

## Slowly Changing Dimensions (SCD) – Typen 1 bis 6

Slowly Changing Dimensions (SCD) beschreiben Methoden zur Verarbeitung **langsam veränderlicher Attributwerte** in einem Data Warehouse. Sie sind entscheidend für die **Historisierung, Analyse und Konsistenz von Dimensionstabellen**.

### SCD Typ 1 – Überschreiben (ohne Historie)

#### ◇ Definition:

Der alte Wert wird **einfach überschrieben**. Änderungen sind **nicht nachvollziehbar**.

#### SQL-Beispiel:

```
sql
Code kopieren
UPDATE DimBenutzer
SET Email = 'neue-email@example.com'
WHERE BenutzerID = 123;
```

#### Anwendungsbeispiel:

- Korrektur eines falschen Namens oder einer E-Mail-Adresse
- Preisaktualisierung bei Produkten, bei denen Historie keine Rolle spielt

#### Einsatz:

- Wenn **nur der aktuelle Zustand** wichtig ist
- Keine Notwendigkeit zur Analyse von Veränderungen

## SCD Typ 2 – Historie bewahren (neue Zeile)

### ◇ Definition:

Bei Änderungen wird **eine neue Zeile eingefügt**. Der alte Datensatz wird mit einem **Ende-Datum geschlossen**.

So entsteht eine **vollständige Historie**.

### SQL-Beispiel:

```
sql
Code kopieren
-- Alten Eintrag schließen
UPDATE DimMitarbeiter
SET Gueltig_bis = '2023-03-31'
WHERE MitarbeiterID = 1 AND Gueltig_bis = '9999-12-31';

-- Neuen Datensatz einfügen
INSERT INTO DimMitarbeiter (MitarbeiterID, Name, Position,
Gueltig_von, Gueltig_bis)
VALUES (2, 'Max M.', 'Teamleitung', '2023-04-01', '9999-12-31');
```

### Beispieltabelle:

ID	Name	Position	Gueltig_von	Gueltig_bis
1	Max M.	Entwickler	01.01.2021	31.03.2023
2	Max M.	Teamleitung	01.04.2023	31.12.9999

### Einsatz:

- Bei **vollständiger Historisierung**
- Für **zeitbasierte Analysen** und **Audits**

## ● SCD Typ 3 – Begrenzte Historie (neue Spalte)

### ◇ Definition:

Nur eine **begrenzte Anzahl** vorheriger Zustände wird gespeichert – meist **nur der letzte**, z. B. in Spalte PreviousTitle.

### 📄 SQL-Beispiel:

```
sql
Code kopieren
UPDATE Mitarbeiter
SET PreviousTitle = CurrentTitle,
    CurrentTitle = 'Director',
    DateTitleChanged = '2024-05-19'
WHERE EmployeeID = 1;
```

### 📊 Beispieltabelle:

EmployeeID	Name	CurrentTitle	PreviousTitle	DateTitleChanged
1	John Doe	Director	Manager	19.05.2024

### 🔑 Einsatz:

- Wenn **nur der letzte bekannte Zustand** relevant ist
- Für **Vergleich alt/neu** bei wenigen Änderungen

## ● SCD Typ 4 – Historientabelle getrennt

### ◇ Definition:

Der aktuelle Datensatz bleibt in der Haupttabelle, während **alle alten Versionen** in einer **separaten History-Tabelle** gespeichert werden.

## SQL-Beispiel:

```
sql
Code kopieren
-- Alte Version archivieren
INSERT INTO DimKunde_History
SELECT * FROM DimKunde WHERE KundeID = 99;

-- Aktuelle Daten überschreiben
UPDATE DimKunde
SET Adresse = 'Neue Straße 5'
WHERE KundeID = 99;
```

## Zwei Tabellen:

- DimKunde → nur aktueller Stand
- DimKunde\_History → vollständige Historie

## Einsatz:

- Trennung von **aktuellen Daten und Historie**
- Nützlich für **Performanceoptimierung**

# ● SCD Typ 5 – Kombination aus Typ 1 + 4

## ◇ Definition:

Aktuelle Änderungen werden in der Haupttabelle **überschrieben** (Typ 1), aber über einen **Verweis auf die Historientabelle (Typ 4)** kann auf vergangene Zustände zugegriffen werden.

## Beispielstruktur:

- DimKunde (enthält aktuelle Daten + Foreign Key auf Historie)
- DimKunde\_History (alle Versionen)

### Einsatz:

- Man möchte **aktuelle Daten schnell zugreifbar** haben
- Aber trotzdem **historische Nachvollziehbarkeit**

## SCD Typ 6 – Hybrid aus Typ 1 + 2 + 3

### ◇ Definition:

Vereint die Vorteile von:

- **Typ 1:** aktuelle Werte überschreiben
- **Typ 2:** vollständige Historie mit Gültigkeitsfeldern
- **Typ 3:** zusätzliche Spalte für letzte bekannte Version

### Tabelle enthält:

ID	Name	Aktueller_Titel	Vorheriger_Titel	Von	Bis
5	Max	Direktor	Teamleitung	2024-01-01	9999-12-31

### Einsatz:

- Wenn **komplexe Historien** nötig sind
- **Audits + Vergleiche + aktuelle Darstellung** in einem

## Zusammenfassung als Tabelle

T y p	Methode	Historie	Struktur	Wann verwenden?
1	Überschreiben	✗	1 Zeile	Nur aktueller Wert zählt
2	Neue Zeile + Zeitstempel	✓	mehrere Zeilen	Vollständige Historie nötig
3	Alte Werte in Extra- Spalte	◆ begrenzt	1 Zeile + alte Spalten	Vergleiche alt/neu für 1–2 Zustände

4	Historie in separater Tabelle	✓	Haupt + History- Tabelle	Trennung von aktueller und alter Sicht
5	Typ 1 + Verweis auf Typ 4	✓	kombiniert	Effizient & nachvollziehbar
6	Kombination von 1 + 2 + 3	✓ ++	alle Methoden kombiniert	Flexibel, aber komplex