

START ML ORM

KARPOV.COURSES

ПРОБЛЕМА

- Приходится “собирать” запрос по кускам некоторым кодом
- В сложных запросах -> чаще hard-coded SQL
- “Неестественно” общаться с БД словарями (таблицы похожи на классы)
- Некрасиво мешать SQL и Python в одном файле
- *Создание и изменение таблиц не версионизируется*

ORM

Object-Relational Mapping. Пытаться решить проблему:

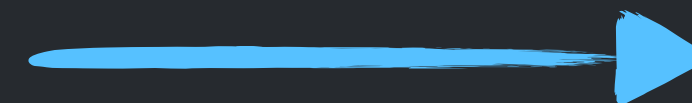
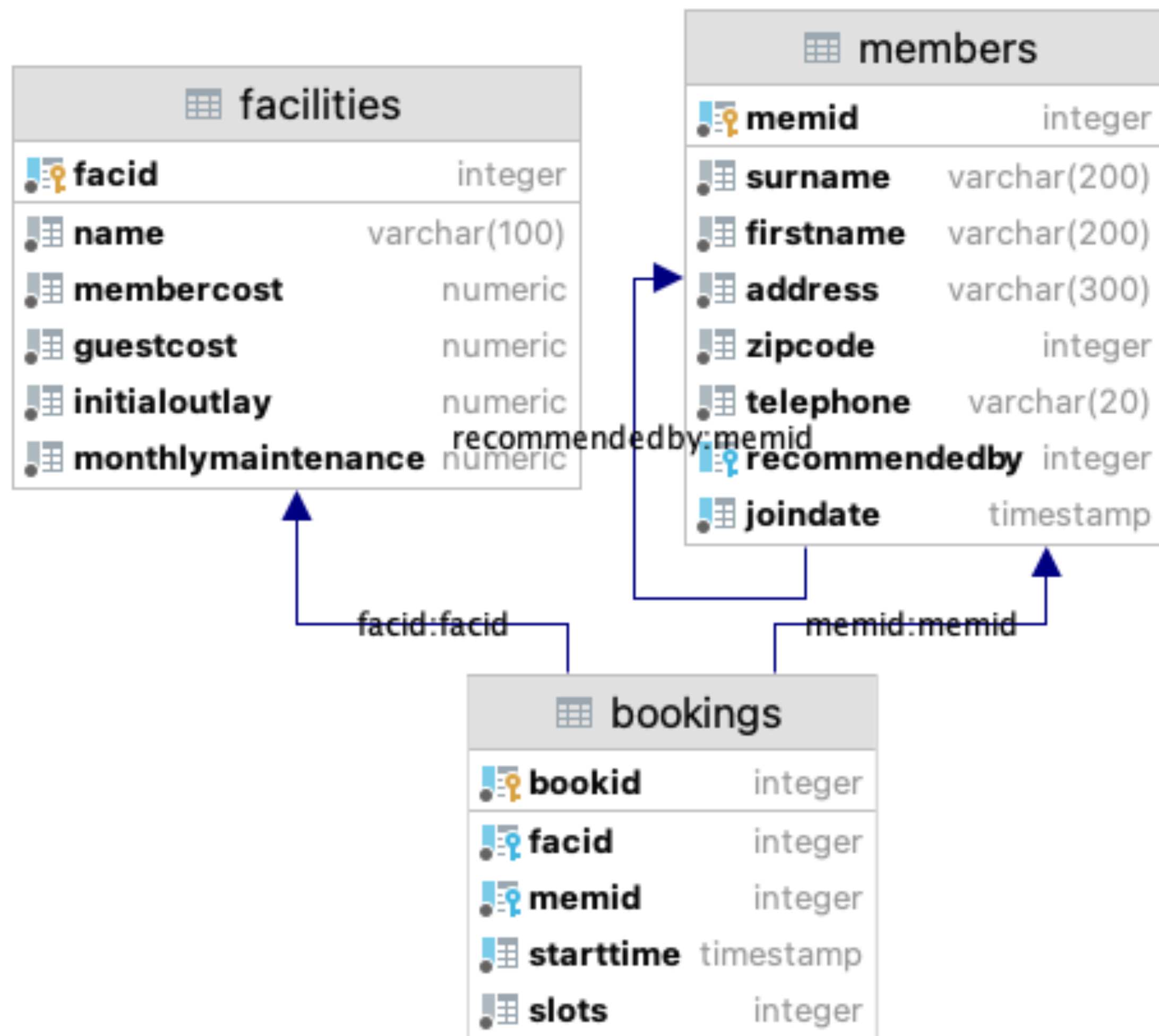
- Все таблицы в СУБД отображаются в классы Python
- Новый класс -> новая таблица в СУБД
- Объекты класса == записи в таблице
- Меняете объект - меняется соответствующая запись
- Прозрачная поддержка многих СУБД
- Но бывают медленнее сырых SQL (нужно руками тюнить)

SQLALCHEMY

Популярная ORM

- Имеет некоторый порог вхождения
- Самодостаточна, т.е. не зависит от кого-либо
- Зрелый инструмент: вероятность багов минимальна
- Имеет рядом тулзу для миграций Alembic

ИДЕЯ ORM



```
class Facility:
    facid = Column(Integer)
    name = Column(String)
    ...

class Member:
    memid = Column(Integer)
    surname = Column(String)
```

ПРИМЕРЫ

```
>>> stmt = select(User, Address).join(User.addresses).order_by(User.id, Address.id)

>>> for row in session.execute(stmt):
...     print(f"{row.User.name} {row.Address.email_address}")
spongebob spongebob@sqlalchemy.org
sandy sandy@sqlalchemy.org
sandy squirrel@squirrelpower.org
patrick pat999@aol.com
squidward stentcl@sqlalchemy.org
```

```
>>> stmt = (
...     select(User).
...     join(User.addresses.and_(Address.email_address != 'foo@bar.com'))
... )
>>> print(stmt)
```

```
SELECT user_account.id, user_account.name, user_account.fullname
FROM user_account
JOIN address ON user_account.id = address.user_id
AND address.email_address != :email_address_1
```