

Many

В предыдущих примерах мы везде использовали постфикс *_one*. Это значит, что Mongo выполняла операцию только с одним документом, даже если условию удовлетворяет несколько документов.

Но в Mongo доступны методы для получения/обновления/удаления множественных документов.

connectors/mongo/select_many.py

```
import asyncio
import os

from motor.motor_asyncio import AsyncIOMotorClient

def get_client():
    return AsyncIOMotorClient(os.getenv("MONGO_URL"))

def get_db():
    return get_client().metaclass

def get_user_collection():
    return get_db().user

async def insert_many():
    bulk = []
    is_tutor = False
    for i in range(15):
        bulk.append(
            {'first_name': f'name{i}', 'last_name': f'last_name{i}', 'is_tutor': is_tutor}
        )
        is_tutor = not is_tutor
    await get_user_collection().insert_many(bulk)

async def find_using_cursor():
    cursor = get_user_collection().find({'is_tutor': {'$eq': True}}).sort('first_name')
    for document in await cursor.to_list(length=5):
        print(document)

async def find_using_async_for():
    cursor = get_user_collection().find({'is_tutor': {'$eq': True}}).sort('first_name')
    async for document in cursor:
        print(document)

async def run():
    await insert_many()
    await find_using_cursor()
    await find_using_async_for()

if __name__ == "__main__":
    asyncio.run(run())
```

Запустить пример можно в [mercury](#):

```
python3 connectors/mongo/select_many.py
```

insert_many. В функцию можно передать массив словарей, и Mongo за один запрос вставит их все.

find_using_cursor и **find_using_async_for.** В Mongo, как и многих БД, есть [cursor](#). С его помощью можно получать не все записи, которые вернул запрос, а частями. Например, запрос вернул 1000 записей, а мы получили только 10 из них.

Доступны 2 синтаксиса:

- **cursor.to_list(length=5).** Тогда, даже если в запросе миллион записей, при вызове `to_list(length=5)` вернется только 5 записей. Мы сможем эффективно обработать их и при этом не заблокировать event loop.
- **async for document in cursor,** асинхронный итератор. По сути `async for` является [синтаксическим сахаром](#) для **cursor.to_list**. Он так же будет получать данные частями и предотвращать получение большого числа строк без возврата управления event loop.

Подробнее про множественные операции можно прочитать в [документации](#).