

# Базы данных

---

## Основы Веб-программирования

Кафедра Интеллектуальных Информационных Технологий, ИНФО, УрФУ

`http://lectures.uralbash.ru/6.www.sync/2.coding/9.databases/index.html`

**DB-API** - общий правила работы с БД.  
**реп-249**

- Connection
- Cursor
- Типы данных
- Исключения

**Connection** - устанавливает соединение с БД.

- **close()**- Закрывает соединение с базой данных.
- **commit()** - Завершает транзакцию.
- **rollback()** - Откатывает начатую транзакцию
- **cursor()** - Возвращает объект-курсор, использующий данное соединение. Если база данных не поддерживает курсоры, модуль сопряжения должен их имитировать.

## DB-API 2.0 - Connection. Пример

```
import sqlite3  
conn = sqlite3.connect('example.sqlite')
```

```
import sqlite3  
conn = sqlite3.connect(':memory:')
```

**Курсор** — объект базы данных, который позволяет приложениям работать с записями «по одной», а не сразу с множеством, как это делается в обычных SQL командах



- **execute()** - Закрывает соединение с базой данных.
- **executemany()** - Выполняет серию запросов или команд.
- **callproc()** - Откатывает начатую транзакцию
- **cursor()** - Вызывает хранимую процедуру.

## DB-API 2.0 - Cursor. Пример

```
c = conn.cursor()

# Создание таблицы
c.execute('''CREATE TABLE stocks
(date text, trans text,
symbol text, qty real, price real)''')

# Добавление записи
c.execute(
    "INSERT INTO stocks VALUES "
    "('2006-01-05', 'BUY', 'RHAT', 100, 35.14)"
)
```

## DB-API 2.0 - Cursor. Пример

```
# Сохранение (commit) изменений
conn.commit()

# Закрытие соединения.
# Если изменения не были сохранены метод( commit),
# то данные пропадут.
conn.close()
```

## DB-API 2.0 - Cursor. Пример

```
# Запись сразу нескольких объектов за раз
purchases = [
    ('2006-03-28', 'BUY', 'IBM', 1000, 45.00),
    ('2006-04-05', 'BUY', 'MSFT', 1000, 72.00),
    ('2006-04-06', 'SELL', 'IBM', 500, 53.00),
]
c.executemany(
    'INSERT INTO stocks VALUES (?, ?, ?, ?, ?)',
    purchases
)
```

- **fetchone()** - Возвращает следующую запись (в виде последовательности) из результата запроса.
- **fetchall()** - Возвращает все (или все оставшиеся) записи результата запроса.
- **fetchmany()** - Возвращает следующие несколько записей из результатов запроса.

## DB-API 2.0 - Cursor. Пример

```
# Никогда так не делайте — не безопасно!  
symbol = 'RHAT'  
c.execute(  
    "SELECT * FROM stocks WHERE symbol = '%s'" % symbol  
)  
  
# Правильно  
t = ('RHAT',)  
c.execute(  
    'SELECT * FROM stocks WHERE symbol=?',  
    t  
)  
print(c.fetchone())
```

- **Date**
- **Time**
- **Timestamp**
- **DateFromTicks**
- **TimeFromTicks**
- **Binary**

# DB-API 2.0 - Исключения

StandardError

└─Warning

└─Error

└─InterfaceError (a problem with the db api)

└─DatabaseError (a problem with the database)

└─DataError (bad data, values out of **range**, etc.)

└─OperationalError (the db has an issue out of o

└─IntegrityError

└─InternalError

└─ProgrammingError (something wrong with the ope

└─NotSupportedError (the operation **is not** support



## DB-API 2.0 - Кто использует

- SQLite
- PostgreSQL (psycopg2, txpostgres, ...)
- FireBird (fdb)
- MySQL (mysql-python, PyMySQL, ...)
- MS SQL Server (adodbapi, pymssql, mxODBC, pyodbc, ...)
- Oracle (cx\_Oracle, mxODBC, pyodbc, ...)
- и другие

<http://wiki.python.org/moin/DatabaseInterfaces>