

ORM vs Raw SQL

Pros of raw SQL

Compatible with any backend (Java, C++, Python, etc)

Portable, easier to maintain

No need to install, manage, and learn ORM

Helps new contributors get up to speed faster

Database design is not influenced by ORM

Pros of ORM

Lower barrier to entry if unfamiliar with SQL

Increases development speed, great for rapid prototyping

Provides consistent, object-oriented structure and naming patterns

Usually fewer lines of code to write

May provide integration with migrations tool

Raw SQL VS. ORM



Raw SQL	ORM
Lower-level direct queries	Higher-level abstractions
Fewer tools to learn and manage	Convenient
Better for complex queries	Better for simple queries
Functional approach	Object-oriented approach

Raw SQL VS. ORM



Raw SQL	ORM
Lower-level direct queries	Higher-level abstractions
Fewer tools to learn and manage	Convenient
Better for complex queries	Better for simple queries
Functional approach	Object-oriented approach



Beware: Object-relational impedance mismatch

How to choose?

It depends on....

Resources (time, number of developers)

Scale (load, complexity)

Expertise with SQL vs. backend language

It's all SQL!

ORMs provide a framework for working with SQL

Only some backends use an ORM

Even when using an ORM, it's still important to know SQL

You must learn SQL to use relational databases