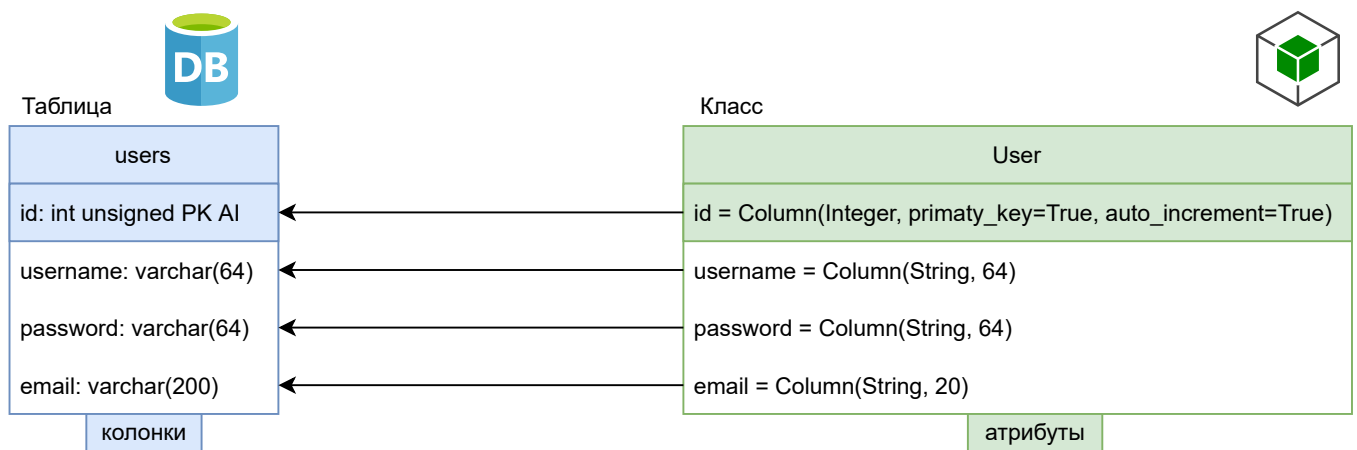


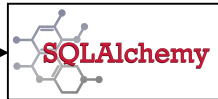
ORM



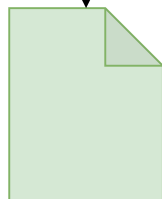
Установка зависимости

В pycharm Открываем терминал через Alt + F12

```
pip install sqlalchemy
```



import sqlalchemy



test_db.py

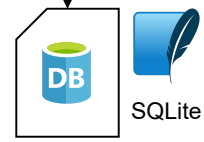
driver
нужно качать



driver
нужно качать



driver
встроенный



test.db

```
# Путь - `/test.db` (В текущей папке)
dsn = "sqlite:///test.db"

# Точка входа в БД.
engine = create_engine(dsn)
```



Engine

```
# Создаем декларативную основу для будущих классов
class Base(DeclarativeBase):
    pass
```

Создаем отображение класса на будущую таблицу

```
class User(Base):
    __tablename__ = "users"
    id = Column(Integer, primary_key=True, autoincrement=True)
    username = Column(String(64), unique=True, nullable=False) # nullable=False это NOT NULL
    password = Column(String(64), nullable=False)
    email = Column(String(200), unique=True, nullable=False)
```

Создаем таблицы в базе, используя engine

```
def create_tables():
    # Создаем таблицы, которые унаследованы от `Base`
    Base.metadata.create_all(engine)
```

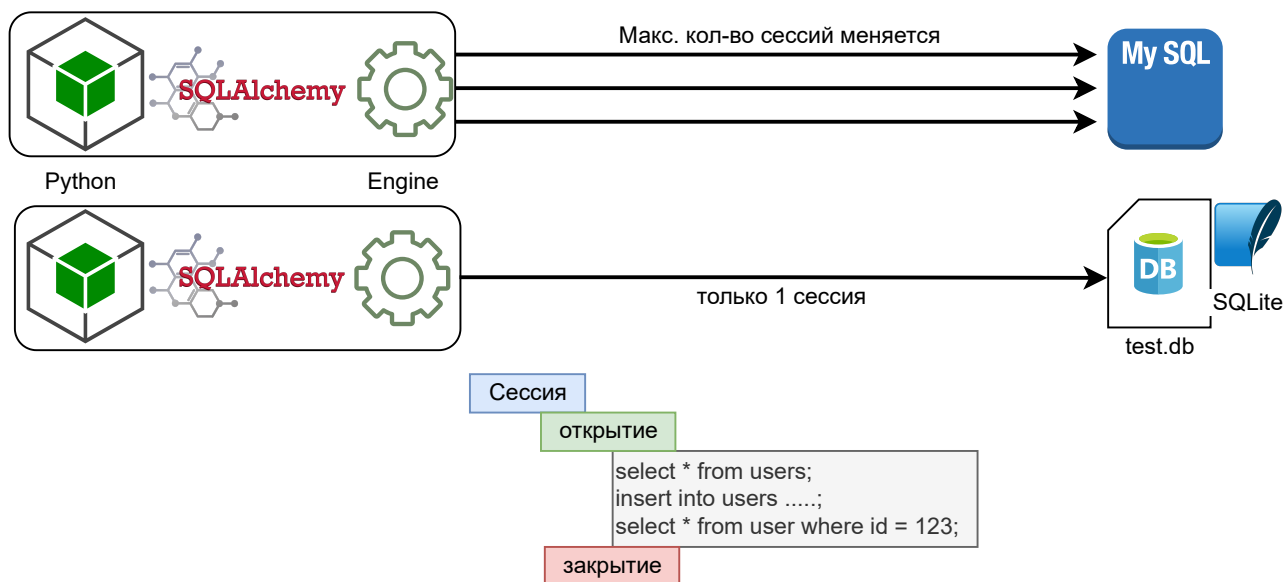
Engine



test.db



Сессии



Отношения моделей

```
class User(Base):
    __tablename__ = "users"
    id = Column(Integer, primary_key=True, autoincrement=True)
    # nullable=False это NOT NULL
    username = Column(String(64), unique=True, nullable=False)
    password = Column(String(64), nullable=False)
    email = Column(String(200), unique=True, nullable=False)

    # Далее внутренние связи для SQLAlchemy
    orders = relationship("Order", back_populates="user")
```

```
class Order(Base):
    __tablename__ = "orders"
    id = Column(Integer, primary_key=True, autoincrement=True)
    user_id = Column(Integer, ForeignKey("users.id"))
    product_id = Column(Integer, ForeignKey("products.id"))

    # Далее внутренние связи для SQLAlchemy
    product = relationship(Product, back_populates="orders")
    user = relationship(User, back_populates="orders")
```

```
class Product(Base):
    __tablename__ = "products"
    id = Column(Integer, primary_key=True, autoincrement=True)
    name = Column(String(64))
    price = Column(Float)

    # Далее внутренние связи для SQLAlchemy
    orders = relationship("Order", back_populates="product")
```

Название одинаковое

Название одинаковое

Название одинаковое

Название одинаковое