

# 데이터베이스 처리

- I. 데이터베이스 개념
- II. 데이터베이스 구축
- III. 데이터 입력과 조회

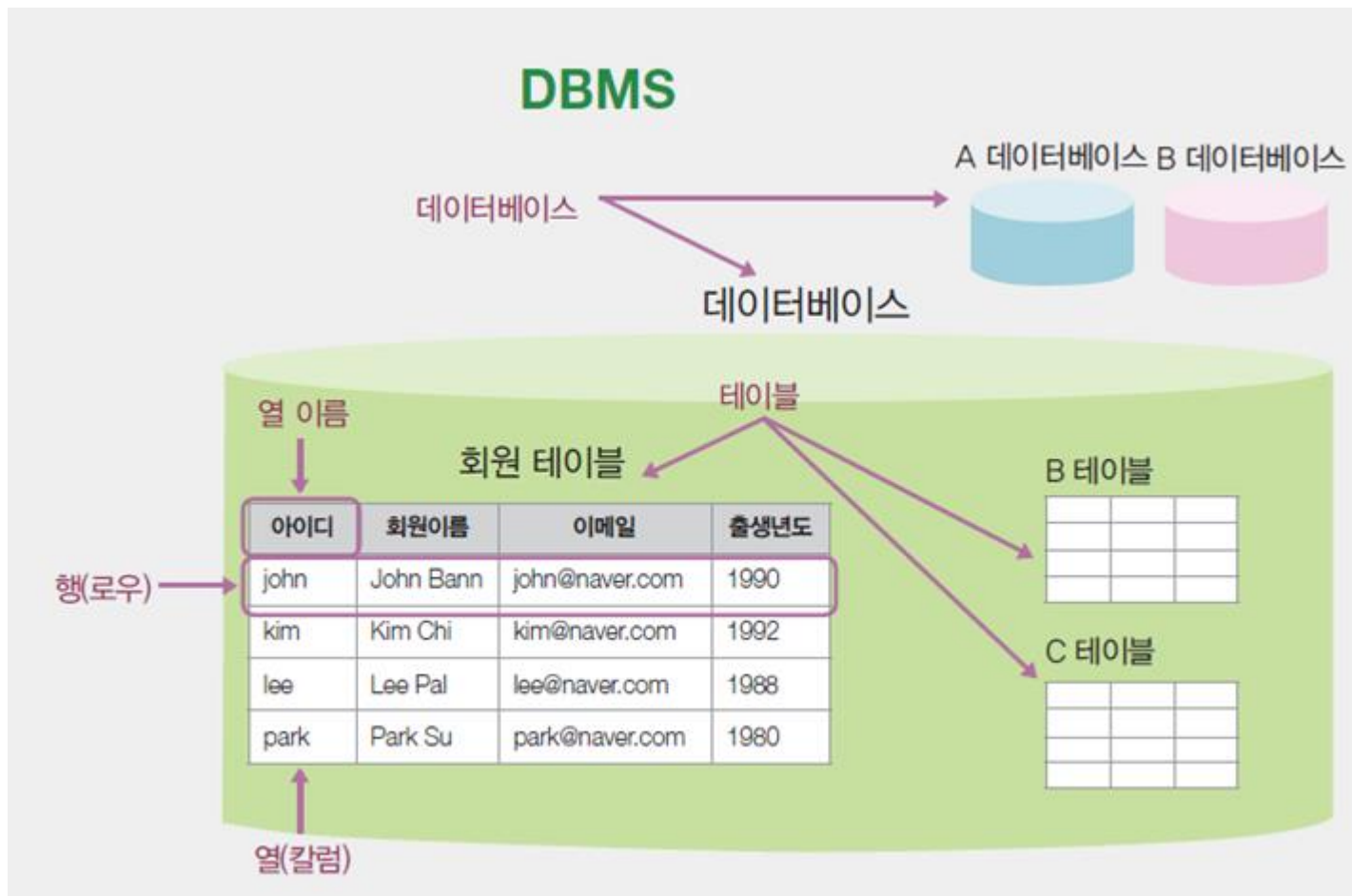
## ❖ 데이터베이스 이해

- 대량의 데이터가 발생하는 현대 사회에서 데이터베이스를 사용하면 수십억, 수백억 건의 데이터를 저장하고 무리 없이 운영할 수 있음
- 데이터베이스 소프트웨어(DBMS) - Oracle, SQL Server, MySQL, Access, SQLite 등

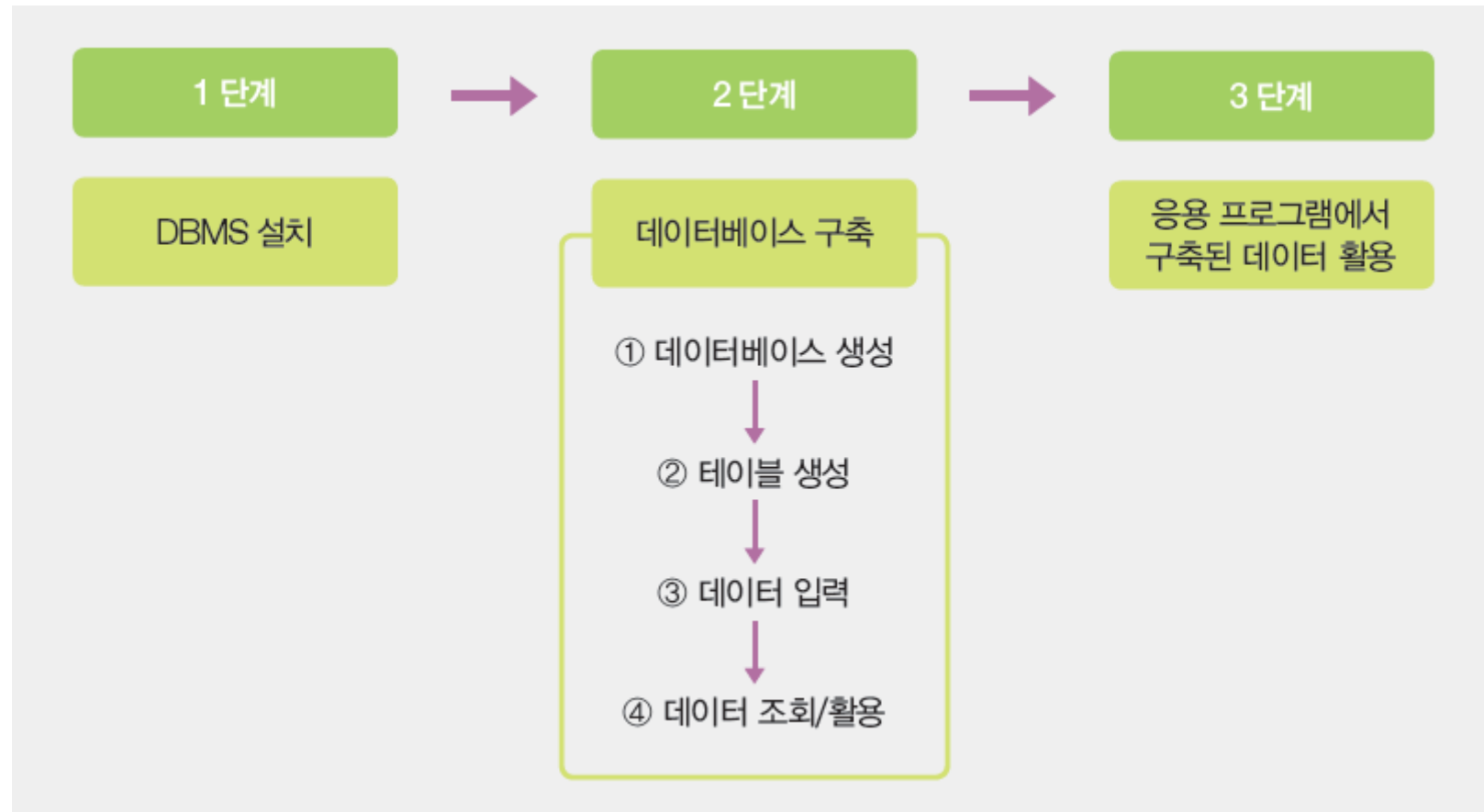
## ❖ 데이터베이스 기본 기법

- 데이터베이스 정의
  - ✓ 대량의 데이터를 체계적으로 저장해 여러 사용자가 서로 공유할 수 있는 시스템
- 관계형 데이터베이스
  - ✓ DBMS는 계층형(Hierarchical), 망형(Network), 관계형(Relational), 객체지향형(Object-Oriented), 객체관계형(Object-Relational) 등의 유형으로 나뉨
  - ✓ 그 중 관계형(Relational) DBMS가 가장 많이 사용됨

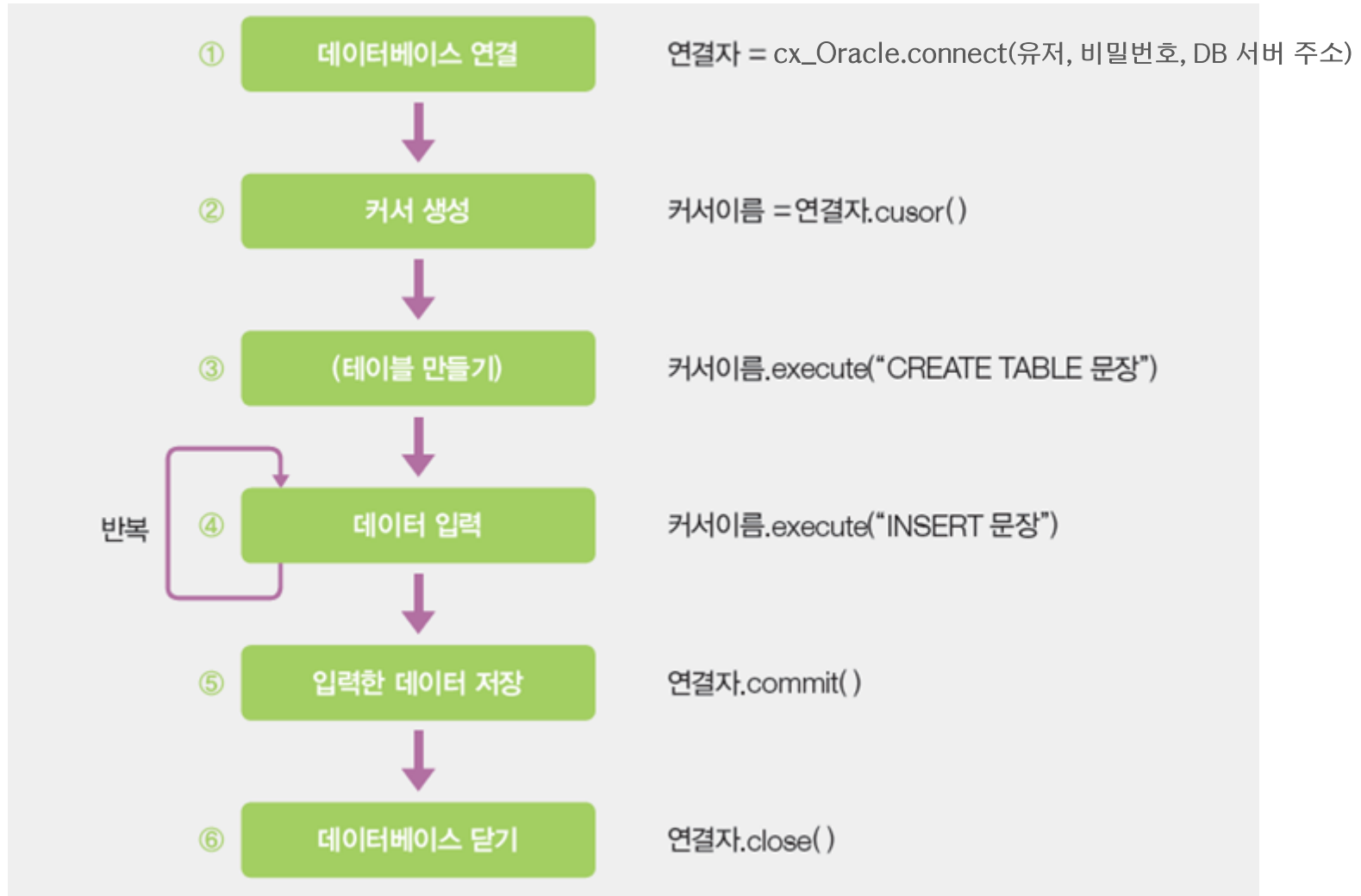
## ❖ DBMS 구성도



### ❖ 데이터베이스 구축 및 운영 절차



## ❖ 파이썬에서 데이터 입력하는 코딩 순서



- 오라클 db 연동을 위한 모듈 다운로드 및 설치 (윈도우 키 + R)

- ✓ `python -m pip install cx_Oracle --upgrade`
- ✓ `pip install cx_Oracle`

- ① 데이터베이스 연결

- ✓ 오라클을 사용하기 위해 관련 모듈인 cx\_Oracle를 임포트한 후,  
`cx_Oracle.connect('유저', '비밀번호', '데이터베이스 서버 주소')`으로 데이터베이스와 연결

```
import cx_Oracle
con = cx_Oracle.connect('c##user1', 'master', 'localhost/orcl')
```

- `con = cx_Oracle.connect('c##user1/master@localhost/orcl')` 도 동일함

- ② 커서 생성

- ✓ 커서(Cursor)는 데이터베이스에 SQL문을 실행 또는 실행 결과를 돌려받는 통로
- ✓ ①에서 연결한 연결자에 커서를 만들어야 함

```
cur = con.cursor()
```

#### ■ ③ 테이블 만들기

- ✓ 테이블을 만드는 SQL문을 커서이름.execute() 함수의 매개변수로 넘겨주면, SQL문이 데이터베이스에 실행

```
cur.execute("CREATE TABLE userTable (id char(10), userName char(15), email char(15), birthYear int)")
```

#### ■ ④ 데이터 입력

- ✓ 데이터 입력도 SQL문을 사용해야 하므로 커서이름.execute() 함수를 사용
- ✓ 데이터 입력은 필요한 만큼 반복함

```
cur.execute("INSERT INTO userTable VALUES('john', 'John Bann', 'john@naver.com', 1990)")
cur.execute("INSERT INTO userTable VALUES('kim', 'Kim Chi', 'kim@daum.', 1992)")
cur.execute("INSERT INTO userTable VALUES('lee', 'Lee Pal', 'lee@paran.com', 1988)")
cur.execute("INSERT INTO userTable VALUES('park', 'Park Su', 'park@gmail.com', 1980)")
```



- ⑤ 입력한 데이터 저장

- ✓ ④에서 입력한 4건의 데이터는 임시로 저장된 상태임
- ✓ 이를 확실하게 저장하는 것을 커밋(Commit)이라고 함

```
con.commit( )
```

- ⑥ 데이터베이스 닫기

```
con.close( )
```

## ❖ 데이터 입력 프로그램 구현

```
# C:/doit/dbora_insert.py
import cx_Oracle
import os
os.putenv('NLS_LANG', '.UTF8')      ← 한글 지원 방법

## 변수 선언 부분 ##
con, cur = None, None
data1, data2, data3, data4 = "", "", "", ""
sql = ""

## 메인 코드 부분 ##
con = cx_Oracle.connect('c##user1', 'master', 'localhost/orcl') ← 데이터베이스 연결
cur = con.cursor()          ← 커서 생성

while (True):               ← 무한 반복하면서 data1~data4 입력
    data1 = input("사용자ID ==> ")
    if data1 == "":
        break;              ← 입력할 데이터가 없으면, while문 빠져나옴
    data2 = input("사용자이름 ==> ")
    data3 = input("이메일 ==> ")
    data4 = input("출생년도 ==> ")

    sql = "INSERT INTO userTable VALUES('" + data1 + "', '" + data2 + "', '" + data3 + "', '" + data4 + "')"
    cur.execute(sql)         ← 입력한 데이터를 INSERT 문으로 sql 변수에 문자열로 만들

con.commit()                ← 입력한 데이터 저장
con.close()                 ← 연결된 데이터베이스 닫음
```

## ❖ 파이썬에서 데이터 조회하는 코딩 순서



- ③에서는 커서에 SELECT로 조회한 결과를 한꺼번에 저장
- ④에서 조회한 데이터를 한 행씩 fetchone() 함수로 접근한 다음 출력

## ❖ 데이터 조회 프로그램 구현

```
# C:/doit/dbora_select.py
import cx_Oracle
import os
os.putenv('NLS_LANG', '.UTF8')  ← 한글 지원 방법

## 변수 선언 부분 ##
con, cur = None, None
data1, data2, data3, data4 = "", "", "", ""
row = None

## 메인 코드 부분 ##
con = cx_Oracle.connect('c##user1', 'master', 'localhost/orcl')  ← 데이터베이스 연결
cur = con.cursor()  ← 커서 생성

cur.execute("SELECT * FROM userTable")  ← SELECT 문으로 테이블을 조회한 결과는 cur 변수에 들어감
print("사용자ID   사용자이름   이메일   출생연도")
print("-----")

while (True):
    row = cur.fetchone()  ← 조회한 결과의 모든 행을 한 행씩 추출.
    if row == None:  ← fetchone() 함수로 조회된 결과가 저장된 row 변수는 튜플 형식으로 각 행 데이터가 저장
        break;  ← 추출한 데이터 없으면, while문 빠져나옴
    data1 = row[0]
    data2 = row[1]
    data3 = row[2]
    data4 = row[3]
    print("%5s %15s %15s %d" % (data1, data2, data3, data4))  ← 데이터 출력

con.close()  ← 연결된 데이터베이스 닫음
```