Для создания таблицы нужно:

1) Создать объект метадата

2) Создать таблицу

1 Создать объект метадата в файле моделс

From sqlalchemy import MetaData

metadata\_obj = MetaData()

2) Создать таблицу в файле моделс

from sqlalchemy import Table, Column, Integer, String

Workers\_table = Table(

‘workers’, - название таблицу

Metadata\_obj, - объект метадаты

Column(‘id’, Integer, primary\_key=True), - столбец, обязательно один должен быть первичным ключом

Column(‘username’, String),

)

metadata\_obj.create\_all(движок) – создаёт таблицу

metadata\_obj.drop\_all(движок) – удаляет таблицу

Для добавления данных в таблицу:

Есть два способа начало и конец одинаковый, отличается только запрос

From sqlalchemy import text

With engine.connect() as conn:

stmt = """INSERT INTO workers (username) VALUES

('AO Bobr'),

('OOO VOLK');"""

Conn.execut(text(stmt))

Conn.commit() – если чистый sql запрос

From sqlalchemy import select

from models import workers\_table

With engine.connect() as conn:

stmt = insert(workers\_table).values([

{"username": 'AO Bobr'},

{"username": 'OOO VOLK'},

])

Conn.execut(stmt)

Conn.commit() – если sqlalchemy запрос

Можно использовать create, update, insert, delete

Запросы create, update, insert, delete обычно называют stmt

Запросы select обычно называют query

Для получения данных используем select

def select\_workers():  
 with sync\_engine.connect() as conn:  
 query = select(workers\_table)  
 res = conn.execute(query)  
 print(res.first())

Функции обработки селекта

first() – первый результат

all() – Все результаты

one() – один

one\_or\_none – один либо ничего

для изменение используем update

def update\_workers(worker\_id: int = 1, new\_username: str = 'Boba'):  
 with sync\_engine.connect() as conn:  
 stmt = text("""UPDATE workers SET username=:username WHERE id=:worker\_id""")  
 stmt = stmt.bindparams(username=new\_username, worker\_id=worker\_id)  
 conn.execute(stmt)  
 conn.commit()

ef update\_workers(worker\_id: int = 1, new\_username: str = 'Boba'):  
 with sync\_engine.connect() as conn:  
 stmt = update(workers\_table).values(username=new\_username).filter\_by(id=worker\_id)  
 conn.execute(stmt)  
 conn.commit()