



Microsoft Access와 Python을 활용한 데이터베이스 핸들링 정리

1. Microsoft Access란?

- Microsoft Access는 ****데이터베이스 관리 시스템(DBMS)****으로, Microsoft Office 제품군에 포함됩니다.
- **관계형 데이터베이스**로, 테이블을 이용해 데이터를 저장하고, 쿼리를 통해 데이터를 검색하며 관리할 수 있습니다.
- 소규모 데이터베이스 애플리케이션에 적합하며, 비즈니스 및 개인 프로젝트에서 데이터를 쉽게 관리할 수 있도록 도와줍니다.

2. 주요 기능

- **테이블 관리**: 여러 개의 테이블을 생성하여 데이터를 저장하고 관리.
- **쿼리**: 데이터를 조회하거나 검색할 수 있는 기능. SQL(Structured Query Language)을 사용하여 복잡한 쿼리 실행 가능.
- **폼(Form)**: 데이터 입력과 조회를 쉽게 하기 위한 사용자 인터페이스.
- **보고서(Report)**: 데이터를 요약하고 시각적으로 보고할 수 있는 도구.
- **매크로 및 VBA**: 자동화된 작업을 설정하거나 복잡한 프로그래밍 기능 추가 가능.

3. Python으로 Access 핸들링 (ODBC 사용)

Python을 이용하여 Access 데이터베이스를 제어하기 위해 **ODBC** 드라이버를 통해 연결할 수 있습니다.

4. **pyodbc** 라이브러리 설치

먼저, ODBC를 사용하기 위해 **pyodbc** 라이브러리를 설치해야 합니다.

```
pip install pyodbc
```

5. Access 데이터베이스 연결 및 데이터 처리 예시

```

import pyodbc

# Access 파일 경로 설정
database_path = r'C:\path\to\your\database.accdb'

# ODBC 연결 문자열
conn_str = (
    r'DRIVER={Microsoft Access Driver (*.mdb, *.accdb)};'
# ODBC 드라이버
    rf'DBQ={database_path};' # Access 파일 경로
)

# 데이터베이스 연결
conn = pyodbc.connect(conn_str)

# 커서 생성
cursor = conn.cursor()

# 쿼리 실행
cursor.execute("SELECT * FROM TableName")

# 결과 출력
for row in cursor.fetchall():
    print(row)

# 연결 종료
cursor.close()
conn.close()

```

6. 주요 단계 요약

1. **ODBC 드라이버 설치:** Access ODBC 드라이버가 Windows에 설치되어 있어야 함.
2. **파일 경로 지정:** Access 파일의 경로를 지정하여 연결.
3. **SQL 쿼리 실행:** SQL을 사용하여 데이터 검색 또는 조작.
4. **결과 처리:** `cursor.fetchall()` 을 사용해 데이터를 가져와 처리.

7. 활용 가능 작업

- **데이터 조회:** `SELECT` 쿼리를 통해 테이블에서 데이터를 검색.
- **데이터 삽입:** `INSERT INTO` 를 사용해 데이터를 추가.
- **데이터 수정:** `UPDATE` 쿼리를 통해 데이터 업데이트.
- **데이터 삭제:** `DELETE` 쿼리로 데이터를 삭제.

이를 통해 Python과 Microsoft Access를 연동하여 데이터베이스 작업을 자동화하거나 관리할 수 있습니다.