DAMA SAUDI

Chapter 5

Data Modelling and Design

Key Takeaways

Definition of Cardinality

- **Cardinality** defines how many instances of one entity are related to instances of another entity.
- It is used to specify relationships in database systems, determining the nature of the association between tables or entities.

Reference: Cardinality helps in structuring relationships, such as one-to-one, one-to-many, or many-to-many.

Super-type and Sub-type Entity Relationship

- A **sub-type** inherits all attributes and relationships of the super-type.
  - **Sub-type entities must be mutually exclusive** is **not true** of super-type/sub-type relationships.
  - Sub-types may or may not be mutually exclusive, depending on the model.

Reference: Sub-types share some attributes with their super-type, but not all must be mutually exclusive.

Star and Snowflake Data Modeling Schemes

- **Star and Snowflake** are concepts of the **Dimensional** data modeling scheme.
- These models are used in data warehousing for organizing data in a multi-dimensional format.

Reference: Dimensional modeling optimizes query performance in business intelligence systems, often using star and snowflake schemas.

Data Modeling Styles

- **CRUD** is **not** a data modeling style.
- CRUD represents operations (Create, Read, Update, Delete), while modeling styles include **ORM**, **UML**, **IDEF1X**, and **CHEN**.

Reference: CRUD is an operations framework rather than a style of data modeling.

Purpose of a Logical Data Model

- The purpose of a **Logical Data Model** is to **define the structure of data elements and set relationships between them**.
  - It is used to represent how data should be structured within a system.

Reference: Logical models bridge the gap between conceptual understanding and physical implementation.

Keys in Data Models

- **Logical Key** is **not** considered a key in data modeling.
  - Common keys include **Primary Key**, **Foreign Key**, **Surrogate Key**, and **Alternate Key**.

Reference: Logical Key refers to abstract definitions rather than being a concrete key used in database systems.

Synonym for Relation in Relational Data Model

- In the relational data model, the best synonym for a "relation" is **Table**.
    - A relation represents a set of data that can be described as a table with rows and columns.

Reference: Relations (or tables) are the core structure in relational databases, organizing data into rows (tuples) and columns (attributes).

Relationship Labels in Database Technology

- **Relationship labels** are **verb phrases describing business rules in each direction between two entities.**
  - They define how entities are related and interact with one another within the database.

Reference: Relationship labels clarify how different data elements are linked and used in business processes.

Business Rules in Data Models

- A business rule **defines constraints on what can and cannot be done**.
- Business rules specify the conditions that must be met for processes and operations within the system.

Reference: Business rules play a critical role in ensuring data integrity and guiding system behavior.

Use of Foreign Keys from Code Tables

- A database using **foreign keys from code tables for column values** is implementing **Reference Data**.
  - Reference data is often stored in code tables and referenced by other tables using foreign keys.

Reference: This approach ensures consistency and accuracy of data values across systems.

Number of Data Models Expected in Projects

- More **physical data models** than **logical data models**, and more **logical data models** than **conceptual data models**, can be expected in projects.
- As projects move from high-level abstraction to implementation, more detailed models (physical) are required.

Reference: Projects typically start with a single conceptual model but require multiple logical and physical models for different systems or components.

Definition of a Surrogate Key

- A **surrogate key** is a **unique identifier attached to each record, which may be used as a primary key**.
  - It is usually an automatically generated value that does not have a business meaning.

Reference: Surrogate keys simplify identifying records in large datasets, especially when natural keys are not ideal.

False Statement About Business Rules

- **Data rules cannot be shown on a data model** is **false**.
  - Data rules, such as constraints and relationships, can indeed be represented in data models.

Reference: Business rules guide data design and can be modeled as constraints or relationships in a data model.

Business Rule Not on a Logical Data Model

- **Customer Last Name requires a non-unique index to improve retrieval performance** should **not appear on a logical data model**.
  - Indexing is a physical implementation concern, not a logical data model concern.

Reference: Logical models focus on structure and relationships, not performance-related implementation details.

Purpose of Denormalization in Data Models

- Denormalization is done to **optimize overall database performance across both data access and data update requests**.
  - It adds redundancy to improve query performance but comes with trade-offs in terms of data integrity.

Reference: Denormalization balances the need for faster data retrieval against potential complexities in data management.

'X' in Information Engineering Subtype Discriminator

- The 'X' in the subtype discriminator symbol in information engineering means **Exclusive**.
  - It indicates that an entity can belong to only one sub-type in the hierarchy.

Reference: The 'X' helps enforce constraints in sub-type relationships, ensuring clarity in entity hierarchies.

Recursive Relationship

- A **recursive relationship** is **unary** (involves one entity) and is also referred to as **self-referencing**.
- This relationship occurs when an entity has a relationship with itself, such as in hierarchical structures.

Reference: Recursive relationships are common in organizational hierarchies and other self-referencing data structures.

Bank Example: Many-to-Many Relationship

- The relationship type where **each Customer may own one or many Accounts, and each Account must be owned by one or many Customers** is **many-to-many**.
- This type of relationship involves linking multiple instances of two entities.

Reference: Many-to-many relationships require intermediary tables in relational databases to manage the links between entities.

Data Model Components

- **Data model components** include **Keys, Relationships, Attributes, Entities, and Facts**.
  - These components define the structure and relationships of data within a database.

Reference: Data models help visualize and organize data, ensuring clarity in how entities relate to one another.

Alternate Key

- An **alternate key** is a **candidate key not selected to be the primary key**.
- It can be used as a unique identifier but is not the main key chosen for the table.

Reference: Alternate keys provide additional options for uniquely identifying records in a relational database.

Best Relationship Type for Management Hierarchy

- The best relationship type for the example where **an employee may work for one other employee and may manage one or more employees** is **recursive**.
- This relationship is commonly used to model hierarchical structures like management chains.

Reference: Recursive relationships effectively model hierarchical or self-referencing structures in data models.

Entity Synonym in Physical Model

- The most common term for "entity" at the physical level of a model is **Table**.
- Entities represent data objects, and at the physical level, they are implemented as tables in relational databases.

Reference: Physical models translate entities into tables, defining how data will be stored in the database.

Properties of a Logical Data Model

- A **Logical Data Model** includes:
  - **Relationship cardinality**, **attributes**, and **primary keys**.
  - It is **technology-independent**, meaning it is abstract and not tied to any specific database platform.
  - The property that is **not true** for a logical data model is **technology-dependent**.

Reference: Logical data models focus on the logical structure of data without considering the underlying technology that will implement the model.

Super-type and Sub-type Entity Relationship

- In a **super-type/sub-type** relationship:
  - **Sub-types** inherit all attributes and relationships from the **super-type**.
  - Attributes and relationships shared by all sub-types are modeled in the super-type.
  - A common misconception is that **sub-type entities must be mutually exclusive**, which is not necessarily true.

Reference: Super-type/sub-type relationships are used to model entities with common characteristics while allowing for specific sub-type differences.