ORM Tools

1 Definition and Importance

ORM (Object-Relational Mapping) tools are tools that convert relational database to object-oriented approach and the developer can manipulate the database without writing raw SQL queries and instead use the functions that the ORM provide.

Hence the developer will use those functions, it will simplify database operation (CRUD) and it will enhance the readability and writability and make the development easier and faster.

2 Some of the popular ORM Tools

We will see different ORM tools and also tools that are not exactly ORM tools, but it serves the same goal.

2.1 Exposed

It's a lightweight ORM which means it has the basic features of ORM but without the overhead of full ORM that may reduce the performance, and it offers two approaches which are SQL DSL (Domain Specific Language) and DAO (Data Access Object).

The DSL keeps SQL concepts while adding type safety and the DAO provides an object-oriented approach similar to ORM.

Advantages:

1- No optimization overhead because it's lightweight
2- Easier integration
3- Provides two approaches

Disadvantages:

1- Low flexibility
2- Relatively new and still evolving

2.2 Hibernate

It’s an ORM that implements JPA (Java Persistence API) and has more features on top of it, also it has HQL (Hibernate Query Language) that is similar to SQL but it’s object-oriented language

Advantages:

1- High flexibility
2- Object-oriented approach
3- Large community

Disadvantages:

1- Complexity
2- Performance overhead
3- Steep learning curve

2.3 jOOQ

It’s an internal DSL and source code generator which offers the ability to write SQL queries using Java code and make it type-safe.

Advantages:

1- Compile time type safety.
2- Code generation.
3- Easy integration.

Disadvantages:

1- No Built-in Relationship Management
2- Require a good knowledge of SQL.
3- Bigger amount of code.

2.4 Ktorm

A lightweight ORM designed for Kotlin and provides a SQL DSL.

Advantages:

1- Designed for Kotlin so it supports its features such as extension functions and lambda expressions.
2- Type-safe because it uses DSL.
3- No optimization overhead

Disadvantages:

1- Low flexibility
2- Smaller community
3- No Built-in Relationship Management

2.5 SQLdelight

It's a type-safe SQL tool that generates APIs from SQL statements that must be written in a certain file.

Advantages:

1- High performance because of direct database access
2- Type safety

Disadvantages:

1- Requires good knowledge of SQL because it requires writing raw SQL
2- No Built-in Relationship Management