

Object Relational Impedence Mismatch

Object oriented programming languages store data in the form of interconnected graphs of objects whereas RDBMS represent data in a table like format

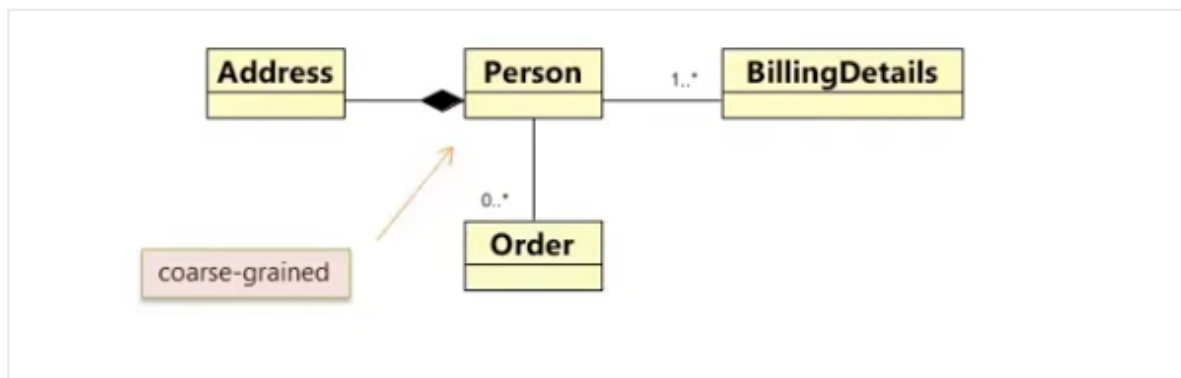
So OOP languages -> Object model
RDMBS -> relational model

Because both the models differ significantly in the way they represent data, when we load or store object model data in a relational database we come across 5 different types of mismatch - ORIM or Paradigm mismatch
Object Model and the Relational Model do not work very well together

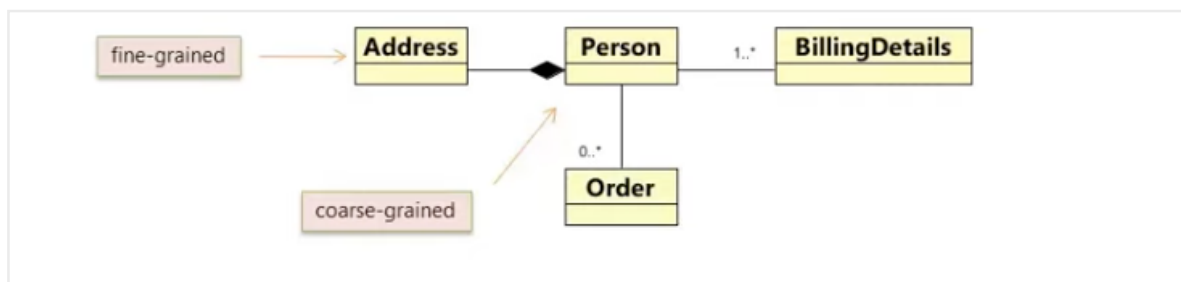
1. Granularity

Extent to which a system can be broken down into small parts or components

In an e-commerce application a Person object is a coarse-grained object because it contains a lot of details about the person, such as the person's address, billingDetails, orders



Another granular Object in an e-commerce application is Address which is fine-grained because it contains limited details or details specific to an address

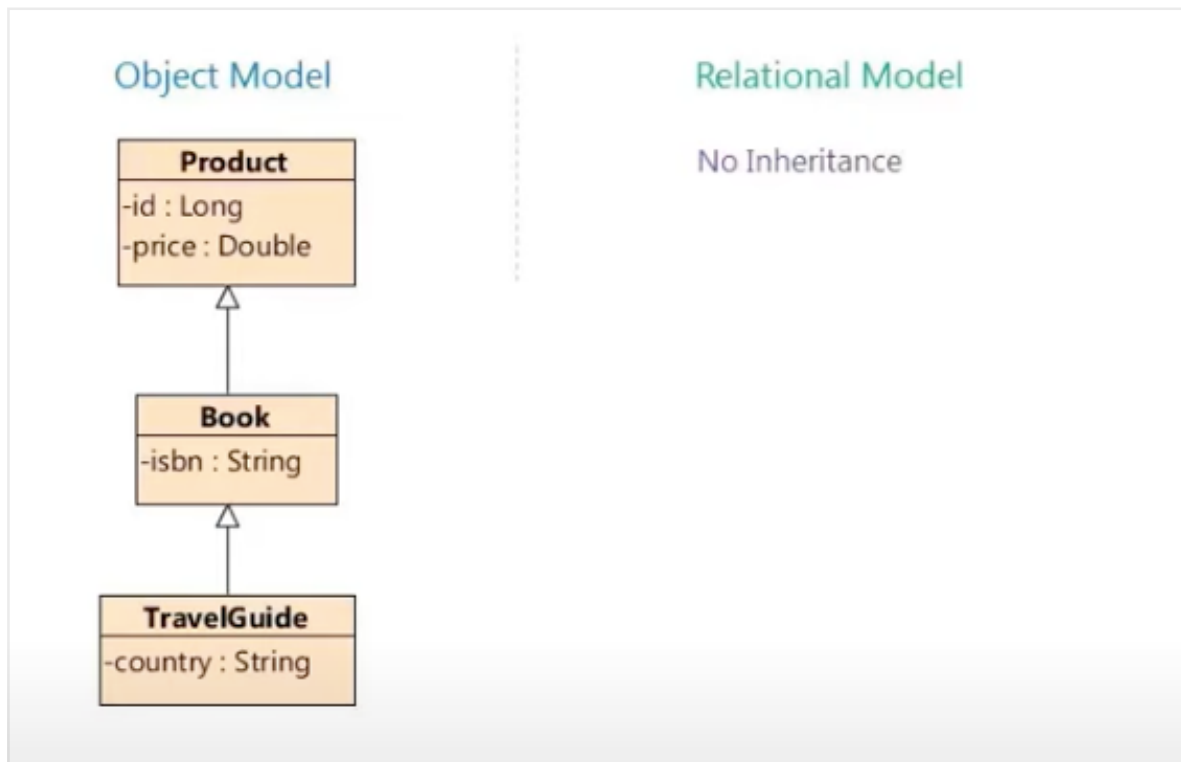


Object Model	Relational Model
---------------------	-------------------------

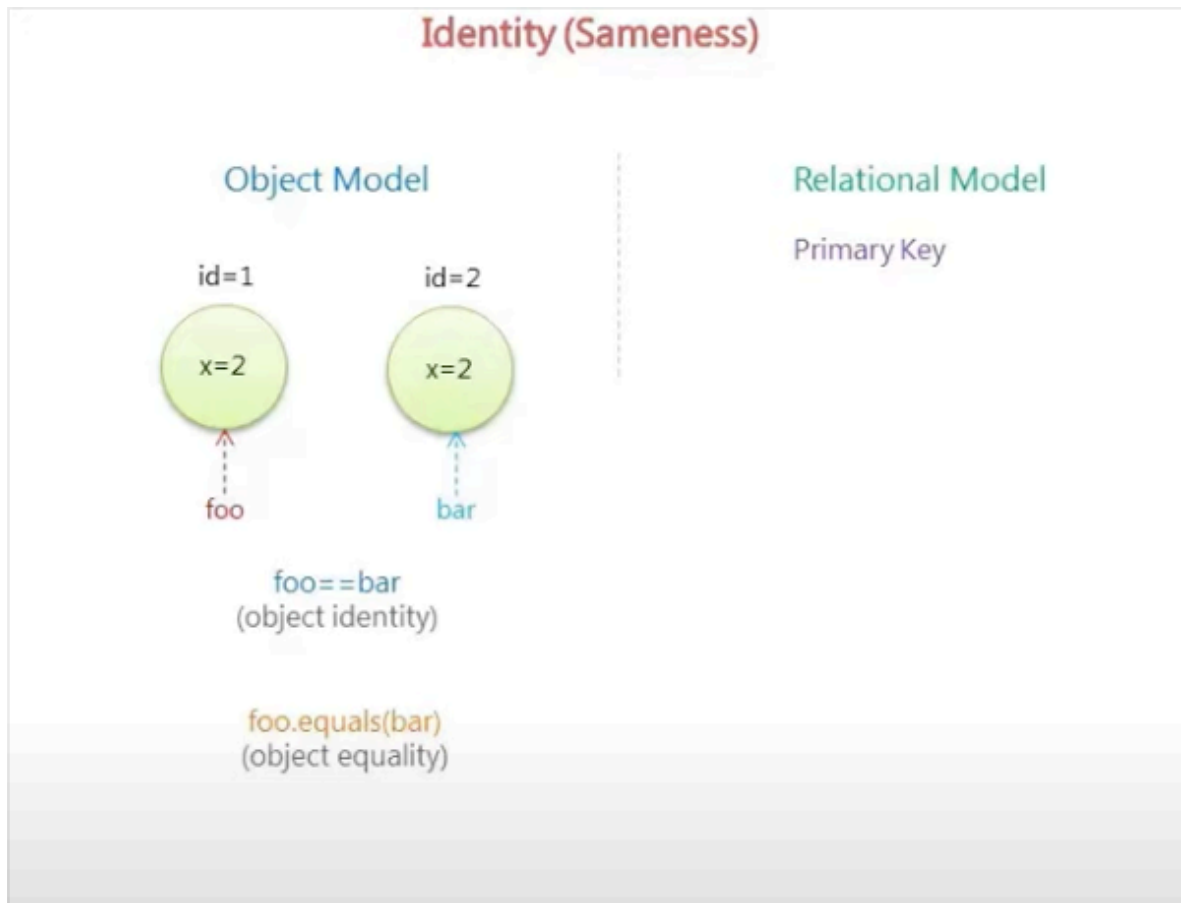
Various levels of granularity	2 levels of granularity (tables and columns)

So when we decide to store data related to an object in a relational database, we will actually have far more classes than tables.

Inheritance Mismatch

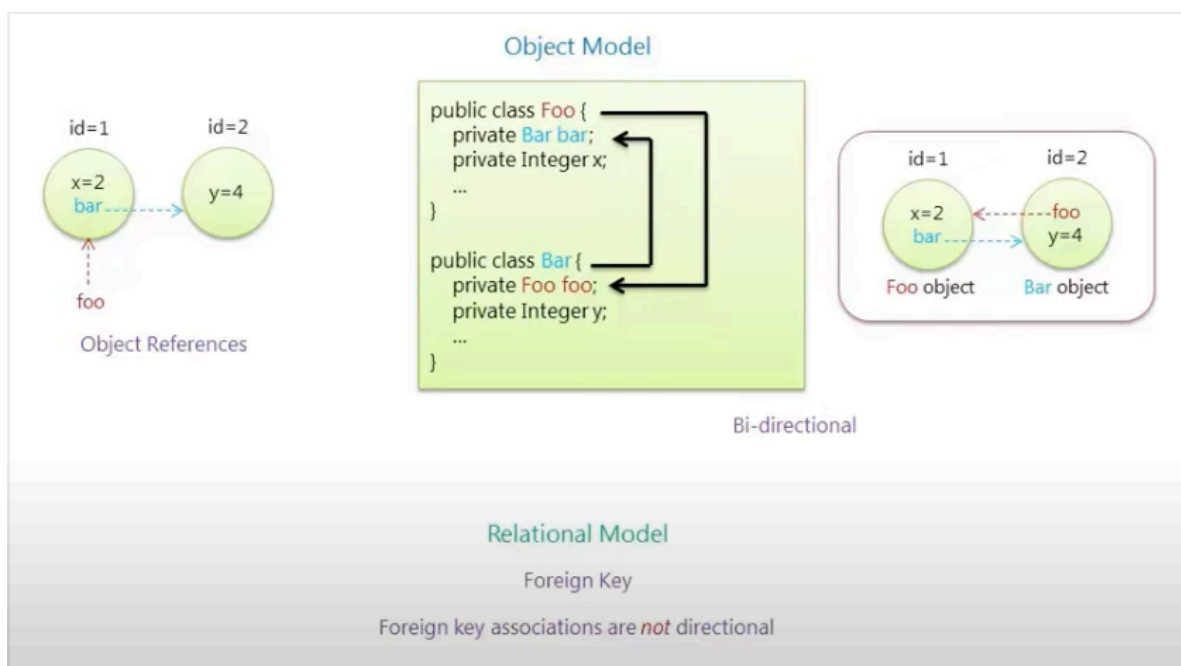


Identity Mismatch



In a relational database, the sameness of two entities is decided by their primary keys,

Associations Mismatch



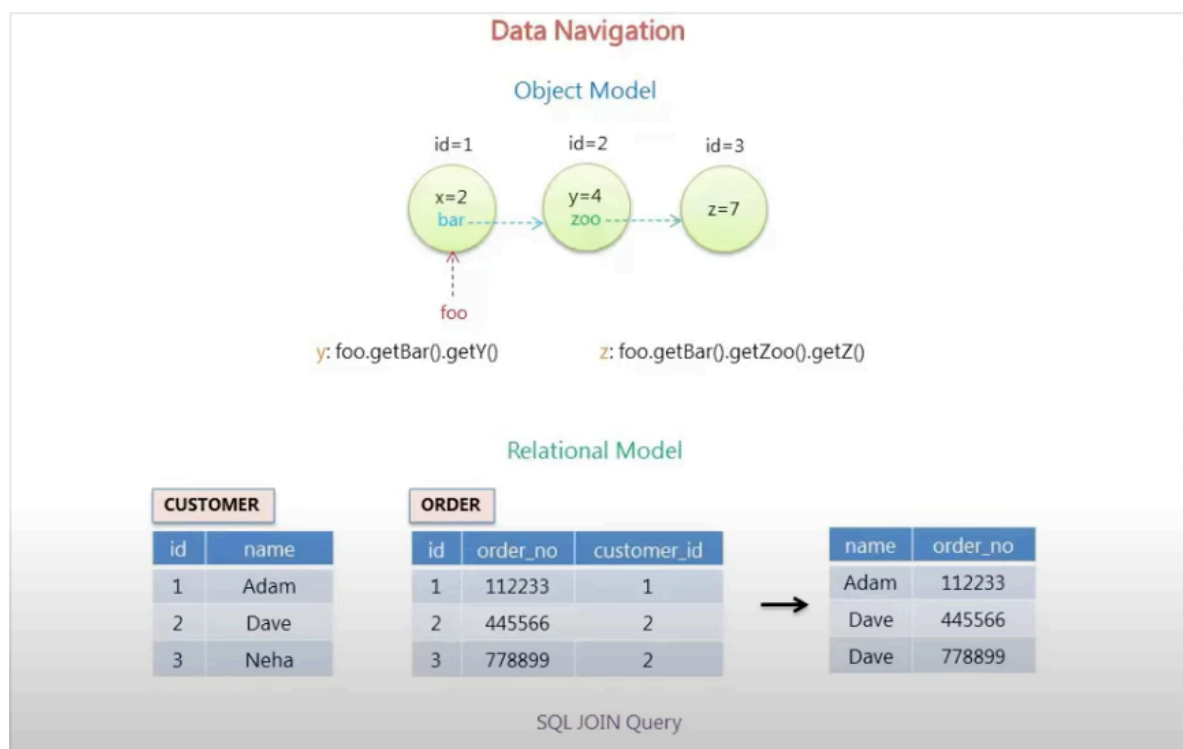
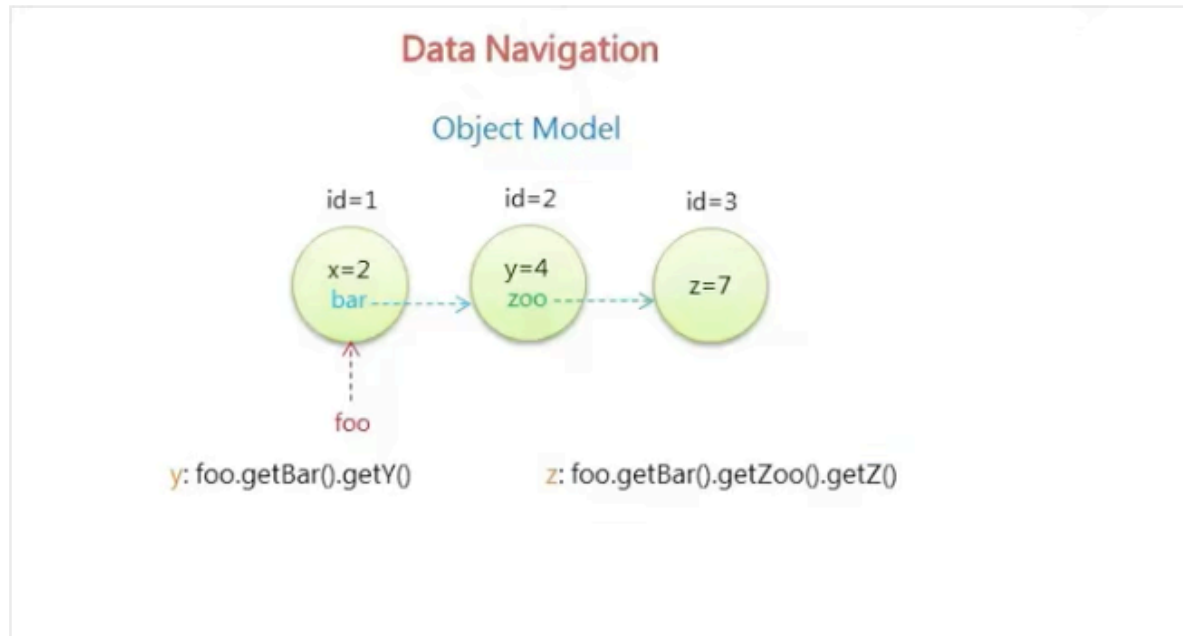
In Object models -> references

In relational models -> foreign keys to associate two entities

Object references are directional

But foreign key associations are not directional, for example joins are symmetric in nature.

Data Navigation



In an object model, there is no way we can access the value of z without getting the value of y and that of y without accessing the bar object.

But in a relational model, this will be highly underperformant, we would want to make as less calls to the database as possible and hence, we perform joins, or

write join queries in order to get the data as efficiently as possible.