**Chapter 2**

**Fundamentals**

**Object –Relational Mismatch**

The object world is designed to solve specific problems: booking a flight, transferring money, shopping for books, and so on. Technical solutions are designed with a domain model in mind to solve these problems. The domain model defines the objects representing a real-world problem. A flight reservation, bank account, and book search are the examples of domain objects. These domain objects are expected to be stored and retrieved from a database.

Most of the real-world applications cannot exist without some form of persistence. We would go crazy if our banks said that they lost our money because they hadn’t stored our deposits in a durable storage space. Etc.

Of all the choices available, relational database persistence is the industry standard of durable storage.

Relational databases store the data in a two-dimensional format: rows and columns. The data relationships are expressed in the form of foreign keys.

The problem arises when we start thinking of storing the domain objects into a relational database. The objects in our application are not in a row-column format; they hold state using attributes (variable). So, unfortunately, they can’t be stored as is. At a very high level, this mismatch is called object-relational impedance mismatch.

There are few fundamental differences between the object and relational models, which are discussed briefly here.