

[What’s New in SQLAlchemy 2.0? — SQLAlchemy 2.0 Documentation](https://docs.sqlalchemy.org/en/20/changelog/whatsnew_20.html)

the two most significant front-facing portions of SQLAlchemy are the **Object Relational Mapper (ORM)** and the **Core**.

Core contains the breadth of SQLAlchemy’s SQL and database integration and description services, the most prominent part of this being the **SQL Expression Language**.

The SQL Expression Language is a toolkit on its own, independent of the ORM package, which provides a system of constructing SQL expressions represented by composable objects, which can then be “executed” against a target database within the scope of a specific transaction, returning a result set. Inserts, updates and deletes (i.e. [DML](https://docs.sqlalchemy.org/en/20/glossary.html#term-DML)) are achieved by passing SQL expression objects representing these statements along with dictionaries that represent parameters to be used with each statement.

The ORM builds upon Core to provide a means of working with a domain object model mapped to a database schema. When using the ORM, SQL statements are constructed in mostly the same way as when using Core, however the task of DML, which here refers to the persistence of business objects in a database, is automated using a pattern called [unit of work](https://docs.sqlalchemy.org/en/20/glossary.html#term-unit-of-work), which translates changes in state against mutable objects into INSERT, UPDATE and DELETE constructs which are then invoked in terms of those objects. SELECT statements are also augmented by ORM-specific automations and object-centric querying capabilities.