

## fooddb 세팅 및 실행 과정 정리

### MySQL 데이터베이스 생성

```
mysql -u root -p
CREATE DATABASE fooddb CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
SHOW DATABASES;
exit;
```

- ◆ **역할:** 한글까지 저장 가능한 `fooddb` 데이터베이스 생성

### SQL 스크립트(ranking.sql) 실행

```
mysql --default-character-set=utf8mb4 -u root -p fooddb < "C:\food_project\ranking.sql"
```

- ◆ **역할:** `ranking.sql` 안의 테이블 및 초기 데이터 자동 등록  
(에러 없이 넘어가면 성공)

### Python 코드(food\_db.py) 연결 설정

```
conn = pymysql.connect(
    host='localhost',
    user='root',
    password='비밀번호',
    db='fooddb',
    charset='utf8mb4',
    cursorclass=pymysql.cursors.DictCursor
)
```

- ◆ **역할:** Python과 MySQL 연결  
(`pymysql` 라이브러리 사용)

### 실행 및 확인

#### CMD나 VS Code 터미널에서

```
python food_db.py
```

- ◆ **정상일 때 출력:** - "연결 성공" - 또는 SQL 결과(테이블 데이터)

---

## 오류 발생 시 빠른 점검표

상황	해결 방법
파일을 찾을 수 없습니다	<code>ranking.sql</code> 경로 확인 또는 <code>"</code> 로 감싸기
Incorrect string value (한글 깨짐)	<code>--default-character-set=utf8mb4</code> 옵션 사용
Unknown database	<code>CREATE DATABASE fooddb;</code> 다시 실행
연결 실패 (Access denied)	MySQL 계정/비밀번호 확인
Python에서 한글 깨짐	<code>charset='utf8mb4'</code> 추가

---

## 최종 구조 요약

```
C:\food_project
├─ food_db.py
├─ ranking.sql
└─ (MySQL 내부에 fooddb)
```

---

## 완성 후 실행 순서 요약

1. MySQL 실행
2. `fooddb` 생성
3. `ranking.sql` 적용
4. `python food_db.py` 실행