

connect once, open a cursor, then every query travels through it

Connecting Python to MySQL

- the `mysql.connector` module gives Python the ability to talk to a MySQL database
- `import mysql.connector` brings the module in, it usually needs installing first with a package manager
- `connect()` opens the link: host, user, password and database are the four details it needs
- the object `connect()` returns represents the open connection itself, and stays needed until the program is done with the database

Opening a connection

Q. connect Python to a MySQL database named *school*, running on the local machine, user *root* with password *pwd*

- 1 import `mysql.connector` first
- 2 call `connect()` with host, user, password and database as keyword arguments
- 3 store the result, it is the connection object every later step needs

Answer

```
mydb =  
mysql.connector.connect(host="localhost",  
user="root", password="pwd",  
database="school")
```

The cursor object

- a cursor is what actually sends SQL to the database and brings results back, the connection alone cannot do that
- `mydb.cursor()` creates one, from an already open connection object
- `execute(sql)` hands one SQL statement to the cursor and runs it
- one connection can open more than one cursor, but a single cursor is enough for most simple programs



four stops in order: open the connection, open a cursor from it, run a query, then commit, that last stop only when the query changed data