SCD Type 1 & Type 2 — Interview Guide

Comprehensive summary of Slowly Changing Dimensions, including concepts, SQL implementations, dbt patterns, and interview questions.

Concept Overview

Slowly Changing Dimensions (SCD) manage historical attribute changes in dimension tables.
• SCD Type 1: Overwrites old data, no history retained.
• SCD Type 2: Inserts new record per change, tracks history with start_date/end_date and is_current flag.

Type	Behavior	Use Case
SCD1	Overwrite existing record	Data correction
SCD2	Insert new version of record	Track historical changes

SQL Implementation Examples

SCD Type 1 (Overwrite):

```
merge into dim_customer t using stg_customer s on t.customer_id = s.customer_id when matched then update set name = s.name, city = s.city when not matched then insert (customer_id, name, city) values (s.customer_id, s.name, s.city);
```

SCD Type 2 (Preserve History):

```
merge into dim_customer t using stg_customer s on t.customer_id = s.customer_id and t.is_current = true when matched and (s.city <> t.city or s.name <> t.name) then update set t.end_date = current_date - 1, t.is_current = false when not matched by target then insert (customer_id, name, city, start_date, end_date, is_current) values (s.customer_id, s.name, s.city, current_date, '9999-12-31', true);
```

Interview Questions & Answers

Q: What's the key difference between SCD1 and SCD2?
A: SCD1 overwrites data with the latest value. SCD2 adds a new record per change to maintain historical versions.

Q: When would you prefer SCD1?
A: When history isn't needed, and only the latest data is relevant.

Q: What columns are required for SCD2?
A: start_date, end_date, is_current, and a surrogate key.

Q: How to detect changes efficiently?
A: Compare hashes (e.g., MD5) of tracked attributes between source and target.

Q: How to query record valid on a date?
A: SELECT * FROM dim_customer WHERE '2024-03-01' BETWEEN start_date AND end_date;

Q: What’s SCD3?
A: Tracks limited history (current + previous values only).

Q: How to handle late-arriving data?
A: Reopen the prior record, adjusting start_date and end_date.

Q: Common challenges?
A: Overlapping dates, late data, merge conflicts, and performance.

dbt Incremental Example

```
{{ config(materialized='incremental', unique_key='customer_id',
incremental_strategy='merge') }} with src as (select * from {{ ref('stg_customer')
}}) select customer_id, name, city, current_timestamp() as load_ts from src
```