WISSEN TECHNOLOGY

# DIFFERENT WAYS OF REPRESENTING HISTORY IN DATABASE

Exploring Different Techniques for Effectively Representing History in Databases

# WHY REPRESENT HISTORY ?

### Auditability & Compliance
Example: Financial transactions must be recorded for audits to prevent fraud.

### Data Recovery & Debugging
If a system crashes or data is accidentally modified, historical data helps restore previous states.

### Business Intelligence & Trend Analysis
A retail company tracks yearly sales trends to predict future demand and adjust inventory.

### Versioning & Change Tracking
Used in collaboration platforms (Google Docs, GitHub) to maintain versions.

# TEMPORAL DATABASES

## What is a Temporal Database?

A Temporal Database is a type of database that tracks and stores data changes over time.

## Temporal Databases allow us to:

- Track the history of changes
- Know what data looked like at a specific time
- Help make decisions based on past data.

## Types of Time in Temporal Databases

**Transaction Time (System Time)**

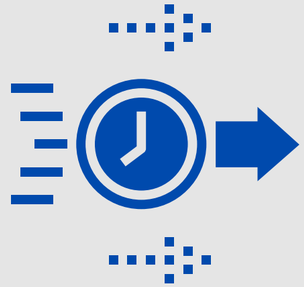
Refers to when data is stored or changed in the database.

Example: If a record is updated today, the transaction time is today.

**Valid Time (Business Time)**

Refers to when the data is true or valid in the real world.

Example: If a customer's address changed on January 1st but it was updated in the system on February 1st, the valid time is January 1st.



# SLOWLY CHANGING DIMENSIONS

## Types of SCD

1

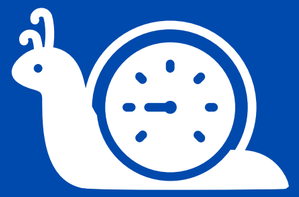
**SCD Type 1 (Overwrite Data)**
Simply update the record with new data, losing the old value.
Example: Updating a customer's phone number in a database.

2

**SCD Type 2 – Store historical data with a new row**
A new row is created for each change, keeping track of all previous versions
Example: When a customer changes their address, a new row is added with validity dates.

3

**SCD Type 3 – Store historical data in new columns**
Instead of adding new rows, add extra columns to store past values.
Example: If a product's price changes, store the old price and current price in separate columns.



# APPEND-ONLY LOGS

## What is a Temporal Database?

A Temporal Database is a type of database that tracks and stores data changes over time.

Existing data remains unchanged to ensure historical accuracy and traceability.

## How Append-Only Logs Work?

- New entries are always added at the end of the log.
- Old records are never modified or deleted.

| Transaction_ID | User | Action | Amount | Timestamp |
|---|---|---|---|---|
| 1 | Alice | Deposit | 100 | 2024-02-05 10:00 AM |
| 2 | Bob | Withdraw | 50 | 2024-02-05 10:05 AM |
| 3 | Alice | Deposit | 200 | 2024-02-05 10:10 AM |

**Bank Transactions Example**

# VERSIONING SYSTEMS

## What is a Versioning System?

A Versioning System is a method to track changes to files or data over time.

## Types of  Versioning System

### Local Versioning

Save different versions manually on your computer.

**Example:** File_1.docx, File_2.docx, File_3.docx

### Centralized Version Control

A single server stores all versions of the data.

**Example:** Subversion (SVN), Perforce

### Distributed Version Control

Every user has a complete copy of the repository.

**Example:** Git, Mercurial

| Feature | Local Versioning | CVCS (SVN) | DVCS (Git) |
|---|---|---|---|
| Works Offline | ✅ Yes | ❌ No | ✅ Yes |
| Requires Central Server | ❌ No | ✅ Yes | ❌ No |
| Supports Branching | ❌ No | ⚠️ Limited | ✅ Yes |
| Best for Teams | ❌ No | ⚠️ Yes (But centralized) | ✅ Yes (Fully distributed) |

# OTHER EXAMPLES

## EVENT SOURCING

- Captures every change as an event that can be replayed to reconstruct history.
- Used in finance, audit systems, and distributed systems.

## AUDIT TABLES

- Maintain a separate table logging changes to track modifications.
- Includes columns like timestamp, action (INSERT/UPDATE/DELETE), old & new values.

## TEMPORAL TABLES

(SQL STANDARD)

- SQL supports SYSTEM_VERSIONED TABLES, automatically tracking row changes over time.

WISSEN
TECHNOLOGY

# THANK YOU