

## [ 3-1. 파이썬과 데이터베이스 연결하기 ]

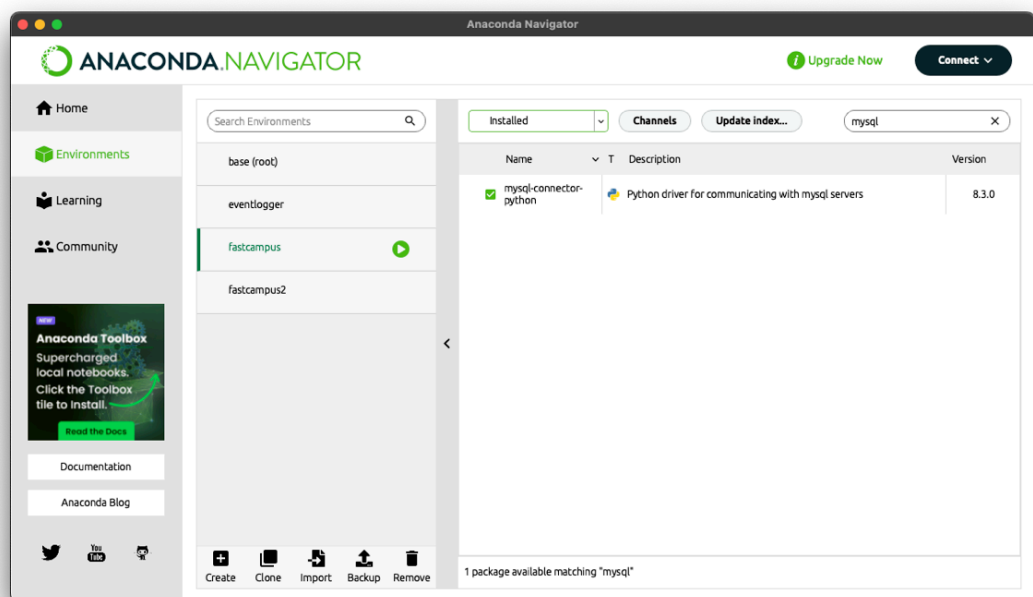
### 1. mysql.connector 라이브러리

#### 1-1. mysql.connector 라이브러리 소개

- mysql.connector는 Python에서 MySQL 서버와 상호 작용하기 위한 공식 MySQL 드라이버입니다.
- mysql.connector 라이브러리를 사용하면 MySQL 서버와의 연결을 설정하고 해제할 수 있으며, SQL 쿼리를 실행하여 데이터베이스에서 데이터를 가져오거나 업데이트 할 수 있습니다.

#### 1-2. mysql.connector 라이브러리 설치

- ANACONDA Environments에서 "mysql-connector-python" 설치



## 2. MySQL 연결하기

### 2-1. 연결

- mysql.connector 라이브러리의 connect 메서드에 연결 정보를 입력하여 connector 객체 생성.
- 연결 정보는 'host명', 'user명', 'port번호' 등이 있음.

### 2-2. 연결 해제

- 데이터베이스 연결은 시스템 자원을 사용하므로, 사용이 끝난 경우에는 꼭 연결을 해제해줘야 함.
- 연결을 해제하면 해당 자원이 다른 프로세스나 연결에 사용될 수 있게 됨.
- 연결을 해제하지 않고 계속 사용하면 메모리 누출과 같은 문제가 발생할 수 있음.

```
In [ ]: import mysql.connector
import pandas as pd

# MySQL 서버에 연결
# 호스트, 사용자 이름, 비밀번호, 데이터베이스 이름을 알맞게 변경하세요
connection = mysql.connector.connect(
    host="localhost",
    user="root",
    port="3306",
    password="fastcampus1!",
    database="cashflow"
)
cursor = connection.cursor(dictionary=True)

if connection.is_connected():
    print("MySQL 데이터베이스에 성공적으로 연결되었습니다.\n")

query = "SELECT * FROM accounts_info"
cursor.execute(query)
result = pd.DataFrame(cursor.fetchall())
print(result)

# 연결 종료
cursor.close()
connection.close()
```

MySQL 데이터베이스에 성공적으로 연결되었습니다.

	id	account_id	bank_name	account_number	account_holder	account_type	\
0	1	hana1234	하나은행	12345-01-1234	FC호텔	보통예	
금							
1	2	hana4321	하나은행	12345-01-4321	FC호텔	보통예	
금							
2	3	shhn4567	신한은행	54321-21-4567	FC호텔	정기예	
금							
	branch_name	account_alias	description		opening_date	closing_date	\
0	여의도점	매출계좌	매출대금	입금계좌	2024-01-02		Non
e							
1	여의도점	운영계좌	운영비용	지출 계좌	2024-01-02		Non
e							
2	여의도점	정기예금	10억원	1년 정기예금	2024-01-02		Non
e							
	created_at		deleted_at				
0	2024-04-10	23:33:35	None				
1	2024-04-10	23:33:35	None				
2	2024-04-10	23:33:35	None				

### 3. context manager를 이용하여 DB연결 클래스 만들기

## 3-1. Context Manager 소개

### 3-1-1. Context Manager 개요

- Context Manager는 파이썬에서 리소스의 할당 및 해제를 자동으로 관리하는 데 사용되는 기능입니다.
- Context Manager는 "with"문과 함께 사용되며, 일반적으로 파일, 네트워크 연결, 데이터베이스 연결과 같은 리소스의 생성 및 정리에 사용됩니다.
- Context Manager를 사용하면 리소스를 명시적으로 할당하고 해제하는 코드를 작성하지 않아도 됩니다. 대신, "with"문을 사용하여 리소스를 사용하고, "with" 블록이 종료될 때 자동으로 리소스가 정리됩니다.

### 3-1-2. Context Manager 정의하기

- Context Manager는 "**enter()**"와 "**exit()**" 두가지 메서드를 정의하는 방식으로 설정합니다.
- **enter()** 메서드 : with 블록이 실행되기 전에 호출되며, 리소스를 할당하고 리소스를 반환합니다. **enter()** 메서드가 반환하는 값은 as 키워드로 지정된 변수에 할당됩니다.
- **exit()** 메서드 : with 블록이 실행을 완료하거나 예외가 발생할 때 호출됩니다. 이 메서드는 예외 처리와 리소스의 정리를 담당합니다. 예외가 발생하지 않았을 경우 None을 반환하고, 예외가 발생했을 경우 True 또는 False 값을 반환하여 예외 처리를 결정할 수 있습니다.

## 3-2. Connector 클래스 만들기

```
In [ ]: import mysql.connector

class Connector:
    def __enter__(self):
        self.connection = mysql.connector.connect(
            host="localhost",
            user="root",
            port="3306",
            password="fastcampus1!",
            database="cashflow"
        )
        self.cursor = self.connection.cursor(dictionary=True)
        return self

    def __exit__(self, exc_type, exc_value, traceback):
        self.cursor.close()
        self.connection.close()
```

```
In [ ]: with Connector() as conn:
        query = "SELECT * FROM accounts_info"

        conn.cursor.execute(query)
        result = pd.DataFrame(conn.cursor.fetchall())
        print(result)
```

	id	account_id	bank_name	account_number	account_holder	account_type	\
0	1	hana1234	하나은행	12345-01-1234	FC호텔	보통예	
금							
1	2	hana4321	하나은행	12345-01-4321	FC호텔	보통예	
금							
2	3	shhn4567	신한은행	54321-21-4567	FC호텔	정기예	
금							

	branch_name	account_alias	description	opening_date	closing_date	\	
0	여의도점	매출계좌	매출대금	입금계좌	2024-01-02	Non	
e							
1	여의도점	운영계좌	운영비용	지출	계좌	2024-01-02	Non
e							
2	여의도점	정기예금	10억원	1년	정기예금	2024-01-02	Non
e							

	created_at	deleted_at
0	2024-04-10 23:33:35	None
1	2024-04-10 23:33:35	None
2	2024-04-10 23:33:35	None

## 4. Connector 모듈 만들기

### 4-1. Connector 모듈 작성

- connector.py 파일 생성
- Connector 클래스 내용 작성 후 저장

### 4-2. Connector 모듈 사용하기

```
In [ ]: from connector import Connector as Connector_py

        with Connector_py() as conn:
            query = "SELECT * FROM accounts_info"

            conn.cursor.execute(query)
            result = pd.DataFrame(conn.cursor.fetchall())
            print(result)
```

	id	account_id	bank_name	account_number	account_holder	account_type	\
0	1	hana1234	하나은행	12345-01-1234	FC호텔	보통예	
금							
1	2	hana4321	하나은행	12345-01-4321	FC호텔	보통예	
금							
2	3	shhn4567	신한은행	54321-21-4567	FC호텔	정기예	
금							

	branch_name	account_alias	description	opening_date	closing_date	\
0		여의도점	매출계좌	매출대금 입금계좌	2024-01-02	Non
e						
1		여의도점	운영계좌	운영비용 지출 계좌	2024-01-02	Non
e						
2		여의도점	정기예금	10억원 1년 정기예금	2024-01-02	Non
e						

	created_at	deleted_at
0	2024-04-10 23:33:35	None
1	2024-04-10 23:33:35	None
2	2024-04-10 23:33:35	None