Partitioning Tables in YugabyteDB

Denis Magda

Friday, June/24/2022

YFTT
YUGABYTEDB
FRIDAY
TECH TALKS

yugabyteDB

Agenda

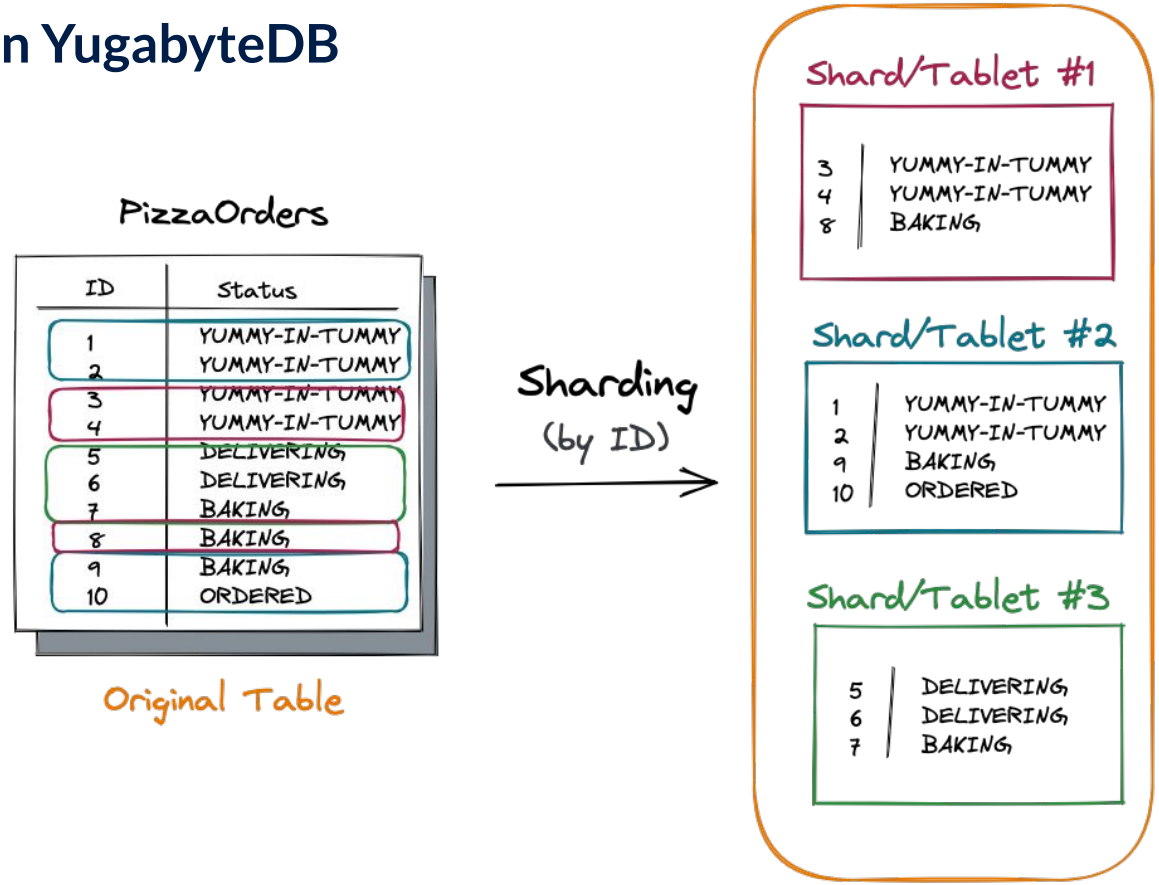
- Partitioning vs. Sharding
- Types of Table Partitioning in YugabyteDB
- Partition Pruning and Maintenance

Partitioning vs. Sharding

Sharding

- A mechanism to distribute load across servers in a distributed system.
- Splits tables into smaller parts (shards), and this is **automatic**.
- Shards are internal to the system; end-users do not directly do any "shard" level operations.

Sharding in YugabyteDB

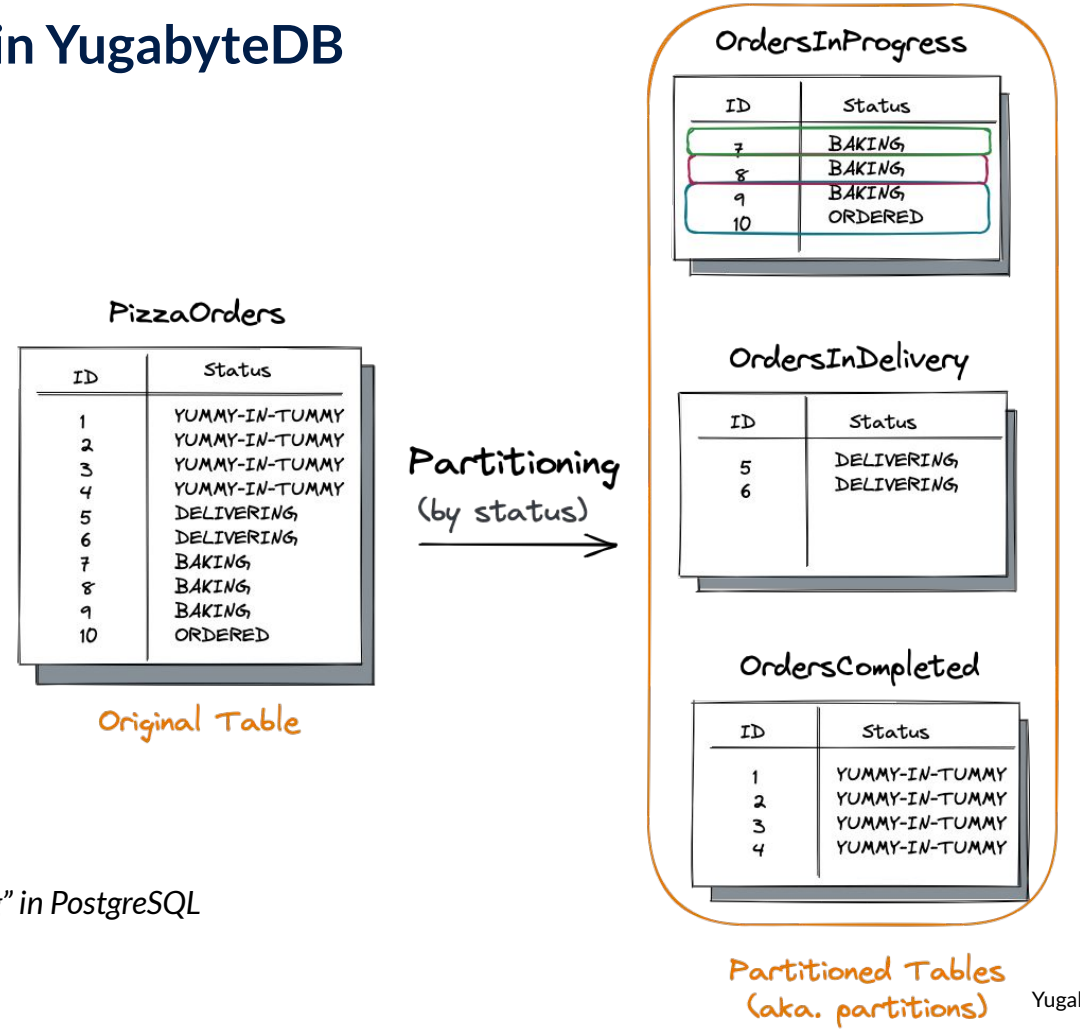


What you call "partitioning" in Cassandra, Ignite, SingleStore

Partitioning

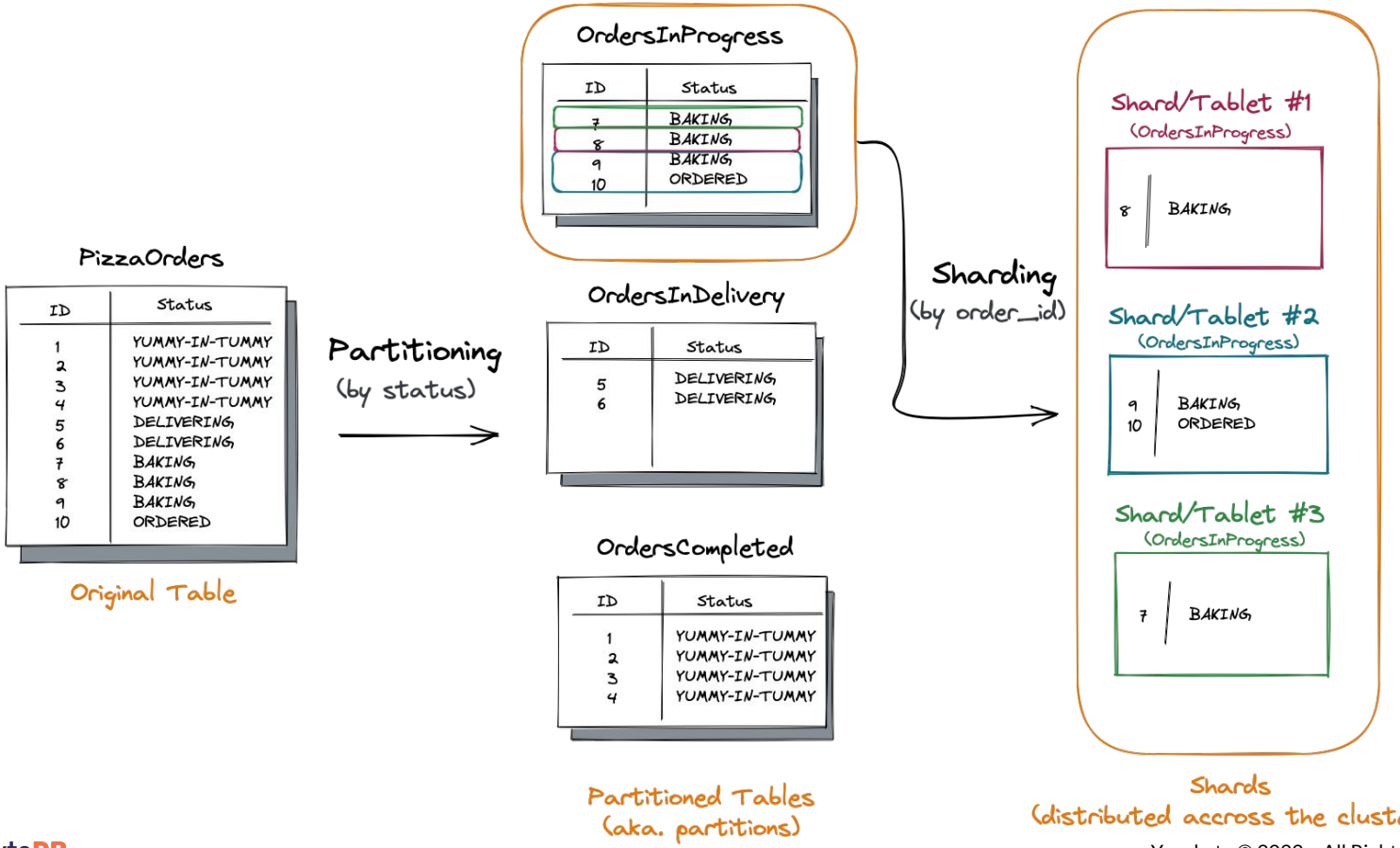
- A declarative way of splitting a large table into more manageable independent tables.
- Commonly available in single-node & distributed DBs (e.g., Postgres, Oracle, YugabyteDB).
- DDL statements can manipulate individual partitions.
- Simplifies manageability (e.g., dropping a partition via DDL is much cheaper dropping all its rows via DML).

Partitioning in YugabyteDB



Identical to "partitioning" in PostgreSQL

Partitioning + Sharding in YugabyteDB



Types of Table Partitioning in YugabyteDB

Types of Table Partitioning in YugabyteDB

	Description	PostgreSQL	YugabyteDB
Hash Partitioning	A table is partitioned by specifying a modulus and a remainder for each partition.	✔	✔
List Partitioning	A table is partitioned by explicitly listing which key value(s) appear in each partition.	✔	✔
Range Partitioning	A table is partitioned into ranges defined by one or more key columns.	✔	✔
Multilevel Partitioning	A table is partitioned by one method and then each partition is further subdivided into subpartitions.	✔	✔
Geo-partitioning	You can partition a table using one of the above methods. Then, you can pin these partitions to the desired locations.	❌	✔

Partition Pruning and Maintenance

Partition Pruning and Constraint Exclusion

- Excludes unnecessary partitions from the execution
- Partition Pruning:
  - Uses table's partition bounds
  - Applied at planning & execution phases
- Constraint Exclusion:
  - Uses table's CHECK constraints
  - Applied at the planning phase
- Works with constant and supplied arguments

```
# flags that control the behaviour
SET enable_partition_pruning = on;
SET constraint_exclusion = partition;
```

Partition Maintenance: What can you do with Partitioned Tables?

- Partitions can be…
  - Dropped
  - Added
  - Detached
  - Attached

```
#detach an existing partition
ALTER TABLE PizzaOrders
    DETACH PARTITION orders_2022_05;

#add a new partition
CREATE TABLE orders_2022_07
    PARTITION OF PizzaOrders
    FOR VALUES FROM('2022-07-01') TO ('2022-08-01');
```

Table Partitioning: Things to consider

- Indexes and unique constraints enforce integrity at the partition level
- Primary key needs to include the partition key
- Creating a foreign key reference on a partitioned table is not supported
- More here

Resources

- GitHub Gist with sample commands
- Table Partitioning, PostgreSQL docs
- Table Partitioning, YugabyteDB docs
- Sharding and. Partitioning in YugabyteDB (article)

Distributed SQL Essentials: Sharding and Partitioning in YugabyteDB

Franck Pachot

November 19, 2021

A distributed SQL database provides a service where you can query the global database without knowing where the rows are. You connect to any node, without having to know the cluster topology. You query your tables, and the database will determine the best access to your data, whether it's close to your client or geographically distant.

The organization of data, whether co-located or partitioned, is the most important consideration for high performance, high availability, and geo-distribution. YugabyteDB has several features that help here. These features can be divided into three levels of physical organization:

Thank You

Join us on Slack: yugabyte.com/slack (#yftt channel)

Star us on Github: github.com/yugabyte/yugabyte-db

YFTT YUGABYTEDB FRIDAY TECH TALKS

yugabyteDB