**Data Modeling Review & Enhancement**

**1. Peer-Review of ERDs & Star Schemas**

**What is Peer-Review?**

A process where data models are reviewed by peers to find errors, inconsistencies, or improvement areas before final implementation.

**Key Focus Areas**

* **Entity Relationship Diagrams (ERDs):**
  + Check each entity has a primary key (PK).
  + Verify relationships use foreign keys (FK).
  + Validate cardinalities (1:1, 1:M, M:N).
  + Ensure attribute naming follows conventions.
  + Detect redundant or missing entities.
* **Star Schema Review:**
  + Confirm one central fact table surrounded by denormalized dimension tables.
  + Ensure dimensions use surrogate keys.
  + Avoid snowflake schemas unless justified.
  + Identify degenerate dimensions (facts stored in fact table as attributes).
  + Check for junk dimensions (combine low-cardinality attributes).

**Benefits of Peer Review**

* Improves model accuracy and completeness
* Ensures consistency across the team
* Facilitates knowledge sharing

**2. Grain & SCD Checklist**

**Grain**

* **Definition:** The level of detail recorded in a fact table.
* **Why Important?** It determines how facts relate to dimensions and affects query results.
* **Checklist:**
  + Clearly define grain (e.g., one row per sales line item).
  + Avoid mixing granularity in one fact table.
  + Align dimensions to the fact’s grain.

**Slowly Changing Dimensions (SCD)**

* **Purpose:** Track changes in dimension attributes over time.
* **Types:**
  + **Type 0:** No changes allowed.
  + **Type 1:** Overwrite old data (no history).
  + **Type 2:** Keep history by adding new rows with effective dates and surrogate keys.
  + **Type 3:** Store limited history in the same row with extra columns.
* **Checklist for SCD Type 2:**
  + Surrogate keys in dimension tables.
  + Add columns: start\_date, end\_date, current\_flag.
  + Use hashing or change detection for identifying updates.

**3. Refactor Models from Feedback**

**Why Refactor?**

To improve clarity, consistency, and performance of data models based on peer or stakeholder feedback.

**Common Refactoring Activities**

* Rename entities and attributes for clarity and naming consistency.
* Add surrogate keys where missing.
* Adjust data types and precision.
* Normalize or denormalize tables based on usage patterns.
* Remove redundant relationships and circular references.

**Best Practices**

* Keep dimension tables flat for star schema performance.
* Document all changes and rationales.

**4. Mapping Logical to Physical Model**

**Logical Model**

* Abstract design showing entities, attributes, and relationships without technical details.

**Physical Model**

* Concrete implementation with tables, columns, constraints, data types, and indexes.

**Mapping Guidelines**

| **Logical Concept** | **Physical Implementation** |
| --- | --- |
| Entity | Table |
| Attribute | Column |
| Identifier (PK) | Primary Key constraint |
| Relationship (FK) | Foreign Key constraint |
| Derived Attribute | Computed column or view |

**Additional Considerations**

* Choose appropriate data types for storage and performance.
* Define indexes on keys and frequently filtered columns.
* Plan partitions and clustering if data volume is large.
* Consider storage format (e.g., row vs column store).

**5. Mini-Lab Fixes**

**Common Issues to Fix**

* Incorrect cardinalities (e.g., M:N not properly modeled).
* Misaligned grain in fact tables.
* Missing surrogate keys in dimensions.
* Ambiguous or inconsistent naming.
* Incorrect nullability and default values.

**Approach**

* Review feedback comments carefully.
* Revisit grain definitions and ensure consistency.
* Add surrogate keys and update related foreign keys.
* Rename tables/columns to match conventions.
* Validate and enforce constraints.

**Summary**

Mastering data modeling review and enhancement ensures a robust, scalable, and performant data warehouse design. Understanding grain, SCDs, and physical mapping bridges the gap between conceptual design and implementation.