Let's explain Slowly Changing Dimension Type 2 (SCD Type 2) with a very simple, plain example that illustrates the concepts clearly step-by-step.

**Scenario**

A customer has an address stored in a data warehouse. Over time, the customer moves to a new address. We want to keep all historical addresses of the customer, not just the current one.

**Initial Table: Customer Dimension Before Change**

| **Customer\_ID** | **Address** | **Start\_Date** | **End\_Date** | **Is\_Current** |
| --- | --- | --- | --- | --- |
| 1 | 123 Elm St | 2023-01-01 | 9999-12-31 | 1 |

* One record for Customer 1 at "123 Elm St".
* Start\_Date is when this version became valid.
* End\_Date is when this version stops being valid (default far future date means current).
* Is\_Current indicates this is current address.

**New Data: Customer Moves on 2025-09-22**

| **Customer\_ID** | **Address** | **Start\_Date** |
| --- | --- | --- |
| 1 | 456 Oak St | 2025-09-22 |

**Step 1: Expire Old Record**

Old address validity ends when the new one starts, so update the old record:

sql

**UPDATE** CustomerDimension

**SET** End\_Date = '2025-09-22',

Is\_Current = 0

**WHERE** Customer\_ID = 1

AND Is\_Current = 1;

After this update:

| **Customer\_ID** | **Address** | **Start\_Date** | **End\_Date** | **Is\_Current** |
| --- | --- | --- | --- | --- |
| 1 | 123 Elm St | 2023-01-01 | 2025-09-22 | 0 |

**Step 2: Fix Overlapping Dates (if needed)**

Imagine due to prior errors, End\_Date is after new Start\_Date. For example, old record ends 2025-09-25 instead of 2025-09-22 causing overlap.

This update fixes that:

sql

**UPDATE** CustomerDimension **AS** tgt

**SET** End\_Date = DATEADD(**day**, -1, src.Start\_Date)

**FROM** CustomerStaging **AS** src

**WHERE** tgt.Customer\_ID = src.Customer\_ID

AND tgt.Is\_Current = 0

AND tgt.End\_Date > src.Start\_Date;

It changes End\_Date from 2025-09-25 to 2025-09-21 — one day before the new record starts.

**Step 3: Insert New Record**

Add the new address record starting on 2025-09-22:

sql

**INSERT** **INTO** CustomerDimension (Customer\_ID, Address, Start\_Date, End\_Date, Is\_Current)

**VALUES** (1, '456 Oak St', '2025-09-22', '9999-12-31', 1);

**Final Table:**

| **Customer\_ID** | **Address** | **Start\_Date** | **End\_Date** | **Is\_Current** |
| --- | --- | --- | --- | --- |
| 1 | 123 Elm St | 2023-01-01 | 2025-09-21 | 0 |
| 1 | 456 Oak St | 2025-09-22 | 9999-12-31 | 1 |

**Summary**

* The old address record ends one day before the new address starts (no overlap).
* A new record is inserted for the new address with an open End\_Date.
* History of changes is preserved with accurate validity periods.
* This lets you query the address of the customer valid on any date.

This is the core idea behind SCD Type 2 and the reason for the two updates plus insert steps.

If visual timelines or more analogies would help, please ask!

Related

How does SCD Type 2 handle overlapping date ranges

What are common mistakes in implementing SCD Type 2

How can I visualize changes in historical data effectively

What tools support automated SCD Type 2 management

How do surrogate keys impact data retrieval in SCD Type 2

How does SCD Type 2 handle overlapping date ranges

How does SCD Type 2 handle overlapping date ranges

SCD Type 2 handles overlapping date ranges primarily through the careful management of the start and end dates of dimension records. The goal is to ensure that for any given business key (such as a customer ID), the validity periods of the records do not overlap, thus maintaining data integrity and accurate historical tracking.

Here's how SCD Type 2 manages this:

1. Setting End Dates for Old Records: When a change occurs, the current active record is "closed" by setting its end date to the day before the new record's start date. This prevents the old and new records from having overlapping validity periods.
2. Adjusting Overlapping Records: If issues exist where an old record’s end date extends beyond the new record’s start date (due to data errors or prior processing), an update is performed to trim the old record’s end date to one day before the new start date. This adjustment removes any overlap.
3. Inserting New Records with Clear Validity: New records are inserted with start dates equal to the change date and with open-ended end dates (e.g., '9999-12-31') to mark them as current.
4. Validation Queries: Often, self-joins or window functions are applied post-loading to detect any overlapping date ranges that might have slipped through, enabling correction.

**Concrete Example:**

* An old address record valid from 2023-01-01 to 2025-09-25.
* A new address record valid from 2025-09-22 onward exists.
* Since 2025-09-22 to 2025-09-25 overlaps, SCD Type 2 updates the old record’s end date to 2025-09-21 to eliminate overlap.

This approach guarantees clean, non-overlapping historical records suitable for temporal queries and accurate change tracking.

**Overlapping date ranges in SCD Type 2** can be prevented by following these best practices:

1. Set End Date as One Day Before Next Start Date  
   When updating an old record, always set its END\_DATE to one day before the new record's START\_DATE. This prevents overlapping validity periods. For example, if the new record starts on 2025-09-22, the old record ends on 2025-09-21.

**Problem: Overlapping Date Ranges in SCD Type 2**

In SCD Type 2, each record has a start date and an end date defining its validity period. Sometimes, due to previous data issues or processing errors, the end date of an old record overlaps with the start date of the new record, causing incorrect historical data.

**Example Data Before Fix**

| **CUSTOMER\_ID** | **ADDRESS** | **START\_DATE** | **END\_DATE** | **IS\_CURRENT** |
| --- | --- | --- | --- | --- |
| 1 | 111 Main St | 2023-01-01 | 2025-09-25 | 0 |
| 1 | 222 Oak St | 2025-09-22 | 9999-12-31 | 1 |

**The Overlap**

* Old record is valid until 2025-09-25.
* New record is valid from 2025-09-22.
* Dates overlap by 4 days (from 2025-09-22 to 2025-09-25).
* This creates confusion about which address was correct in this period.

**Fix: Correct the END\_DATE for the old record**

The update statement:

sql

**UPDATE** Customer\_Dimension **AS** tgt

**SET** END\_DATE = DATEADD(**day**, -1, src.START\_DATE)

**FROM** Customer\_Staging **AS** src

**WHERE** tgt.CUSTOMER\_ID = src.CUSTOMER\_ID

AND tgt.IS\_CURRENT = 0

AND tgt.END\_DATE > src.START\_DATE;

* For the old record, it sets END\_DATE to one day before the new record's START\_DATE.
* So, END\_DATE becomes 2025-09-21.
* This removes any overlap.