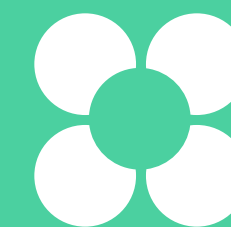


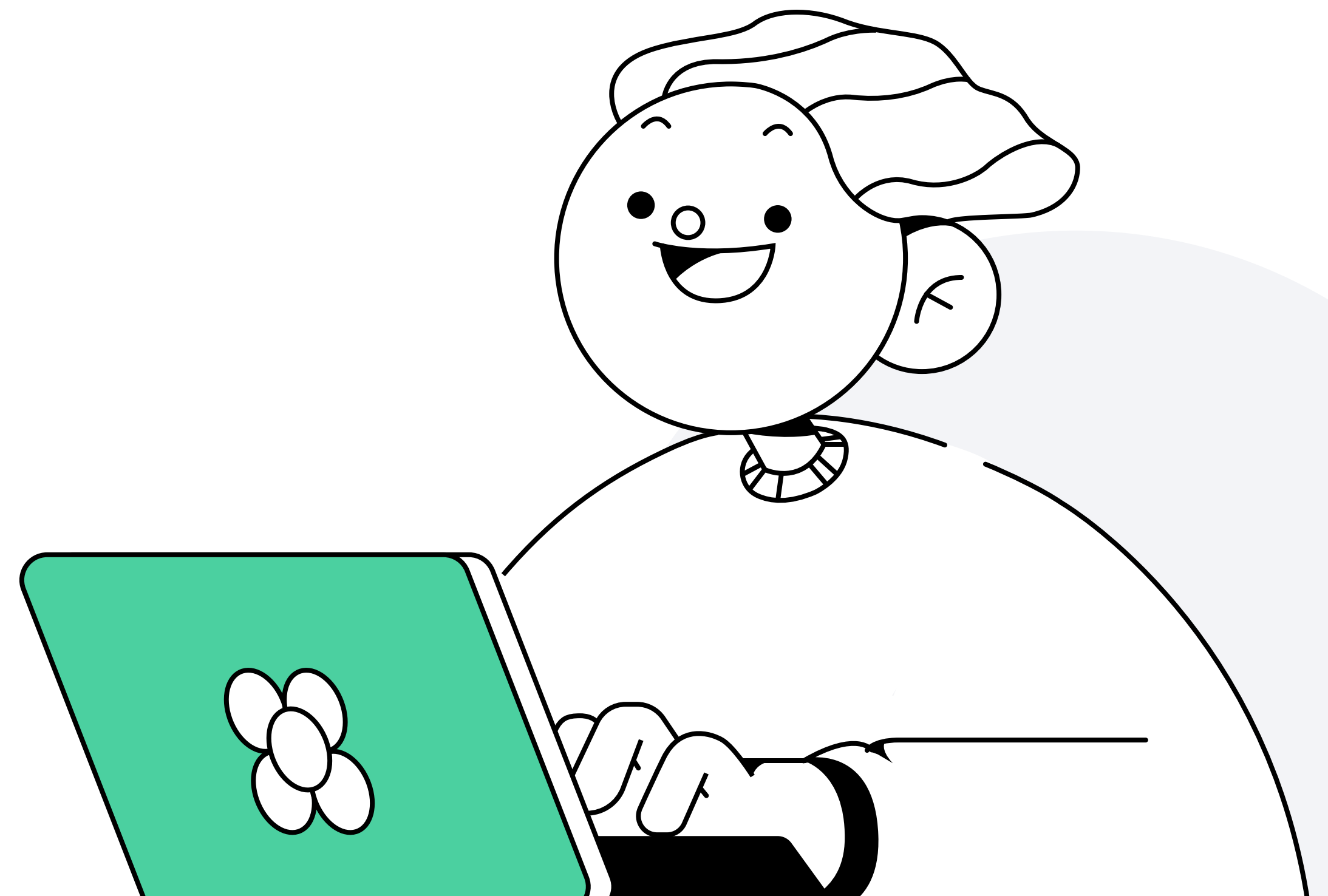
Python и БД. ORM

Александр Иванов
Senior backend engineer в OneSoil



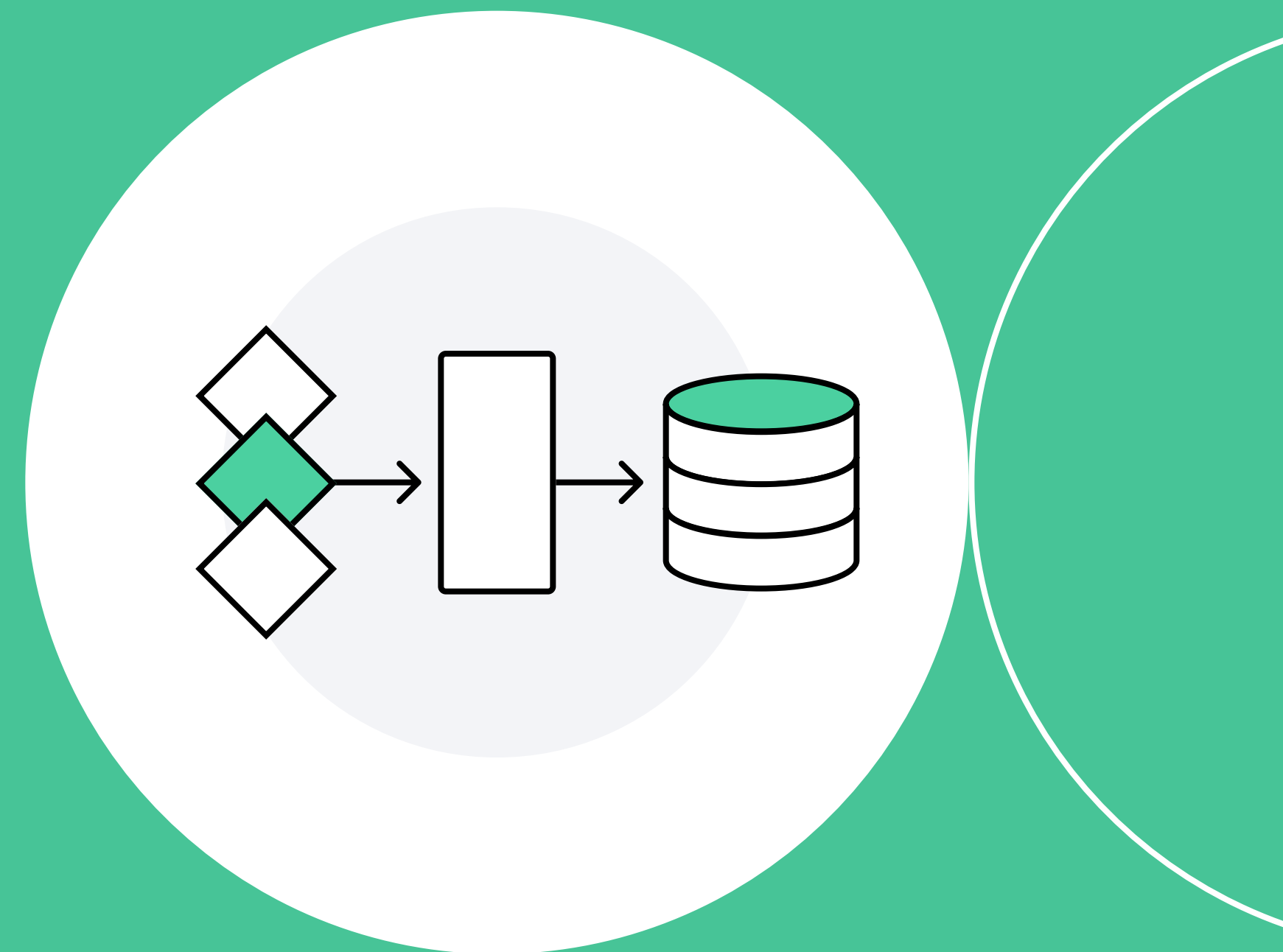
План занятия

- 1 Концепт ORM
- 2 Абстракции ORM



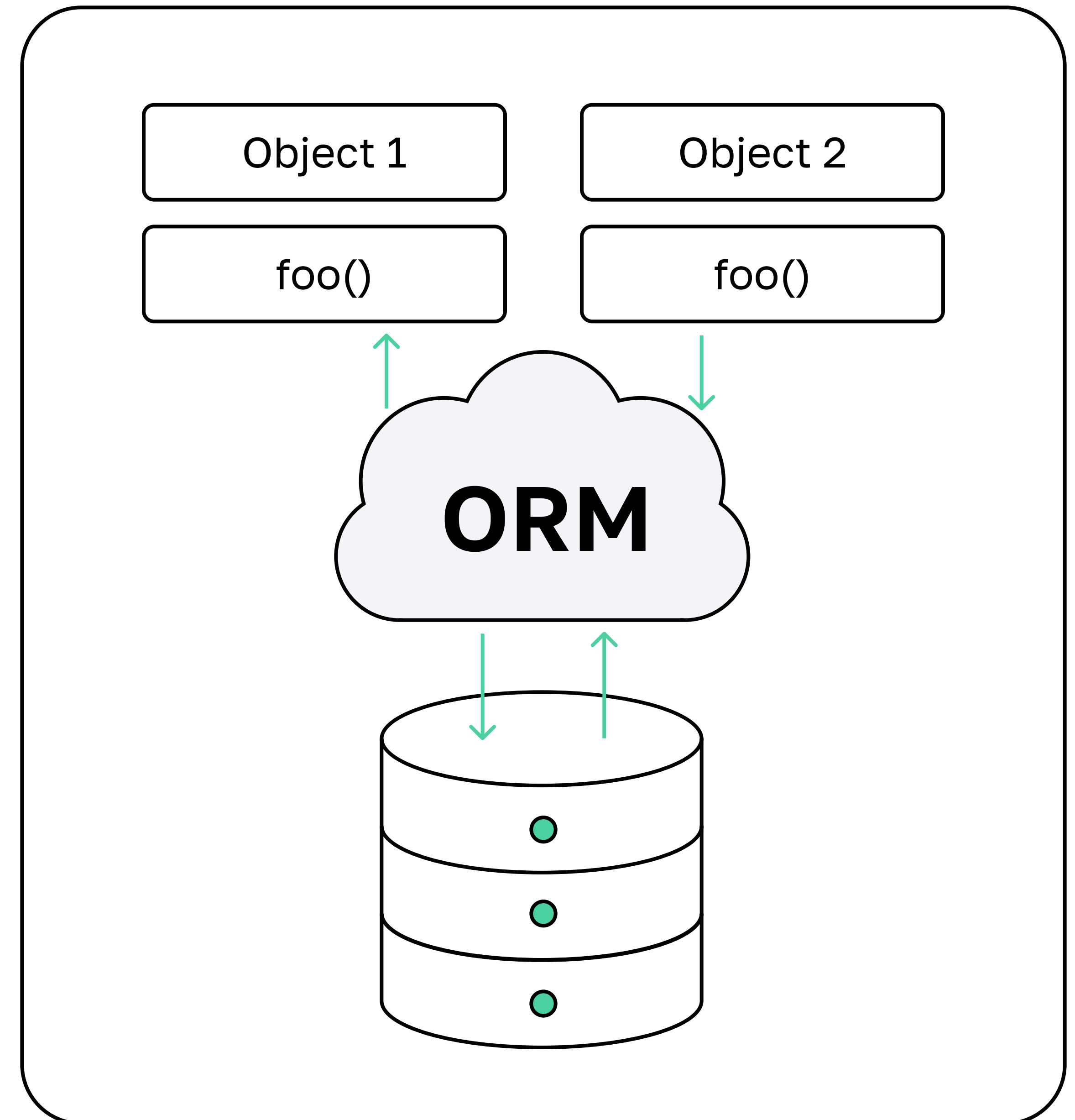
Концепт ORM

Александр Иванов
Senior backend engineer в OneSoil



Object-Relational Mapping

- ORM (объектно-реляционное отображение) — это дополнительный способ взаимодействия с БД из кода, который работает с таблицами и запросами к БД как с классами, объектами и методами в ООП



SQLAlchemy

- Популярная ORM для Python – SQLAlchemy.
pip install sqlalchemy

Таблица artist	
id	name
1	Queen
2	AC/DC
3	The Beatles

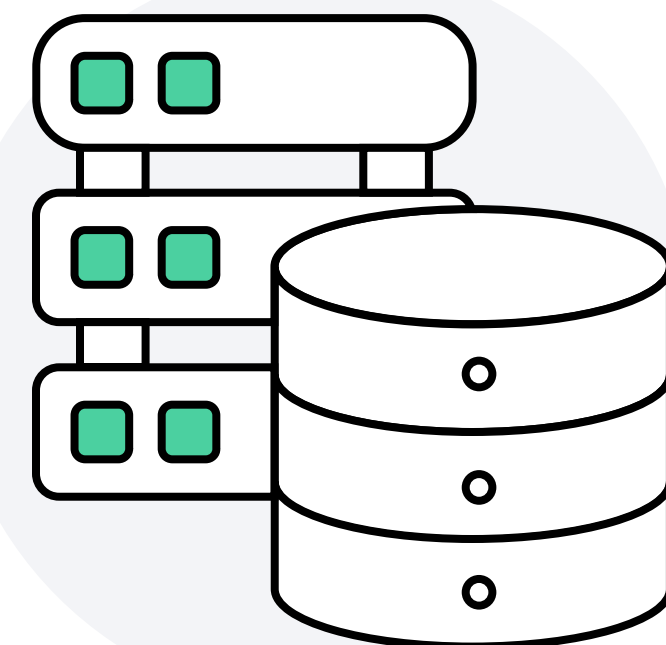
ORM
↔

```
import sqlalchemy as sq
from sqlalchemy.ext.declarative import declarative_base
Base = declarative_base()
class Artist(Base):
    __tablename__ = 'artist'
    id = sq.Column(sq.Integer, primary_key=True)
    name = sq.Column(sq.String)
queen = Artist(name='Queen')
ac_dc = Artist(name='AC/DC')
beatles = Artist(name='The Beatles')
```

- ! ORM работает поверх стандартных SQL-запросов.
Все правила и закономерности реляционных связей остаются

Абстракции ORM

Александр Иванов
Senior backend engineer в OneSoil



Плюсы и минусы использования ORM

- Не обязательно знать SQL и специальные функции СУБД
- Возможность десериализации (распаковки) результата из БД в удобном формате для API или обработки. Не нужно придумывать сложные парсеры массивов, работаем с готовыми структурами данных
- Один и тот же код может работать в разных БД, если на это рассчитана ORM
- Разработчиками ORM продуманы примитивные вопросы безопасности, экранирования и оптимизации запросов
- Изначально не очевидны итоговые запросы
- Ограничения и читаемость при построении сложных, многоуровневых запросов