Object and Relational Models

by Sophia

WHAT'S COVERED

This tutorial explores the use of object-oriented data models created in the mid-1980s through modern approaches to big data in three parts:

1. Object-Oriented Models
2. Extended Relational Models
3. Modern Big Data Challenges

1. Object-Oriented Models

As more complex real-world problems developed in the late 20th century, there was a need for newer data models. Starting in the mid-1980s, the object-oriented and object-object relational data models came into use. Unlike the entity-relationship model, the object-oriented data model (OODM) stores both the data and the relationships in a single structure within an object. In other words, objects are created in which both data and its relationships are all contained in one place. This means that the objects contain the factual content, but unlike the relational model, they also contain information about the relationships between the facts, giving the facts within the objects greater meaning. The OODM is the basis of the object-oriented database management system (OODBMS).

The concept of the OODM was a data model that closely represented the real world. In many ways, the object in an OODM is equivalent to an entity relationship's entity (a record). The attributes describe the properties of the object that contains them (similar to the columns within a table). Objects that have similar characteristics are grouped into classes, where they have a shared structure of attributes as well as shared behaviors or methods. These methods define an available real-world action related to the object. In many ways, they are similar to procedures or functions in other programming languages.

↱ EXAMPLE For example, one method could be defined for finding an artist's name and a second method defined for setting the artist's name within the object.

These classes are then organized in a class hierarchy that resembles an upside-down tree. Each class has a single parent, but a parent can have multiple children. Notice that this is quite similar to the structure in the hierarchical data model.

THINK ABOUT IT

Why might the hierarchical data model have become useful again for this type of data? How might older approaches to organizing data become relevant again even in future models?

2. Extended Relational Models

With more need to handle complex data representations, object/relational (O/R) and Extensible Markup Language (XML) evolved from the relational model starting in the mid-1990s. Many of the relational model vendors created the extended relational data model (ERDM) as a way to add the object-oriented model features in a more straightforward relational database structure. This gave birth to a new, fifth generation of relational databases that would be able to support object-oriented features like objects. Many of the databases that we see today fall under this realm of object/relational.

Extensible Markup Language (XML) was used as the standard to store and exchange structured and unstructured data. With O/R databases being able to add support for XML-based documents, it made the use of these files more efficient. Although relational and object/relational databases do meet most current processing needs, there is still a new generation of databases that have been emerging to address particular business criteria.

---

3. Modern Big Data Challenges

The current generation of emerging data models from the early 2000s to the early 2020s focuses on NoSQL and big data. These include the key-value store, wide-column store, document-oriented, and graph stores. In today's world, there are mountains of data that are stored, and it’s not always possible to fit some of the unstructured or social media data into the conventional structure of rows and columns. Many of these needs are unique, and there isn’t a single solution to meet all of the needs of organizations when it comes to data management. In the current NoSQL databases, JavaScript Object Notation (JSON) has emerged as a replacement for XML as the means to store structured and unstructured data.

SUMMARY

The object-relational and **object-oriented** databases reflected the shift to object-oriented computer programming languages in the 1980s. More recently, the need to manage, organize, and share large amounts of unstructured data led to the development of **extended relational models** like object/relational (O/R) and NoSQL. JSON is now the preferred format to address **modern big data challenges**.

Source: Authored by Vincent Tran