Appendix

실습 소프트웨어 설치

A Oracle Database 26ai 설치
B 오라클 개발 도구 및 샘플 데이터베이스 설치
C 파이썬, oracledb 라이브러리, Flask 설치
D SQL Developer Data Modeler
E 오라클 클라우드
F 데이터베이스 관리

A Oracle Database 26ai 설치

1 소개

Oracle AI Database 26ai FREE는 Oracle Database의 무료 버전으로, 개발·학습·테스트 환경에 적합하다. 이 버전은 상용 Oracle Database와 동일한 코드 기반을 사용하며, 다음과 같은 리소스 제한이 있다.

항목	제한
CPU	최대 2CPU
RAM	최대 2GB
저장소	최대 12GB

Oracle AI Database 26ai FREE의 주요 특징은 다음과 같다.

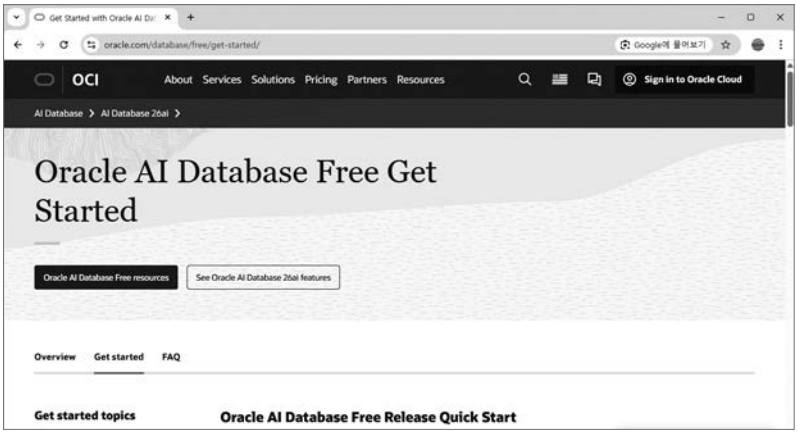
- AI Vector Search : 벡터 데이터를 저장하고 검색하는 기능 제공
- SELECT AI : 자연어 입력을 통해 SQL 쿼리를 생성
- JSON Relational Duality : JSON 데이터와 관계형 데이터 간 통합 뷰 제공
- Property Graph : 그래프 데이터베이스 기능 제공
- True Cache : 읽기 전용 인메모리 캐시 기능

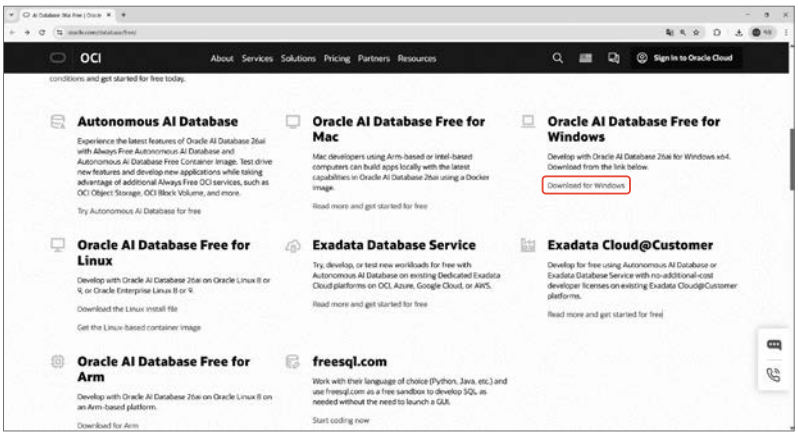
2 다운로드

Oracle AI Database 26ai FREE는 Windows와 Linux 환경에서는 직접 설치할 수 있으며, macOS 환경에서는 Docker를 통해 사용할 수 있다.

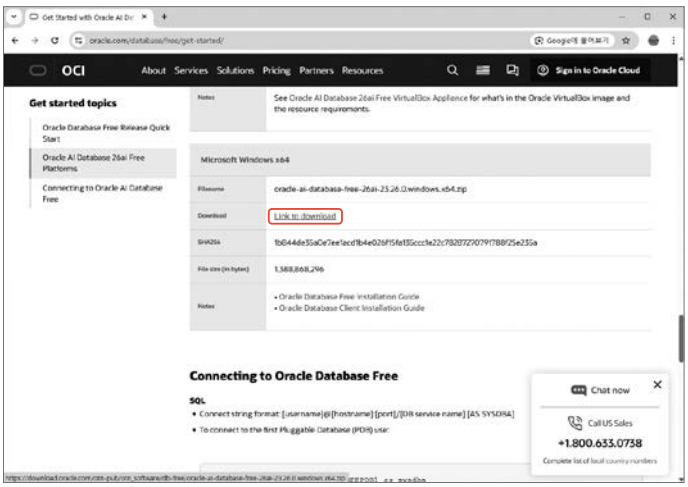
Windows

❶ 웹 브라우저에서 Oracle 다운로드 페이지(https://www.oracle.com/database/free/get-started/)에 접속 후 스크롤을 내려 [Download for Windows]를 클릭한다.





❷ 운영체제에 맞는 버전(Windows x64: oracle-ai-database-free-26ai-23.26.0.windows.x64.zip)을 선택한다.



TIP 영문(USA) 사이트와 한국어(Korean) 사이트에서 표시되는 파일 이름은 다를 수 있으나, 실제로는 동일한 파일이므로 어느 것을 사용해도 무방하다.

macOS(Docker 사용)

macOS에서는 Oracle Database를 직접 설치할 수 없으므로 Docker 컨테이너 이미지를 사용한다.

❶ Docker Desktop 다운로드 페이지(https://www.docker.com/products/docker-desktop/)에 접속한다.

❷ Mac 프로세서에 맞는 버전을 다운로드한다.

- Apple Silicon(M1/M2/M3/M4) : Mac with Apple chip
- Intel Mac : Mac with Intel chip

❸ Docker 설치 후 터미널에서 pull 명령을 실행하여 Oracle Database 컨테이너 이미지를 다운로드한다.

```
docker pull container-registry.oracle.com/database/free:latest-lite
```

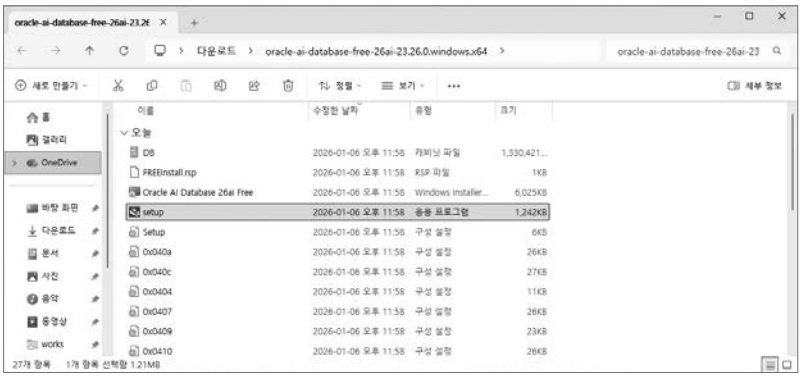
TIP Oracle Container Registry의 이미지는 Oracle 계정 없이도 다운로드할 수 있다.

3 설치

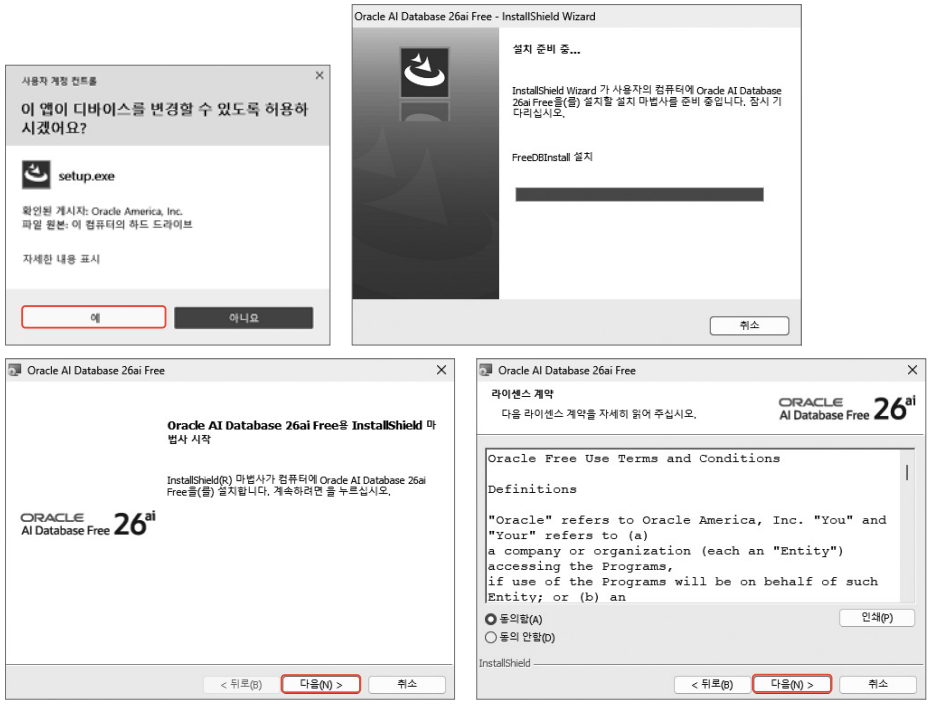
Windows 설치

❶ 다운로드한 zip 파일의 압축을 해제한다.

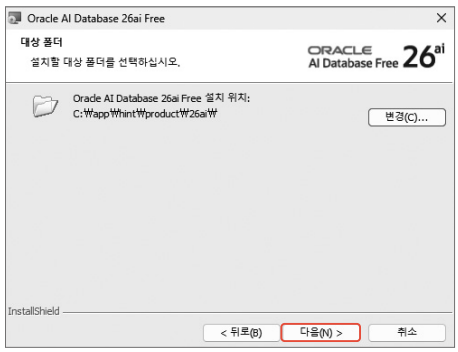
❷ setup.exe 파일을 더블클릭하여 설치 프로그램을 실행한다.



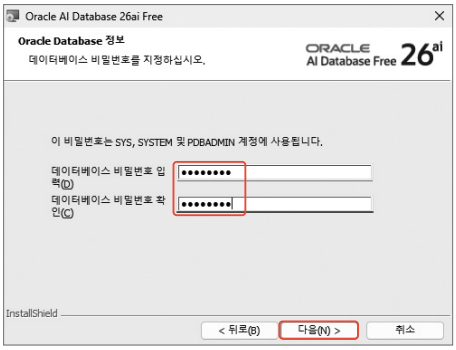
사용자 계정 컨트롤 화면이 나타나면 [예]를 선택한다.



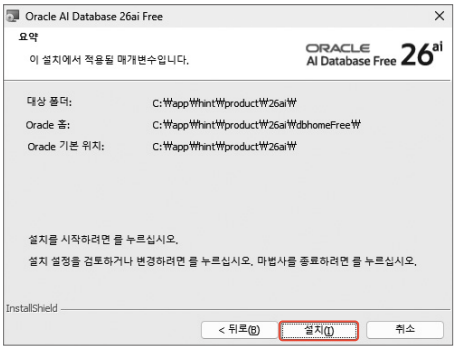
❸ 시스템 요구 사항을 확인한 후 설치 경로를 지정한다. 기본 경로는 C:\app\사용자계정\product\26ai\이다.

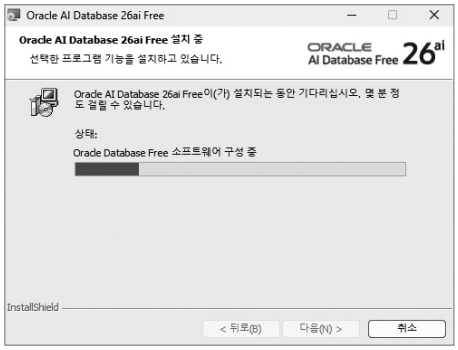


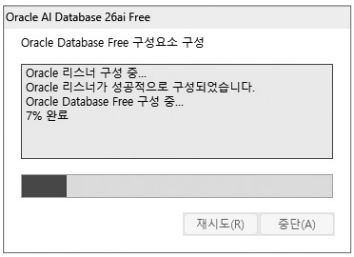
❹ 데이터베이스 비밀번호를 Manager1로 설정한다. 이 비밀번호는 SYS, SYSTEM, PDBADMIN 계정에 공통으로 적용된다.



❺ 설치 요약 화면에서 [설치] 버튼을 클릭한다.

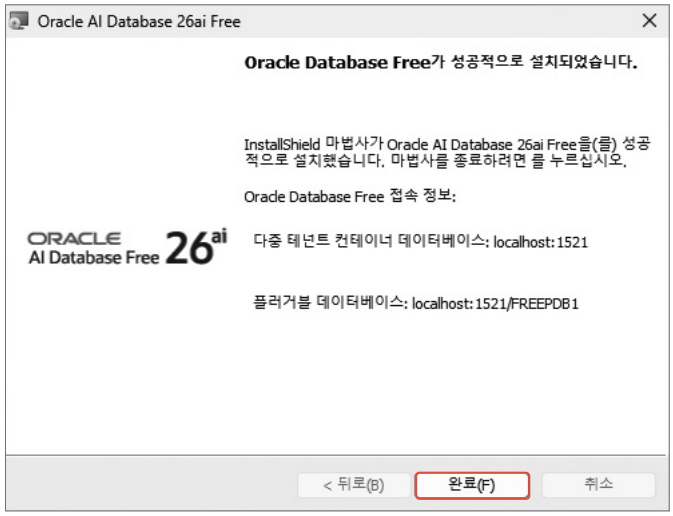






설치 중 Windows 보안 화면이 나타나면 [허용]을 선택한다.

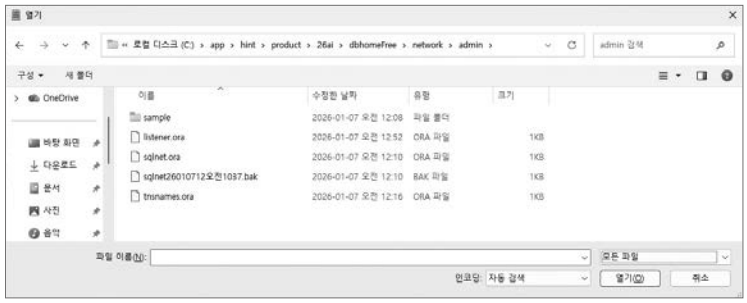
❻ 설치가 완료되면 컨테이너 데이터베이스 FREE와 플러거블 데이터베이스 FREEPDB1이 자동으로 생성된다.



❼ **(추가 사항) 리스너 설정**

Oracle 26ai는 기본적으로 로컬 IP를 사용해 접속할 수 있도록 설정되어 있다. 설치 후 안내되는 localhost로 접속되지 않는 경우에는 리스너 설정을 아래와 같이 변경한다. 본문에서는 설치 완료 후 localhost로 접속하는 것을 기준으로 설명한다. Windows의 경우 메모장을 사용하여 다음 리스너 설정 파일을 편집한다.

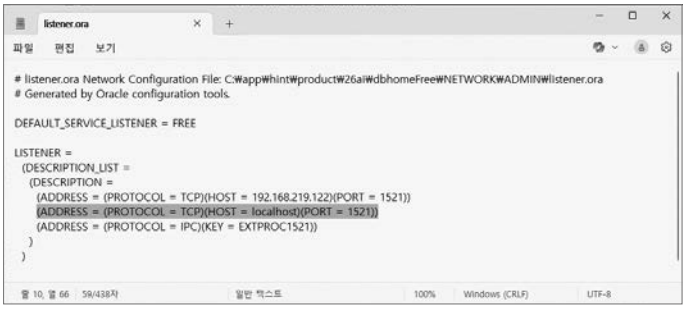
```
C:\app\유저계정명\product\26ai\dbhomeFree\network\admin\listener.ora
```



TIP 유저계정명은 Windows에 로그인한 사용자 이름이다. 위 예시는 'hint' 계정으로 로그인한 경우이다.

listener.ora 파일을 열어 다음 내용을 추가한 후 저장하고, 시스템을 재부팅한다.

```
(ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
```

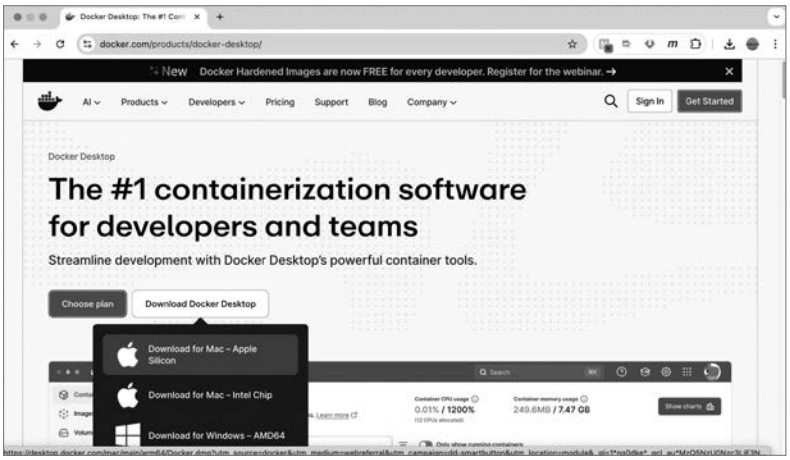


macOS 설치(Docker 사용)

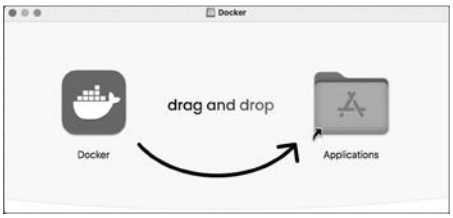
macOS에서는 Oracle Database를 직접 설치할 수 없으므로 Docker 컨테이너를 사용해 실행한다. Apple Silicon(M1/M2/M3/M4) Mac에서도 사용할 수 있다.

❶ Docker Desktop 설치

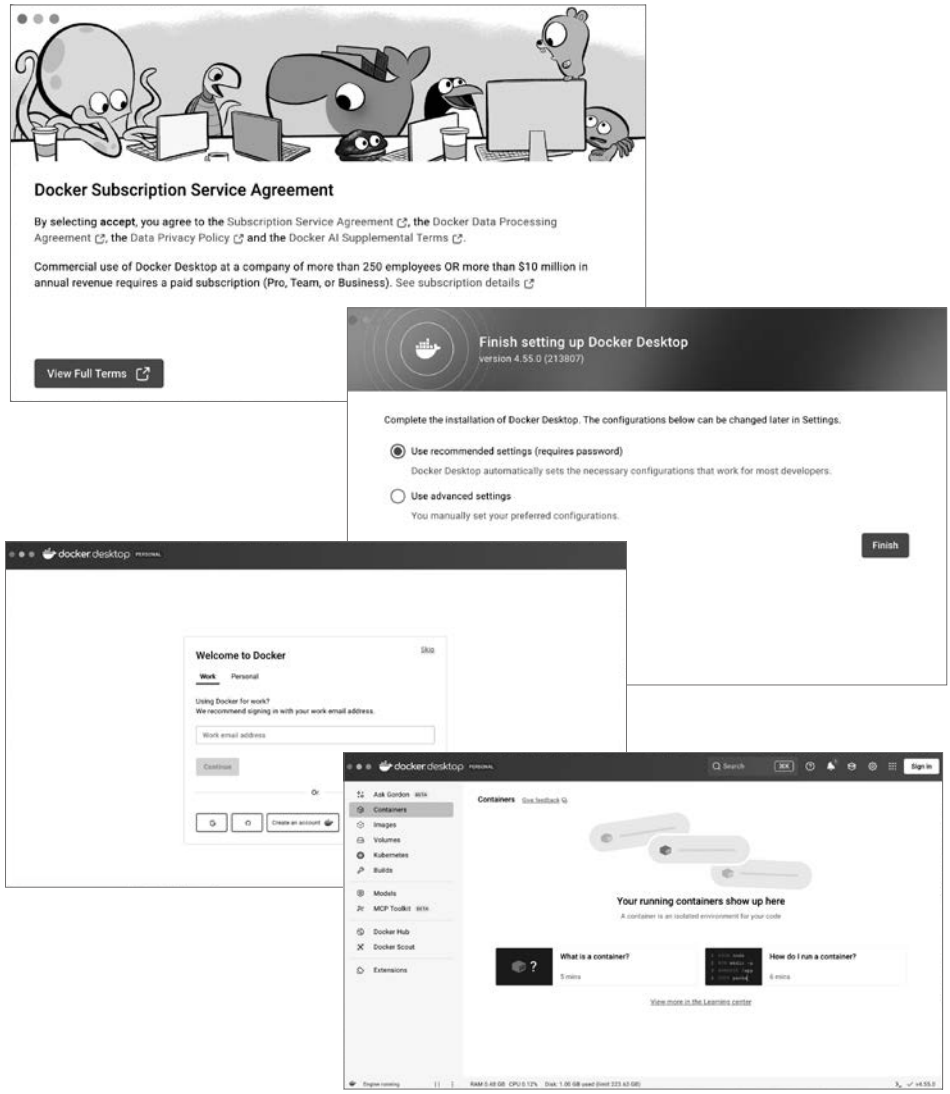
Docker 공식 웹사이트(https://www.docker.com/products/docker-desktop/)에 접속하여 본인의 Mac 사양(Apple Silicon 또는 Intel Chip)에 맞는 버전을 선택하여 다운로드한다.



다운로드한 .dmg 파일을 실행한 후, Docker 아이콘을 Applications 폴더로 드래그하여 설치를 완료한다.



Docker Desktop을 실행하고 서비스 약관 동의(Accept) 및 권장 설정(Use recommended settings)을 완료한다.

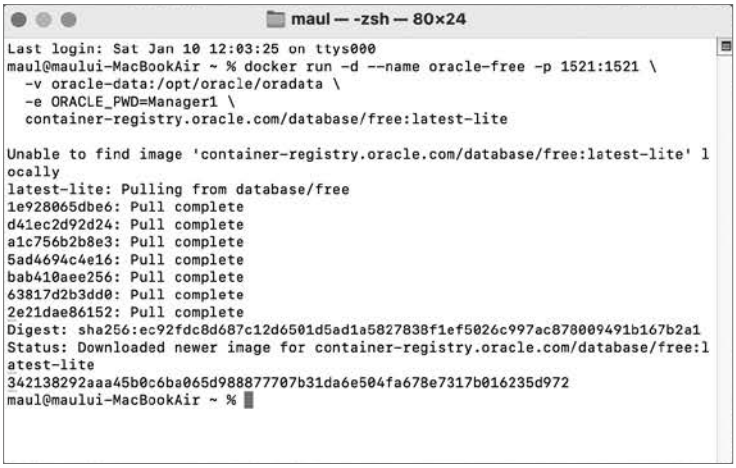


❷ Oracle Database 컨테이너 실행

터미널(Terminal)을 열고 아래 명령어를 입력하여 오라클 컨테이너를 생성하고 실행한다.

```
docker run -d --name oracle-free -p 1521:1521 \
   -v oracle-data:/opt/oracle/oradata \
   -e ORACLE_PWD=Manager1 \
   container-registry.oracle.com/database/free:latest-lite
```

TIP Apple Silicon Mac에서는 latest-lite 또는 latest-l 태그를 사용해야 ARM64 아키텍처가 지원된다.

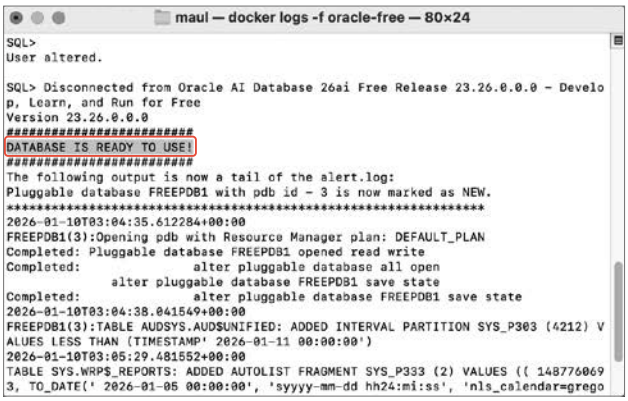


❸ 데이터베이스 준비 상태 확인

컨테이너가 내부적으로 데이터베이스 생성을 마칠 때까지 기다려야 한다. 아래 명령어로 로그를 확인한다.

```
docker logs -f oracle-free
```

화면에 'DATABASE IS READY TO USE!'라는 메시지가 나타나면 데이터베이스가 완전히 준비된 것이다. 이때 Ctrl+C를 눌러 로그 보기를 종료한다.



❹ 접속 정보

SQL Developer 등 외부 도구에서 접속할 때 아래 정보를 사용한다.

호스트 이름	본인 PC의 IP 또는 localhost(리스너 설정한 경우)
포트 번호	1521
접속 이름	FREEPDB1
사용자 이름	system
비밀번호	Manager1 (ORACLE_PWD로 설정한 값)

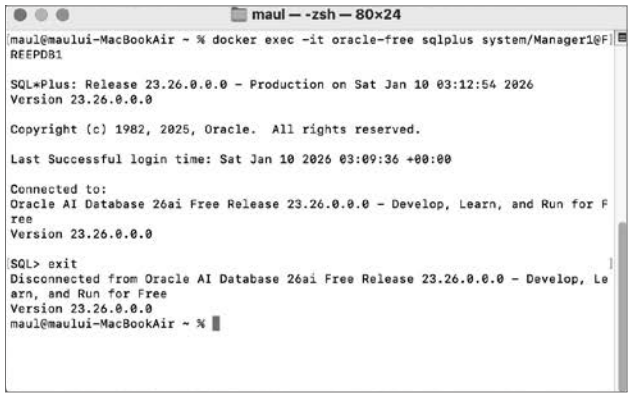
❺ 컨테이너 관리 명령어

독커 컨테이너를 관리할 때 주로 사용하는 명령어는 다음과 같다.

```
# 컨테이너 시작
docker start oracle-free

# 컨테이너 중지
docker stop oracle-free

# 컨테이너 상태 확인
docker ps -a
```

TIP 생성된 데이터는 oracle-data 볼륨에 저장되므로, 컨테이너를 삭제하거나 재시작해도 데이터는 유지된다.

```
# SQL*Plus 접속(컨테이너 내부)
docker exec -it oracle-free sqlplus system/Manager1@FREEPDB1
```



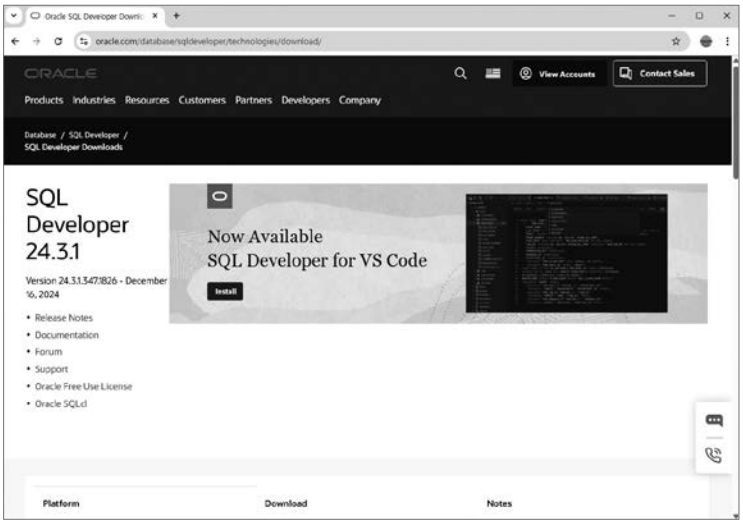
B 오라클 개발 도구 및 샘플 데이터베이스 설치

1 SQL Developer

SQL Developer는 Oracle에서 제공하는 무료 통합 개발 환경(IDE)이다. SQL 작성, 데이터 조회, 데이터베이스 관리 등 다양한 기능을 제공하며, Windows, macOS, Linux 환경에서 모두 사용할 수 있다.

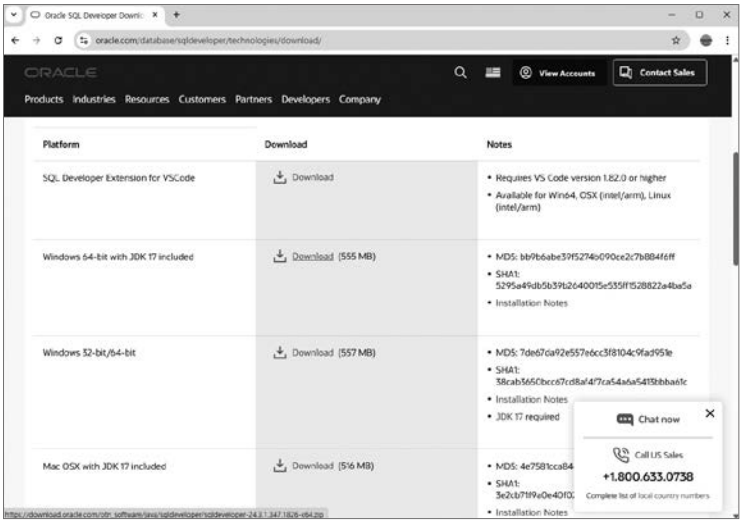
다운로드 및 설치

❶ Oracle SQL Developer 다운로드 페이지(https://www.oracle.com/database/sqldeveloper/technologies/download/)에 접속한다.



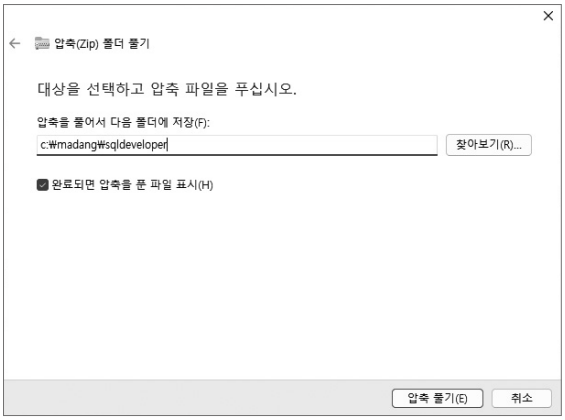
❷ 운영체제에 맞는 버전을 다운로드한다.

- **Windows** : Windows 64-bit with JDK 17 included
- **macOS(Apple Silicon)** : Mac ARM64 버전

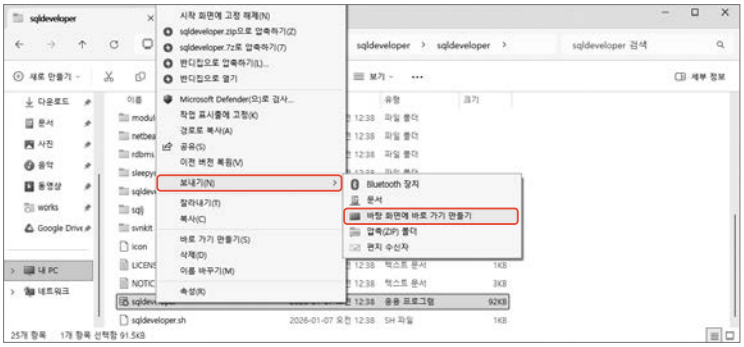


TIP JDK가 포함된 버전(with JDK included)을 다운로드하면 별도의 Java 설치가 필요 없다.

❸ 다운로드한 zip 파일의 압축을 C:\madang\sqldeveloper 폴더에 해제한다.

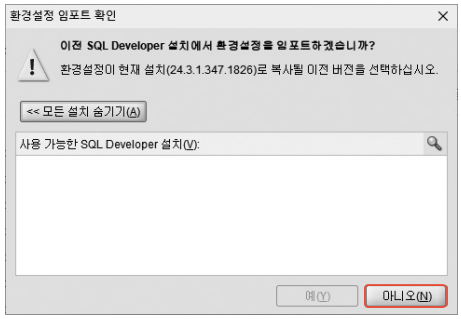


❹ sqldeveloper 폴더 안의 sqldeveloper.exe 파일을 오른쪽 마우스로 클릭하여 바탕화면에 바로가기를 생성한 후 실행한다.

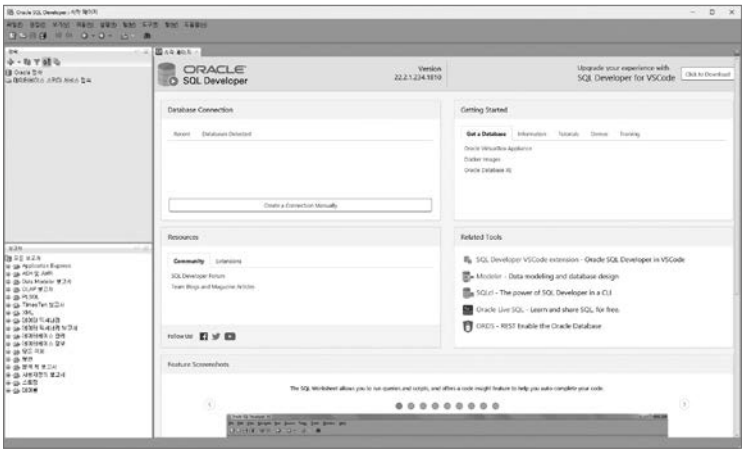


TIP Windows 11의 경우, 해당 파일을 오른쪽 마우스로 클릭한 후 [추가 옵션 표시] → [보내기] → [바탕화면에 바로 가기 만들기]를 클릭하여 바로가기를 생성한 후 실행하면 편리하다.

❺ 프로그램이 처음 실행되면서 환경설정 임포트 여부를 묻는 창이 나타나면 [아니오]를 선택한다.

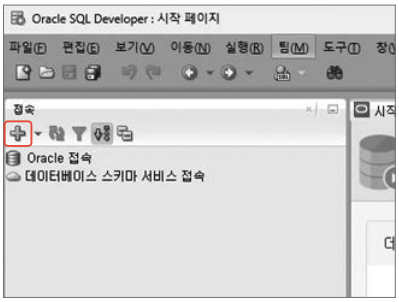


❻ 잠시 기다리면 SQL Developer의 메인 화면(Welcome Page)이 나타나며, 이제 데이터베이스 접속 설정을 진행할 준비가 완료된다.



접속 설정

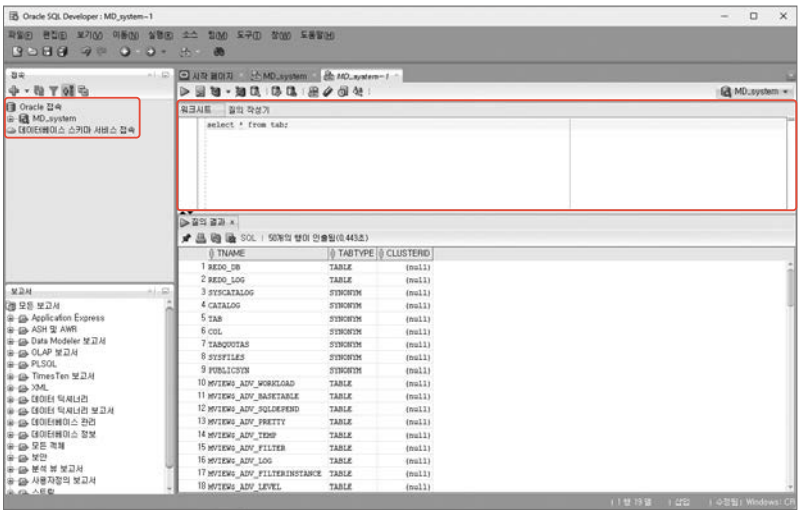
❶ SQL Developer가 실행되면 좌측의 [접속(Connections)] 패널에서 [+] 버튼을 클릭한다.



❷ 다음과 같이 접속 정보를 입력한 뒤 [테스트] 버튼을 클릭하여 좌측 하단에 '상태: 성공'이 표시되는지 확인하고 [접속] 버튼을 클릭한다.

접속 이름	MD_system
사용자 이름	system
비밀번호	Manager1 (비밀번호 저장 체크)
호스트 이름	오라클이 설치된 PC의 IP 또는 localhost(리스너 설정한 경우)
포트	1521
서비스 이름	FREE

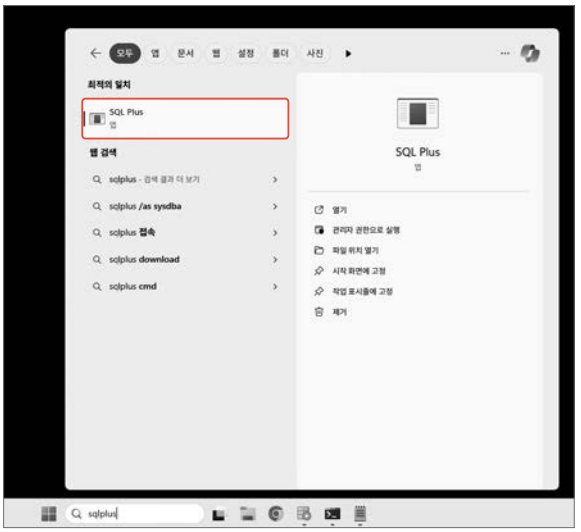
❸ 접속이 완료되면 좌측 [Connections] 패널에 MD_system 항목이 생성되며, 우측에 SQL을 작성하고 실행할 수 있는 워크시트 창이 나타난다.



2 SQL*Plus

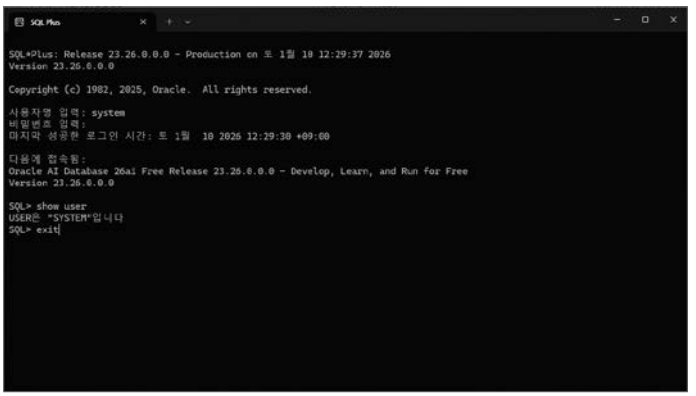
SQL*Plus는 Oracle에서 제공하는 명령줄 기반 SQL 도구이다. Oracle Database 설치 시 자동으로 포함된다.

❶ 시작 메뉴에서 'SQL Plus'를 검색하여 실행한다. 사용자명에 system, 비밀번호에 Manager1을 입력하여 접속한다.



❷ 기본적인 명령어는 다음과 같다.

```
SHOW USER              -- 현재 사용자 확인
SELECT * FROM tab;     -- 테이블 목록 조회
DESC 테이블명          -- 테이블 구조 확인
EXIT                   -- 종료
```



3 madang 사용자 계정과 샘플 데이터 설치

본 교재에서 사용하는 마당서점 예제 데이터베이스를 설치한다. 먼저 시스템(system) 계정으로 접속하여 madang 사용자 계정을 생성한 후, 해당 계정에 샘플 데이터를 입력한다.

다음은 madang 사용자 계정을 생성하기 위한 스크립트이다.

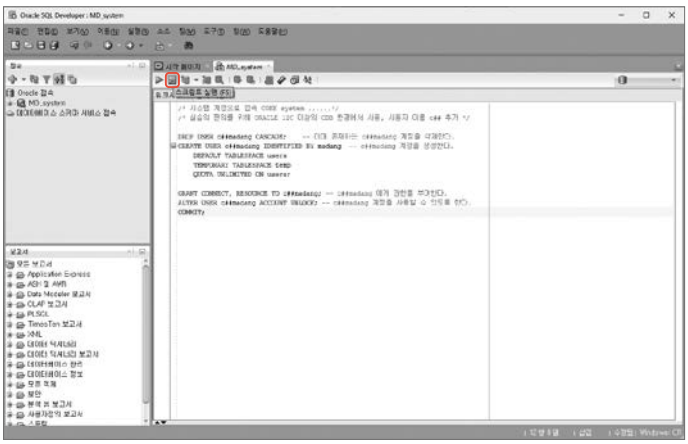
demo_madang_system.sql

```
DROP USER c##madang CASCADE;                -- 이미 존재하는 c##madang 계정을 삭제한다.
CREATE USER c##madang IDENTIFIED BY madang -- c##madang 계정을 생성한다.
   DEFAULT TABLESPACE users
   TEMPORARY TABLESPACE temp
   QUOTA UNLIMITED ON users;
GRANT CONNECT, RESOURCE TO c##madang;       -- c##madang 계정에 권한을 부여한다.
ALTER USER c##madang ACCOUNT UNLOCK;        -- c##madang 계정을 사용할 수 있도록 한다.
COMMIT;
```

madang 사용자 계정 생성

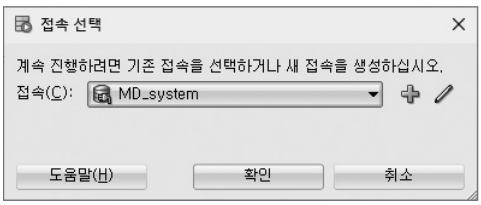
❶ SQL Developer에서 앞에서 설정한 MD_system 접속으로 로그인한다.

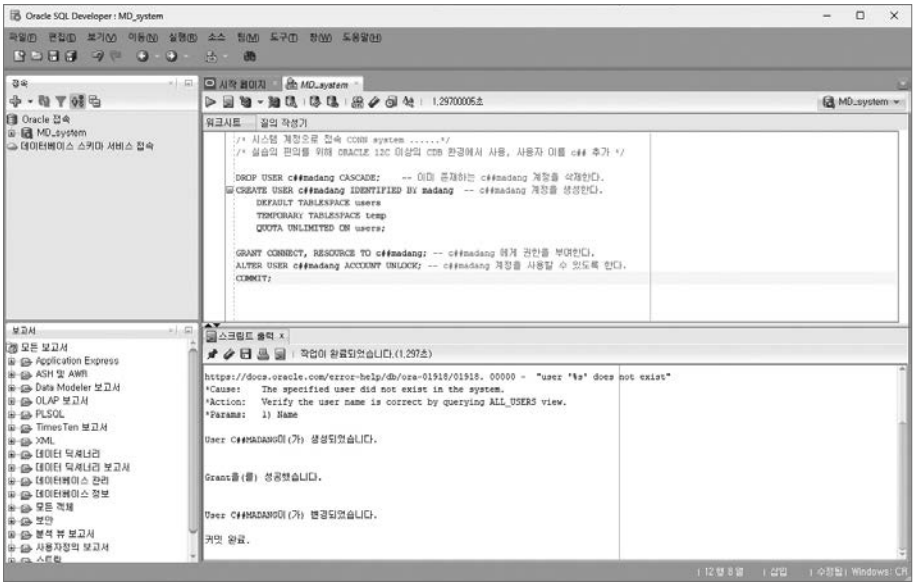
❷ demo_madang_system.sql 파일을 열고 스크립트를 실행한다(F5 또는 (스크립트실행) 버튼).



▲ madang 사용자 생성 스크립트 실행 화면

이때 접속 선택 창이 나타나면 MD_system 접속을 선택한다.





▲ MD_system으로 접속하여 스크립트를 실행한 결과

madang 샘플 데이터 설치

❶ madang 계정이 생성되면, 아래 정보를 입력하여 새 접속(MD_madang)을 생성한다.

접속 이름	MD_madang
사용자 이름	c##madang
비밀번호	madang / 비밀번호 저장 체크
호스트 이름	오라클이 설치된 PC의 IP 또는 localhost(리스너 설정한 경우)
포트	1521
서비스 이름	FREE

❷ [테스트] 버튼을 클릭해 접속 성공을 확인한 뒤 [저장], [접속]을 차례로 클릭한다.

❸ MD_madang 접속 워크시트에서 demo_madang.sql 파일을 열고 스크립트를 실행([F5])한다.

demo_madang.sql

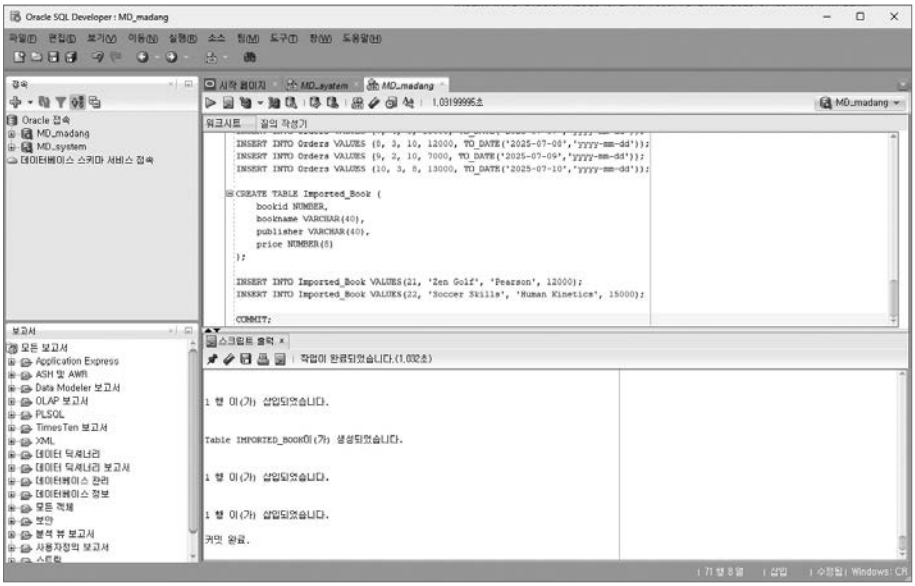
```
/* c##madang 으로 접속해서 실행한다. */
/* SQL*Plus 명령창에서 실행할 경우 다음과 같이 접속 */
/* conn c##madang/madang; */

DROP TABLE Orders CASCADE CONSTRAINT;
```

```
DROP TABLE Book CASCADE CONSTRAINT;
DROP TABLE Customer CASCADE CONSTRAINT;
DROP TABLE Imported_Book CASCADE CONSTRAINT;

CREATE TABLE Book (
   bookid NUMBER(2) PRIMARY KEY,
   bookname VARCHAR2(40),
   publisher VARCHAR2(40),
   price NUMBER(8)
);

CREATE TABLE Customer (
   custid NUMBER(2) PRIMARY KEY,
   name VARCHAR2(40),
   address VARCHAR2(50),
   phone VARCHAR2(20)
);

CREATE TABLE Orders (
   orderid NUMBER(2) PRIMARY KEY,
   custid NUMBER(2) REFERENCES Customer(custid),
   bookid NUMBER(2) REFERENCES Book(bookid),
   saleprice NUMBER(8),
   orderdate DATE
);

/* Book, Customer, Orders 데이터 생성 */
INSERT INTO Book VALUES(1, '축구의 역사', '굿스포츠', 7000);
INSERT INTO Book VALUES(2, '축구 아는 여자', '나무수', 13000);
INSERT INTO Book VALUES(3, '축구의 이해', '대한미디어', 22000);
INSERT INTO Book VALUES(4, '골프 바이블', '대한미디어', 35000);
INSERT INTO Book VALUES(5, '피겨 교본', '굿스포츠', 8000);
INSERT INTO Book VALUES(6, '배구 단계별기술', '굿스포츠', 6000);
INSERT INTO Book VALUES(7, '야구의 추억', '이상미디어', 20000);
INSERT INTO Book VALUES(8, '야구를 부탁해', '이상미디어', 13000);
INSERT INTO Book VALUES(9, '올림픽 이야기', '삼성당', 7500);
INSERT INTO Book VALUES(10, 'Olympic Champions', 'Pearson', 13000);

INSERT INTO Customer VALUES (1, '박지성', '영국 맨체스터', '000-5000-0001');
INSERT INTO Customer VALUES (2, '김연아', '대한민국 서울', '000-6000-0001');
INSERT INTO Customer VALUES (3, '김연경', '대한민국 경기도', '000-7000-0001');
INSERT INTO Customer VALUES (4, '추신수', '미국 클리블랜드', '000-8000-0001');
```

```
INSERT INTO Customer VALUES (5, '박세리', '대한민국 대전', NULL);

INSERT INTO Orders VALUES (1, 1, 1, 6000, TO_DATE('2025-07-01','yyyy-mm-dd'));
INSERT INTO Orders VALUES (2, 1, 3, 21000, TO_DATE('2025-07-03','yyyy-mm-dd'));
INSERT INTO Orders VALUES (3, 2, 5, 8000, TO_DATE('2025-07-03','yyyy-mm-dd'));
INSERT INTO Orders VALUES (4, 3, 6, 6000, TO_DATE('2025-07-04','yyyy-mm-dd'));
INSERT INTO Orders VALUES (5, 4, 7, 20000, TO_DATE('2025-07-05','yyyy-mm-dd'));
INSERT INTO Orders VALUES (6, 1, 2, 12000, TO_DATE('2025-07-07','yyyy-mm-dd'));
INSERT INTO Orders VALUES (7, 4, 8, 13000, TO_DATE('2025-07-07','yyyy-mm-dd'));
INSERT INTO Orders VALUES (8, 3, 10, 12000, TO_DATE('2025-07-08','yyyy-mm-dd'));
INSERT INTO Orders VALUES (9, 2, 10, 7000, TO_DATE('2025-07-09','yyyy-mm-dd'));
INSERT INTO Orders VALUES (10, 3, 8, 13000, TO_DATE('2025-07-10','yyyy-mm-dd'));

CREATE TABLE Imported_Book (
   bookid NUMBER,
   bookname VARCHAR(40),
   publisher VARCHAR(40),
   price NUMBER(8)
);

INSERT INTO Imported_Book VALUES(21, 'Zen Golf', 'Pearson', 12000);
INSERT INTO Imported_Book VALUES(22, 'Soccer Skills', 'Human Kinetics', 15000);

COMMIT;
/* c##madang 으로 접속해서 실행한다. */
/* SQL*Plus 명령창에서 실행할 경우 다음과 같이 접속 */
/* conn c##madang/madang; */

DROP TABLE Orders CASCADE CONSTRAINT;
DROP TABLE Book CASCADE CONSTRAINT;
DROP TABLE Customer CASCADE CONSTRAINT;
DROP TABLE Imported_Book CASCADE CONSTRAINT;

CREATE TABLE Book (
   bookid NUMBER(2) PRIMARY KEY,
   bookname VARCHAR2(40),
   publisher VARCHAR2(40),
   price NUMBER(8)
);

CREATE TABLE Customer (
  custid NUMBER(2) PRIMARY KEY,
```

```
  name VARCHAR2(40),
  address VARCHAR2(50),
  phone VARCHAR2(20)
);

CREATE TABLE Orders (
  orderid NUMBER(2) PRIMARY KEY,
  custid NUMBER(2) REFERENCES Customer(custid),
  bookid NUMBER(2) REFERENCES Book(bookid),
  saleprice NUMBER(8),
  orderdate DATE
);

/* Book, Customer, Orders 데이터 생성 */
INSERT INTO Book VALUES(1, '축구의 역사', '굿스포츠', 7000);
INSERT INTO Book VALUES(2, '축구 아는 여자', '나무수', 13000);
INSERT INTO Book VALUES(3, '축구의 이해', '대한미디어', 22000);
INSERT INTO Book VALUES(4, '골프 바이블', '대한미디어', 35000);
INSERT INTO Book VALUES(5, '피겨 교본', '굿스포츠', 8000);
INSERT INTO Book VALUES(6, '배구 단계별기술', '굿스포츠', 6000);
INSERT INTO Book VALUES(7, '야구의 추억', '이상미디어', 20000);
INSERT INTO Book VALUES(8, '야구를 부탁해', '이상미디어', 13000);
INSERT INTO Book VALUES(9, '올림픽 이야기', '삼성당', 7500);

INSERT INTO Book VALUES(10, 'Olympic Champions', 'Pearson', 13000);
INSERT INTO Customer VALUES (1, '박지성', '영국 맨체스터', '000-5000-0001');
INSERT INTO Customer VALUES (2, '김연아', '대한민국 서울', '000-6000-0001');
INSERT INTO Customer VALUES (3, '김연경', '대한민국 경기도', '000-7000-0001');
INSERT INTO Customer VALUES (4, '추신수', '미국 클리블랜드', '000-8000-0001');
INSERT INTO Customer VALUES (5, '박세리', '대한민국 대전', NULL);

INSERT INTO Orders VALUES (1, 1, 1, 6000, TO_DATE('2025-07-01','yyyy-mm-dd'));
INSERT INTO Orders VALUES (2, 1, 3, 21000, TO_DATE('2025-07-03','yyyy-mm-dd'));
INSERT INTO Orders VALUES (3, 2, 5, 8000, TO_DATE('2025-07-03','yyyy-mm-dd'));
INSERT INTO Orders VALUES (4, 3, 6, 6000, TO_DATE('2025-07-04','yyyy-mm-dd'));
INSERT INTO Orders VALUES (5, 4, 7, 20000, TO_DATE('2025-07-05','yyyy-mm-dd'));
INSERT INTO Orders VALUES (6, 1, 2, 12000, TO_DATE('2025-07-07','yyyy-mm-dd'));
INSERT INTO Orders VALUES (7, 4, 8, 13000, TO_DATE('2025-07-07','yyyy-mm-dd'));
INSERT INTO Orders VALUES (8, 3, 10, 12000, TO_DATE('2025-07-08','yyyy-mm-dd'));
INSERT INTO Orders VALUES (9, 2, 10, 7000, TO_DATE('2025-07-09','yyyy-mm-dd'));
INSERT INTO Orders VALUES (10, 3, 8, 13000, TO_DATE('2025-07-10','yyyy-mm-dd'));
```

```
CREATE TABLE Imported_Book (
   bookid NUMBER,
   bookname VARCHAR(40),
   publisher VARCHAR(40),   price NUMBER(8)
);

INSERT INTO Imported_Book VALUES(21, 'Zen Golf', 'Pearson', 12000);
INSERT INTO Imported_Book VALUES(22, 'Soccer Skills', 'Human Kinetics', 15000);

COMMIT;
```



4 SCOTT 사용자 계정과 샘플 데이터 설치

Oracle의 전통적인 학습용 계정인 SCOTT 사용자와 샘플 데이터를 설치한다.

SCOTT 사용자 계정 생성

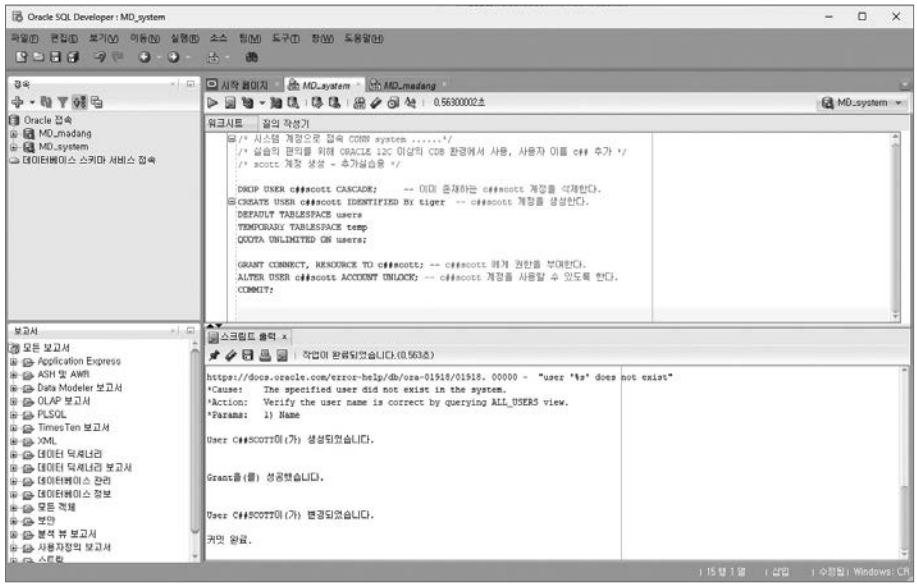
❶ SQL Developer에서 앞에서 설정한 MD_system 접속으로 로그인한다.

❷ demo_scott_system.sql 파일을 열고 스크립트를 실행한다(F5 또는 (스크립트실행) 버튼).

demo_scott_system.sql

```
/* 시스템 계정으로 접속 CONN system ..... */
/* 실습의 편의를 위해 ORACLE 12C 이상의 CDB 환경에서 사용, 사용자 이름 앞에 c## 추가 */
```

```
/* scott 계정 생성 - 추가실습용 */

DROP USER c##scott CASCADE;                  -- 이미 존재하는 c##scott 계정을 삭제한다.
CREATE USER c##scott IDENTIFIED BY tiger     -- c##scott 계정을 생성한다.
DEFAULT TABLESPACE users
TEMPORARY TABLESPACE tempQUOTA UNLIMITED ON users;

GRANT CONNECT, RESOURCE TO c##scott;         -- c##scott에게 권한을 부여한다.
ALTER USER c##scott ACCOUNT UNLOCK;          -- c##scott 계정을 사용할 수 있도록 한다.
COMMIT;
```



SCOTT 샘플 데이터 설치

❶ scott 계정이 생성되면, 아래 정보를 입력하여 새 접속(MD_scott)을 생성한다.

접속 이름	MD_scott
사용자 이름	c##scott
비밀번호	tiger / 비밀번호 저장 체크
호스트 이름	오라클이 설치된 PC의 IP 또는 localhost(리스너 설정한 경우)
포트	1521
서비스 이름	FREE

❷ [테스트] 버튼을 클릭하여 접속 성공을 확인한 후 [저장], [접속]을 차례로 클릭한다.

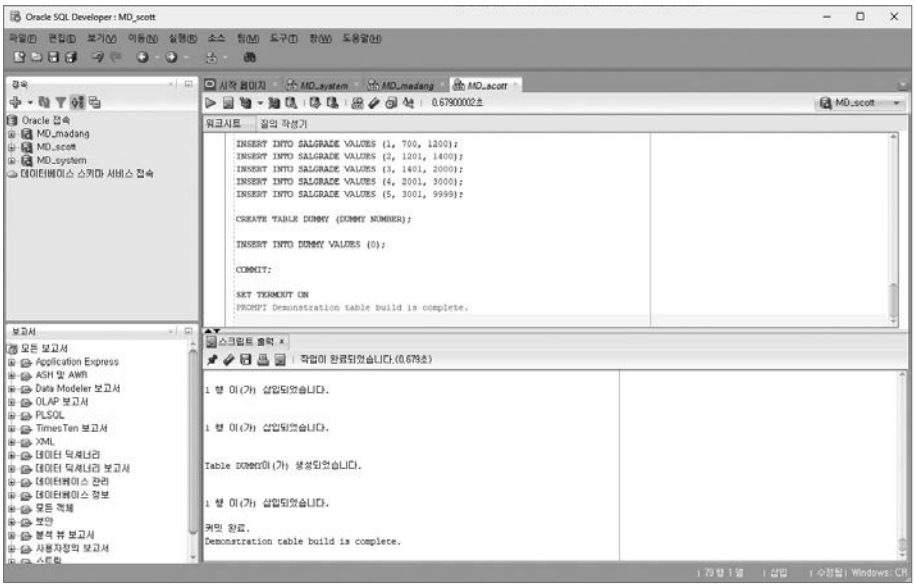
❸ demo_scott.sql 파일을 열고 스크립트를 실행한다(F5 또는 (스크립트실행) 버튼).

scott 샘플 데이터 스크립트: demo_scott.sql

```
/* 시스템 계정으로 접속 CONN system ......*/
/* 실습의 편의를 위해 ORACLE 12C 이상의 CDB 환경에서 사용, 사용자 이름 c## 추가 */
/* scott 계정 생성 - 추가실습용 */
/* c##scott로 접속해서 실행한다. */
/* SQL*Plus 명령창에서 실행할 경우 다음과 같이 접속 */
/* conn c##scott/tiger; */

SET TERMOUT ON
PROMPT Building demonstration tables. Please wait.
SET TERMOUT OFF

DROP TABLE   EMP;
DROP TABLE   DEPT;
DROP TABLE   BONUS;
DROP TABLE   SALGRADE;
DROP TABLE   DUMMY;

CREATE TABLE EMP (
         EMPNO       NUMBER(4) NOT NULL,
         ENAME       VARCHAR2(10),
         JOB         VARCHAR2(9),
         MGR         NUMBER(4),
         HIREDATE    DATE,
         SAL         NUMBER(7, 2),
         COMM        NUMBER(7, 2),
         DEPTNO      NUMBER(2)
         );

INSERT INTO EMP VALUES (7369, 'SMITH', 'CLERK', 7902,TO_DATE('1980-12-17', 'YYYY-
   MM-DD'), 800, NULL, 20);
INSERT INTO EMP VALUES (7499, 'ALLEN', 'SALESMAN', 7698,TO_DATE('1981-02-20',
   'YYYY-MM-DD'), 1600, 300, 30);
INSERT INTO EMP VALUES (7521, 'WARD', 'SALESMAN', 7698,TO_DATE('1981-02-22',
   'YYYY-MM-DD'), 1250, 500, 30);
INSERT INTO EMP VALUES (7566, 'JONES', 'MANAGER', 7839,TO_DATE('1981-04-02',
   'YYYY-MM-DD'), 2975, NULL, 20);
INSERT INTO EMP VALUES (7654, 'MARTIN', 'SALESMAN', 7698,TO_DATE('1981-09-28',
   'YYYY-MM-DD'), 1250, 1400, 30);
```

```
INSERT INTO EMP VALUES (7698, 'BLAKE', 'MANAGER', 7839,TO_DATE('1981-05-01',
   'YYYY-MM-DD'), 2850, NULL, 30);
INSERT INTO EMP VALUES (7782, 'CLARK', 'MANAGER', 7839,TO_DATE('1981-06-09',
   'YYYY-MM-DD'), 2450, NULL, 10);
INSERT INTO EMP VALUES (7788, 'SCOTT', 'ANALYST', 7566,TO_DATE('1982-12-09',
   'YYYY-MM-DD'), 3000, NULL, 20);
INSERT INTO EMP VALUES (7839, 'KING', 'PRESIDENT', NULL,TO_DATE('1981-11-17',
'YYYY-MM-DD'), 5000, NULL, 10);
INSERT INTO EMP VALUES (7844, 'TURNER', 'SALESMAN', 7698,TO_DATE('1981-09-08',
'YYYY-MM-DD'), 1500, 0, 30);
INSERT INTO EMP VALUES (7876, 'ADAMS', 'CLERK', 7788,TO_DATE('1983-01-12', 'YYYY-
MM-DD'), 1100, NULL, 20);
INSERT INTO EMP VALUES (7900, 'JAMES', 'CLERK', 7698,TO_DATE('1981-12-03', 'YYYY-
MM-DD'), 950, NULL, 30);
INSERT INTO EMP VALUES (7902, 'FORD', 'ANALYST', 7566,TO_DATE('1981-12-03',
'YYYY-MM-DD'), 3000, NULL, 20);
INSERT INTO EMP VALUES (7934, 'MILLER', 'CLERK', 7782,TO_DATE('1982-01-23',
'YYYY-MM-DD'), 1300, NULL, 10);

CREATE TABLE DEPT(
        DEPTNO      NUMBER(2),
        DNAME       VARCHAR2(14),
        LOC         VARCHAR2(13)
        );

INSERT INTO DEPT VALUES (10, 'ACCOUNTING', 'NEW YORK');
INSERT INTO DEPT VALUES (20, 'RESEARCH', 'DALLAS');
INSERT INTO DEPT VALUES (30, 'SALES', 'CHICAGO');
INSERT INTO DEPT VALUES (40, 'OPERATIONS', 'BOSTON');

CREATE TABLE BONUS(
        ENAME       VARCHAR2(10),
        JOB         VARCHAR2(9),
        SAL         NUMBER,
        COMM        NUMBER
        );

CREATE TABLE SALGRADE(
        GRADE NUMBER,
        LOSAL NUMBER,
        HISAL NUMBER
        );
```

```
INSERT INTO SALGRADE VALUES (1, 700, 1200);
INSERT INTO SALGRADE VALUES (2, 1201, 1400);
INSERT INTO SALGRADE VALUES (3, 1401, 2000);
INSERT INTO SALGRADE VALUES (4, 2001, 3000);
INSERT INTO SALGRADE VALUES (5, 3001, 9999);

CREATE TABLE DUMMY (DUMMY NUMBER);

INSERT INTO DUMMY VALUES (0);

COMMIT;

SET TERMOUT ON
PROMPT Demonstration table build is complete.
```



C 파이썬, oracledb 라이브러리, Flask 설치

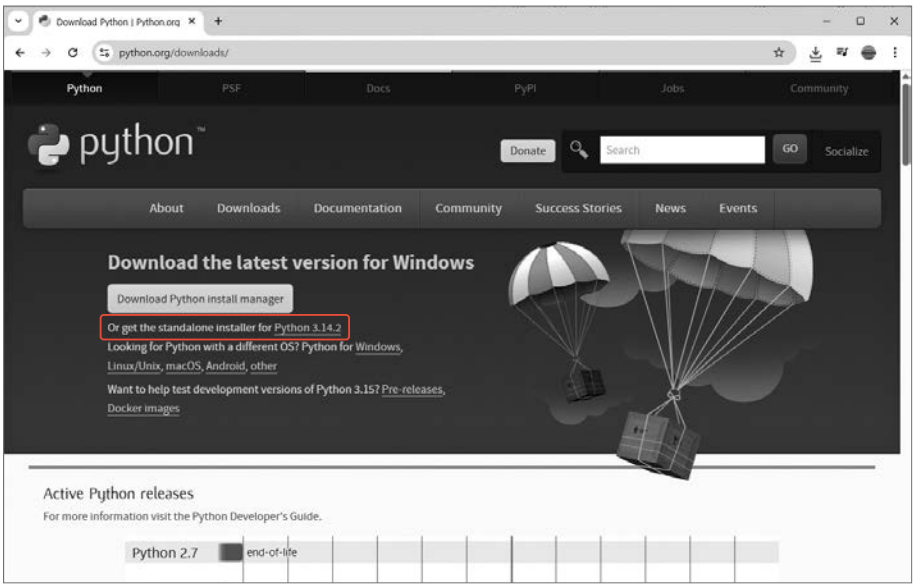
1 파이썬 다운로드 및 설치

파이썬은 데이터베이스 프로그래밍에 널리 사용되는 프로그래밍 언어이다.

다운로드

❶ 파이썬 공식 웹사이트(https://www.python.org/downloads/)에 접속한다.

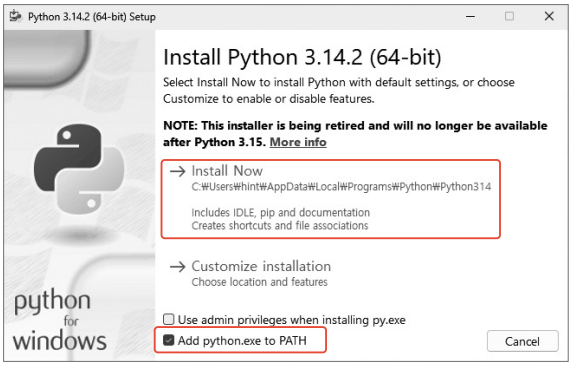
❷ 'Or get the standalone installer for Python 3.14.2'를 클릭하여 최신 버전을 다운로드한다.

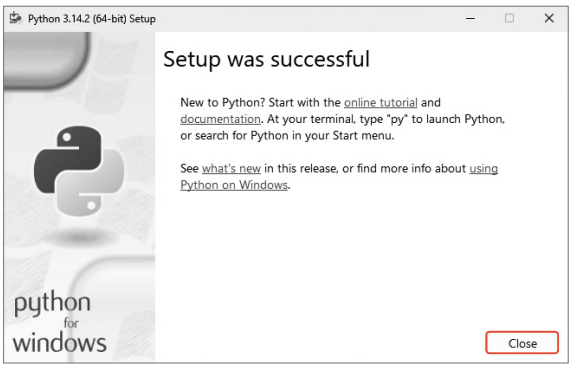


설치 – Windows

❶ 설치 파일을 실행하고 [Add python.exe to PATH] 옵션을 체크한다.

❷ [Install Now]를 클릭하여 설치를 완료한다.





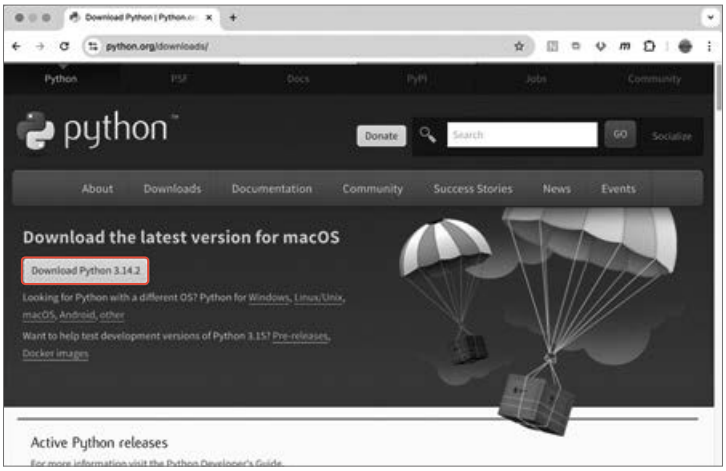
❸ 명령 프롬프트에서 설치 여부를 확인한다.

```
python   --version
pip      --version
```

TIP 명령 프롬프트는 Windows 검색 또는 실행 창(⊞키+ R)에서 cmd를 입력한 후 Enter 키를 누르면 실행할 수 있다.

설치 - macOS

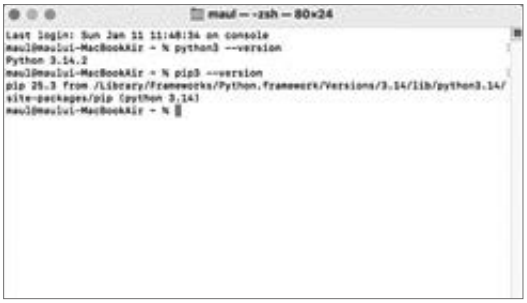
❶ 다운로드한 .pkg 파일을 더블클릭하여 설치한다.



❷ 설치 마법사의 안내에 따라 설치를 진행한다.

❸ 터미널에서 설치 여부를 확인한다.

```
python3      --version
pip3         --version
```



TIP macOS에서는 python 대신 python3, pip 대신 pip3 명령을 사용한다.

2 oracledb 라이브러리

python-oracledb는 Oracle Database에 연결하기 위한 파이썬 드라이버이다. 기존의 cx_Oracle을 대체하는 드라이버로, Thin 모드와 Thick 모드를 지원한다.

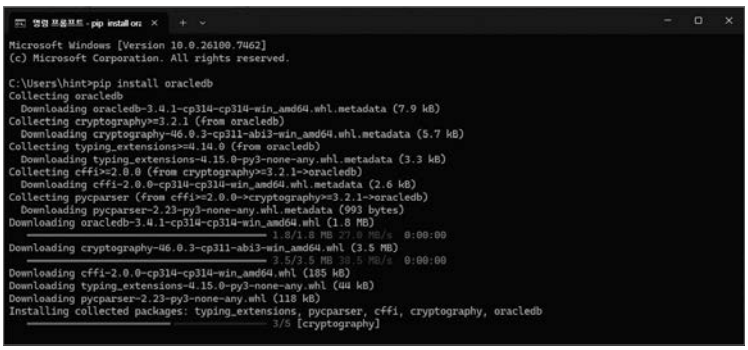
주요 특징은 다음과 같다.

- Thin 모드: Oracle Client 설치 없이 직접 데이터베이스에 연결(기본값)
- Thick 모드: Oracle Client 라이브러리를 사용한 고급 기능 지원
- Python DB-API 2.0 표준 준수
- 비동기(asyncio) 프로그래밍 지원

oracledb 라이브러리 설치

❶ 터미널(또는 cmd)에서 다음 명령을 입력한다.

```
pip install oracledb
```



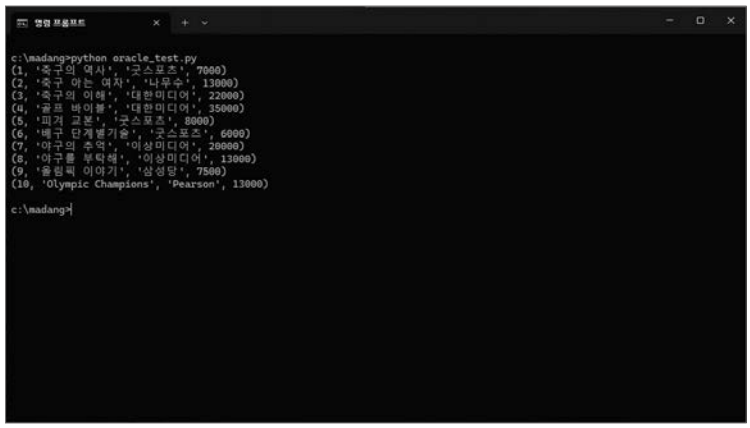
❷ C:₩madang 폴더에 다음 소스코드를 작성하여 저장한다.

연결 테스트 / booklist.py

```
import oracledb

# 데이터베이스 연결
conn = oracledb.connect(
   user='c##madang',
   password='madang',
   dsn='localhost:1521/FREE'
)

# 커서 생성 및 쿼리 실행
cursor = conn.cursor()
cursor.execute('SELECT * FROM Book')

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

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

❸ 명령 프롬프트에서 python booklist.py를 입력하여 도서 목록이 출력되는지 확인한다.

```
c:\madang>python booklist.py
```



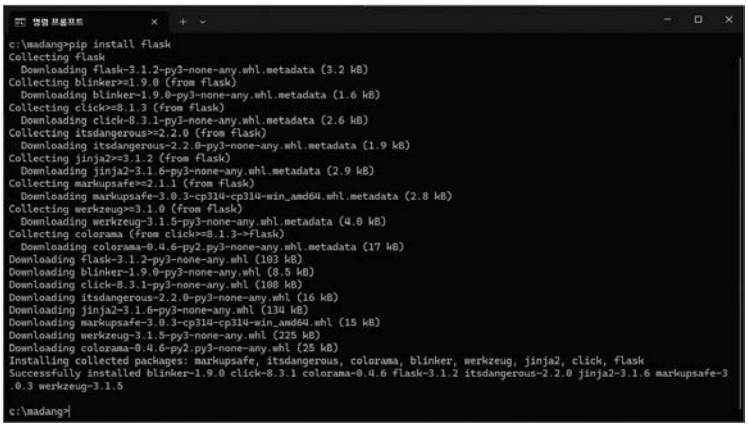
3 Flask

Flask는 파이썬 기반의 경량 웹 프레임워크이다. 데이터베이스와 연동된 웹 애플리케이션을 쉽게 개발할 수 있다.

Flask 설치

pip install flask 명령으로 웹 프레임워크를 설치한다.

```
pip install flask
```



웹 애플리케이션 예제 작성

메모장을 사용하여 madang 폴더에 다음 소스를 저장한다.

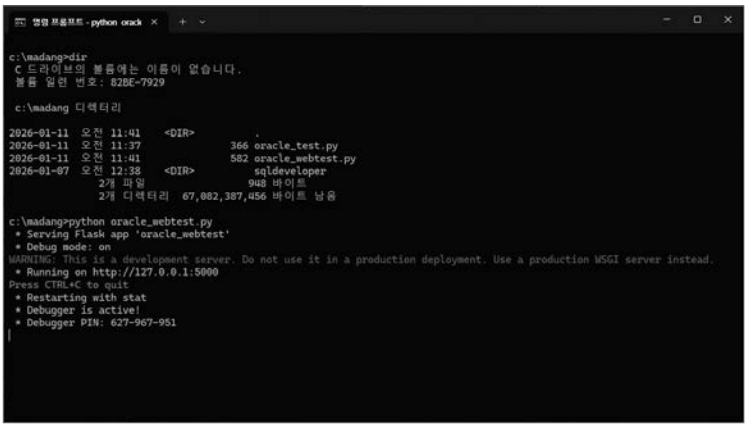
booklist_web.py

```
from flask import Flask
import oracledb

app = Flask(__name__)

@app.route('/')
def index():
   conn = oracledb.connect(
       user='c##madang', password='madang',
       dsn='localhost:1521/FREE')
   cursor = conn.cursor()
   cursor.execute('SELECT bookname, price FROM Book')
   result = '<h1>마당서점 도서목록</h1><ul>'
```

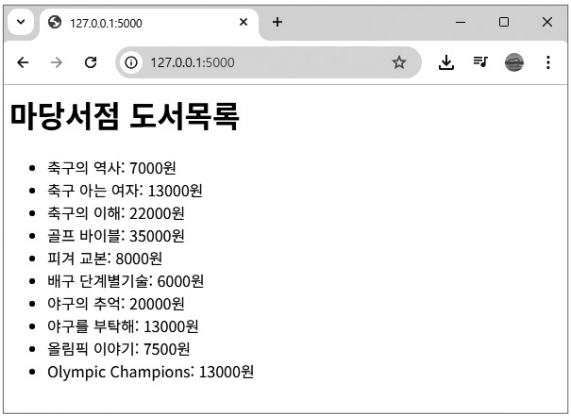
```
    for bookname, price in cursor:
        result += f'<li>{bookname}: {price}원</li>'
    result += '</ul>'
    cursor.close()
    conn.close()
    return result

if __name__ == '__main__':
    app.run(debug=True)
```

실행

```
python booklist_web.py
```



웹 브라우저에서 http://127.0.0.1:5000에 접속하여 실행 결과를 확인한다.



D SQL Developer Data Modeler

1 다운로드

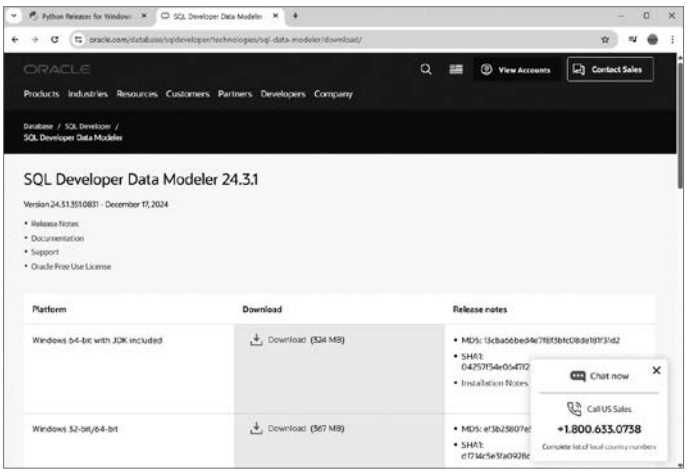
SQL Developer Data Modeler는 오라클에서 제공하는 무료 데이터 모델링 도구이다. ER 다이어그램 작성, 논리적 · 물리적 모델 설계, DDL 스크립트 생성 등의 기능을 제공하며, Windows, macOS, Linux 환경에서 모두 사용할 수 있다.

❶ Oracle SQL Developer Data Modeler 다운로드 페이지(https://www.oracle.com/database/sqldeveloper/technologies/sql-data-modeler/download/)에 접속한다.

❷ 운영체제에 맞는 버전을 선택한다.

- Windows : Windows 64-bit with JDK 17 included
- macOS(Intel) : Mac OSX
- macOS(Apple Silicon) : Mac ARM64

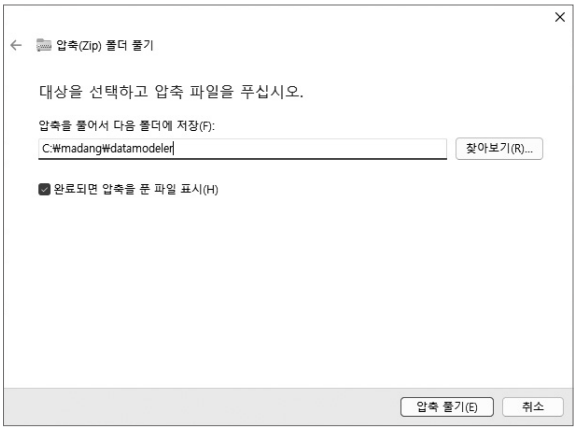
TIP JDK가 포함된 버전을 다운로드하면 Java를 별도로 설치할 필요가 없다.



2 설치

Windows

❶ 다운로드한 .zip 파일의 압축을 C:\madang\datamodeler 폴더에 해제한다.



❷ datamodeler 폴더 안의 datamodeler.exe 파일을 직접 실행하거나, 바로가기를 바탕화면에 생성한 후 실행한다.

macOS

❶ 다운로드한 .dmg 파일을 더블클릭한다.

❷ DataModeler.app을 Applications 폴더로 드래그한다.

❸ Applications 폴더에서 DataModeler를 실행한다.

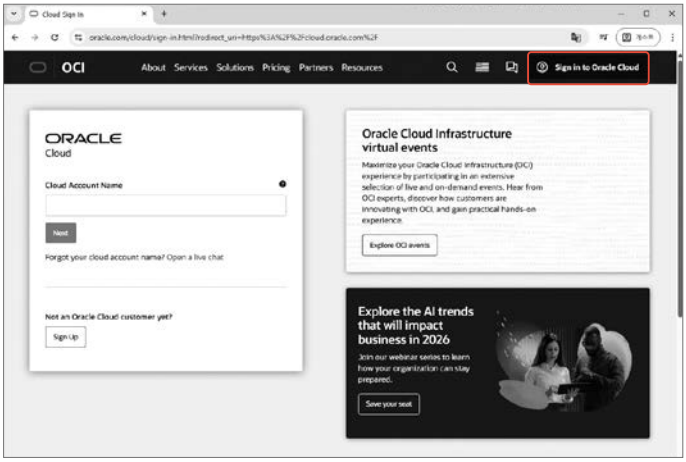
E 오라클 클라우드

1 Oracle Autonomous Database 클라우드 가입

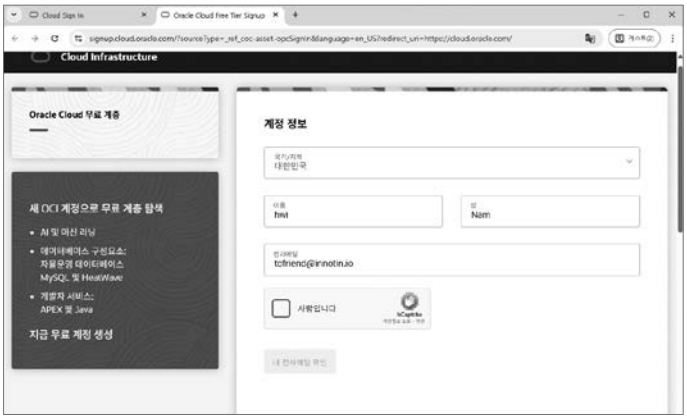
Oracle Cloud Infrastructure(OCI)는 무료로 사용할 수 있는 Always Free Tier를 제공한다. 이를 통해 Autonomous Database를 무료로 생성하여 사용할 수 있다.

클라우드 계정 생성 및 로그인

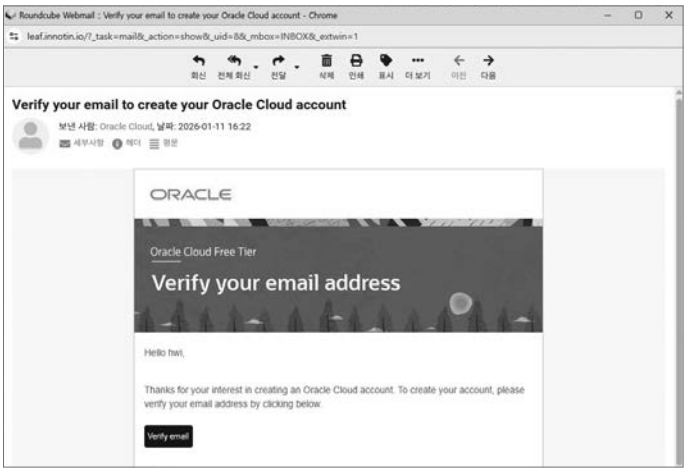
❶ Oracle Cloud 홈페이지(https://cloud.oracle.com)에 접속한 후 [Sign Up]을 클릭한다.



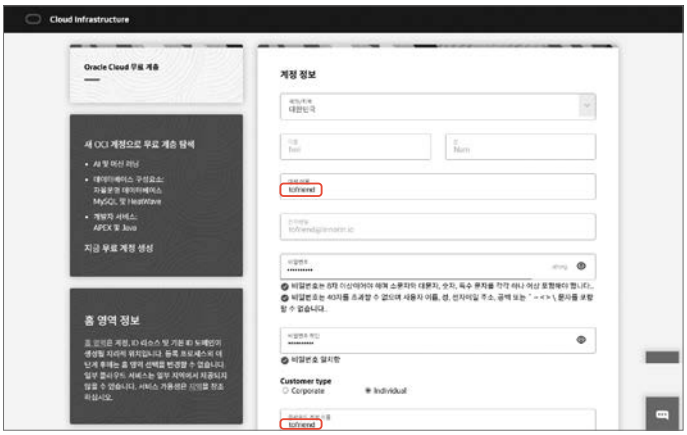
❷ 이메일, 이름, 국가/지역 정보를 입력한다.



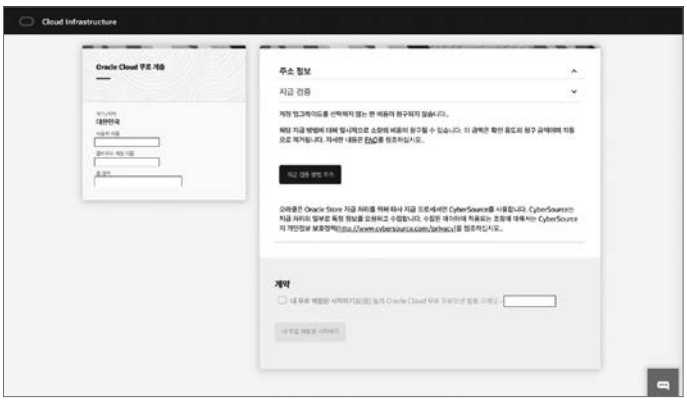
❸ 이메일 인증을 완료한다.



❹ 전송받은 이메일을 확인한 후 Verify email을 클릭하여 가입 절차를 계속 진행한다.

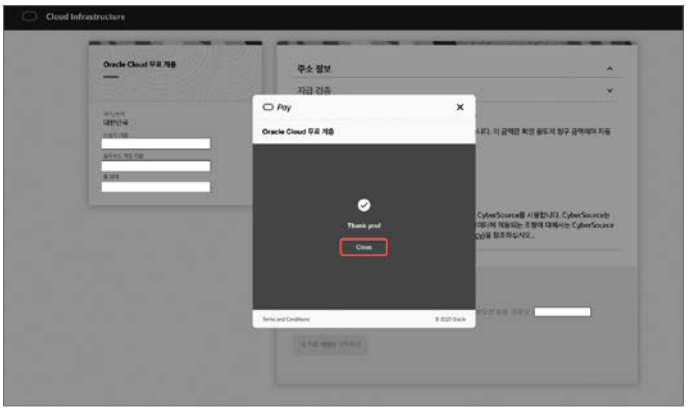


❺ 주소 및 과금 정보를 입력한다.

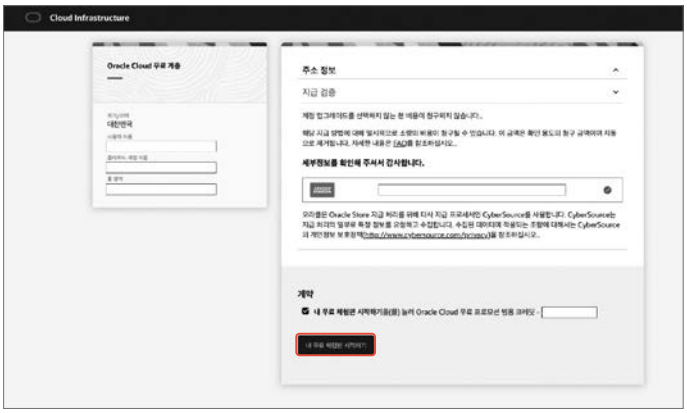


❻ 결제 정보를 입력한다. 카드 정보는 본인의 실명 정보와 동일해야 하며, 카드 인증이 완료되면 인증 완료 화면이 나타난다. 인증 완료 화면에서 [CLOSE]를 클릭한다.

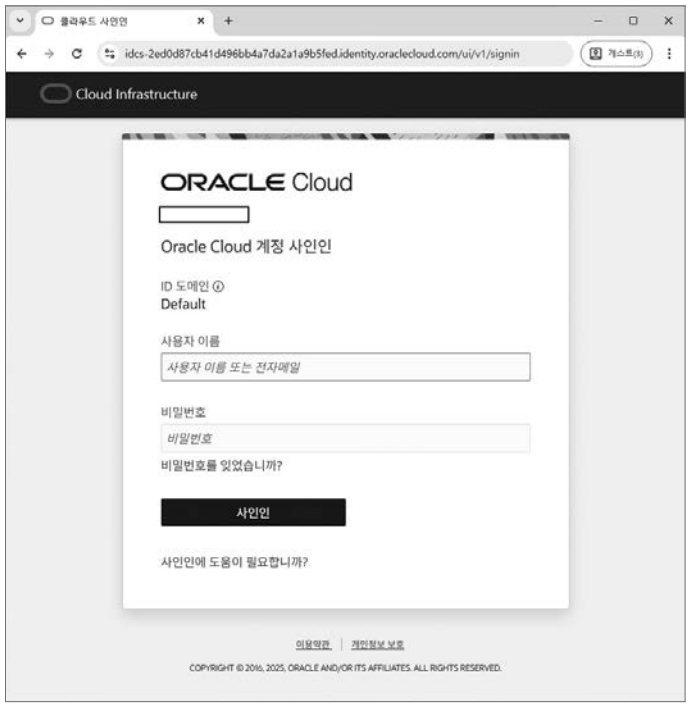
TIP Always Free 서비스만 사용 시 과금되지 않는다.



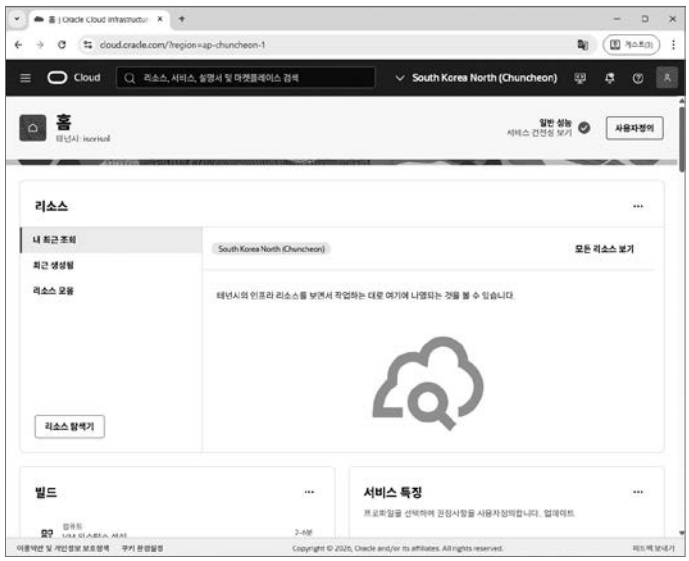
❼ 확인 창에서 [내 무료 체험판 시작하기]를 클릭하면 계정 생성이 완료된다.



❽ 계정 생성이 완료되면 로그인을 진행한다.



❾ 첫 로그인 시 2단계 인증(MFA) 설정을 진행해야 한다. Oracle Authenticator 앱을 설치하거나 다른 인증 방식(FIDO 등)을 사용할 수 있다. 2단계 인증을 완료하면 오라클 클라우드 콘솔에 접속할 수 있다.

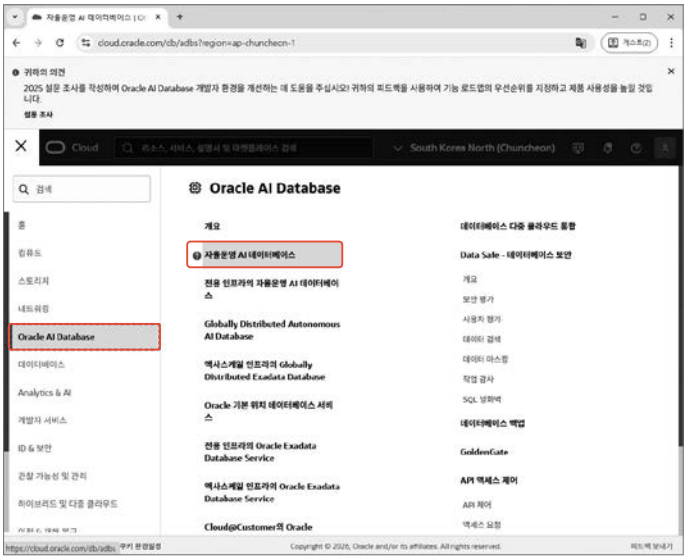


2 데이터베이스 생성

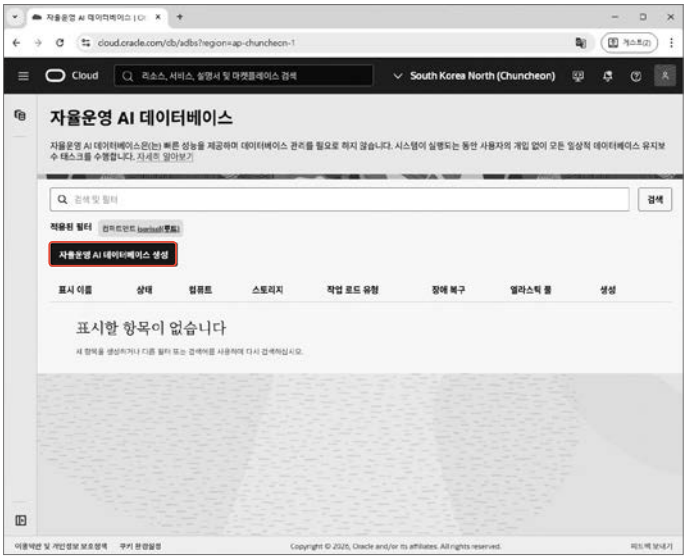
자율운영 Autonomous Database 생성

❶ Oracle Cloud Console (https://cloud.oracle.com)에 로그인한다.

❷ 좌측 메뉴에서 [Oracle AI Database] → [자율운영 AI 데이터베이스]를 선택한다.



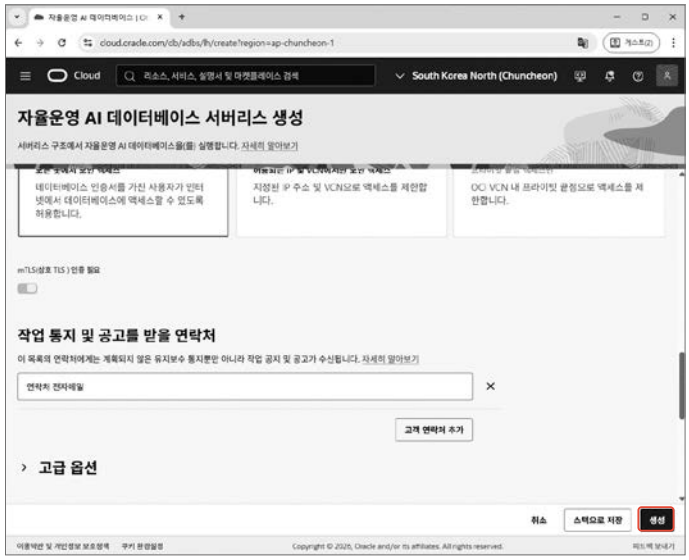
❸ [자율 운영 AI 데이터베이스 생성] 버튼을 클릭한다.



❹ 다음 정보를 입력한다.

Display name	MadangDB
Database name	madangdb
작업 로드 유형(Workload type)	트랜잭션 처리(Transaction Processing)
데이터베이스 구성	Always Free 체크(무료 사용)
데이터베이스 버전	26ai
ADMIN Password	복잡한 비밀번호 설정
네트워크 액세스	모든 곳에서 보안 액세스

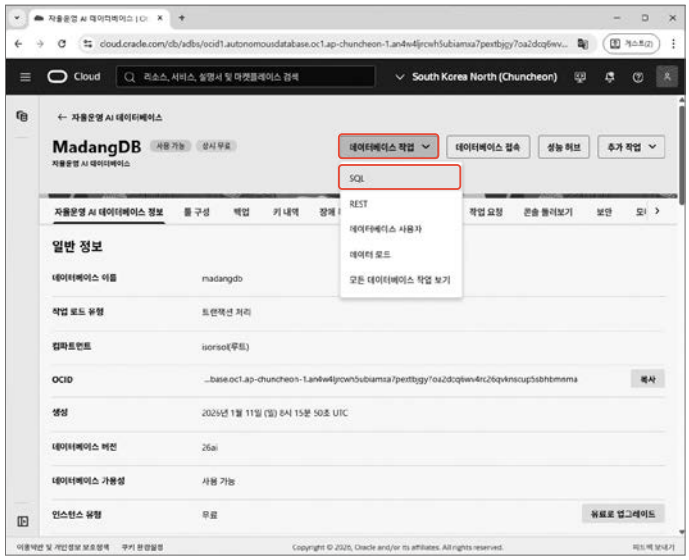
❺ [생성] 버튼을 클릭한다.



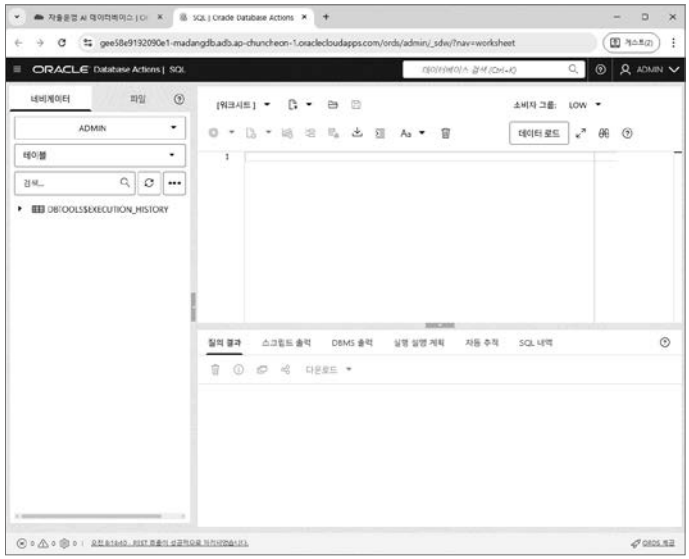
❻ 프로비저닝이 완료되면 상태가 Available로 변경된다.

데이터베이스 작업(Database Actions) 접속

❶ 생성된 데이터베이스의 상세 페이지에서 [데이터베이스 작업] → [SQL]을 클릭한다.

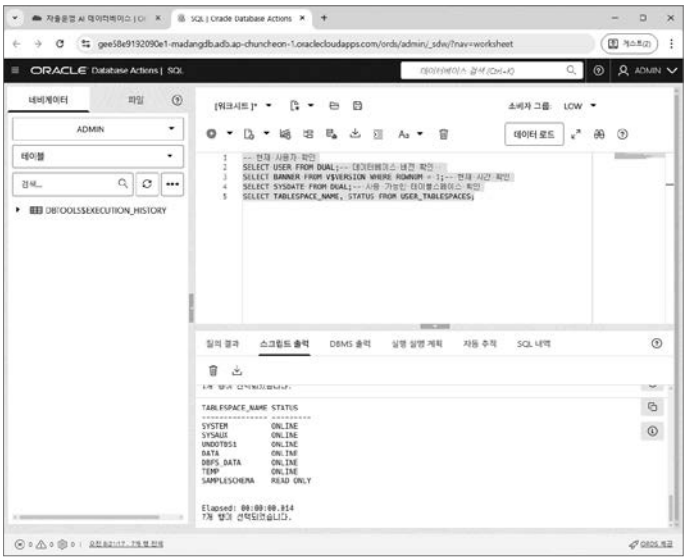


❷ ADMIN 계정으로 로그인한다.



❸ SQL Worksheet에서 다음 SQL 문을 실행하여 정상적으로 생성되었는지 확인한다.

```
-- 현재 사용자 확인
SELECT USER FROM DUAL;          -- 데이터베이스 버전 확인
SELECT BANNER FROM V$VERSION WHERE ROWNUM = 1;       -- 현재 시간 확인
SELECT SYSDATE FROM DUAL;     -- 사용 가능한 테이블스페이스 확인
SELECT TABLESPACE_NAME, STATUS FROM USER_TABLESPACES;
```



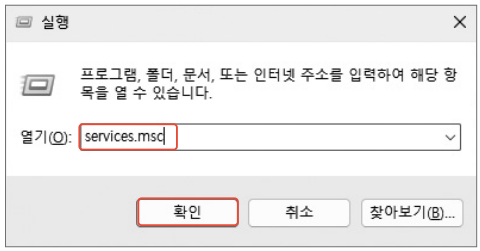
F 데이터베이스 관리

1 데이터베이스 시작과 종료(STARTUP, SHUTDOWN)

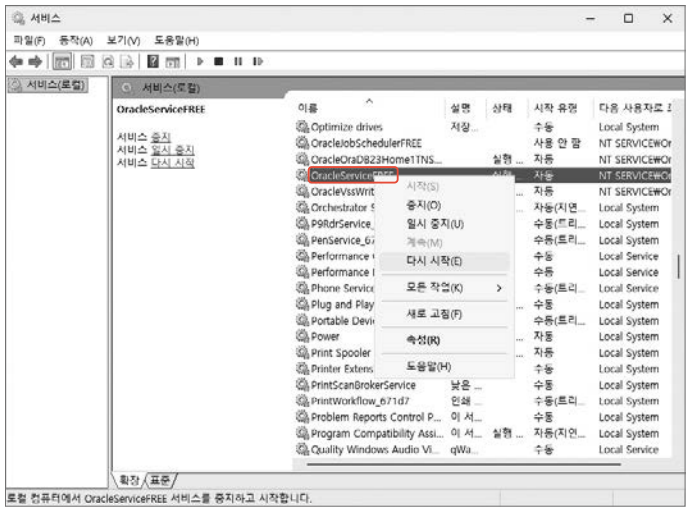
Oracle Database는 Windows 서비스 관리자 또는 명령어를 통해 시작하고 종료할 수 있다.

Windows 서비스 사용

❶ [⊞ 키+R]을 눌러 실행 창을 열고 services.msc를 입력한 뒤 [확인]을 클릭한다.



❷ 서비스 목록에서 OracleServiceFREE를 찾는다.



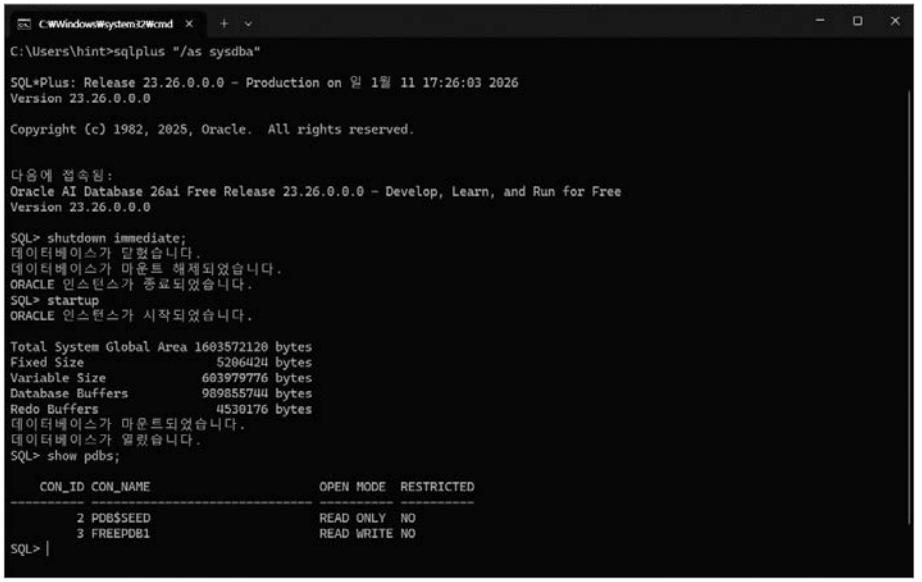
❸ 오른쪽 마우스를 클릭하여 [시작] 또는 [중지]를 선택한다.

TIP OracleOraDBXXHome1TNSListener 서비스도 함께 관리해야 한다. 리스너가 중지되면 원격 접속이 불가능하다.

SQL*Plus 명령어 사용 - cmd

명령 프롬프트에서 SYSDBA 권한으로 접속하여 데이터베이스를 관리한다.

```
sqlplus "/as sysdba"
```



데이터베이스 종료

```
-- 일반 종료 (모든 트랜잭션 완료 후 종료)
SHUTDOWN NORMAL;

-- 즉시 종료 (현재 트랜잭션 롤백 후 종료)
SHUTDOWN IMMEDIATE;

-- 강제 종료 (즉시 종료, 복구 필요할 수 있음)
SHUTDOWN ABORT;
```

데이터베이스 시작

```
-- 일반 시작 (마운트 → 오픈)
STARTUP;

-- 단계별 시작
STARTUP NOMOUNT;        -- 인스턴스만 시작
ALTER DATABASE MOUNT;   -- 데이터베이스 마운트
ALTER DATABASE OPEN;    -- 데이터베이스 오픈
```

PDB(Pluggable Database) 관리

Oracle 12c 이상부터 도입된 멀티테넌트 환경에서는 PDB를 별도로 관리해야 한다.

```
-- PDB 상태 확인
SHOW PDBS;

-- 특정 PDB 오픈
ALTER PLUGGABLE DATABASE FREEPDB1 OPEN;

-- 모든 PDB 오픈
ALTER PLUGGABLE DATABASE ALL OPEN;

-- PDB 자동 시작 설정
ALTER PLUGGABLE DATABASE FREEPDB1 SAVE STATE;
```

2 데이터베이스 관리 도구 – 엔터프라이즈 매니저

Oracle Enterprise Manager Database Express(EM Express)는 오라클 데이터베이스에 내장된 웹 기반 관리 도구이다. 별도의 설치 없이 웹 브라우저를 통해 데이터베이스를 모니터링하고 관리할 수 있다.

EM Express 주요 기능

- 성능 허브: 실시간 및 과거 성능 데이터 모니터링
- 저장소 관리: 테이블스페이스, 데이터파일 관리
- 보안 관리: 사용자, 역할, 권한 관리
- 구성 관리: 초기화 파라미터 설정