

하나증권 : THE NEXT

프로젝트 실행 가이드

CONTENTS

목차

01

개요

프로젝트 파일 링크

02

프론트엔드

프론트엔드 파일 실행 방법

03

백엔드

백엔드 서버 실행 방법

04

데이터베이스

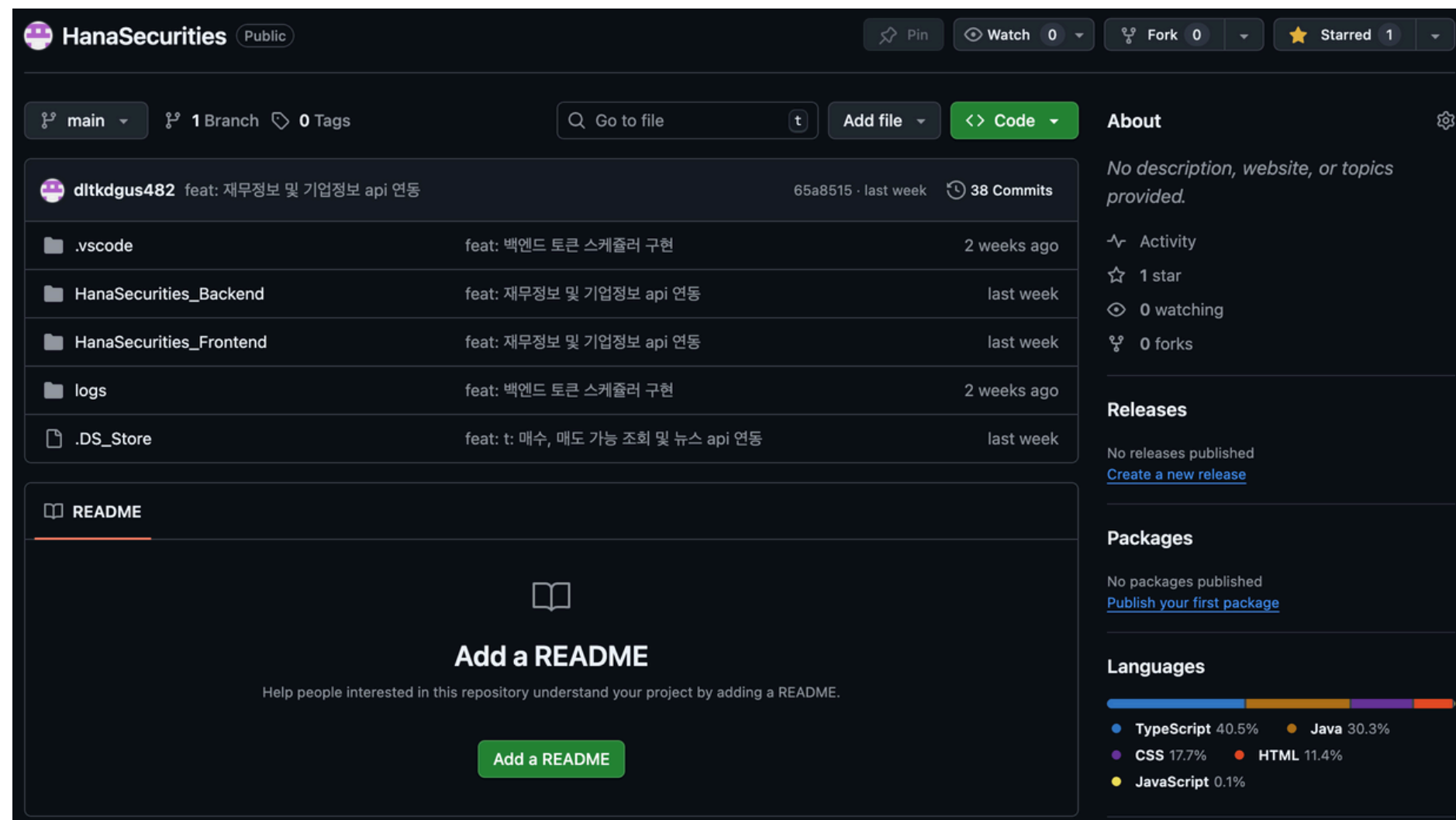
DBMS 실행 및
초기 데이터 삽입 방법

01 개요

프로젝트 파일 링크



1. 프로젝트 전체 파일은 <https://github.com/dltkdgus482/HanaSecurities> 에서 확인하실 수 있습니다. (프론트엔드, 백엔드, DB 및 초기 데이터, 프로젝트 산출물 모두 포함)



[하나증권 : THE NEXT 프로젝트 파일 저장소]

2. 설치 과정 중 오류나 문의사항이 생기시면 언제든지 아래의 이메일로 연락 부탁드립니다.

dltkdgus482@naver.com

02

프론트엔드

프론트엔드 파일 실행 방법



1. 우선, cmd 창에서 `node -v`를 입력해 node.js가 설치되어 있는지 확인하고, 없으면 node.js의 공식 홈페이지 (<https://nodejs.org/ko/>)에 접속하여 node.js를 설치합니다.

```
Last login: Sat Oct 11 12:52:41 on ttys010
sanghyeon@isanghyeon-ui-noteubug ~ % node -v
v23.10.0
```

[node.js가 설치되어 있는지 확인]



[node.js 공식 홈페이지 화면]

2. node.js를 설치하면 npm이 자동으로 설치되기 때문에 cmd 창에서 `npm install -g yarn`을 입력하여 yarn 패키지 매니저를 설치합니다.

```
sanghyeon@isanghyeon-ui-noteubug ~ % npm install -g yarn  
changed 1 package in 866ms
```

[npm으로 yarn을 설치]

3. Github에서 클론받은 파일의 최상단에서 HanaSecurities_Frontend 폴더로 이동하여 터미널 창에서 yarn install을 입력하여 의존성을 설치해줍니다.

```
sanghyeon@isanghyeon-ui-noteubug dev % git clone https://github.com/dltkdgus482/HanaSecurities.git
'HanaSecurities'에 복제합니다...
remote: Enumerating objects: 1475, done.
remote: Counting objects: 100% (101/101), done.
remote: Compressing objects: 100% (77/77), done.
remote: Total 1475 (delta 15), reused 77 (delta 15), pack-reused 1374 (from 1)
오브젝트를 받는 중 : 100% (1475/1475), 45.27 MiB | 4.10 MiB/s, 완료.
델타를 알아내는 중 : 100% (583/583), 완료.
sanghyeon@isanghyeon-ui-noteubug dev % ls
HanaSecurities
sanghyeon@isanghyeon-ui-noteubug dev % cd HanaSecurities/HanaSecurities_Frontend
sanghyeon@isanghyeon-ui-noteubug HanaSecurities_Frontend % yarn install
> YN0088: A new stable version of Yarn is available: 4.10.3!
> YN0088: Upgrade now by running yarn set version 4.10.3

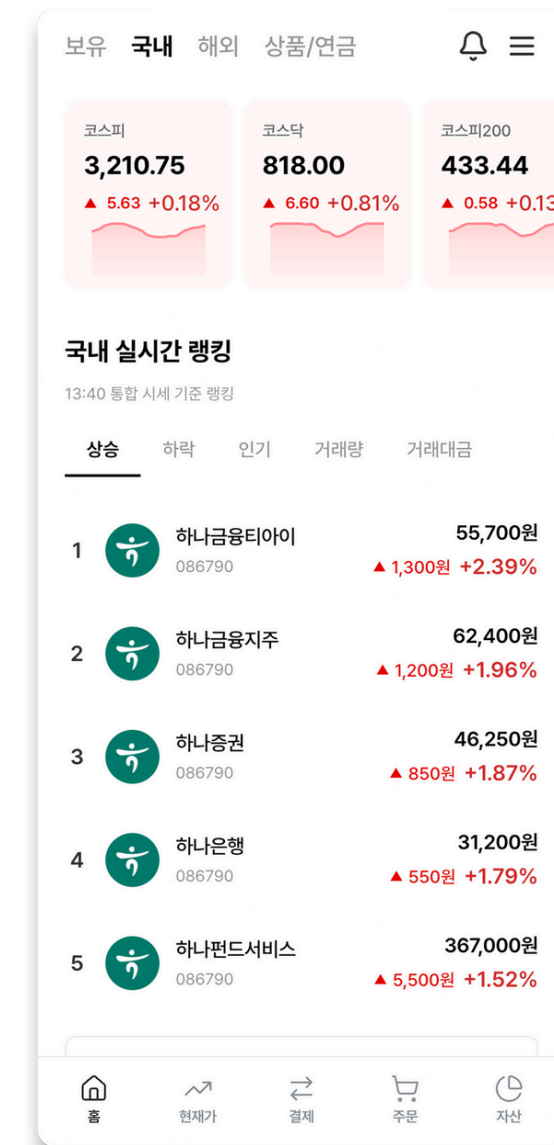
> YN0000: · Yarn 4.6.0
> YN0000: [ Resolution step
> YN0000: [ Completed
> YN0000: [ Fetch step
> YN0000: [ Completed in 0s 279ms
> YN0000: [ Link step
> YN0000: [ ESM support for PnP uses the experimental loader API and is therefore experimental
> YN0007: [ esbuild@npm:0.25.9 must be built because it never has been before or the last one failed
> YN0000: [ Completed in 1s 47ms
> YN0000: · Done with warnings in 1s 450ms
```

[yarn으로 패키지 의존성 설치]

4. 패키지 설치가 완료되면 터미널 창에 yarn dev를 입력하고 http://localhost:5173/에 접속하면 메인 화면이 보입니다.

```
sanghyeon@isanghyeon-ui-noteubug HanaSecurities_Frontend % yarn dev  
  
VITE v7.1.5 ready in 1956 ms  
  
→ Local:   http://localhost:5173/  
→ Network: use --host to expose  
→ press h + enter to show help
```

[로컬 웹서버 실행]



[하나증권 : THE NEXT 메인 화면]

02

백엔드

백엔드 서버 실행 방법



1. Github에서 클론받은 파일 최상단에서 HanaSecurities_Backend/src/main/resources 폴더에 .env 파일을 만들어 환경변수를 설정해야합니다. 필요한 환경변수의 목록과 역할은 동일 폴더에 .env.example 파일을 통해 확인할 수 있습니다.

```
# =====  
# Database Configuration (Oracle)  
# =====  
DB_URL=your_db_url  
DB_USERNAME=your_db_username  
DB_PASSWORD=your_db_password  
  
# =====  
# 한국투자증권 환경변수 설정  
# =====  
API_APP_KEY=your_app_key  
API_APP_SECRET=your_app_secret  
API_BASE_URL=https://openapi.koreainvestment.com:9443  
  
# 한국투자증권 주식 거래용 환경변수  
API_ACCOUNT_NUMBER=your_account_number  
API_ACCOUNT_PRODUCT_CODE=01  
  
# =====  
# 키움증권 환경변수 설정  
# =====  
KIWOOM_APP_KEY=your_app_key  
KIWOOM_SECRET_KEY=your_secret_key  
KIWOOM_API_BASE_URL=https://api.kiwoom.com  
  
# =====  
# 네이버 API 설정  
# =====  
API_TOKEN_NAVER_CLIENT_ID=your_client_id  
API_TOKEN_NAVER_CLIENT_SECRET=your_client_secret  
  
# =====  
# WebAuthn 지문 인증 설정  
# =====  
WEBAUTHN_RP_ID=localhost  
WEBAUTHN_RP_NAME=Hana Securities
```

[.env.example 파일 내용]

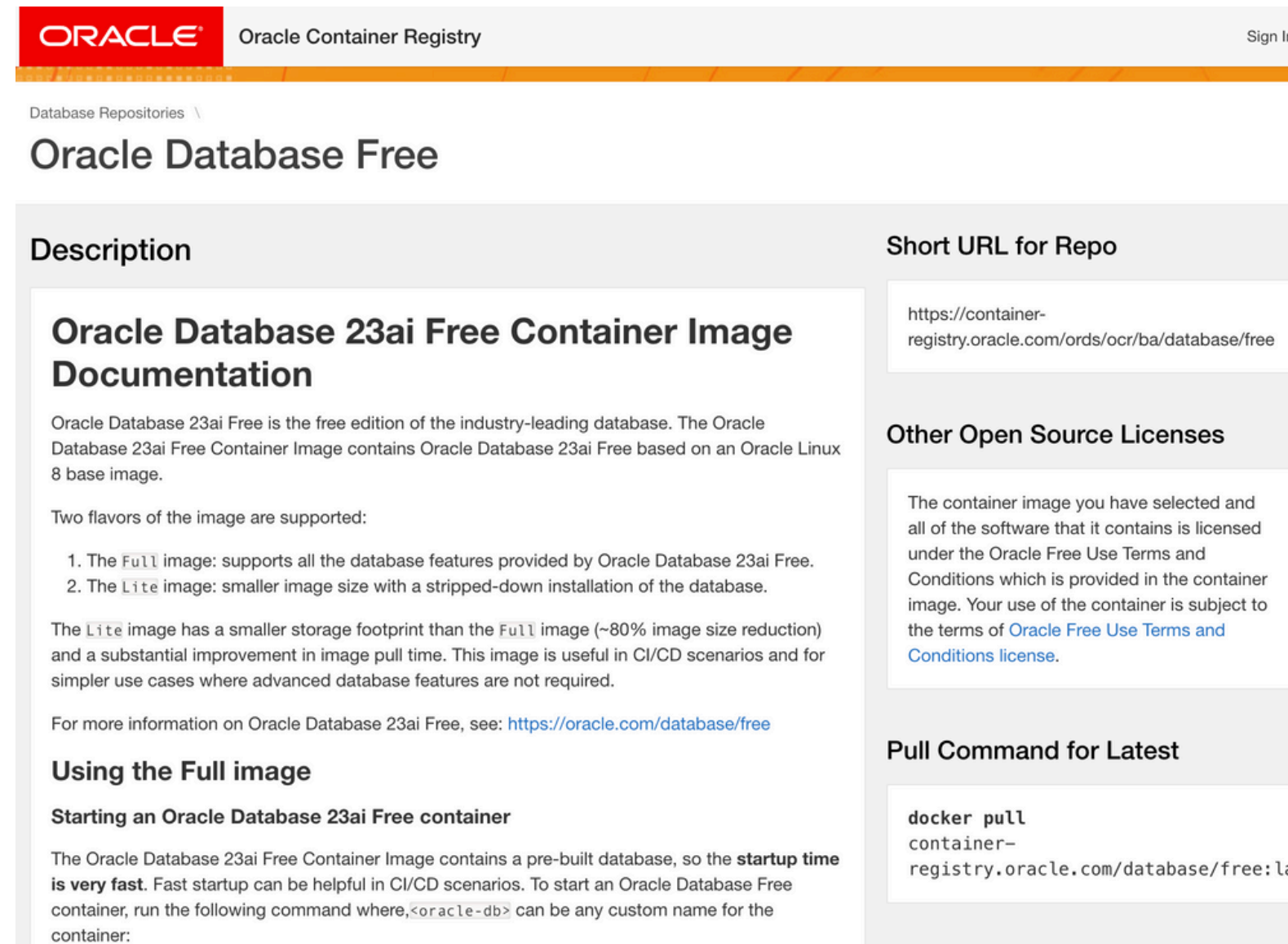
03

데이터베이스

시연 시나리오
시연 영상



1. Oracle 공식 홈페이지에서 제공하는 Docker 이미지를 사용하였습니다.



The screenshot shows the Oracle Container Registry page for the Oracle Database 23ai Free Container Image. The page is titled "Oracle Database Free" and includes a "Description" section with the following content:

Oracle Database 23ai Free Container Image Documentation

Oracle Database 23ai Free is the free edition of the industry-leading database. The Oracle Database 23ai Free Container Image contains Oracle Database 23ai Free based on an Oracle Linux 8 base image.

Two flavors of the image are supported:

1. The **Full** image: supports all the database features provided by Oracle Database 23ai Free.
2. The **Lite** image: smaller image size with a stripped-down installation of the database.

The **Lite** image has a smaller storage footprint than the **Full** image (~80% image size reduction) and a substantial improvement in image pull time. This image is useful in CI/CD scenarios and for simpler use cases where advanced database features are not required.

For more information on Oracle Database 23ai Free, see: <https://oracle.com/database/free>

Using the Full image

Starting an Oracle Database 23ai Free container

The Oracle Database 23ai Free Container Image contains a pre-built database, so the **startup time is very fast**. Fast startup can be helpful in CI/CD scenarios. To start an Oracle Database Free container, run the following command where `<oracle-db>` can be any custom name for the container:

Short URL for Repo

```
https://container-registry.oracle.com/ords/ocr/ba/database/free
```

Other Open Source Licenses

The container image you have selected and all of the software that it contains is licensed under the Oracle Free Use Terms and Conditions which is provided in the container image. Your use of the container is subject to the terms of [Oracle Free Use Terms and Conditions license](#).

Pull Command for Latest

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

[오라클 컨테이너 저장소 홈페이지 화면]

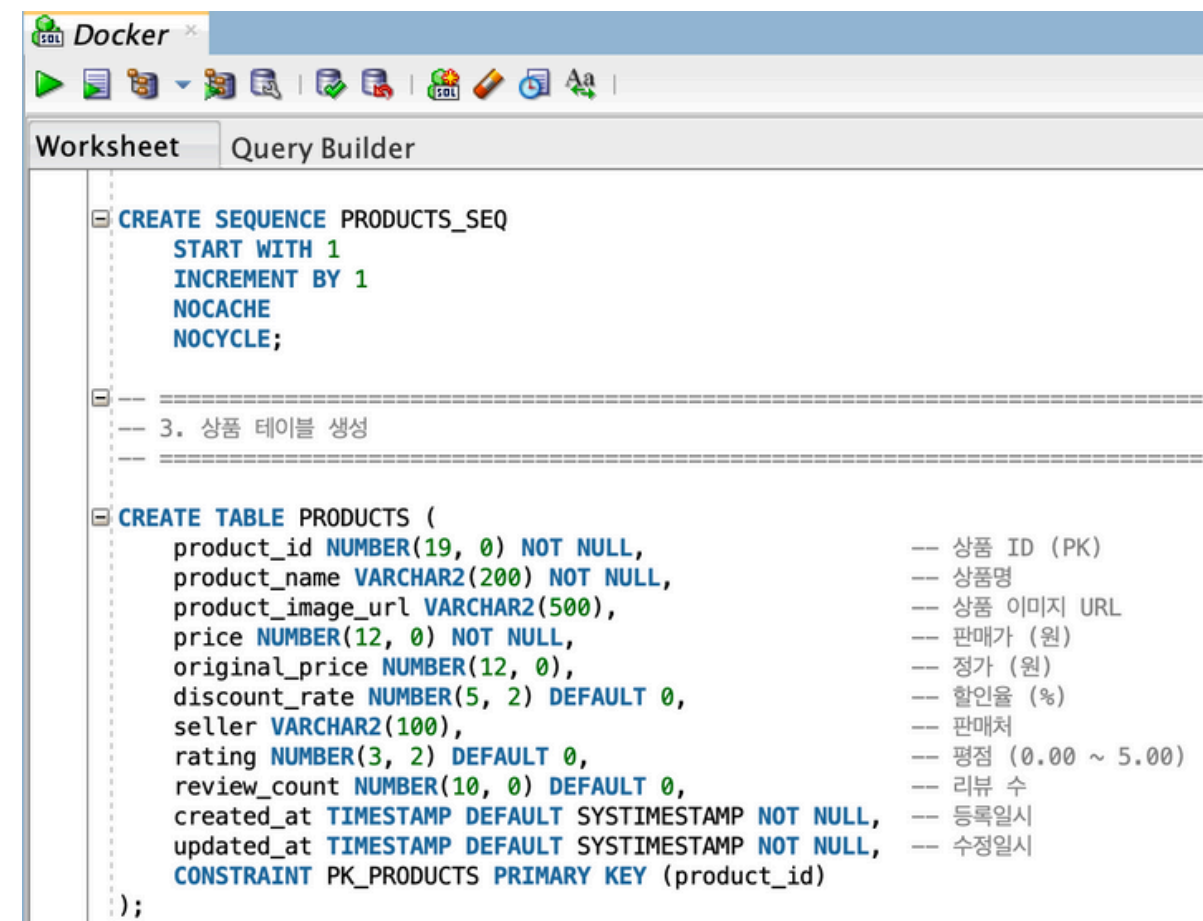
2. 터미널 창에서 `docker run -d -p 1521:1521 -e ORACLE_PWD={{원하는 비밀번호}}`
`container-registry.oracle.com/database/free` 를 입력하여 오라클 컨테이너를
실행시킵니다. (클라우드 등 다른 방법으로 Oracle DBMS를 실행하셔도 됩니다.)

```
sanghyeon@isanghyeon-ui-noteubug ~ % docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	PORTS	NAMES	
bc3cde9ccd81	container-registry.oracle.com/database/free	"/bin/bash -c \$ORACL..."	2 days ago
Up 30 hours (healthy)	0.0.0.0:1521->1521/tcp, [::]:1521->1521/tcp	oracle	

[터미널 창에서 오라클 컨테이너가 실행중인 것을 확인]

3. Github에서 클론받은 파일 최상단에서 HanaSecurities_Database 폴더에 존재하는 HanaSecurities_Init.sql, HanaSecurities_Insert.sql 을 실행시켜 데이터베이스를 초기화합니다. (테이블 및 주식 종목 초기 데이터 포함)



```
CREATE SEQUENCE PRODUCTS_SEQ
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE;

--
-- 3. 상품 테이블 생성
--

CREATE TABLE PRODUCTS (
    product_id NUMBER(19, 0) NOT NULL,
    product_name VARCHAR2(200) NOT NULL,
    product_image_url VARCHAR2(500),
    price NUMBER(12, 0) NOT NULL,
    original_price NUMBER(12, 0),
    discount_rate NUMBER(5, 2) DEFAULT 0,
    seller VARCHAR2(100),
    rating NUMBER(3, 2) DEFAULT 0,
    review_count NUMBER(10, 0) DEFAULT 0,
    created_at TIMESTAMP DEFAULT SYSTIMESTAMP NOT NULL,
    updated_at TIMESTAMP DEFAULT SYSTIMESTAMP NOT NULL,
    CONSTRAINT PK_PRODUCTS PRIMARY KEY (product_id)
);
```

--- 상품 ID (PK)
--- 상품명
--- 상품 이미지 URL
--- 판매가 (원)
--- 정가 (원)
--- 할인율 (%)
--- 판매처
--- 평점 (0.00 ~ 5.00)
--- 리뷰 수
--- 등록일시
--- 수정일시

[테이블 생성 및 초기 데이터 삽입 스크립트 실행 화면]

감사합니다