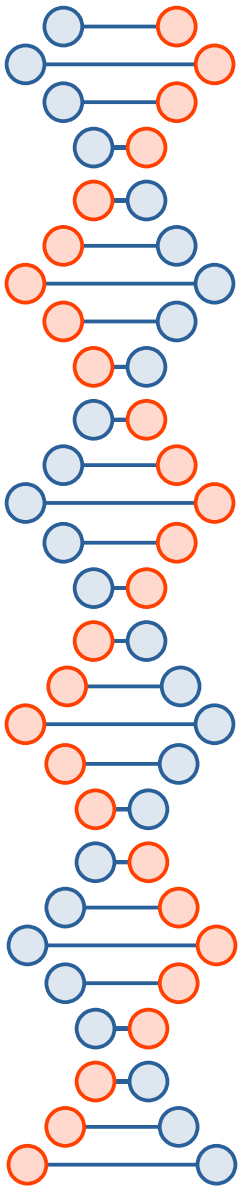


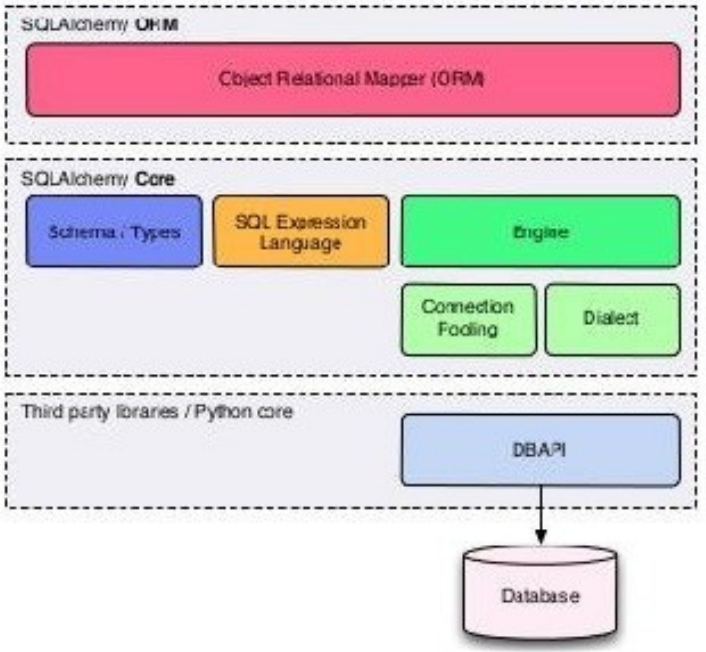
What is SQLAlchemy

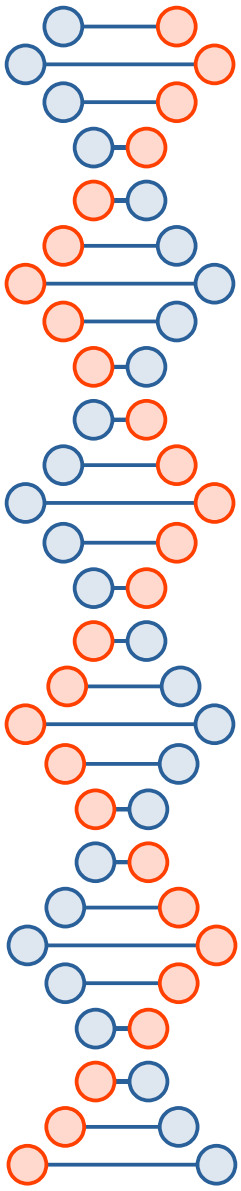
SQLAlchemy is a popular SQL toolkit and Object Relational Mapper.

- It is written in Python and gives full power and flexibility of SQL to an application developer. It is an open source and cross-platform software released under MIT license.
- SQLAlchemy is famous for its object-relational mapper (ORM), using which, classes can be mapped to the database, thereby allowing the object model and database schema to develop in a cleanly decoupled way from the beginning.

SQLAlchemy Overview

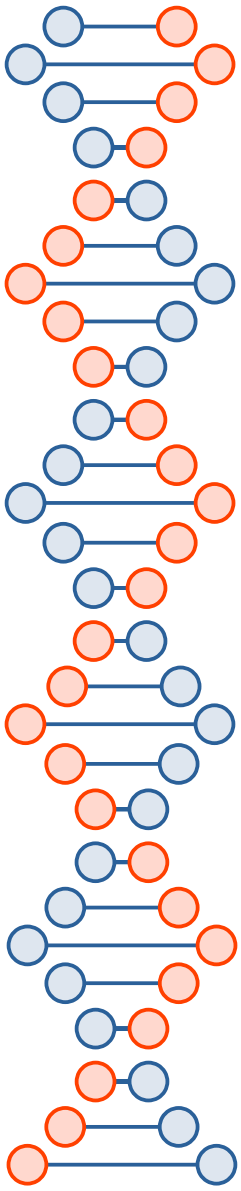
SQLAlchemy consists of the **Core** and the **ORM**





SQLAlchemy is like an Onion

Can be learned from the inside out, or outside in



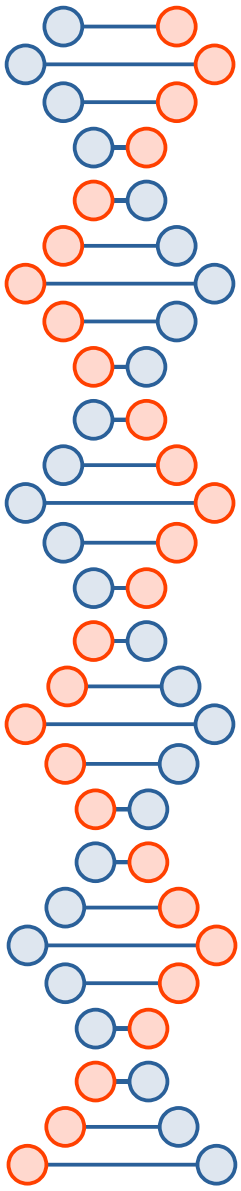
What happens ?

As size and performance of SQL databases start to matter, they behave less like object collections.

As abstraction in object collections starts to matter, they behave less like tables and rows.

Solutions -

SQLAlchemy aims to accommodate both of these principles.

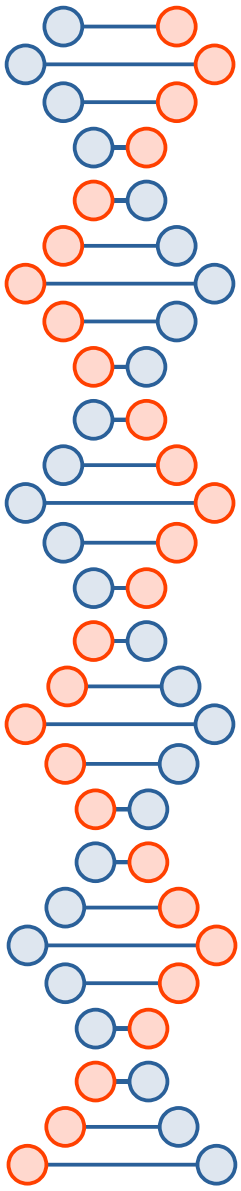


ORM ?

ORM (Object Relational Mapping) is a programming technique for converting data between incompatible type systems in object-oriented programming languages. Usually, the type system used in an Object Oriented (OO) language like Python contains non-scalar types.

These cannot be expressed as primitive types such as integers and strings.

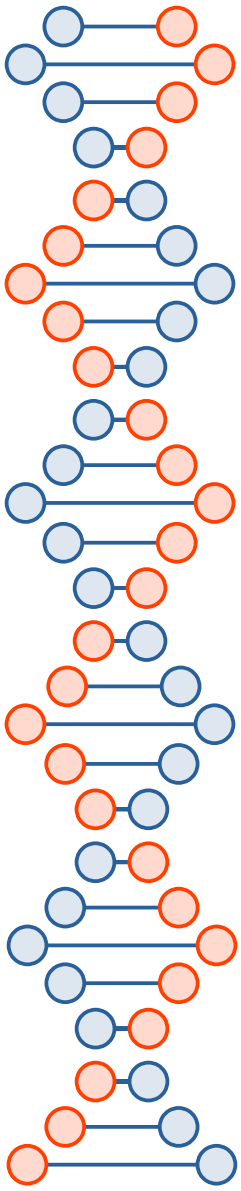
OO programmer has to convert objects in scalar data to interact with backend database.



How ORM helps ?

In an ORM system, each class maps to a table in the underlying database.

Instead of writing tedious database interfacing code yourself, an ORM takes care of these issues for you while you can focus on programming the logics of the system.



SQLAlchemy Core – expression language

SQLAlchemy core includes

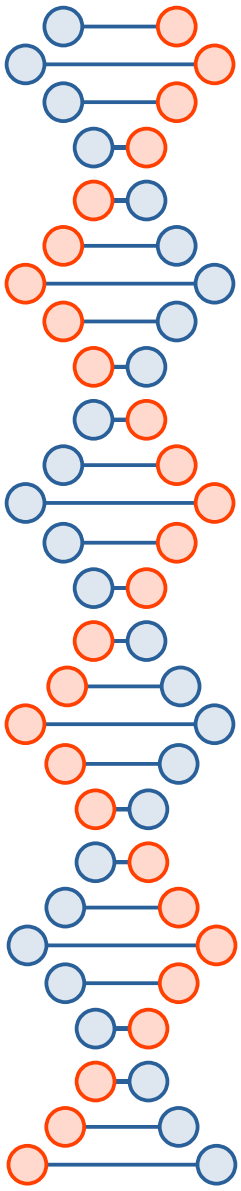
a) SQL rendering engine,

b) DBAPI integration,

c) Transaction integration

d) Schema description services.

SQLAlchemy core uses SQL Expression Language that provides a schema-centric usage paradigm whereas SQLAlchemy ORM is a domain-centric mode of usage.

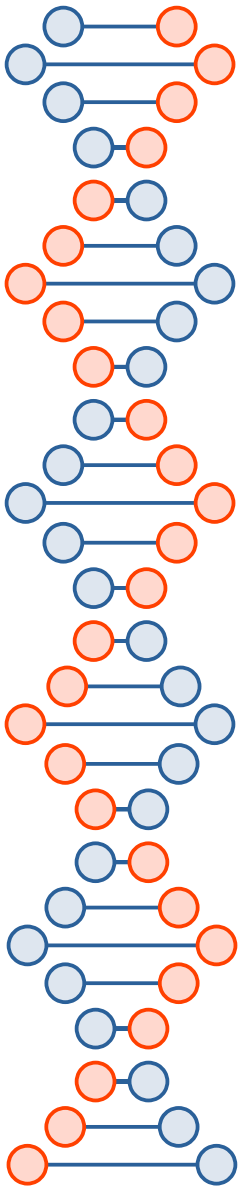


SQL Expression

SQL Expression Language presents a system of representing relational database structures and expressions using Python constructs.

It presents a system of representing the primitive constructs of the relational database directly.

It presents a system of representing the primitive constructs of the relational database directly



SqlAlchemy Core – Conn to DB

Engine class connects a Pool and Dialect together to provide a source of database connectivity and behavior. An object of Engine class is instantiated using the create_engine() function.

e.g

dialect[+driver]://user:password@host/dbname

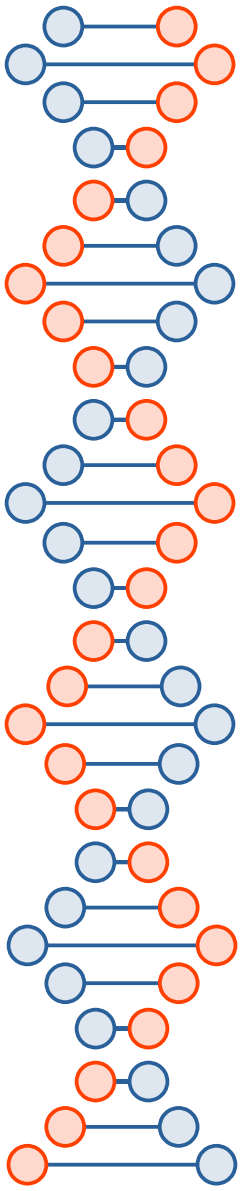
mysql+pymysql://<username>:<password>@<host>/<dbname>

```
from sqlalchemy import create_engine

engine = create_engine('sqlite:///college.db', echo=True)
```

For a MySQL database, use the below command:

```
engine = create_engine("mysql://user:pwd@localhost/college",echo=True)
```



create_engine returns ?

connect()	Returns connection object
execute()	Executes a SQL statement construct
begin()	Returns a context manager delivering a Connection with a Transaction established. Upon successful operation, the Transaction is committed, else it is rolled back
dispose()	Disposes of the connection pool used by the Engine
driver()	Driver name of the Dialect in use by the Engine
table_names()	Returns a list of all table names available in the database
transaction()	Executes the given function within a transaction boundary