Asg1-RelORM

Yengkong Sayaovong

IFT 300

Intermediate DB Mgmt Systems

10/18/2023

**Contrasting Halpin's ORM Approach and the Relational Approach**

There are many differences between Halpin's Object-Role Modeling (ORM) approach and the traditional relational approach to database design in relation to Halpin's approach. A comparison of these two methodologies is based on a number of factors, such as overarching themes, ORM principles, and relational principles, which reveal the differences between the two.

**Overarching Theme: Information Base vs. Database**

It is Halpin's ORM approach that revolves around creating an Information Base as opposed to a traditional database, in order to accomplish this. In this linguistic shift, information is viewed more as truths and rules, rather than merely as a collection of data that can be accessed. A subtle distinction between ORM and the relational approach occurs in its emphasis on capturing a more comprehensive representation of the real-world domains and relationships in contrast to the relational approach's emphasis on structured data storage.

**ORM Concepts:**

**ORM Diagram Features:** ORM diagrams feature Entity Types, Relationship Types, and their interconnections, as opposed to the table-centric approach of relational models. There is an emphasis on the logical and structural relationships that exist between objects and the roles they play in society.

**Entity Type and Domain Entity:** In the ORM approach, an entity type corresponds to a real-world object, which is aligned with the concept of entities in the relational approach. ORM places a strong emphasis, however, on the roles that entities play within relationships as part of the overall relationship.

**Domain Value, Domain Value Type:** In ORM tools, Domain Values are like attributes in relational databases in the sense that they represent attributes or properties of Domain Entities, such as entities **in** the system. The main focus here is on the role that values play within the structure of entities, and that role is the focus of the paper.

**Constraints - Uniqueness, Mandatory:**

In ORM, a constraint such as uniqueness and mandatory participation is enforced through a set of rules. The concept of uniqueness is similar to that of a primary key in a relational database, as it ensures that entities have distinct roles to perform. Like the mandatory constraint in relational databases, mandatory participation requires the presence of a particular role or relationship.

**Relational Concepts:**

**Relation and Tuple:** A Relation corresponds to a table in the relational approach, and each Tuple signifies a row within that table. This mirrors the practice of organizing data into tables with rows and columns.

**Primary Key:** In the relational approach, the primary key serves as a unique identifier for each tuple, ensuring data integrity and eliminating duplicate entries. This aligns with ORM's uniqueness constraint.

**Foreign Key:** A Foreign Key establishes connections between tables by referencing the primary key of another table, akin to ORM's Relationship Types that link entities.

**Candidate Key:** Concerning the uniqueness requirement, the candidate key is similar to ORM's uniqueness enforcement, though it may lack the same semantic depth in the relational approach.

**Summary of Approaches:**

Relational approaches and ORMs operate at different abstraction levels. The ORM approach emphasizes conceptual level, whereas the relational approach emphasizes logical level. Relational design begins with the arrangement of data tables, while ORM begins with capturing the essence of real-world concepts and their relationships. Thus, beginning with a relational approach may result in a more rigid structure and a potential disconnect with real-world semantics, whereas ORM facilitates a more natural design.

**Reference:**

Halpin, T. A. (2015). *Object-role modeling fundamentals : a practical guide to Data Modeling with ORM*. Technics Publications.

Murach, J., & Syverson, B. (2023). *Murach’s SQL Server 2022 for Developers*.