

Pony ORM позволяет работать с содержимым базы данных на уровне объектов. Основным отличием от других мапперов, таких как Django и SQLAlchemy является необычайно удобный синтаксис написания SQL запросов любой сложности на Питоне. При этом Пони транслирует их в SQL. Для того, чтобы пользоваться маппером, нужно сначала объявить сущности, которые отображаются на таблицы в базе данных, а затем сгенерировать маппинг. При этом Пони может либо установить маппинг с уже существующими таблицами в базе данных, либо сгенерировать таблицы для каждой сущности. Когда маппинг построен, можно выбирать существующие объекты из базы или создавать новые.

The Pony ORM let's you work with content databases at the object level. It differentiates itself fundamentally from other mappers, such as Django and SQLAlchemy, by its unusually convenient syntax for writing SQL queries of any complexity in Python. Pony does this by translating them into SQL. To use the mapper, you must first of all declare the entities displayed in the database tables, and then generate the mapping. Pony can either establish the mapping with already existing tables in the database, or generate tables for each entity. Once the mapping has been constructed, you can select existing objects from the database or create new ones.

Любой объектно-реляционный маппер имеет базовый функционал работы с объектами, который обычно называется аббревиатурой CRUD, расшифровывающейся как четыре операции Create/Read/Update/Delete, то есть создание объекта и сохранение его в базе данных, получение объекта из базы данных, изменение и удаление объекта.

Any object-relational mapper has the basic database functions for working with objects, generally indicated with the abbreviation CRUD, which stands for the four operations Create, Read, Update and Delete: creation of the object and saving it in the database, retrieval of the object from the database, changing the object, and deletion of the object.

В Пони также имеются все операции CRUD, синтаксис которых мы сейчас и рассмотрим.

Pony has all of the CRUD operations as well. Let us now consider their syntax.

Создание объекта (CREATE)  
Object creation (CREATE)

Создание объекта в Пони выглядит так же, как и создание обычного объекта в языке Питон, например:

The creation of an object in Pony takes place in the same manner as creating a normal object in the Python language. For example:

```
customer1 = Customer(login="John", password="***", name="John",
```

```
email="john@google.com")
```

При создании объекта в Пони все параметры нужно указывать как именованные аргументы. Если у атрибута задано значение по умолчанию, при создании объекта этот атрибут можно не указывать.

In Pony, all parameters must be specified as named arguments when the object is created. If a default value is set for an attribute, then that attribute need not be specified on object creation.

Любой инстанс сущности, который Пони создает в оперативной памяти, принадлежит текущей транзакции. Первоначально объект создается только в памяти программы, а для того чтобы сохранить его в базе данных, надо завершить транзакцию с помощью команды `commit()`.

Any instance of an entity, which Pony creates in operational memory, belongs to the current transaction. The initial object is created only in program memory, and in order to save it in the database, you must complete the transaction with the help of the “commit()” command.

В некоторых объектно-реляционных мапперах для сохранения объекта нужно вызывать его метод `save()`. Это неудобно, потому что программист должен сам отслеживать, какие объекты он создал или изменил и не забывать вызывать у них метод `save()`. Пони сама отслеживает, какие объекты были созданы или изменены во время выполнения транзакции, поэтому при использовании Пони не нужно вызывать `save` для каждого объекта - все изменения будут сохранены автоматически при завершении транзакции операцией `commit()`.

In some object-relational mappers, you must call an object's “save()” method in order to save it. This is inconvenient, because the programmer him/herself has to track which objects s/he has created or changed and not forget to call their “save()” method. Pony does the tracking itself, of which objects have been created or changed during run-time, so that with Pony you don't need to call `save` for each object – all changes are automatically saved on completion of the “commit()” transactional operation.