Individuals

[Ahmed Mohamed Mostafa] [Link]

"Object to Relational Mapping" is what **ORM** stands for.

- ❖ You utilise the **Object** portion with your programming language (python in this case)
- ❖ A Relational Database Manager System makes up the **Relationship** part (A database that is) Although there are various types of databases, relational databases are the most common (you know tables, columns, pk fk etc eg Oracle MySQL, MS-SQL)
- ❖ Finally, you build a bridge between your objects and your tables in the **Mapping** section.

Pros:

- You write your data model in only one place, which makes it easier to update, maintain, and reuse the code.
- It forces you to write MVC code, making your code alot cleaner.
- Uses OOP.

Cons:

- Takes time to set up and learn to use.
- Abstracts the DB, which can be hard for unfamiliar programmers.

[Ahmed M.Osman] [Link]
Individually (BAD CAT)

Introduction

Object-Relational Mapping (ORM) is a technique that lets you query and manipulate data from a database using an object-oriented paradigm. When talking about ORM, most people are referring to a *library* that implements the Object-Relational Mapping technique, hence the phrase "an ORM".

An ORM library is a completely ordinary library written in your language of choice that encapsulates the code needed to manipulate the data, so you don't use SQL anymore; you interact directly with an object in the same language you're using.

Using ORM saves a lot of time because:

- **DRY: You write your data model in only one place, and it's easier to update, maintain, and reuse the code.**
- **lot of stuff is done automatically, from database handling to I18N.**

- **It forces you to write MVC code, which, in the end, makes your code a little cleaner.**
- **You don't have to write poorly-formed SQL (most Web programmers really suck at it, because SQL is treated like a "sub" language, when in reality it's a very powerful and complex one).**
- **Sanitizing; using prepared statements or transactions are as easy as calling a method.**

If you want to try an ORM library in Web programming, you'd be better off using an entire framework stack

[Esraa Hussein] []

What is ORM?

Object Relational Mapping (is the bridge between databases and object-oriented programming. The ORM equips you with object-oriented tools to run commands that you would usually run on databases. It masks out the complicated intricacies of the databases and lets you manipulate them with your choice of programming language (must support object-oriented programming).

Pros

1. It is easier to query using ORM than MySQL.
2. Hides the complex processes under-the-hood and only lets you worry about the high-level implementation.
3. There is no need to change the code if the underlying database changes.

Cons

1. Learning an ORM from scratch is time-consuming.
2. You cannot directly dive into an ORM without learning MySQL. Even though the ORM helps you abstract the complex details, it is pertinent to know the consequence of each command under-the-hood.
3. You may run into performance issues while coding complex queries using an ORM.

Example

There is a wide range of ORMs in different object-oriented programing languages. Let’s consider the following MySQL query:

```
 $ value = SELECT * FROM collection WHERE day = 'Monday'
```

The above query retrieves a value from the table collection in which columns (day) are equal to Monday.

In an ORM, this query would look something like this:

```
value = collection.query(day = 'Monday'))
```

Team

[Ahmed Hisham][Link]

[Reem Sameh][Link]

[Nagy Nabil][Link]

[Yasser Ramadan][Link]

1- What is a database ORM?

ORM stands for (Object-Relational Mapping)
it's a library make you can read and write in database using the language of your project (so you don’t need to use SQL language any more), what help to make using database more simple and faster

because the database be a part of your project tables be classes and records be objects

2- Advantages of using database ORM

1- Speed-up Development-eliminates that need for repetitive SQL code.

2- Reduce Development Time.

3- Reduce Development Costs.

4- Overcomes vendor specific SQL differences- the ORM knows how to write vendor specific SQL so you don't have to.

3-Problems or disadvantages of using ORMs

1-They should be learned

2-Performance is a problem when queries get complicated (experts can sometimes prefer using their own sql queries)

3-Writing complex queries could be a pain when using ORMs

4-Examples of common database ORMs used with node.js

	Popularity	Repo Activity	Support	Maturity and Stability	Supported Databases
Prisma	Popular	Very active	Strong	Relatively new	MySQL, PostgreSQL, MSSQL, & SQLite (MongoDB, CockroachDB, and PlanetScale in Preview support)
TypeORM	Very popular	Active	Good	Mature	MySQL, MariaDB, PostgreSQL, CockroachDB, SQLite, Microsoft SQL Server, Oracle, SAP Hana, sql.js
sequelize	popular	Active	Good	Mature	ORM for Oracle, Postgres, MySQL, MariaDB, SQLite and SQL Server

5. Creating database with ORM example[prisma]

Prisma has its own declarative schema language, you write schema to express your data and relations in human readable way
> note prisma can handle database migrations if you edit your schema
```
prisma migrate
```
Prisma can auto detect your existing database schema through the cli command
```
prisma db pull
```
to connect to postgreSQL database you need to configure a datasource block in **prisma schema**
```schema.prisma
datasource db {
  provider = "postgresql"
  url      = env("POSTGRES_DB")
}
create model in **prisma.schema** to express data , relations and constraints
```
model Device {
  id   Int   @id @default(autoincrement())
  name String
}
```
to interact with the prisma in the server tell prisma to generate client
```
prisma generate
```
import the generated client in your server and here you go
For more reading PostgreSQL database connector (Reference) | Prisma Docs
```

References:

https://stackoverflow.com/questions/4667906/the-advantages-and-disadvantages-of-using-orm

https://stackoverflow.com/questions/1279613/what-is-an-orm-how-does-it-work-and-how-should-i-use-one

https://www.prisma.io/dataguide/database-tools/top-nodejs-orms-query-builders-and-database-libraries