapshots

Better Querying through File Skipping

Partitioning Allows Skipping Sets of Files

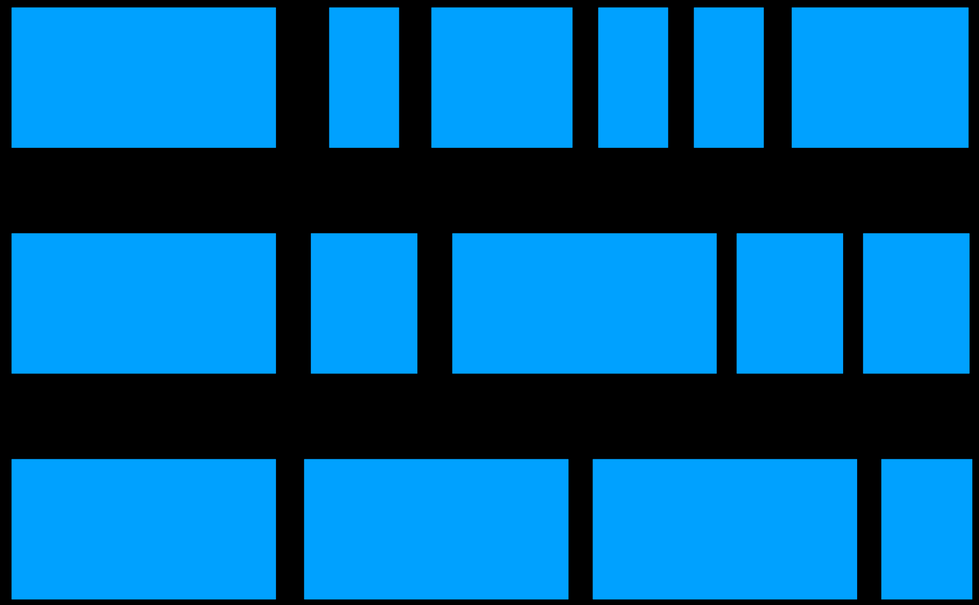


File Metrics Allows Skipping Individual Files



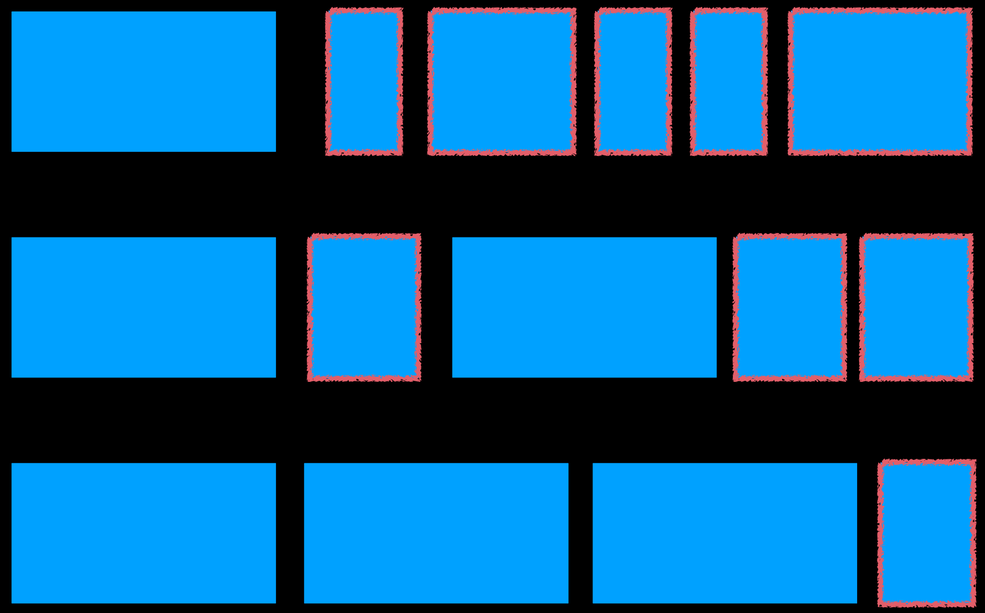
Column	Min	Max
ID	20	75
Name	Cara	Harry
ZIP	30021	30080

Possible Improvements?



Possible Improvements?

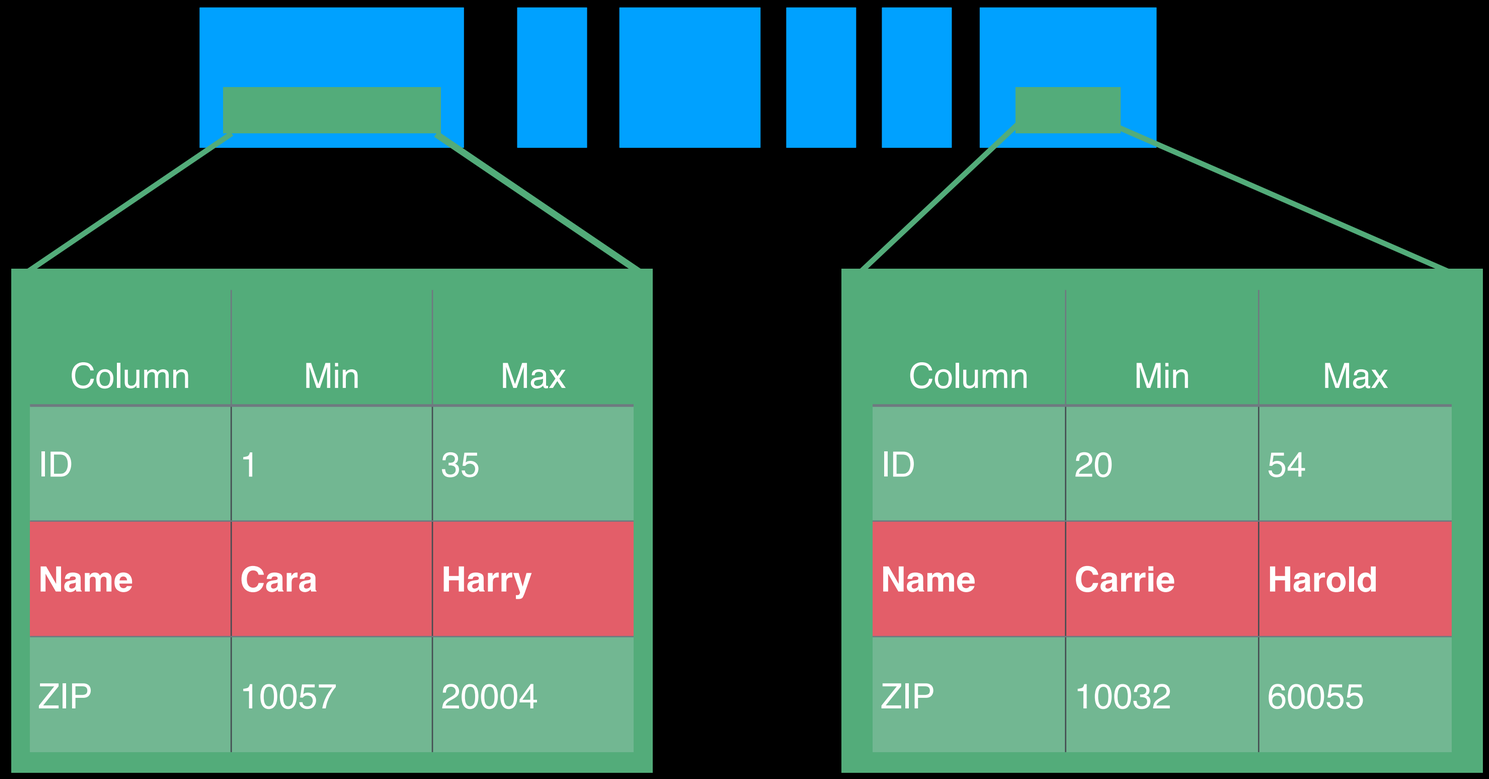
Small Files



- Reading optimized for large files
- Opening file cost
- More metrics to check

Possible Improvements?

Overlapping Metrics

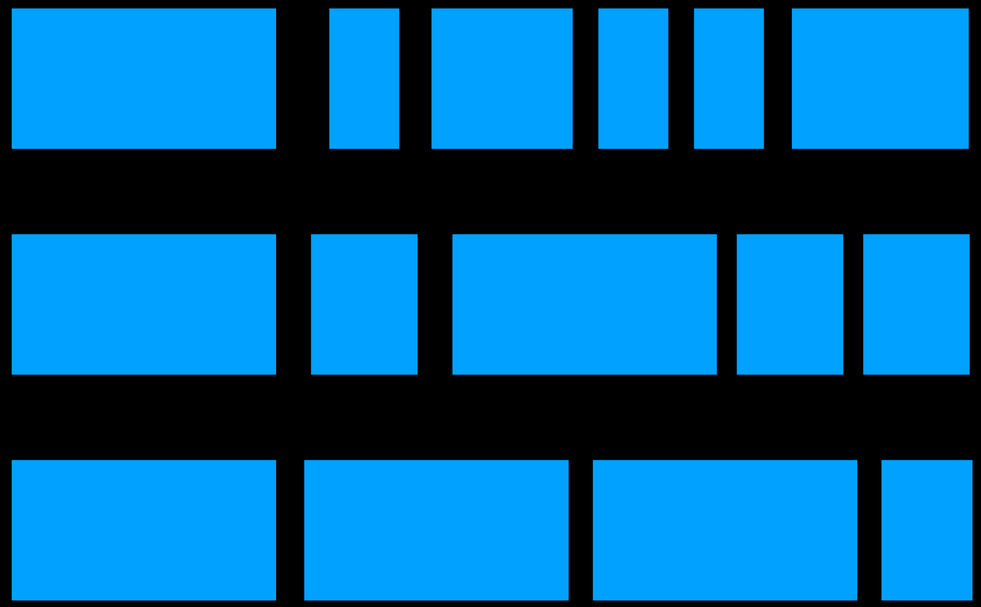


Column	Min	Max
ID	1	35
Name	**Cara**	**Harry**
ZIP	10057	20004

Column	Min	Max
ID	20	54
Name	**Carrie**	**Harold**
ZIP	10032	60055

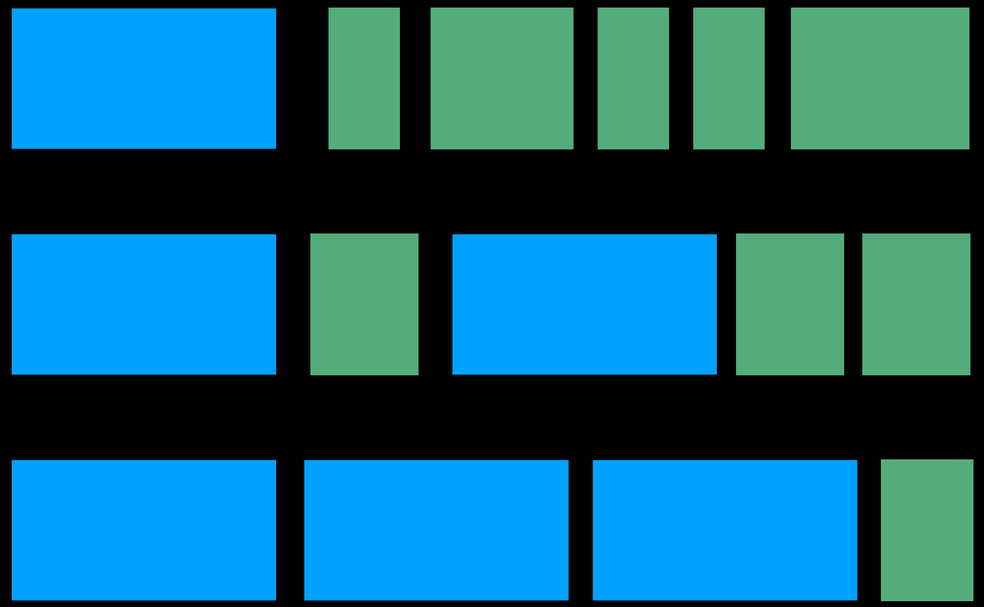
Rewrite Data Files for Optimization

Rewrite Data Files uses a *Strategy* which:



Rewrite Data Files for Optimization

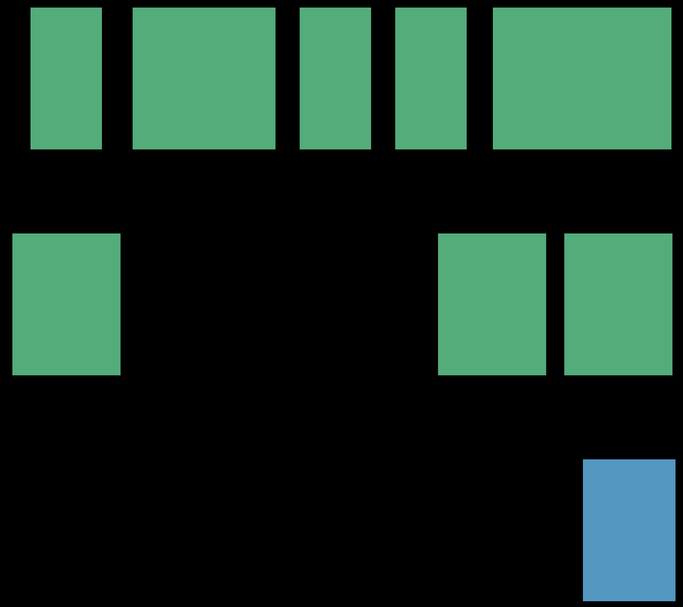
Rewrite Data Files uses a *Strategy* which:



1. Identify files for optimization

Rewrite Data Files for Optimization

Rewrite Data Files uses a *Strategy* which:



1. Identify files for optimization
2. Determine if a partition should be optimized

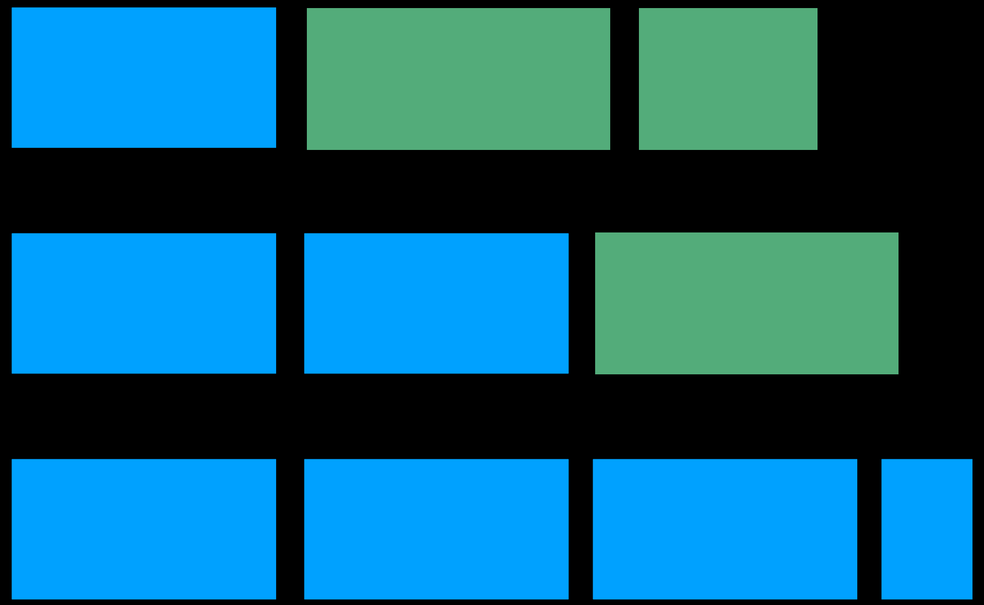
Rewrite Data Files for Optimization

Rewrite Data Files uses a *Strategy* which:

1. Identify files for optimization
2. Determine if a partition should be optimized
3. Rewrite the files within that partition

Rewrite Data Files for Optimization

Rewrite Data Files uses a *Strategy* which:



1. Identify files for optimization
2. Determine if a partition should be optimized
3. Rewrite the files within that partition

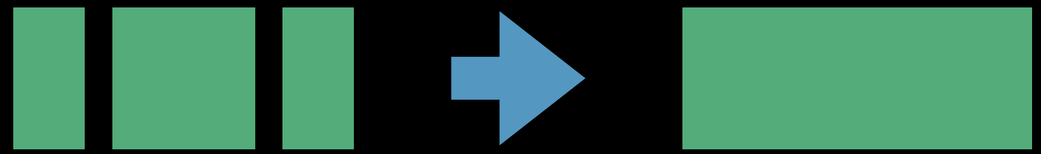
Rewrite Data Files for Optimization

Strategies

Rewrite Data Files for Optimization

Strategies

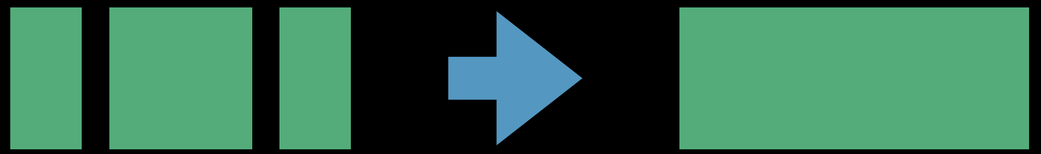
1. Bin-Pack
   Size-based optimization
   Selects files based on size
   Reads files and rewrites



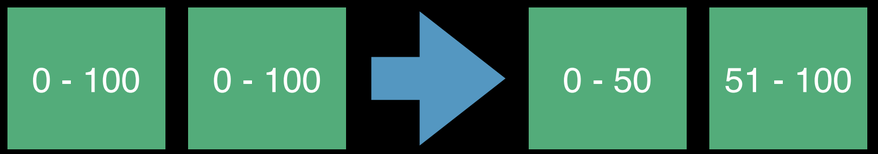
Rewrite Data Files for Optimization

Strategies

1. Bin-Pack
   Size-based optimization
   Selects files based on size
   Reads files and rewrites



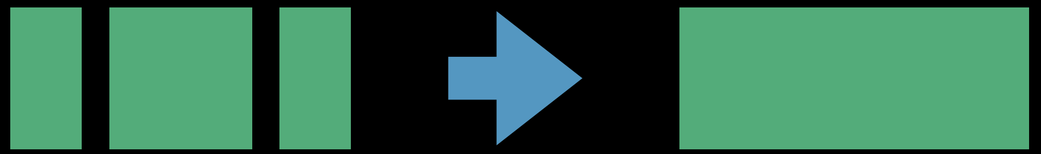
2. Sort
   Metric based optimization
   Selects files based on size / Unsortedness*
   Reads files, sorts based on a column, and rewrites



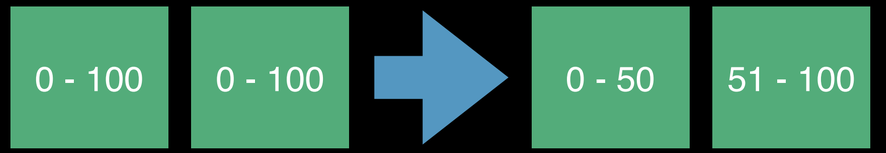
Rewrite Data Files for Optimization

Strategies

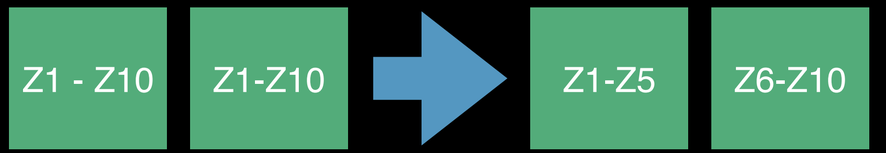
1. Bin-Pack
   Size-based optimization
   Selects files based on size
   Reads files and rewrites



2. Sort
   Metric based optimization
   Selects files based on size / Unsortedness*
   Reads files, sorts based on a column, and rewrites

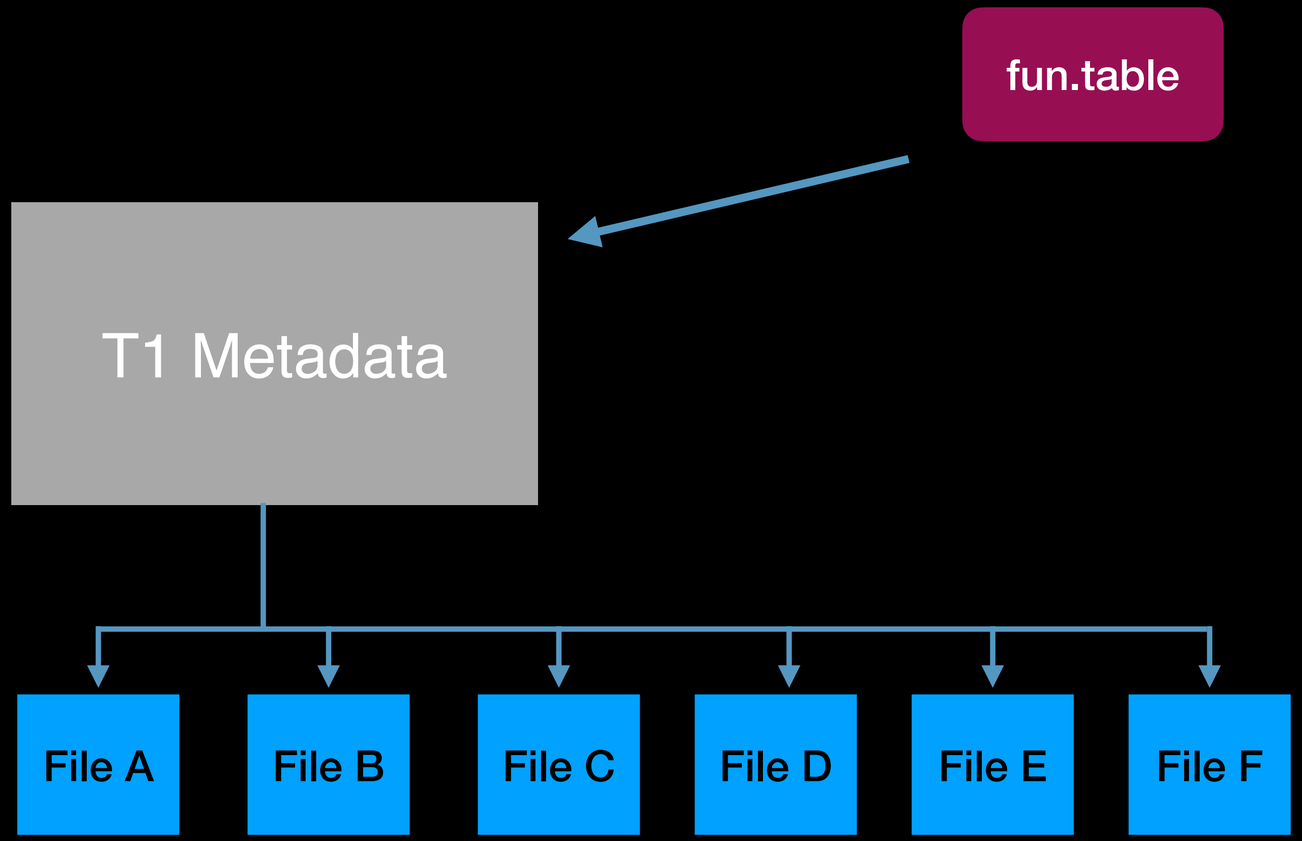


3. ZOrder*
   Multi-dimensional metric optimization
   Selects files based on size
   Reads files, sorts on ZOrder function, and rewrites

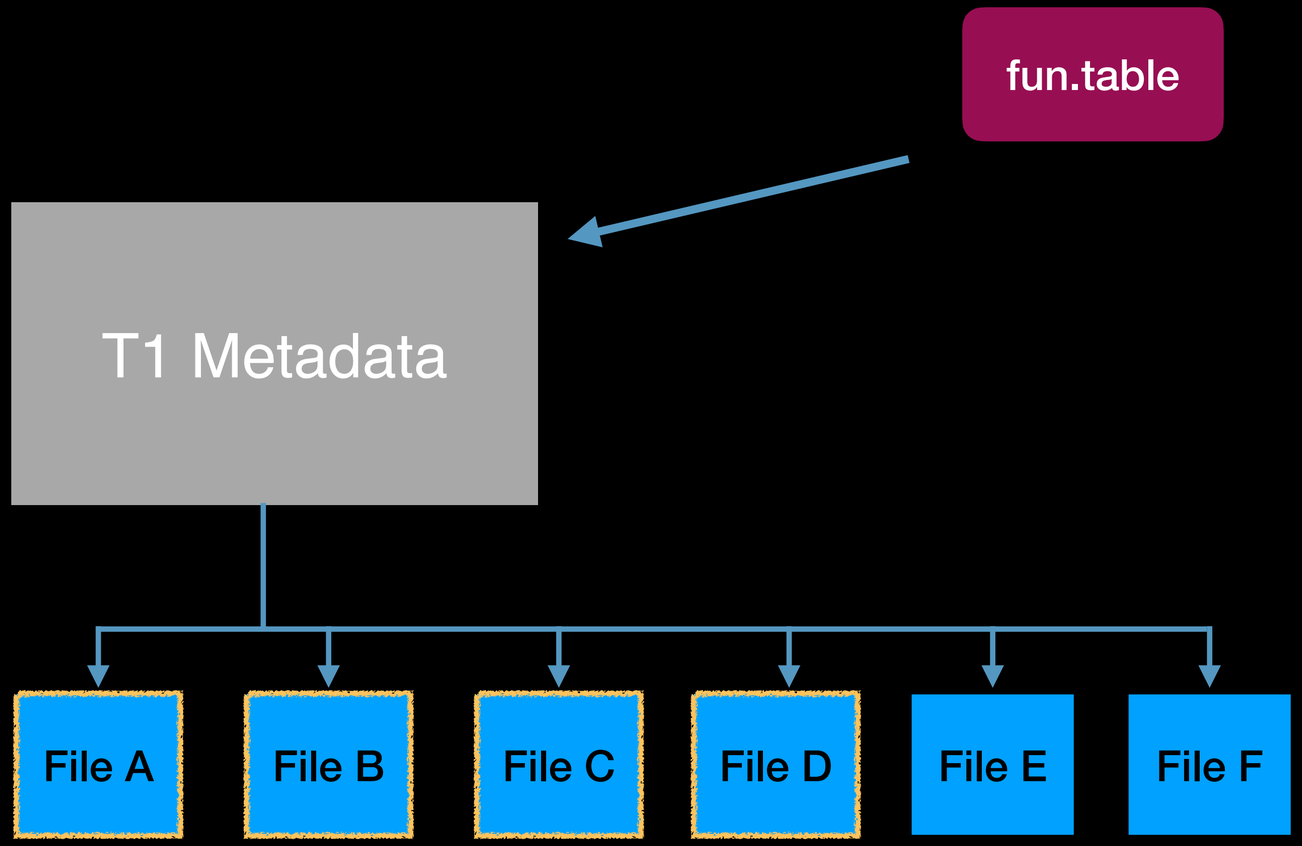


Iceberg Maintains Table History

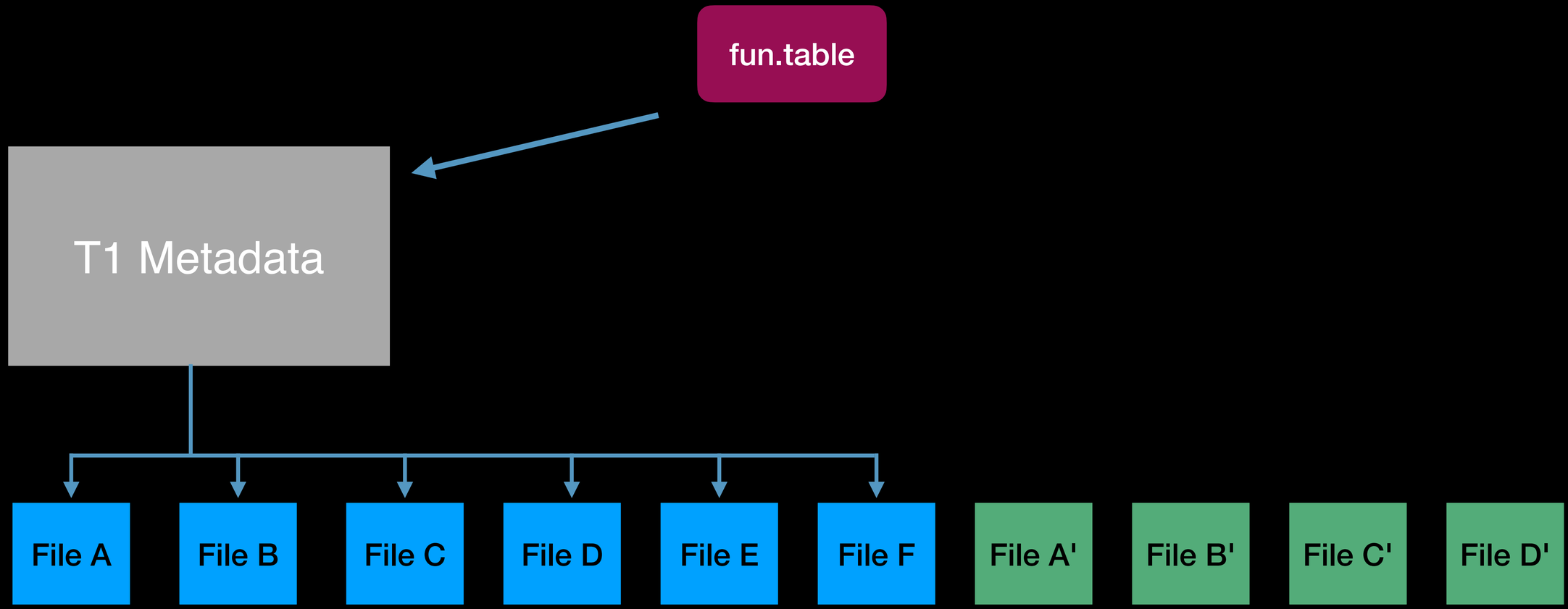
Rewrite Creates a New Snapshot



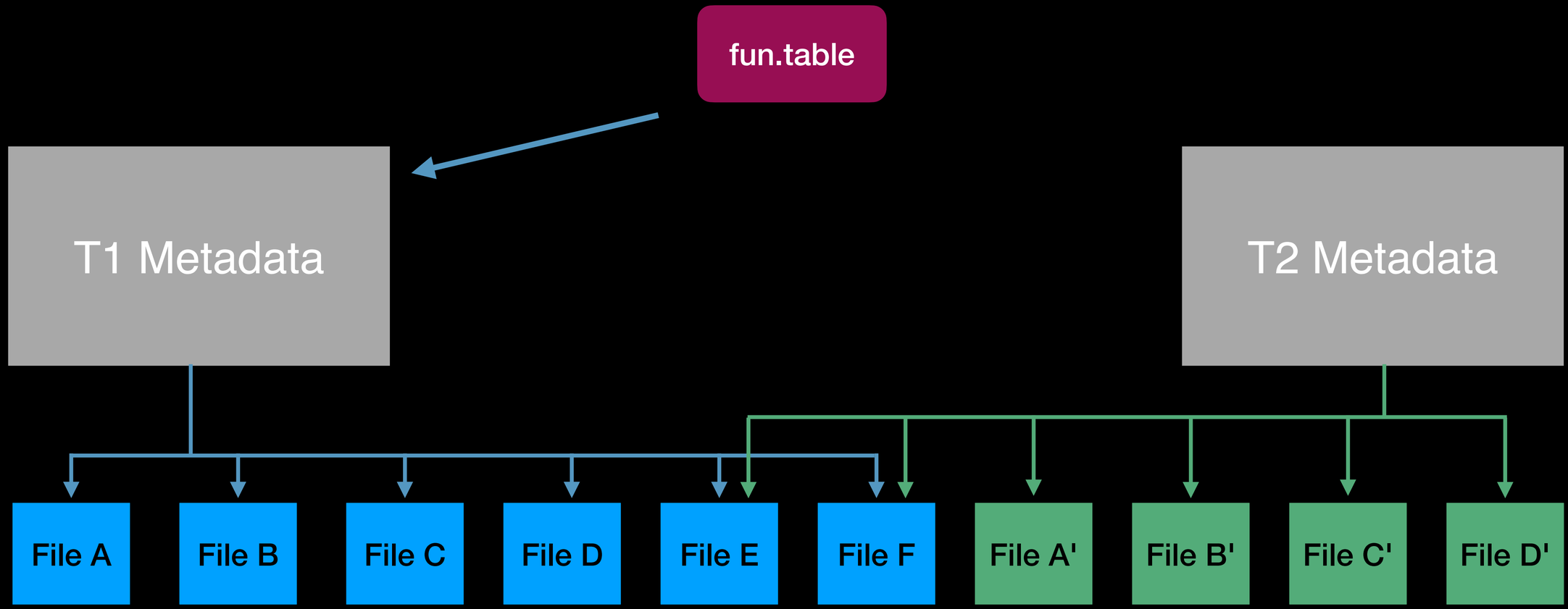
Rewrite Creates a New Snapshot



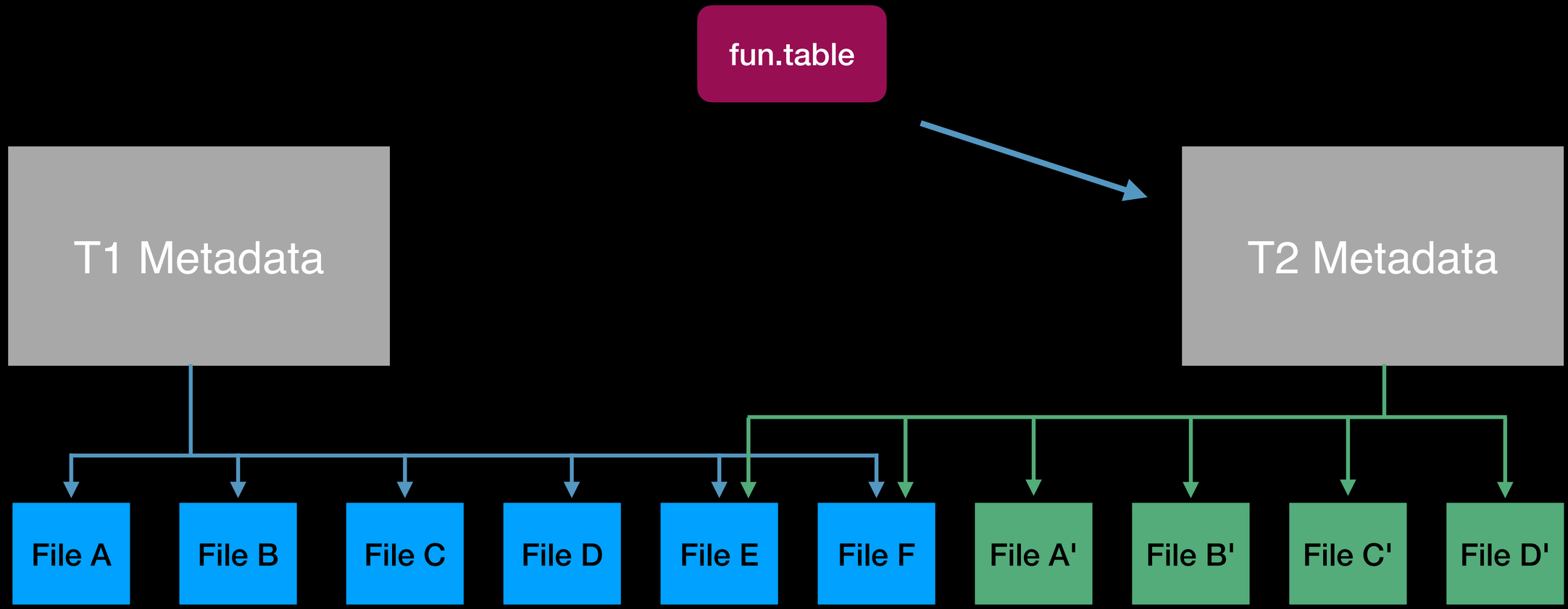
Rewrite Creates a New Snapshot



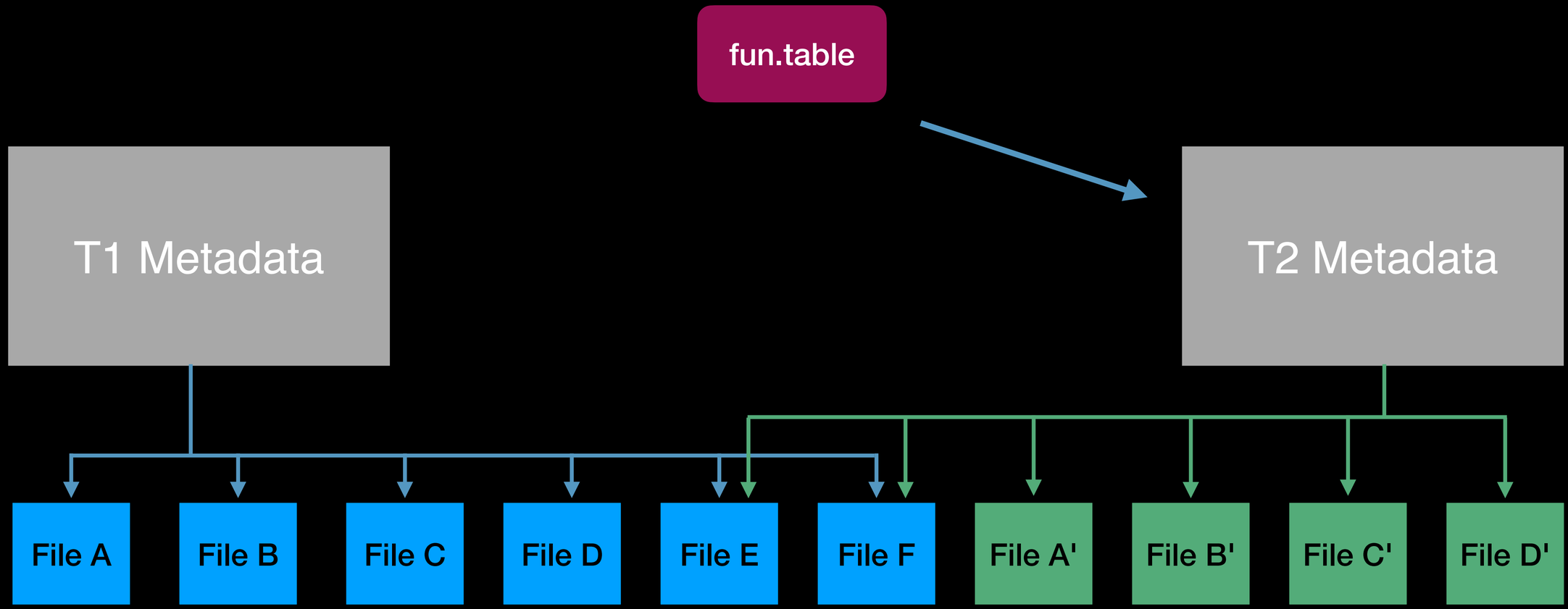
Rewrite Creates a New Snapshot



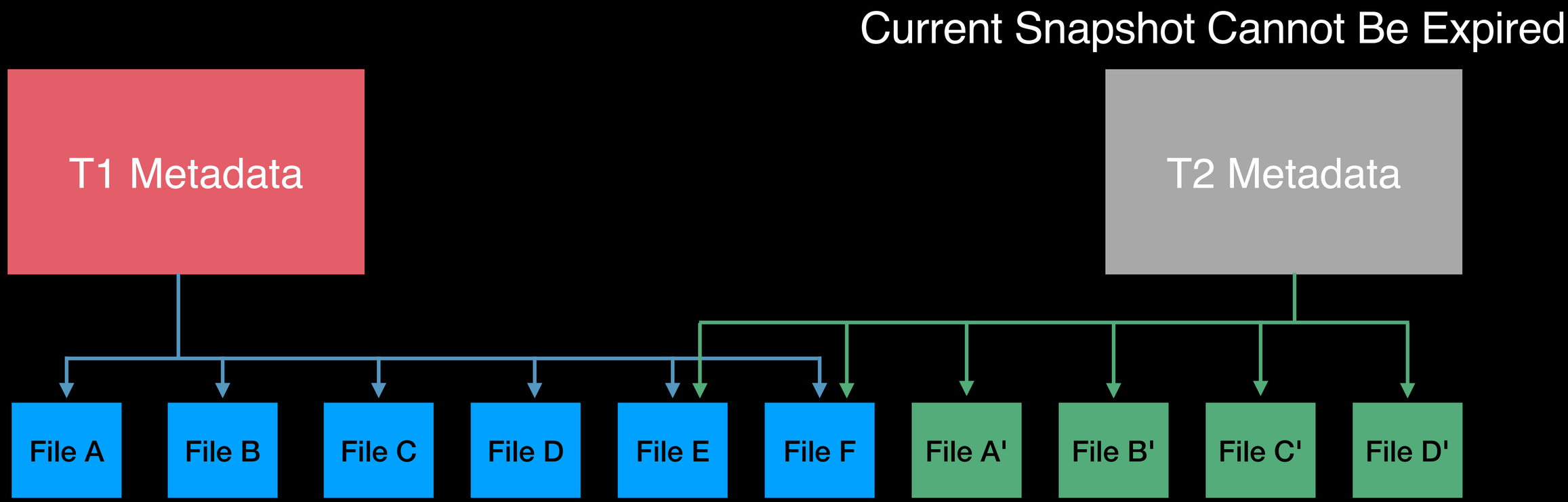
Rewrite Creates a New Snapshot



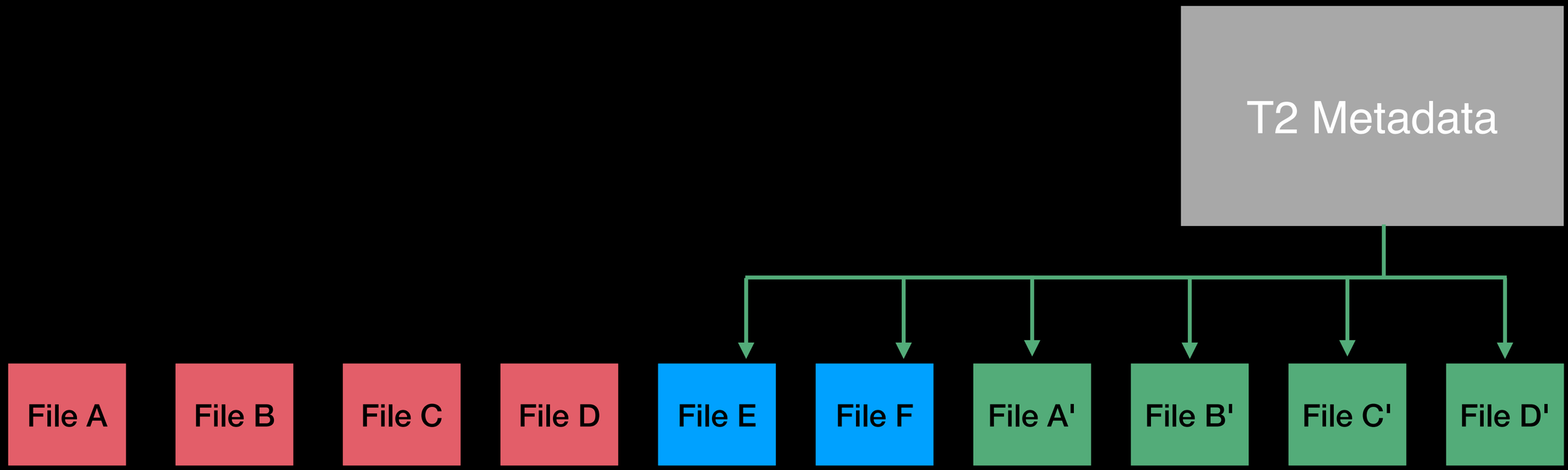
Expire Snapshots Safely Removes Files



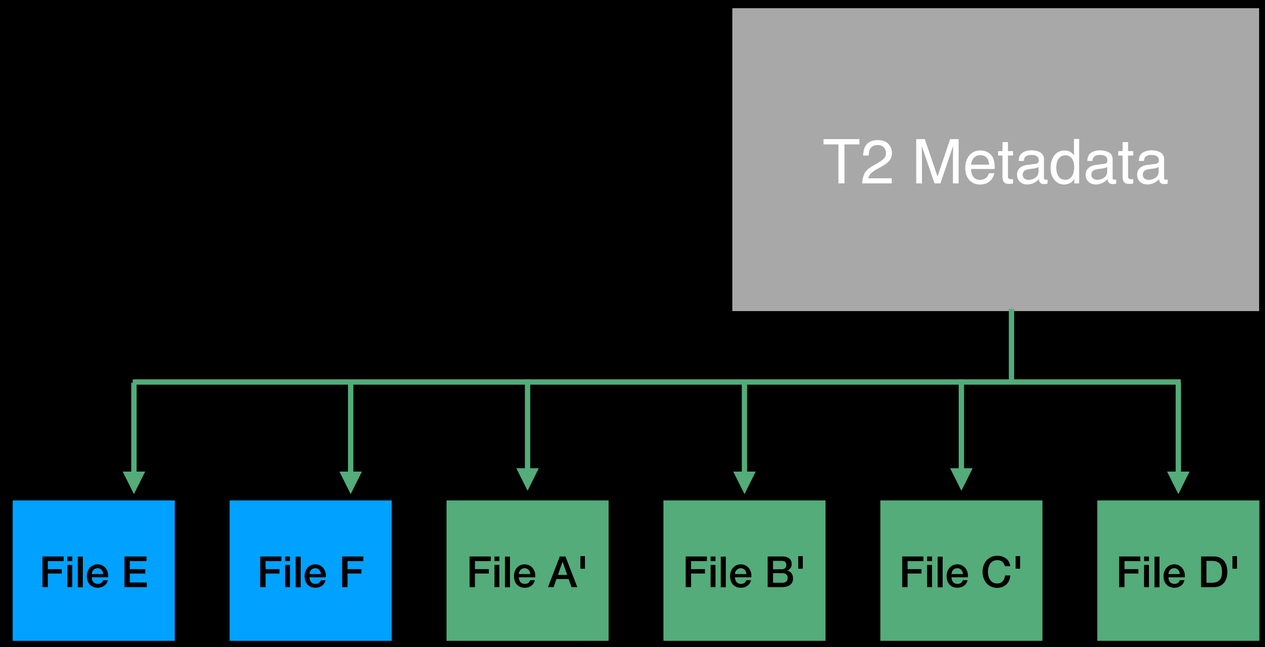
Mark Snapshots for Removal



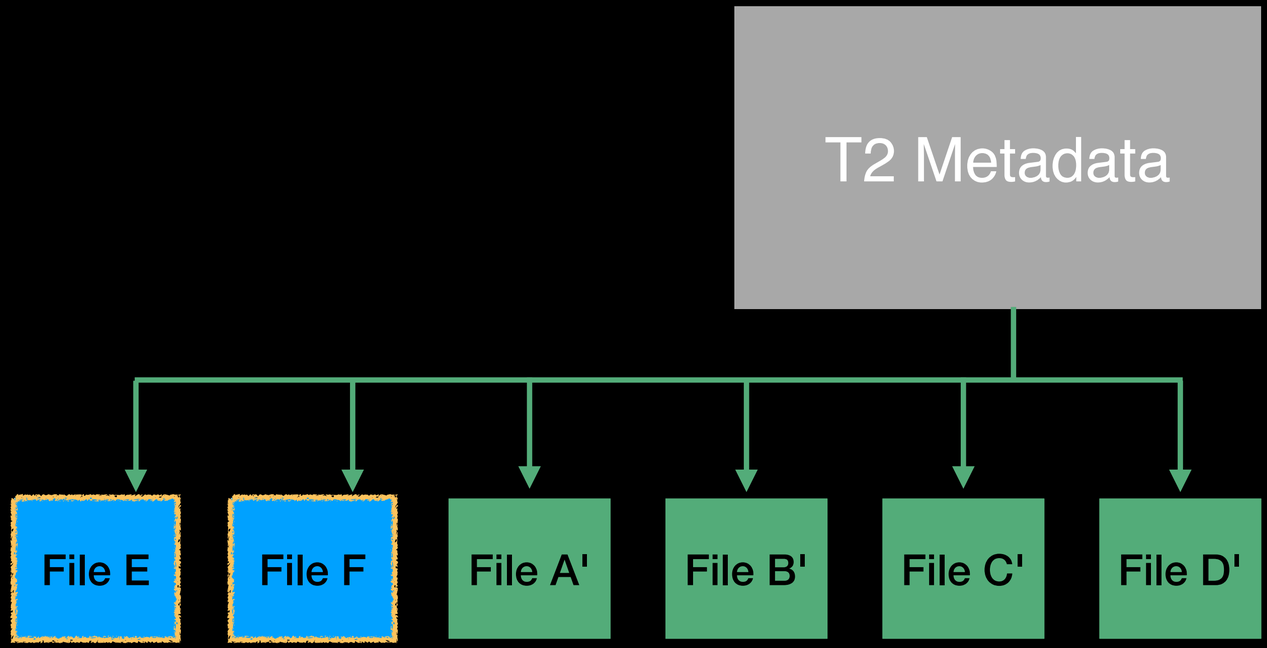
Remove Unreachable Files



Files Are Physically Removed



Old Files are Retained if Needed



Reclaiming Space After Large Rewrites

1. Rewrite data files
   Make a lot of new files
2. Expire snapshot older than now
   Removes history and all no longer needed files

Lots of Work Left - Join us!

Apache Iceberg is an Open Source Project

All are welcome!

https://iceberg.apache.org/

apache-iceberg.slack.com - Invite link on main page