ONE PAGE SUMMARY

IT & Software › SQL › Temporal Tables

# Introduction to Temporal Tables

Master the basics of Temporal Tables in SQL!

## DEFINITION

> Temporal tables are special tables in SQL that automatically track historical data changes over time. They allow users to query data as it existed at any point in the past.

## KEY CONCEPTS

### System-Versioned Tables

These tables automatically keep track of changes with start and end timestamps.

### History Table

A separate table that stores the historical data for system-versioned tables.

### Temporal Queries

Queries that allow you to retrieve data as it was at a specific point in time.

### Data Retention

Policies that determine how long historical data is kept in the history table.

### Current vs Historical Data

Current data reflects the latest state, while historical data shows past states.

## EXAMPLES

→ SELECT * FROM Employees FOR SYSTEM_TIME AS OF '2023-01-01';

→ INSERT INTO Orders VALUES (...); -- Automatically tracked in history table

→ UPDATE Products SET Price = 20 WHERE ProductID = 1;

## MEMORY TIPS

★ Think 'time travel' for data: you can go back to see past states.

★ Remember 'current' and 'history' as two sides of the same table.

★ Use 'versioned' to recall that these tables keep versions of data.

## COMMON MISTAKES

✗ Confusing current data with historical data in queries.

✗ Neglecting to set up the history table when creating a temporal table.

✗ Forgetting to specify time periods in temporal queries.

## QUICK RECAP

Temporal tables in SQL help manage and query historical data efficiently. They consist of system-versioned tables and a history table, allowing users to track changes over time easily.