

[Opensource Basic Project]

오픈소스 기초 프로젝트

- 10주차 : Database

2024. 5. 22(수).

우창우

Dr.woo@chungbuk.ac.kr

□ 주차별 강의계획

※ 4.10(국회의원선거날)

주차(변경 전)	주차(변경 후)	강의계획	과제	번호	제출 마감일
1주(03.06)	1주(03.06)	Python Started	VScode 설치화면, "Hello World" 출력화면 및 작성코드	1	03.06~03.09(토)
2주(03.13)	2주(03.13)	Variables and simple data types	다이아몬드 모양, 진수 변환 프로그램 및 동전 변환 프로그램의 출력화면/작성코드	2	03.13~03.16(토)
3주(03.20)	3주(03.20)	Control Statement	종합계산기, 구구단 출력 프로그램의 출력화면/작성코드	3	03.20~03.23(토)
4주(03.27)	4주(03.27)	Lists, Dictionaries, String	음식 궁합 출력, 문자열 거꾸로 출력 프로그램의 출력화면/작성코드	4	03.27~03.30(토)
5주(04.03)	5주(04.03)	Functions and module	GitHub Copilot Extension 설치화면, 로또번호 추첨 출력화면/작성코드	5	04.03~04.06(토)
6주(04.10)	6주(04.17)	Classes	팀프로젝트 과제 제출양식.hwp (조당 1개)	6	04.17~04.20(토)
7주(04.17)	7주(04.24)	Window programming	사진앨범 프로그램의 출력화면/작성코드	7	04.24~04.27(토)
8주(04.24)	8주(05.01)	Midterm (반영비율 20%)	-	-	-

□ 주차별 강의계획

※5.15(부처님오신날)

주차(변경 전)	주차(변경 후)	강의계획	과제	번호	제출 마감일
9주(05.01)	9주(05.08)	Files and Exceptions	사칙연산문제지 10개 만들기 프로그램 출력화면/작성코드	8	05.08~05.11(토)
10주(05.08)	10주(05.22)	Database	파이썬에서 데이터베이스(SQL Lite)를 입력하고 조회하는 프로그램	9	05.22~05.25(토)
11주(05.15)	11주(05.29)	Data visualization		10	05.29~06.01(토)
12주(05.22)	12주(06.05)	Quiz (반영비율 10%), Team project implementation I			
13주(05.29)	13주(06.12)	Team project implementation II			
14주(06.05)	14주(06.17)	Team project implementation III			
15주(06.12)	15주(06.18)	Project Presentation (반영비율 50%)			



Chapter 13. 데이터베이스

Selection 01. 이 장에서 만들 프로그램

Selection 02. 데이터베이스의 기본

Selection 03. 데이터베이스의 구축

Selection 04. 데이터의 입력과 조회

Selection 01. 이 장에서 만들 프로그램

1. 데이터베이스에 연결하고 입력/조회하는 프로그램

GUI 데이터 입력

사용자ID	사용자이름	이메일	출생연도
john	john Bann	john@naver.com	1990
kim	Kim Chi	kim@daum.net	1992
lee	Lee Pal	lee@paran.com	1988
park	Park Su	park@gmail.com	1980
su	Su Ji	suji@naver.com	1994
ko	Ko Jin	kojin@naver.com	1996
hong	hong gil	hong@naver.com	1986

Selection 02. 데이터베이스의 기본

1. 데이터베이스의 개념

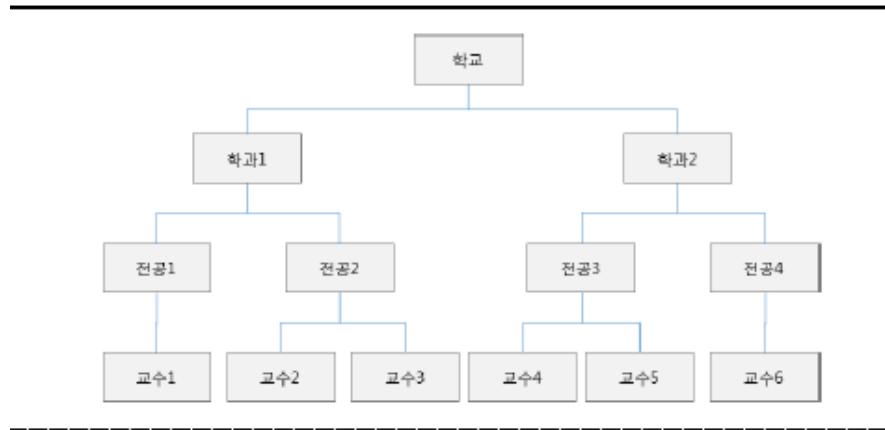
- 여러 사람에 의해 공유되어 사용될 목적으로 통합하여 관리되는 데이터의 집합
- 데이터베이스 소프트웨어 (Database Management System)의 종류
 - MySQL, MariaDB, PostgreSQL, Oracle, SQL Server, DB2, SQLite 등

DBMS	제작사	작동 운영체제	기타
MySQL	Oracle	Unix, Linux, Windows, Mac	오픈 소스(무료), 상용
MariaDB	MariaDB	Unix, Linux, Windows	오픈 소스(무료), MySQL 초기 개발자들이 독립해서 만들
PostgreSQL	PostgreSQL	Unix, Linux, Windows, Mac	오픈 소스(무료)
Oracle	Oracle	Unix, Linux, Windows	상용 시장 점유율 1위
SQL Server	Microsoft	Windows	주로 중/대형급 시장에서 사용
DB2	IBM	Unix, Linux, Windows	메인프레임 시장 점유율 1위
Access	Microsoft	Windows	PC용
SQLite	SQLite	Android, iOS	모바일 전용, 오픈 소스(무료)

Selection 02. 데이터베이스의 기본

2. DBMS의 분류

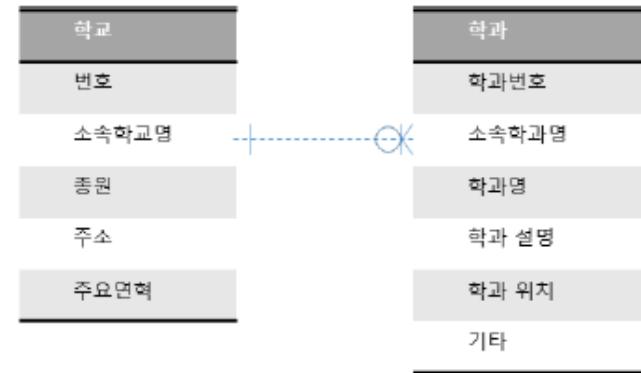
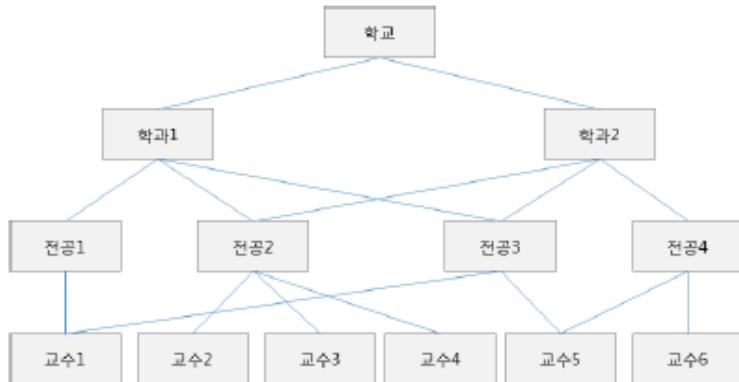
- 관계형 (Relational), 계층형 (Hierarchical), 망형 (Network), 객체지향형 (Object-Oriented), 객체관계형 (Object-Relational) 등
- 계층형 DBMS : 가장 오래된 DBMS의 개념으로 각 계층이 트리형태를 갖고 연결되어 있는 구조로 현재는 사용하지 않음



Selection 02. 데이터베이스의 기본

2. DBMS의 분류

- **망형 DBMS**: 각 계층 하위에 있는 요소들간에 연결되는 유연한 구조로 구성하여 계층형 DBMS의 한계를 극복하고자 했지만 모든 구조를 이해해야 하는 문제가 있음
- **관계형 DBMS**: 가장 일반적으로 사용되는 DBMS의 종류로 줄여서 R-DBMS라고 하며, 테이블로 이루어져 있고 테이블은 열과 행으로 구성되어 있음



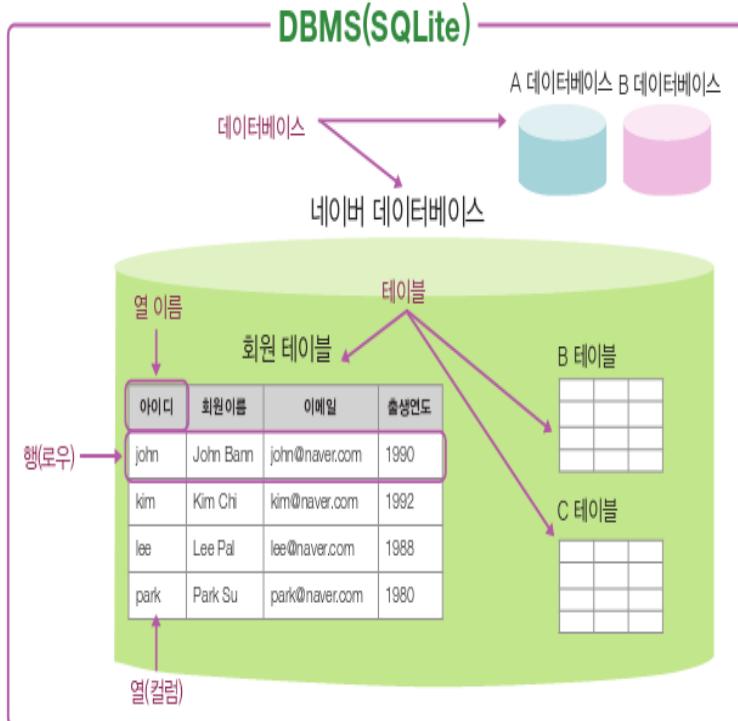
Selection 02. 데이터베이스의 기본

❖ DB-Engines Ranking ('24.5)

Rank			DBMS	Database Model	Score		
May 2024	Apr 2024	May 2023			May 2024	Apr 2024	May 2023
1.	1.	1.	Oracle	Relational, Multi-model	1236.29	+2.02	+3.66
2.	2.	2.	MySQL	Relational, Multi-model	1083.74	-3.99	-88.72
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	824.29	-5.50	-95.80
4.	4.	4.	PostgreSQL	Relational, Multi-model	645.54	+0.49	+27.64
5.	5.	5.	MongoDB	Document, Multi-model	421.65	-2.31	-14.96
6.	6.	6.	Redis	Key-value, Multi-model	157.80	+1.36	-10.33
7.	7.	8.	Elasticsearch	Search engine, Multi-model	135.35	+0.57	-6.28
8.	8.	7.	IBM Db2	Relational, Multi-model	128.46	+0.97	-14.56
9.	9.	11.	Snowflake	Relational	121.33	-1.87	+9.61
10.	10.	9.	SQLite	Relational	114.32	-1.69	-19.54
11.	11.	10.	Microsoft Access	Relational	104.92	-0.49	-26.26
12.	12.	12.	Cassandra	Wide column, Multi-model	101.89	-1.97	-9.25
13.	13.	13.	MariaDB	Relational, Multi-model	93.21	-0.60	-3.66
14.	14.	14.	Splunk	Search engine	86.45	-2.26	-0.19
15.	17.	18.	Databricks	Multi-model	78.61	+2.28	+14.66
16.	15.	16.	Microsoft Azure SQL Database	Relational, Multi-model	77.99	-0.41	-1.21
17.	16.	15.	Amazon DynamoDB	Multi-model	74.07	-3.50	-7.04
18.	18.	17.	Hive	Relational	61.17	-1.41	-12.44
19.	19.	20.	Google BigQuery	Relational	60.38	-1.52	+5.51
20.	20.	21.	FileMaker	Relational	48.20	-1.53	-3.80
140.	143.	141.	Tibero	Relational	1.78	-0.03	-0.63

Selection 02. 데이터베이스의 기본

3. 데이터베이스 관련 용어



- 데이터 : john, lee@naver.com, 1980 등 단편적인 정보
- 테이블 : 회원 데이터가 표 형태로 표현된 것
- 데이터베이스(DB) : 테이블이 저장되는 저장소
- DBMS : 데이터베이스 관리 시스템 또는 소프트웨어
- 열(컬럼 또는 필드) : 각 테이블은 1 개 이상의 열로 구성
- 열 이름 : 각 열을 구분하는 이름
- 데이터 형식 : 테이블을 생성할 때 열 이름과 함께 지정
- 행(로우) : 실질적인 데이터

Selection 03. 데이터베이스의 구축

1. 데이터베이스 구축 및 운영 과정



Selection 03. 데이터베이스의 구축

1. 데이터베이스 설치

- SQLite 다운로드 (www.sqlite.org/index.html)

The screenshot shows the SQLite Home Page (www.sqlite.org/index.html) in a web browser. The page has a header with the SQLite logo, navigation links (Home, About, Documentation, Download, License, Support, Purchase), and a search bar. The main content area discusses what SQLite is, its stability, cross-platform compatibility, and its use as a database engine. It also mentions the source code being in the public-domain and free for use. A sidebar on the right lists common links to various SQLite documentation sections like Features, SQL Syntax, and C/C++ Interface Spec.

What Is SQLite?

SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day. [More Information...](#)

The SQLite file format is stable, cross-platform, and backwards compatible and the developers pledge to keep it that way through the year 2050. SQLite database files are commonly used as containers to transfer rich content between systems [1] [2] [3] and as a long-term archival format for data [4]. There are over 1 trillion (1e12) SQLite databases in active use [5].

SQLite source code is in the public-domain and is free to everyone to use for any purpose.

Latest Release

Version 3.45.3 (2024-04-15). [Download](#) [Prior Releases](#)

Common Links

- Features
- When to use SQLite
- Getting Started
- Try it live!
- Prior Releases
- SQL Syntax
 - Pragmas
 - SQL functions
 - Date & time functions
 - Aggregate functions
 - Window functions
 - Math functions
 - JSON functions
- C/C++ Interface Spec
 - Introduction
 - List of C-language APIs
- The TCL Interface Spec
- Quirks and Gotchas

Selection 03. 데이터베이스의 구축

1. 데이터베이스 설치

- sqlite-tools-win-x64-3.45.3 버전

SQLite Download Page

Precompiled Binaries for Mac OS X (x86)

- [sqlite-tools-osx-x64-3450300.zip](#) (3.52 MiB) A bundle of command-line tools for managing SQLite database files, including the [command-line shell](#) program, the [sqldiff](#) program, and the [sqlite3 analyzer](#) program. (64-bit)
(SHA3-256: e5d25078f8d1d480ba77080db9993c7b6afc450c73391e97a40e56f92a326b9e)

Precompiled Binaries for Windows

- [sqlite-dll-win-x86-3450300.zip](#) (1.01 MiB) 32-bit DLL (x86) for SQLite version 3.45.3.
(SHA3-256: b5d01cccf1cea00b4250bc26d3941ba29b16eaaaa7ce9fe9f7776b8e4491ff2f)
- [sqlite-dll-win-x64-3450300.zip](#) (1.26 MiB) 64-bit DLL (x64) for SQLite version 3.45.3.
(SHA3-256: 9631cadef7d48498a468be6dd0dba825af3121f0980109601168f695f4273688)
- [sqlite-tools-win-x64-3450300.zip](#) (4.77 MiB) A bundle of command-line tools for managing SQLite database files, including the [command-line shell](#) program, the [sqldiff.exe](#) program, and the [sqlite3_analyzer.exe](#) program. 64-bit.
(SHA3-256: 429cca813471884ff3c9b51697ecc1e6750fb947ab5cd91efb098c6fb1f4053)

Precompiled Binaries for .NET

- [System.Data.SQLite](#) Visit the [System.Data.SQLite.org](#) website and especially the [download page](#) for source code and binaries of SQLite for .NET.

WebAssembly & JavaScript

- [sqlite-wasm-3450300.zip](#) (823.09 KiB) A precompiled bundle of `sqlite3.wasm` and its JavaScript APIs, ready for use in web applications.
(SHA3-256: 45d3520084b16a0e51fbef52f18c10780dfa81aa6209131c3014be72a38d3c9)

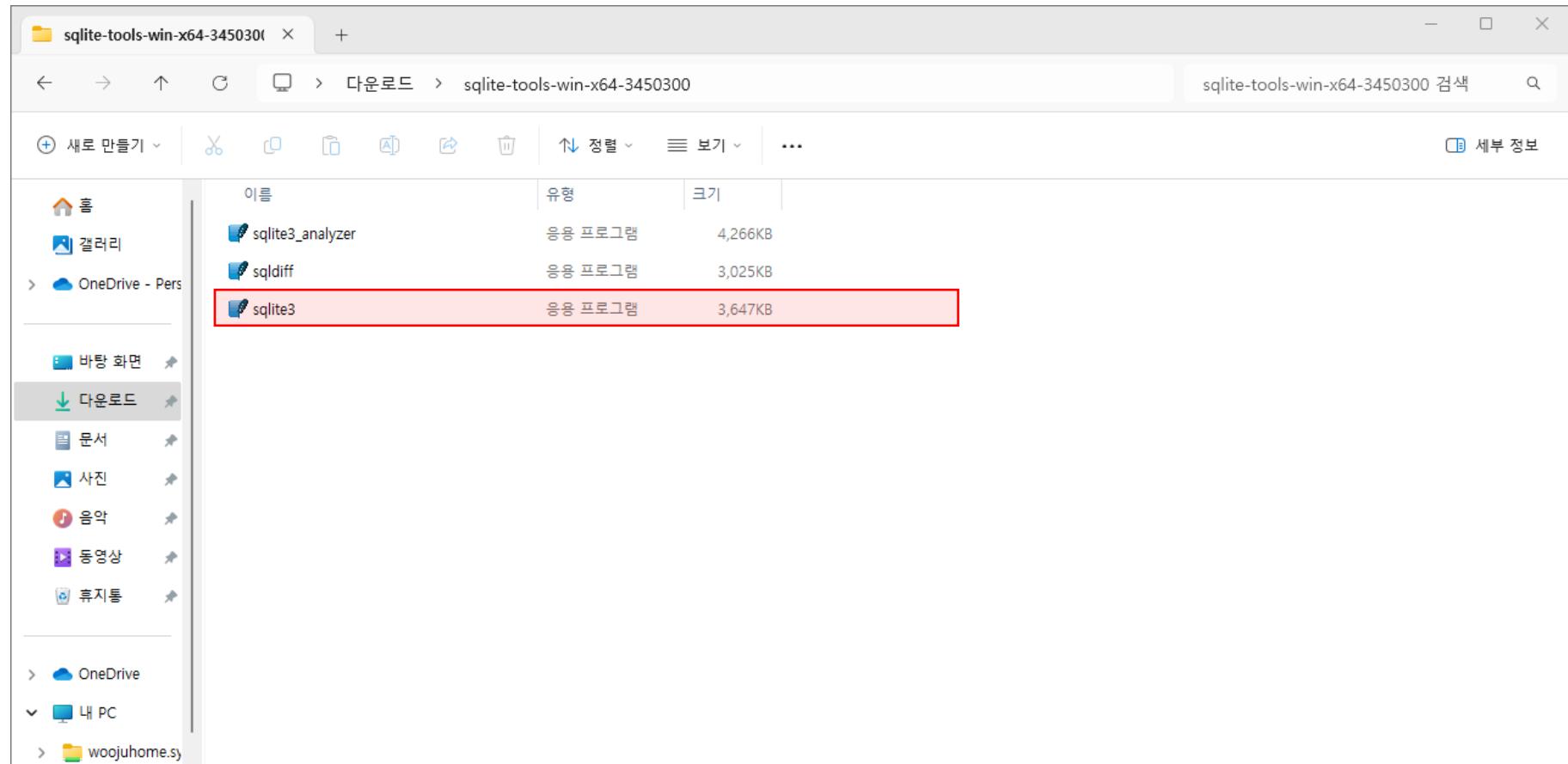
Alternative Source Code Formats

- [sqlite-src-3450300.zip](#) (12.50 MiB) Snapshot of the complete (raw) source tree for SQLite version 3.45.3. See [How To Compile SQLite](#) for usage details

Selection 03. 데이터베이스의 구축

1. 데이터베이스 설치

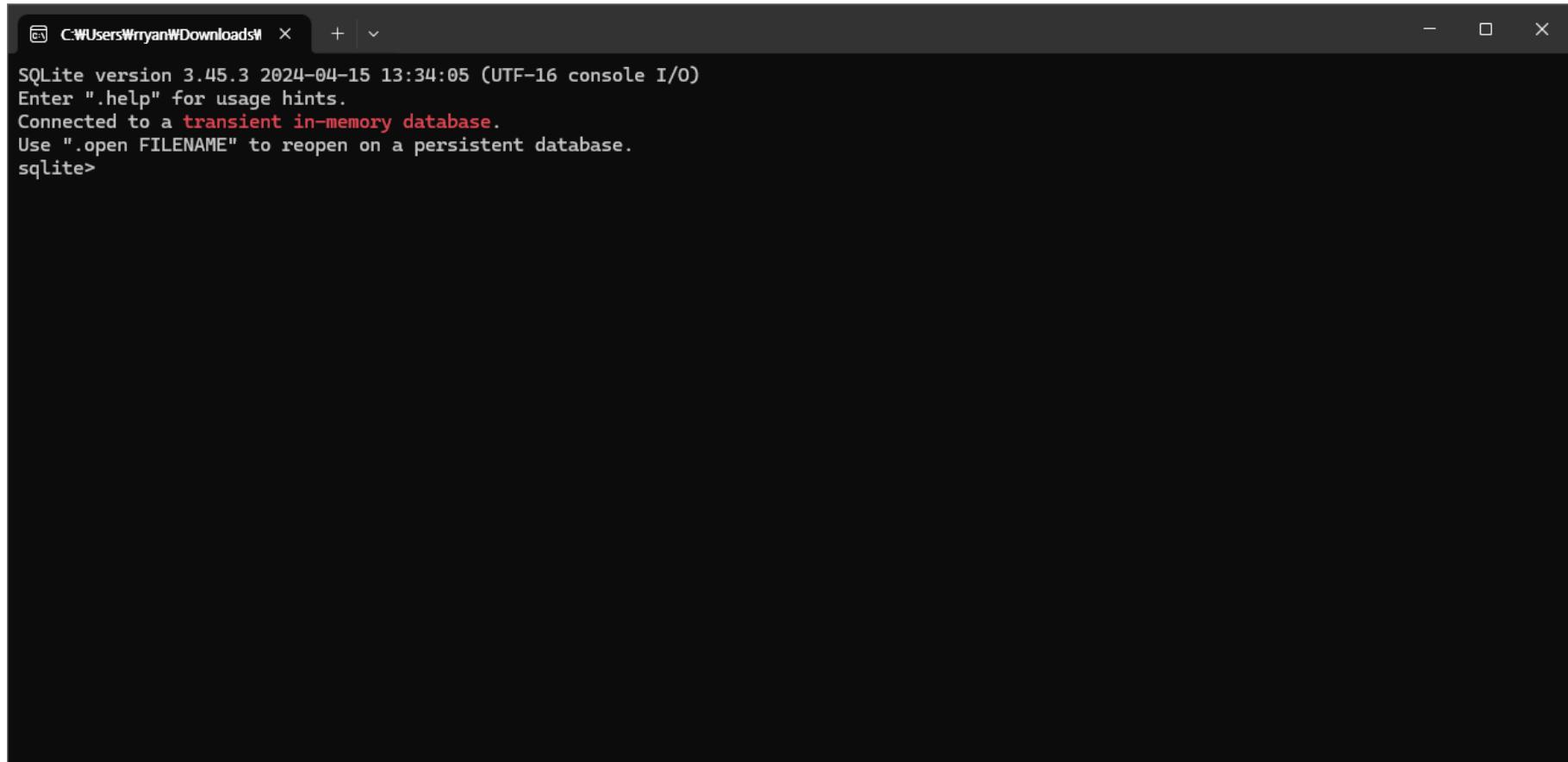
- SQLite 접속(sqlite3.exe 실행)



Selection 03. 데이터베이스의 구축

1. 데이터베이스 설치

- SQLite 접속(sqlite3.exe 실행)



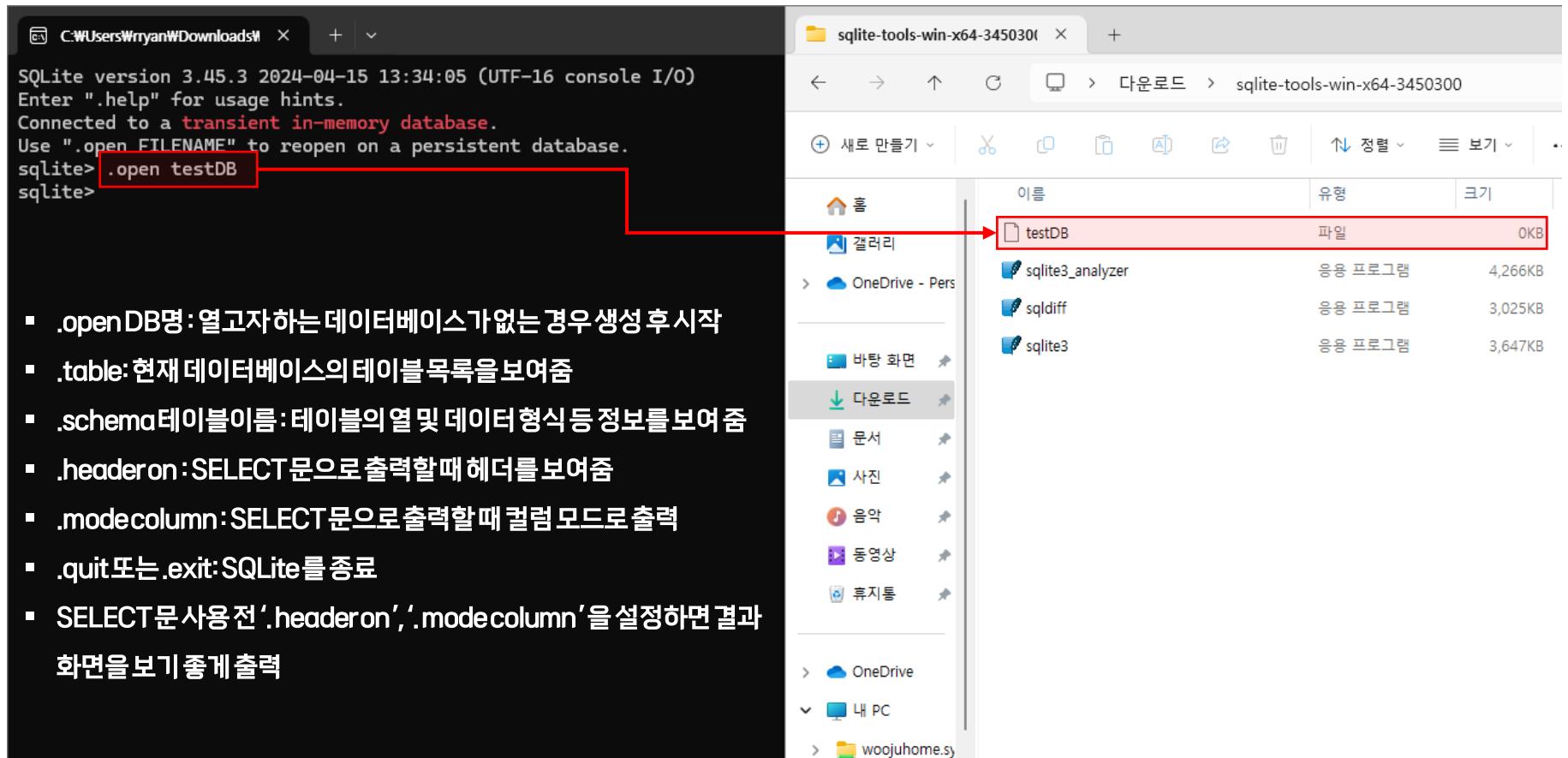
A screenshot of a Windows terminal window titled "C:\Users\ryan\Downloads\". The window shows the output of the SQLite command-line interface. The text in the terminal is as follows:

```
SQLite version 3.45.3 2024-04-15 13:34:05 (UTF-16 console I/O)
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>
```

Selection 03. 데이터베이스의 구축

2. 데이터베이스 구축

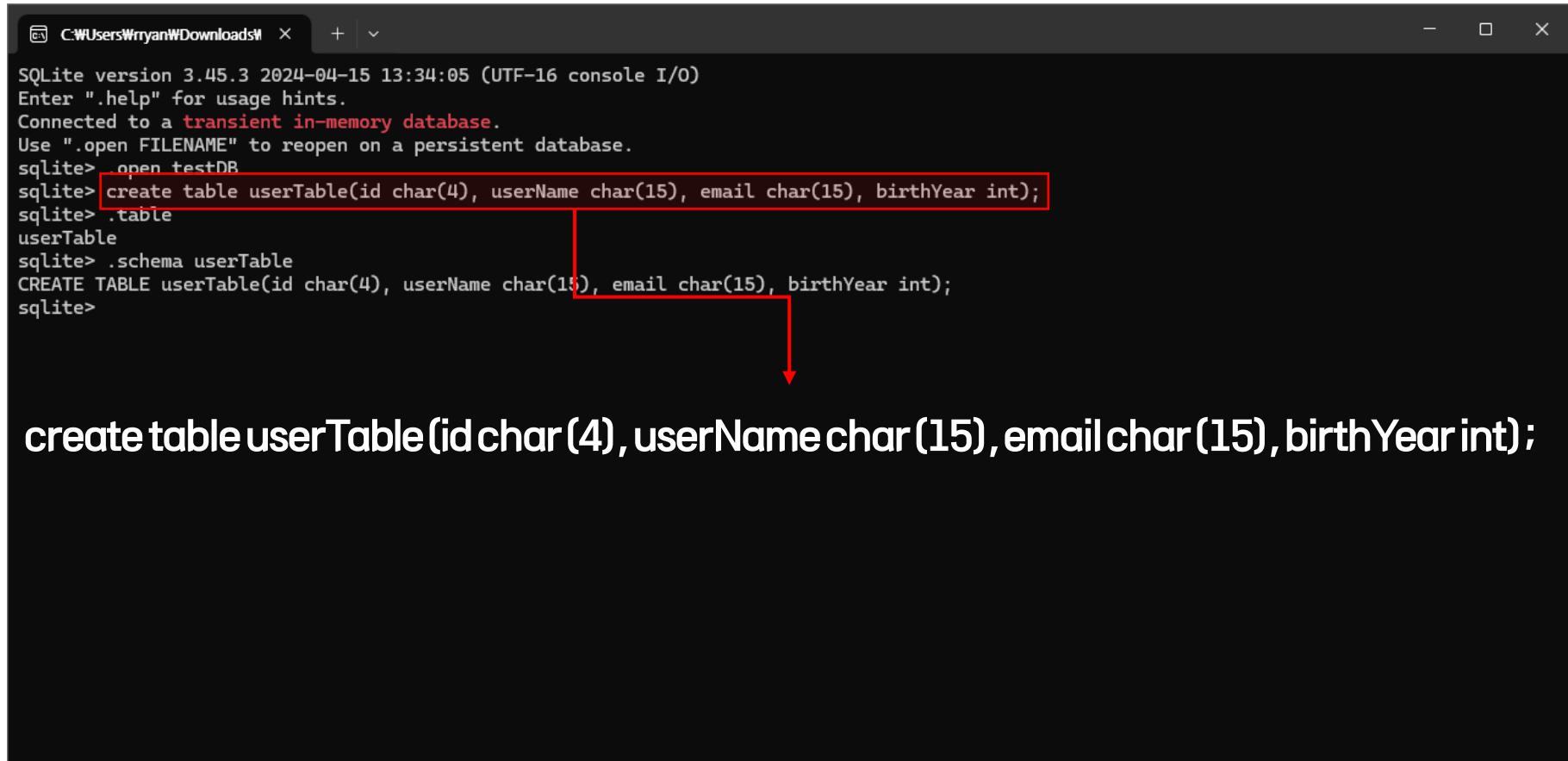
- 데이터베이스 생성 : '.open 데이터베이스이름' 명령어를 통한 DB 구축



Selection 03. 데이터베이스의 구축

2. 데이터베이스 구축

- 테이블 생성 : `create table` 테이블이름(컬럼명1 데이터형식, 컬럼명2 데이터형식…);



The screenshot shows a terminal window for SQLite version 3.45.3. The user has opened a database named 'testDB' and created a table named 'userTable' with four columns: id (char(4)), userName (char(15)), email (char(15)), and birthYear (int). A red box highlights the `create table` command, and a red arrow points from this command to the resulting schema definition below it.

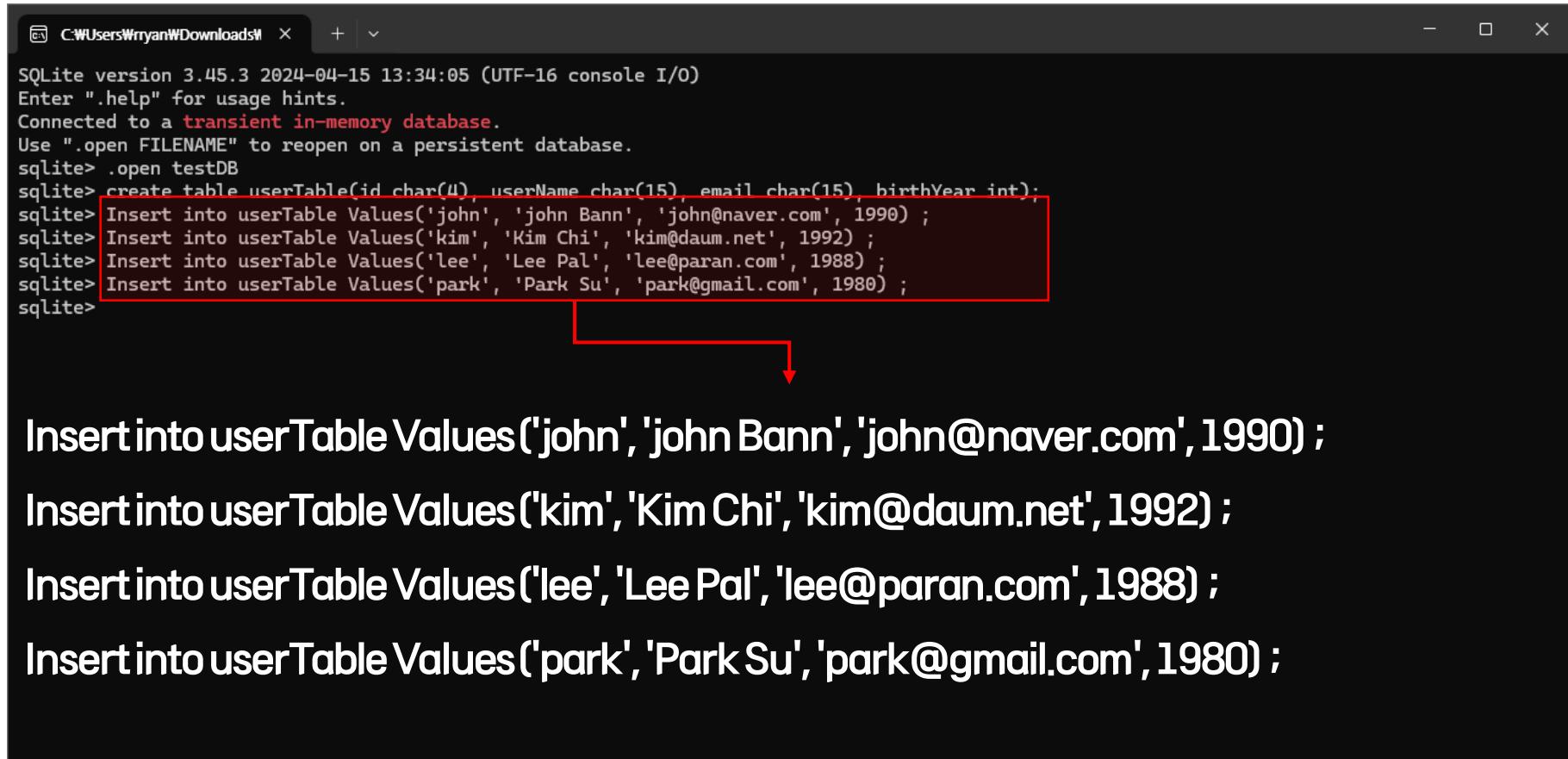
```
SQLite version 3.45.3 2024-04-15 13:34:05 (UTF-16 console I/O)
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open testDB
sqlite> create table userTable(id char(4), userName char(15), email char(15), birthYear int);
sqlite> .table
userTable
sqlite> .schema userTable
CREATE TABLE userTable(id char(4), userName char(15), email char(15), birthYear int);
sqlite>
```

`create table userTable(id char(4), userName char(15), email char(15), birthYear int);`

Selection 03. 데이터베이스의 구축

2. 데이터베이스 구축

- 행 데이터 입력 : `insert into 테이블이름 values (값1, 값2, …);`



The screenshot shows an SQLite command-line interface window. The command `.open testDB` has been run, and the database is connected to a transient in-memory database. The user is creating a table named `userTable` with columns `id`, `userName`, `email`, and `birthYear`. Four rows of data are inserted into the table:

```
sqlite> create table userTable(id char(4), userName char(15), email char(15), birthYear int);
sqlite> Insert into userTable Values('john', 'john Bann', 'john@naver.com', 1990) ;
sqlite> Insert into userTable Values('kim', 'Kim Chi', 'kim@daum.net', 1992) ;
sqlite> Insert into userTable Values('lee', 'Lee Pal', 'lee@paran.com', 1988) ;
sqlite> Insert into userTable Values('park', 'Park Su', 'park@gmail.com', 1980) ;
```

A red box highlights the four `Insert` statements, and a red arrow points from this box down to the corresponding text below.

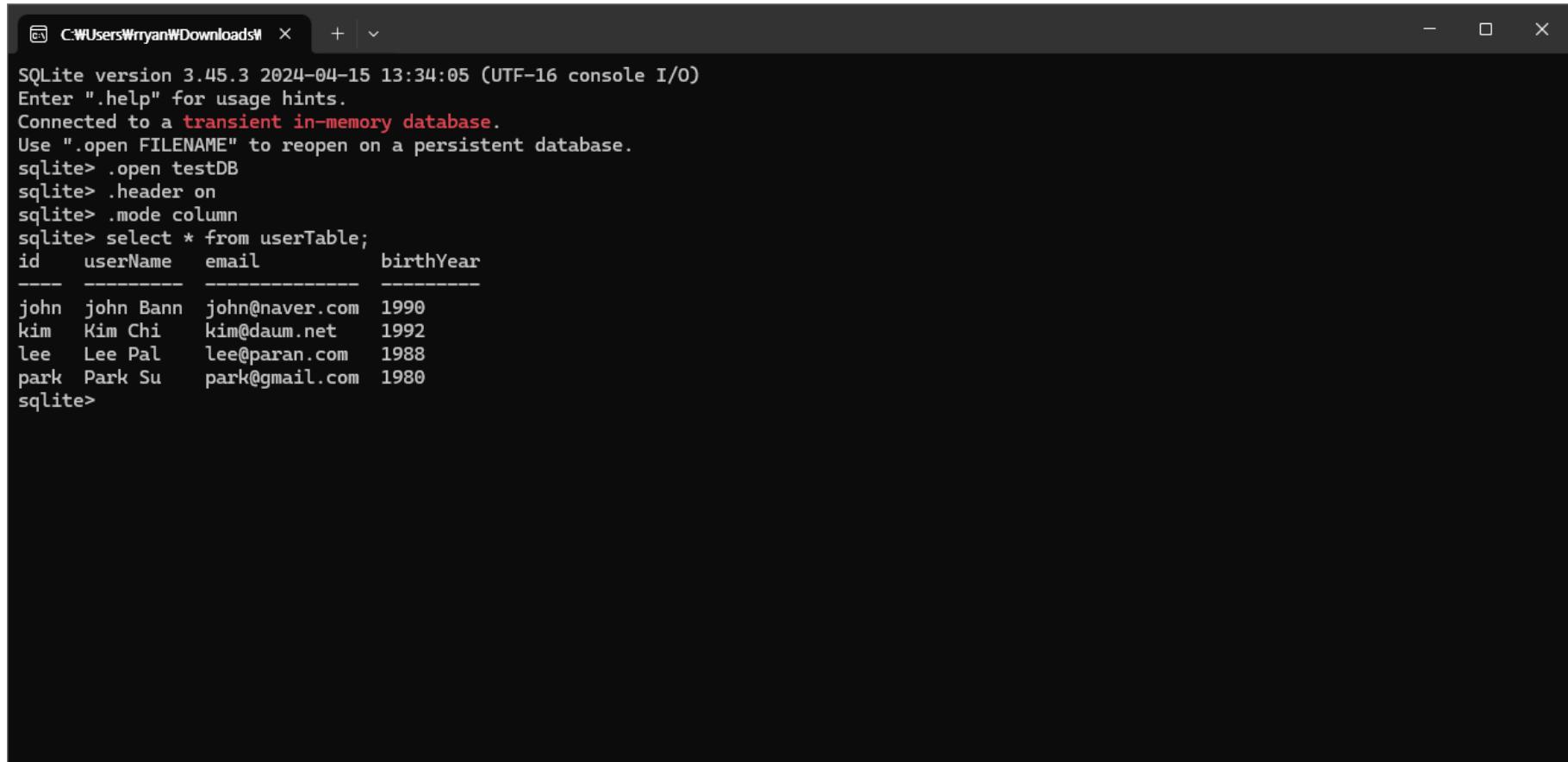
Below the highlighted code, the four inserted rows are listed again:

```
Insert into userTable Values('john', 'john Bann', 'john@naver.com', 1990) ;
Insert into userTable Values('kim', 'Kim Chi', 'kim@daum.net', 1992) ;
Insert into userTable Values('lee', 'Lee Pal', 'lee@paran.com', 1988) ;
Insert into userTable Values('park', 'Park Su', 'park@gmail.com', 1980) ;
```

Selection 03. 데이터베이스의 구축

2. 데이터베이스 구축

- 데이터 조회 : select 컬럼명1, 컬럼명2, … from 테이블이름



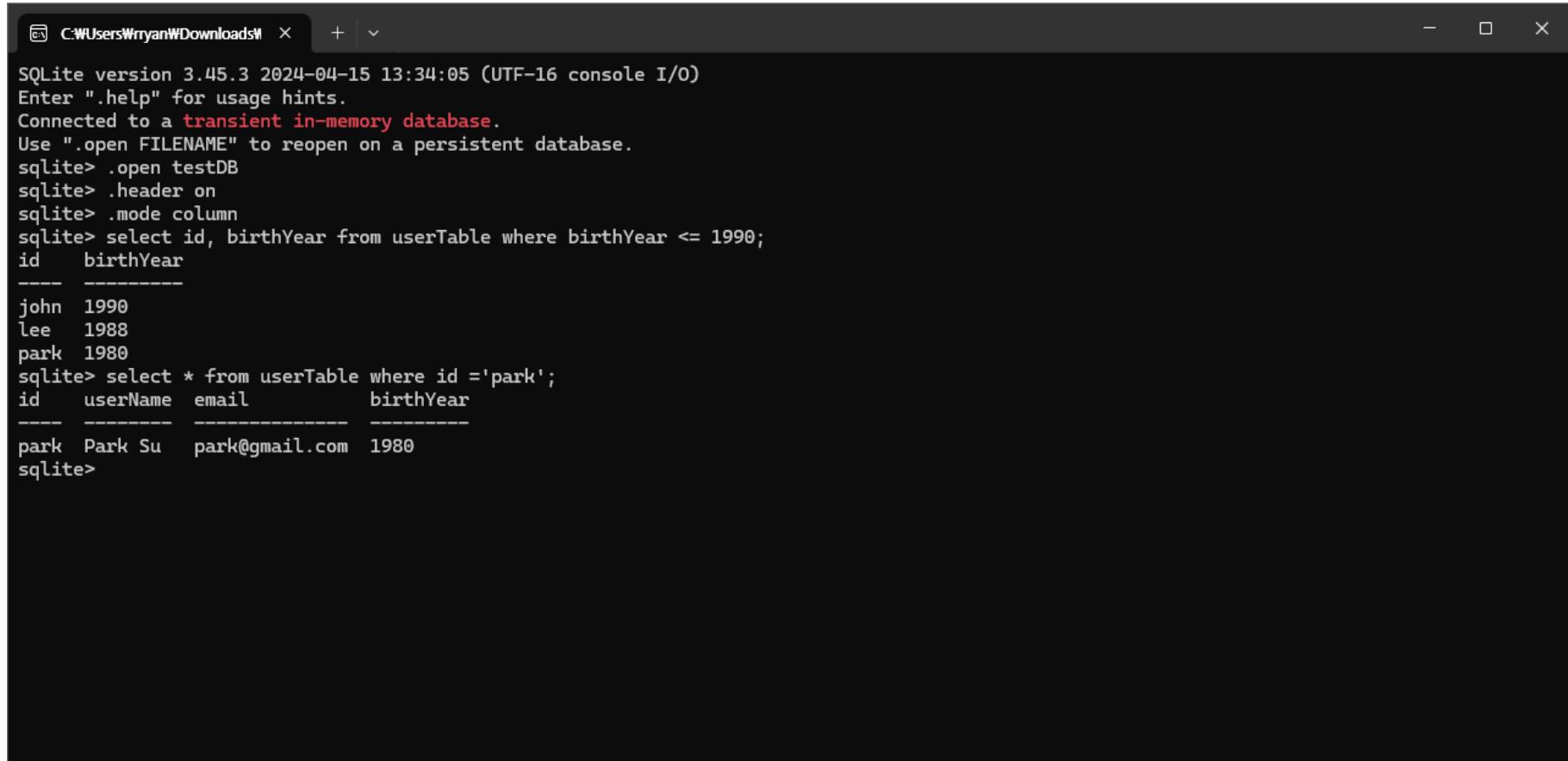
The screenshot shows a terminal window titled 'C:\Users\ryan\Downloads' running an SQLite shell. The output displays the results of a SELECT query on a table named 'userTable'. The table has columns 'id', 'userName', 'email', and 'birthYear'. The data shows five users: john, kim, lee, and park.

```
SQLite version 3.45.3 2024-04-15 13:34:05 (UTF-16 console I/O)
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open testDB
sqlite> .header on
sqlite> .mode column
sqlite> select * from userTable;
id    userName   email        birthYear
---  -----
john  john Bann  john@naver.com  1990
kim   Kim Chi    kim@daum.net   1992
lee   Lee Pal    lee@paran.com  1988
park  Park Su   park@gmail.com 1980
sqlite>
```

Selection 03. 데이터베이스의 구축

2. 데이터베이스 구축

- 조회할 데이터의 조건 설정 : select 컬럼명, … from 테이블이름 where 조건

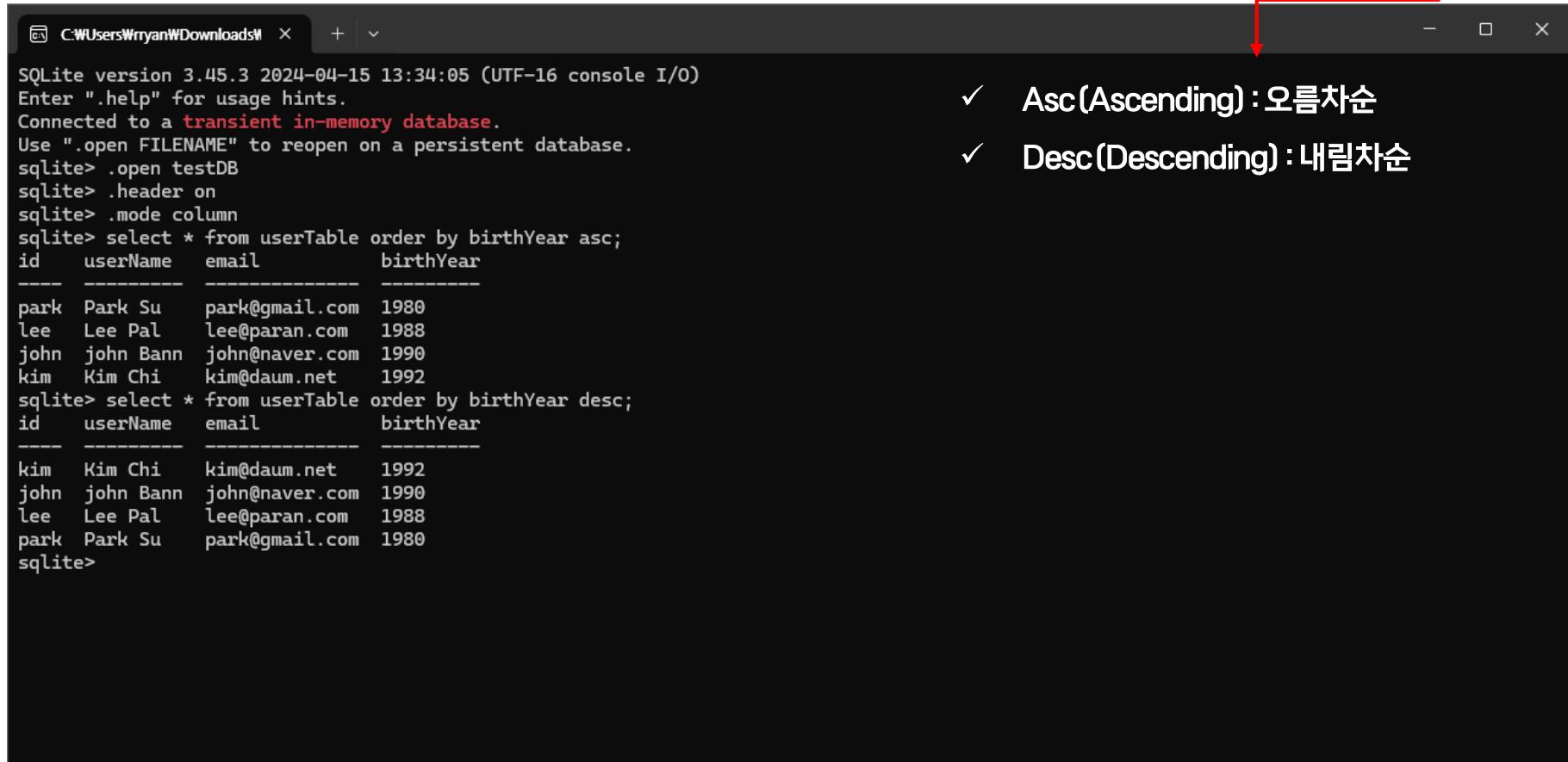


```
C:\Users\ryan\Downloads\ X + V
SQLite version 3.45.3 2024-04-15 13:34:05 (UTF-16 console I/O)
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open testDB
sqlite> .header on
sqlite> .mode column
sqlite> select id, birthYear from userTable where birthYear <= 1990;
id    birthYear
-----
john  1990
lee   1988
park  1980
sqlite> select * from userTable where id ='park';
id    userName   email        birthYear
-----
park  Park Su   park@gmail.com 1980
sqlite>
```

Selection 03. 데이터베이스의 구축

2. 데이터베이스 구축

- 데이터를 컬럼의 값으로 정렬 : `select * from 테이블이름 order by 컬럼명 [asc, desc]`



The screenshot shows an SQLite command-line interface window. The terminal output is as follows:

```
SQLite version 3.45.3 2024-04-15 13:34:05 (UTF-16 console I/O)
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open testDB
sqlite> .header on
sqlite> .mode column
sqlite> select * from userTable order by birthYear asc;
id    userName   email        birthYear
----  -----
park  Park Su   park@gmail.com 1980
lee   Lee Pal   lee@paran.com  1988
john  john Bann john@naver.com 1990
kim   Kim Chi   kim@daum.net  1992
sqlite> select * from userTable order by birthYear desc;
id    userName   email        birthYear
----  -----
kim   Kim Chi   kim@daum.net  1992
john  john Bann john@naver.com 1990
lee   Lee Pal   lee@paran.com  1988
park  Park Su   park@gmail.com 1980
sqlite>
```

Two annotations are present on the right side of the terminal window:

- A red arrow points to the word "asc" in the first query result, with the text "✓ Asc(Ascending) : 오름차순" below it.
- A red arrow points to the word "desc" in the second query result, with the text "✓ Desc(Descending) : 내림차순" below it.

Selection 04. 데이터베이스의 입력과 조회

1. 파이썬에서 데이터를 입력하는 코딩 순서



Selection 04. 데이터베이스의 입력과 조회

1. 파이썬에서 데이터를 입력하는 코딩 순서

- 데이터베이스 연결 : sqlite3 모듈을 임포트 후 sqlite3.connect("db이름")으로 연결

```
import sqlite3  
con = sqlite3.connect("C:/CookPython/naverDB")      # 소스 코드가 저장된 폴더에 생성
```

출력 결과

아무것도 나오지 않음

- 커서(cursor) 생성 : 연결한 db에 SQL 문을 실행 또는 실행된 결과를 돌려받는 커서

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

출력 결과

아무것도 나오지 않음

Selection 04. 데이터베이스의 입력과 조회

1. 파이썬에서 데이터를 입력하는 코딩 순서

- 테이블 생성, 데이터 입력: SQL 문을 커서이름.execute() 함수의 매개변수로 넘겨주면 SQL 문이 데이터베이스에 실행

```
cur.execute("CREATE TABLE userTable (id char(4), userName char(15), email char(15), birthYear int)")
```

출력 결과

```
<sqlite3.Cursor object at 개체번호>
```

〈테이블 생성 SQL〉

```
cur.execute("INSERT INTO userTable VALUES('john', 'John Bann', 'john@naver.com', 1990)")  
cur.execute("INSERT INTO userTable VALUES('kim', 'Kim Chi', 'kim@daum.net', 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)")
```

출력 결과

```
<sqlite3.Cursor object at 개체번호>가 각각 4회 출력됨
```

〈데이터 입력 SQL〉

Selection 04. 데이터베이스의 입력과 조회

1. 파이썬에서 데이터를 입력하는 코딩 순서

- 입력한 데이터 저장-커밋(Commit)

```
con.commit()
```

출력 결과

아무것도 나오지 않음

- 데이터베이스 닫기

```
con.close()
```

출력 결과

아무것도 나오지 않음

Selection 04. 데이터베이스의 입력과 조회

1. 파이썬에서 데이터를 입력하는 코딩 순서

The screenshot shows a Python code editor interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** Search
- Toolbar:** Includes icons for file operations like Open, Save, Find, and others.
- Code Area:** Untitled-1.py (Python script). The code creates a SQLite database and inserts four user records.

```
1 import sqlite3
2
3 con = sqlite3.connect("C:/Users/rryan/Downloads/sqlite-tools-win-x64-3450300/testDB_2")
4
5 cur = con.cursor()
6
7 cur.execute("CREATE TABLE userTable (id char(4), userName char(15), email char(15), birthYear int)")
8
9 cur.execute("Insert into userTable Values('john', 'john Bann', 'john@naver.com', 1990)")
10 cur.execute("Insert into userTable Values('kim', 'Kim Chi', 'kim@daum.net', 1992)")
11 cur.execute("Insert into userTable Values('lee', 'Lee Pal', 'lee@paran.com', 1988)")
12 cur.execute("Insert into userTable Values('park', 'Park Su', 'park@gmail.com', 1980)")
13
14 con.commit()
15
16 con.close()
```

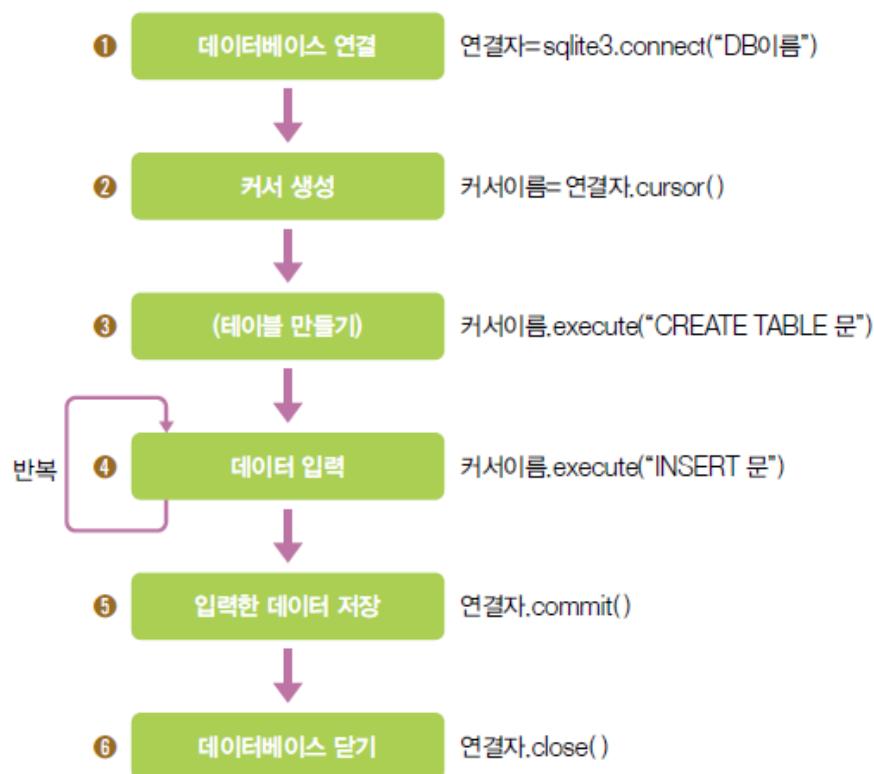
- Bottom Navigation:** PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (underlined), PORTS.
- Terminal Area:** Shows command-line output from running the script.

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe c:/Users/rryan/Downloads/Untitled-1.py
PS C:\Users\rryan>
```

- Side Icons:** User profile, gear, and other settings icons.

Selection 04. 데이터베이스의 입력과 조회

1. 파이썬에서 데이터를 입력하는 코딩 순서



A screenshot of a Python code editor (VS Code) showing a script named "Untitled-1.py". The code creates a SQLite database, defines a user table, and inserts four user records. The code is as follows:

```
import sqlite3
con = sqlite3.connect("C:/Users/rryan/Downloads/sqlite-tools-win-x64")
cur = con.cursor()
cur.execute("CREATE TABLE userTable (id char(4), userName char(15), email char(30))")
cur.execute("Insert into userTable Values('john', 'john Bann', 'john@domain.com')")
cur.execute("Insert into userTable Values('kim', 'Kim Chi', 'kim@dau.com')")
cur.execute("Insert into userTable Values('lee', 'Lee Pal', 'lee@paradise.com')")
cur.execute("Insert into userTable Values('park', 'Park Su', 'park@gmarket.co.kr')")
con.commit()
con.close()
```

The code editor interface shows tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, displaying the command `PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python`.

Selection 04. 데이터베이스의 입력과 조회

2. 데이터 입력 프로그램의 구현

- (예시) 파이썬에서 데이터베이스에 연결한 후 데이터를 입력하는 코드

The screenshot shows a Python code editor interface with a dark theme. The main area displays the following Python script:

```
C:\> Users > rryan > Downloads > Untitled-1.py > ...
1 import sqlite3
2
3 con, cur = None, None
4 data1, data2, data3, data4 = "", "", "", ""
5 sql = ""
6
7 con = sqlite3.connect("C:/Users/rryan/Downloads/sqlite-tools-win-x64-3450300/testDB_2")
8 cur = con.cursor()
9
10 while(True):
11     data1 = input("사용자 ID : ")
12     if data1 == "":
13         break
14     data2 = input("사용자 이름 : ")
15     data3 = input("이메일 : ")
16     data4 = input("출생년도 : ")
17     sql = "insert into userTable values('"+data1+"', '"+data2+"', '"+data3+"', "+data4+ ")"
18     cur.execute(sql)
19
20 con.commit()
21 con.close()
```

The code uses the `sqlite3` module to connect to a SQLite database named `testDB_2`. It enters a loop where it repeatedly prompts the user for a user ID, name, email, and birth year, then inserts the data into a table named `userTable`. Finally, it commits the changes and closes the connection.

At the bottom of the editor, there are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is currently selected. The output window shows the following terminal session:

```
출생년도 : 1994
사용자 ID : ko
사용자 이름 : Ko Jin
이메일 : kojin@naver.com
```

Selection 04. 데이터베이스의 입력과 조회

2. 데이터 입력 프로그램의 구현

- (예시) 파이썬에서 데이터베이스에 연결한 후 데이터를 입력하는 코드

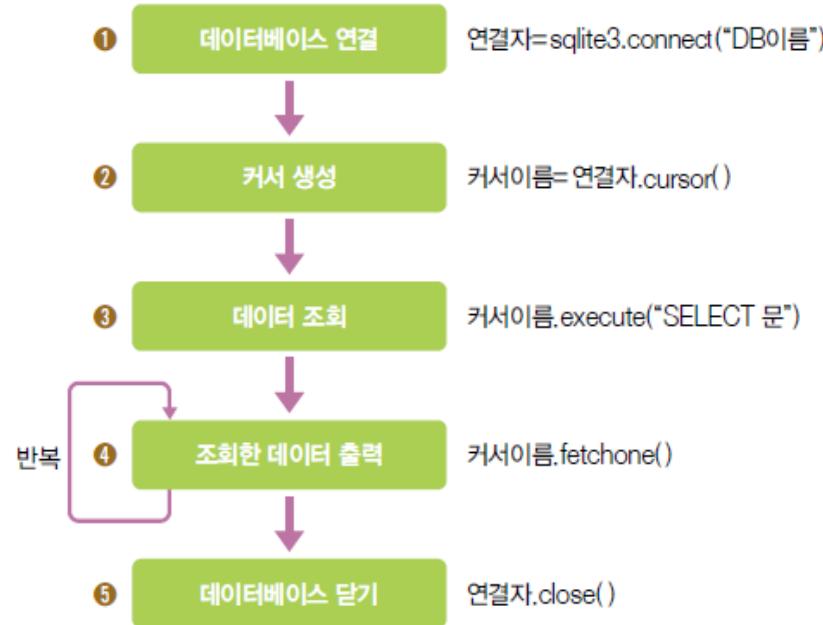
The screenshot shows the VS Code interface with the Terminal tab selected. The terminal window displays the following SQLite session:

```
SQLite version 3.45.3 2024-04-15 13:34:05 (UTF-16 console I/O)
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open testDB_2
sqlite> .header on
sqlite> .mode column
sqlite> select * from userTable;
id    userName   email           birthYear
----  -----     -----          -----
john  john Bann  john@naver.com  1990
kim   Kim Chi   kim@daum.net   1992
lee   Lee Pal   lee@paran.com  1988
park  Park Su   park@gmail.com 1980
sqlite> select * from userTable;
id    userName   email           birthYear
----  -----     -----          -----
john  john Bann  john@naver.com  1990
kim   Kim Chi   kim@daum.net   1992
lee   Lee Pal   lee@paran.com  1988
park  Park Su   park@gmail.com 1980
su    Su Ji     suji@naver.com  1994
ko    Ko Jin    kojin@naver.com 1996
sqlite>
```

A red box highlights the first four rows of the initial query result, and another red box highlights the last two rows of the second query result. A red arrow points from the bottom-left red box to the bottom-right red box, indicating a comparison between the original data and the modified data.

Selection 04. 데이터베이스의 입력과 조회

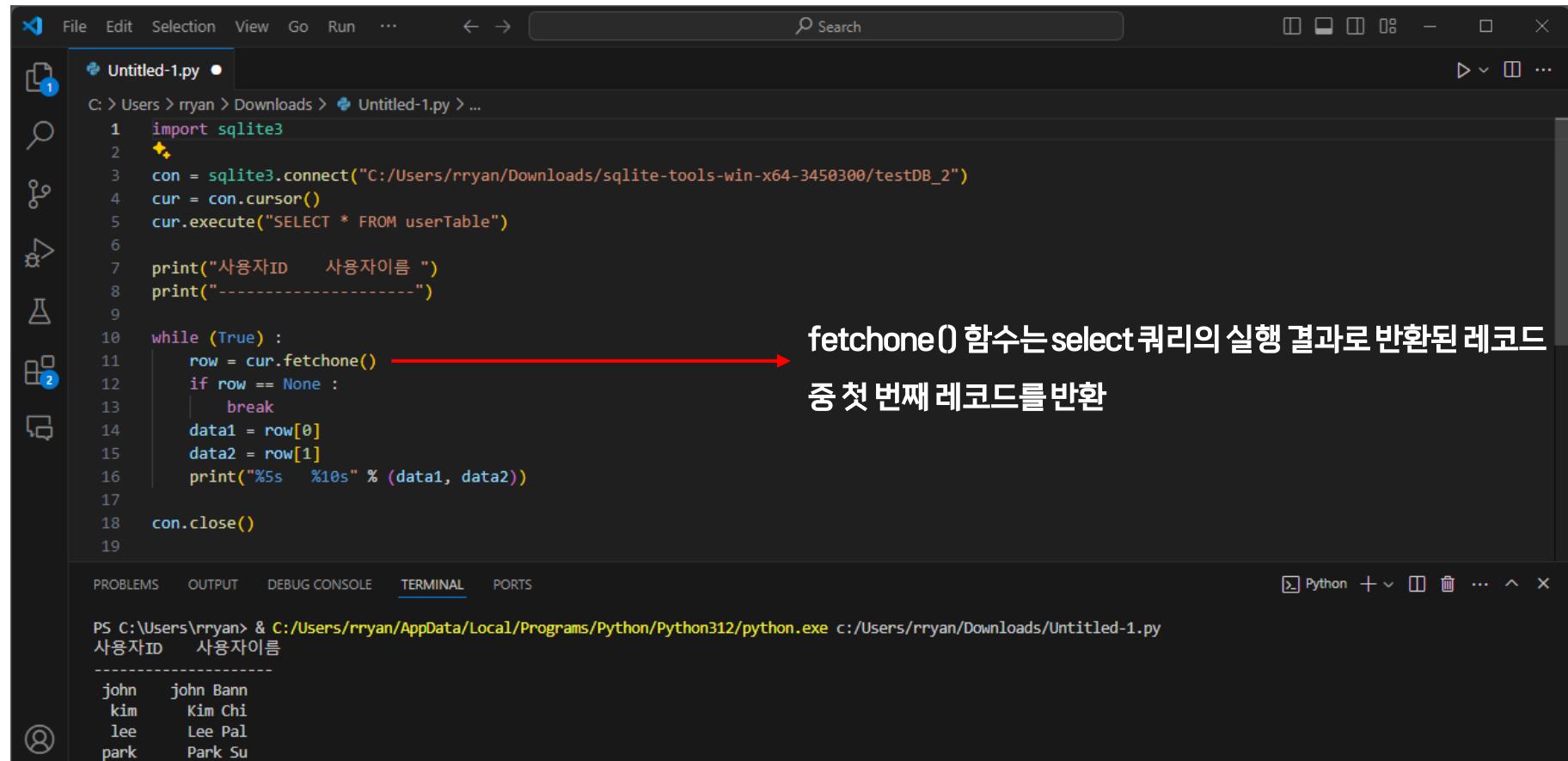
3. 파이썬에서 데이터를 조회하는 코딩 순서



Selection 04. 데이터베이스의 입력과 조회

4. 데이터 조회 프로그램의 구현

- (예시) 파이썬에서 데이터베이스에 연결한 후 데이터를 조회하는 코드



```
C:\> Users > rryan > Downloads > Untitled-1.py > ...
1 import sqlite3
2
3 con = sqlite3.connect("C:/Users/rryan/Downloads/sqlite-tools-win-x64-3450300/testDB_2")
4 cur = con.cursor()
5 cur.execute("SELECT * FROM userTable")
6
7 print("사용자ID      사용자이름 ")
8 print("-----")
9
10 while (True):
11     row = cur.fetchone() ->
12     if row == None:
13         break
14     data1 = row[0]
15     data2 = row[1]
16     print("%5s    %10s" % (data1, data2))
17
18 con.close()

fetchone() 함수는 select 쿼리의 실행 결과로 반환된 레코드
중 첫 번째 레코드를 반환
```

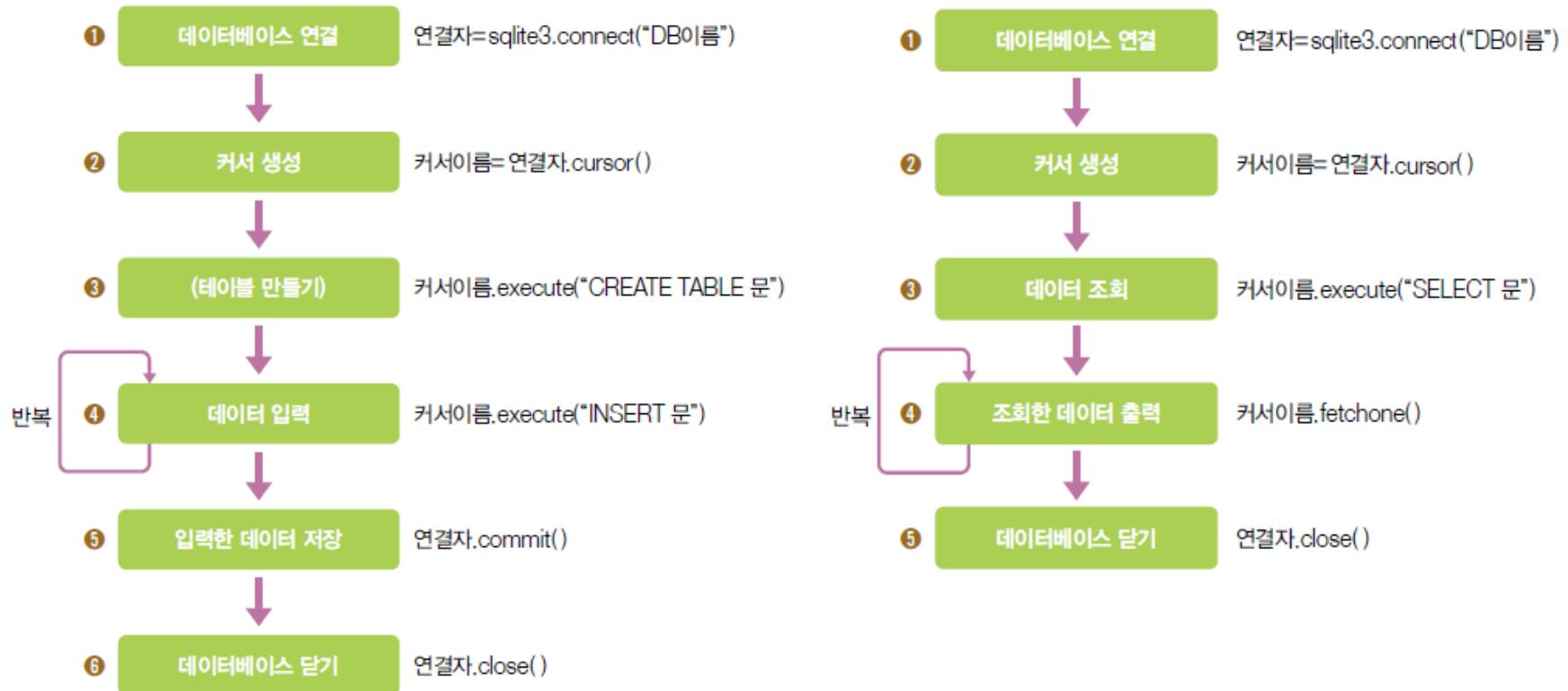
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe c:/Users/rryan/Downloads/Untitled-1.py

사용자ID	사용자이름
john	john Bann
kim	Kim Chi
lee	Lee Pal
park	Park Su

Chapter 13. 요약

- ✓ 데이터베이스는 여러 사람에 의해 공유되어 사용될 목적으로 통합 관리되는 데이터의 집합
- ✓ 데이터베이스 관리 시스템은 데이터베이스를 관리해 주는 시스템 또는 소프트웨어
- ✓ 파이썬에서 SQLite에 데이터를 입력/조회하는 순서는 다음과 같음



감사합니다.

우창우

Dr.woo@chungbuk.ac.kr

실습&과제 : (제출처) Dr.woo@chungbuk.ac.kr, (기한) 5.25(토) 까지

1. 데이터베이스에 연결하고 입력/조회하는 프로그램

GUI 데이터 입력

사용자ID ----- john kim lee park su ko hong	사용자이름 ----- john Bann Kim Chi Lee Pal Park Su Su Ji Ko Jin hong gil	이메일 ----- john@naver.com kim@daum.net lee@paran.com park@gmail.com suji@naver.com kojin@naver.com hong@naver.com	출생연도 ----- 1990 1992 1988 1980 1994 1996 1986

팀프로젝트 현황

조 번호	프로젝트 주제	조장	조원
1조	물리 엔진과 충돌 처리 기능, 그리고 캐릭터 디자인을 반영한 슈팅 게임 (슈퍼 마리오)	조형준	고태경, 김다민
2조	다중 미니게임 클라이언트를 위한 통합 아키텍처 개발	김민혁	전영우, 김정민
3조	팀프로젝트 수행능력 및 파이썬 프로그래밍 역량 강화를 위한 방탈출 게임 개발	홍성진	김태영, 정새연
4조	재학생의 완벽한 한끼 보장을 위한 맛집 서비스 플랫폼 개발	이규민	우태현, 전수혁
5조	날씨에 따른 일정 관리를 위한 캘린더 앱	배정민	박상인, 서범교, 송설희
6조	교내 학생 간의 물품 중고거래를 위한 시스템 SW를 통한 웹사이트 제작	박조현	김건우, 오다영
7조	최근 인기 있는 음악의 키워드를 활용한 시기별 음악을 찾아주는 프로그램 개발	박주현	권정욱, 정현준
8조	파이썬 게임 개발 경험을 쌓기 위한 클루 보드게임 개발	김규현	김준후, 조윤정
9조	학생들을 위한 학습도우미 앱	신종환	신승우, 한강민
11조	학교 졸업사정 Q&A 챗봇	한준영	고태영, 이관학, 육광민
12조	학생의 수업의 참여율 감소 문제를 해결하기 위한 수업 참여도 인공지능 기술개발	윤시훈	전준석, 김민경
13조	AI와 데이터베이스를 활용해 복잡성을 최소화한 화장품 추천 웹사이트	김준호	황지연, 이용희
14조	게임개발 이해도 향상을 위한 고도엔진을 이용한 퍼즐게임 개발	배수환	이한결, 신혜원
15조	학생들의 경제적 부담 완화를 위한 택시 동승 서비스 개발	박성범	이태정, 김민석