**Question : The store wants to keep customer addresses. Propose two architectures for the CUSTOMER\_ADDRESS table, one that will retain changes, and another that will overwrite. Which is type 1, which is type 2?**

**1. Architecture that overwrites changes in the table**

Description: Every time a customer’s address changes, the old address is replaced with the new one. Only the latest address is kept.

Table example (CUSTOMER\_ADDRESS): Before Update for Customer\_id=101

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CUSTOMER\_ID | STREET | CITY | STATE | ZIP |
| 101 | 123 Main St | Cambridge | MA | 02110 |
| 102 | 456 Elm St | Chicago | IL | 60605 |

Table example (CUSTOMER\_ADDRESS): After Update for Customer\_id=101

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CUSTOMER\_ID | STREET | CITY | STATE | ZIP |
| 101 | 123 Main St | Boston | MA | 02110 |
| 102 | 456 Elm St | Chicago | IL | 60605 |

If customer 101 moves from Cambridge to Boston, the row is updated. The Cambridge address is lost.

This is SCD Type: Type 1 🡪 Type 1 = overwrite old values, no history is preserved.

**2. Architecture that retains changes the changes in the table**

Description: When a customer’s address changes, the old address is kept, and the new address is inserted as a new record. You can see the history of addresses.

Table example (CUSTOMER\_ADDRESS):

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CUSTOMER\_ADDRESS\_ID** | **CUSTOMER\_ID** | **STREET** | **CITY** | **STATE** | **ZIP** | **START\_DATE** | **END\_DATE** | **CURRENT\_FLAG** |
| 1 | 101 | 123 Main St | Cambridge | MA | 02110 | 2023-01-01 | 2025-01-01 | N |
| 2 | 101 | 789 Oak St | Boston | MA | 02140 | 2025-01-02 | NULL | Y |
| 3 | 102 | 456 Elm St | Chicago | Il | 60605 | 2024-06-01 | NULL | Y |

You can see the historical addresses of a customer. The CURRENT\_FLAG column shows which address is active.

This is SCD Type: Type 2 🡪 Type 2 = retain history by adding new rows for changes.