Physical Design

by Sophia

WHAT'S COVERED

This tutorial explores last step in designing a database. We will explore physical model design in three parts:

1. Rationale for Physical Design
2. Details of the Database
3. eCommerce Example

1. Rationale for Physical Design

The physical model is the final data model that extends the logical model. This data model describes how the database will be implemented, using a specific database management system. This model is created by the database administrator and the database developers. The purpose here is to implement the database.

By the end of the physical data model process, the entities are translated into tables and attributes are translated into columns. The relationships between the tables are defined, and any physical considerations that may need to change the logical data model are accounted for. This model is completely dependent on the hardware and database management software and should include all table structures including the column name, data types, column constraints, primary keys, foreign keys, and relationships.

It is important to note that the physical data model will be different depending on the relational database management system (e.g. PostgreSQL) that has been selected. Each database will use its own specific data types.

2. Details of the Database

The physical design focuses on data storage, security measures, and performance measures. Before we can define the data storage organization, we have to know the volume of the data and usage patterns. The column data types and sizes are also defined here, so it is important to be aware of the nature of the data. We will also determine the indexes for each table beyond the primary keys, depending on the usage patterns or performance requirements. If there are any anticipated views to create for the database, they would be useful to define now.

With respect to security, we take the time to define the group roles and user roles that can access this database and what level of privileges they should have for each object. This goes beyond the physical data

model itself, but it is a crucial step to take when planning for the physical database design. All of the constraints we have defined as part of the logical data model, such as unique or NOT NULL constraints, would also be implemented as part of the physical data model. Once this model has been created, it should be ready to be implemented in the database.

3. eCommerce Example

Here is a completed physical data model for the eCommerce database. You may notice that a lot of the core elements are identical to the logical data model, with added data types and sizes. In other databases, there may be changes with added constraints depending on the business rules or other criteria being incorporated.

SUMMARY

The **rationale for the physical data model** is to extend the logical data model. We will need to define the **details of the database**, such as the data types, sizes, and constraints specific to the hardware and the database management software. The **ecommerce example** demonstrates how our design for our case study concludes.

Source: Authored by Vincent Tran