**6. Data Modeling & Warehousing (SCD, Fact/Dimension, Schema Design)**

**Scenario:**  
Your Gold layer feeds a BI warehouse (e.g., Redshift, BigQuery, Snowflake, Databricks SQL). You must design a schema to support **historical analysis** (SCD), **fast reporting** (facts/dimensions), and **query optimizations**.

**Baseline approach (expected in interviews):**

* Use **Star Schema**:
  + **Fact tables**: measurable events (e.g., transactions, orders).
  + **Dimension tables**: descriptive attributes (e.g., customers, products).
* Handle **slowly changing dimensions (SCD)** for history:
  + **Type 1** → overwrite (no history).
  + **Type 2** → add new row with surrogate key, mark old row inactive.
* Partition fact tables by **date** for query pruning.

**Example: SCD Type 2 table (customers\_dim):**

| **customer\_sk** | **customer\_id** | **name** | **email** | **valid\_from** | **valid\_to** | **is\_active** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 101 | Alice | [a@x.com](mailto:a@x.com) | 2025-01-01 | 2025-05-31 | 0 |
| 2 | 101 | Alice | alice@new.com | 2025-06-01 | NULL | 1 |

**Advanced considerations:**

* **CDC + SCD merge:** integrate CDC pipeline (Problem 1) directly into dimension tables.
* **Schema evolution:** handle new attributes by altering dimension tables.
* **Surrogate keys:** use surrogate keys (customer\_sk) instead of natural keys for facts.
* **Fact table granularity:** carefully define lowest level (e.g., transaction-level vs daily aggregate).
* **Query performance:**
  + Cluster/partition facts by date.
  + Use materialized views or aggregates for dashboards.
  + Pre-compute rollups for common queries.

**Follow-up questions:**

* When do you choose **Star vs Snowflake schema**?
  + Star → simpler, faster queries (preferred for BI).
  + Snowflake → normalized, saves storage, but slower queries.
* How do you handle **schema changes** (new column in dimension)?
  + Add nullable column; evolve schema with versioning.
* How do you design for **late-arriving facts** (e.g., transactions arriving 3 days late)?
  + Partition by event\_date not ingestion\_date; allow backfills.
* How do you optimize for **BI queries scanning billions of rows**?
  + Partition + clustering, summary tables, Z-order in Delta, materialized views.

✅ Now your **story arc** flows all the way to BI:

“We orchestrate ingestion → Silver → Gold (Problems 3–5), then model Gold into fact/dimension tables (Problem 6) using SCD Type 2 for historical tracking, surrogate keys for consistency, and partitioning/clustering for BI query performance.”

This shows you can **bridge engineering with analytics** — a big differentiator for DE interviews.