



포팅 매뉴얼

1. 빌드 및 배포

기술 스택

AI

Skill	Version
Python	3.9.13
FastAPI	0.103.0
PyTorch	2.2.1
YOLO	v5

Backend

Skill	Version
Java	17
Spring boot	3.2.3
Hibernate	6.4.1
MySQL	8.0.29
Redis	7.2.4

Frontend

Skill	Version
HTML5	
CSS	
JavaScript	
Node.js	20.11.0
NPM	10.2.4
React	10.2.3
Redux	9.1.0

IDE

Skill	Version
Visual Studio Code	1.85.1
IntelliJ IDEA	2023.3.2

Infra

Skill	Version
Linux	5.15.0-1056-aws
Ubuntu	20.04
Docker	26.0.0

Docker-compose	v2.25.0
Nginx	1.18.0
Jenkins	2.440.2

포트 포워딩

```

80 - http
443 - https
8080 - jenkins
3306 : 3306 - MySQL
6379 : 6379 - Redis
3001 : 80 - React (blue)
3002 : 80 - React (green)
8081 : 8080 - Spring boot (blue)
8082 : 8080 - Spring boot (green)
9001:8000 - FAST API

```

디렉토리 구조

```

/home/ubuntu
├── DockerCompose
│   ├── db
│   │   ├── mysql-dev
│   │   │   └── data
│   │   │       ├── #ib_16384_0.dblwr
│   │   │       ├── #ib_16384_1.dblwr
│   │   │       ├── #innodb_temp [error opening dir]
│   │   │       ├── auto.cnf
│   │   │       ├── binlog.000001
│   │   │       ├── binlog.000002
│   │   │       ├── binlog.000003
│   │   │       ├── binlog.index
│   │   │       ├── ca-key.pem
│   │   │       ├── ca.pem
│   │   │       ├── client-cert.pem
│   │   │       ├── client-key.pem
│   │   │       ├── ib_buffer_pool
│   │   │       ├── ib_logfile0
│   │   │       ├── ib_logfile1
│   │   │       ├── ibdata1
│   │   │       ├── ibtmp1
│   │   │       ├── mysql [error opening dir]
│   │   │       ├── mysql.ibd
│   │   │       ├── mysql.sock -> /var/run/mysqld/mysqld.sock
│   │   │       ├── performance_schema [error opening dir]
│   │   │       ├── private_key.pem
│   │   │       ├── public_key.pem
│   │   │       ├── server-cert.pem
│   │   │       ├── server-key.pem
│   │   │       ├── sys [error opening dir]
│   │   │       ├── test_db [error opening dir]
│   │   │       ├── undo_001
│   │   │       ├── undo_002
│   │   │       └── whiskeywiki [error opening dir]
│   │   └── mysql-prod

```

```

├── data
│   ├── #ib_16384_0.dblwr
│   ├── #ib_16384_1.dblwr
│   ├── #innodb_temp [error opening dir]
│   ├── auto.cnf
│   ├── binlog.000001
│   ├── binlog.000002
│   ├── binlog.000003
│   ├── binlog.index
│   ├── ca-key.pem
│   ├── ca.pem
│   ├── client-cert.pem
│   ├── client-key.pem
│   ├── ib_buffer_pool
│   ├── ib_logfile0
│   ├── ib_logfile1
│   ├── ibdata1
│   ├── ibtmp1
│   ├── mysql [error opening dir]
│   ├── mysql.ibd
│   ├── mysql.sock -> /var/run/mysqld/mysqld.sock
│   ├── performance_schema [error opening dir]
│   ├── private_key.pem
│   ├── public_key.pem
│   ├── server-cert.pem
│   ├── server-key.pem
│   ├── sys [error opening dir]
│   ├── undo_001
│   ├── undo_002
│   └── whiskeywikiDB [error opening dir]
├── docker-compose.blue.yml
├── docker-compose.green.yml
├── docker-compose.yml
└── env

```

```

/var/lib/jenkins/workspace/prod-ci-cd/

```

```

├── AI
│   └── object-detection
├── Docs
│   └── images
├── README.md
├── backend
│   ├── Dockerfile.dev
│   ├── Dockerfile.prod
│   ├── build
│   ├── build.gradle
│   ├── gradle
│   ├── gradlew
│   ├── gradlew.bat
│   ├── settings.gradle
│   └── src
├── deploy.sh
├── docker-compose.blue.yml
├── docker-compose.green.yml
├── exec
│   └── DB 덤프 파일
├── frontend
│   └── Dockerfile

```

```
├─ Dockerfile.dev
├─ README.md
├─ env.production
├─ nginx.conf
├─ node_modules
├─ package-lock.json
├─ package.json
├─ public
├─ src
└─ tailwind.config.js
```

/etc/nginx

```
├─ conf.d
│   ├── deploy.sh
│   ├── service-url-back.inc
│   └─ service-url-front.inc
├─ fastcgi.conf
├─ fastcgi_params
├─ koi-utf
├─ koi-win
├─ mime.types
├─ modules-available
├─ modules-enabled
│   ├── 50-mod-http-image-filter.conf -> /usr/share/nginx/modules-available/mod-http-image-filter
│   ├── 50-mod-http-xslt-filter.conf -> /usr/share/nginx/modules-available/mod-http-xslt-filter
│   ├── 50-mod-mail.conf -> /usr/share/nginx/modules-available/mod-mail.conf
│   └─ 50-mod-stream.conf -> /usr/share/nginx/modules-available/mod-stream.conf
├─ nginx.conf
├─ proxy_params
├─ scgi_params
├─ sites-available
│   ├── default
│   └─ whiskeywiki.conf
├─ sites-enabled
│   ├── default -> /etc/nginx/sites-available/default
│   └─ whiskeywiki.conf -> /etc/nginx/sites-available/whiskeywiki.conf
├─ snippets
│   ├── fastcgi-php.conf
│   └─ snakeoil.conf
├─ uwsgi_params
└─ win-utf
```

설정 파일

whiskeywiki.conf(Nginx)

```
server {
    listen 80; #80포트로 받을 때
    server_name www.whiskeywiki.shop whiskeywiki.shop;
    return 301 https://whiskeywiki.shop$request_uri;
}

server {
    listen 80;
    server_name dev.whiskeywiki.shop;
```

```

    location / {
        proxy_pass http://localhost:9091;
    }
}

server {
    listen 80;
    server_name ai.whiskeywiki.shop;

    location / {
        proxy_pass http://localhost:9001;
    }
}

server {
    listen 80;
    server_name dev-fe.whiskeywiki.shop;

    location / {
        proxy_pass http://localhost:4001;
    }

    location /api {
        proxy_pass http://localhost:9091/api;
    }
}

server {
    listen 80;
    server_name jenkins.whiskeywiki.shop;

    location / {
        proxy_pass http://localhost:8080;
    }
}

server {
    listen 443 ssl http2;
    server_name www.whiskeywiki.shop whiskeywiki.shop;
    include /etc/nginx/conf.d/service-url-front.inc;
    include /etc/nginx/conf.d/service-url-back.inc;
    # ssl 인증서 적용하기
    ssl_certificate /etc/letsencrypt/live/www.whiskeywiki.shop/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/www.whiskeywiki.shop/privkey.pem;

    location / {
        proxy_pass $service_url_front;
    }

    location /api { # location 이후 특정 url을 처리하는 방법을 정의
        proxy_pass $service_url_back; # Request에 대해 어디로 리다이렉트하는지
        proxy_redirect off;
        charset utf-8;

        proxy_http_version 1.1;
        proxy_set_header Connection "upgrade";
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Host $http_host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}

```

```

    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header X-NginX-Proxy true;
}

location /ai { # location 이후 특정 url을 처리하는 방법을 정의
    proxy_pass http://localhost:8001; # Request에 대해 어디로 리다이렉트하는지
    proxy_redirect off;
    charset utf-8;

    proxy_http_version 1.1;
    proxy_set_header Connection "upgrade";
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Host $http_host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header X-NginX-Proxy true;
}
}

```

.deploy.sh

```

#!/bin/bash

# 실행 중인 도커 컴포즈 확인
EXIST_A=$(sudo docker-compose -p whiskeywiki-blue -f docker-compose.blue.yml ps | grep Up)

if [ -z "${EXIST_A}" ] # -z는 문자열 길이가 0이면 true. A가 실행 중이지 않다는 의미.
then
    # Green이 실행 중인 경우
    START_CONTAINER=blue
    TERMINATE_CONTAINER=green
    START_PORT_BACK=8081
    START_PORT_FRONT=3001
    TERMINATE_PORT_BACK=8082
    TERMINATE_PORT_FRONT=3002
else
    # Blue가 실행 중인 경우
    START_CONTAINER=green
    TERMINATE_CONTAINER=blue
    START_PORT_BACK=8082
    START_PORT_FRONT=3002
    TERMINATE_PORT_BACK=8081
    TERMINATE_PORT_FRONT=3001
fi

echo "whiskeywiki-${START_CONTAINER} up"

# 실행해야하는 컨테이너 docker-compose로 실행. -p는 docker-compose 프로젝트에 이름을 부여
# -f는 docker-compose파일 경로를 지정
sudo docker-compose -p whiskeywiki-${START_CONTAINER} -f docker-compose.${START_CONTAINER}.yaml up

for cnt in {1..10} # 10번 실행
do
    echo "check server start.."

```

```

# 스프링부트에 등록했던 actuator로 실행되었는지 확인
UP=$(curl -s http://127.0.0.1:${START_PORT_BACK}/api/whiskeywiki/health | grep 'UP')
if [ -z "${UP}" ] # 실행되었다면 break
then
    echo "server not start.."
else
    break
fi

echo "wait 10 seconds" # 10 초간 대기
sleep 10
done

if [ $cnt -eq 10 ] # 10번동안 실행이 안되었으면 배포 실패, 강제 종료
then
    echo "deployment failed."
    exit 1
fi

echo "server start!"
echo "change nginx server port"

# sed 명령어를 이용해서 아까 지정해줬던 service-url.inc의 url값을 변경해줍니다.
# sed -i "s/기존문자열/변경할문자열" 파일경로 입니다.
# 종료되는 포트를 새로 시작되는 포트로 값을 변경해줍니다.
# sudo sed -i "s/${TERMINATE_PORT_BACK}/${START_PORT_BACK}/" /etc/nginx/conf.d/service-url-back
# sudo sed -i "s/${TERMINATE_PORT_FRONT}/${START_PORT_FRONT}/" /etc/nginx/conf.d/service-url-front
echo ">>> 전환할 Port: $START_PORT_BACK"
echo ">>> 전환할 Port: $START_PORT_FRONT"

echo ">>> Port 전환"
# 아래 줄은 echo를 통해서 나온 결과를 | 파이프라인을 통해서 service-url.inc에 덮어쓸 수 있습니다.
echo "set \$service_url_back http://127.0.0.1:${START_PORT_BACK};" | sudo tee /etc/nginx/conf.d/service-url-back
echo "set \$service_url_front http://127.0.0.1:${START_PORT_FRONT};" | sudo tee /etc/nginx/conf.d/service-url-front

# 새로운 포트로 스프링부트가 구동 되고, nginx의 포트를 변경해주었다면, nginx 재시작해줍니다.
echo "nginx reload.."
sudo service nginx reload

# 기존에 실행 중이었던 docker-compose는 종료시켜줍니다.
echo "whiskeywiki-${TERMINATE_CONTAINER} down"
sudo docker-compose -p whiskeywiki-${TERMINATE_CONTAINER} -f docker-compose.${TERMINATE_CONTAINER} down
echo "success deployment"

```

docker-compose-blue.yml

```

version: "3.8"
services:
  backend:
    build:
      context: ./backend
      dockerfile: Dockerfile.prod
    restart: always
    ports:
      - 8081:8080

```

```
frontend:
  build:
    context: ./frontend
    dockerfile: Dockerfile
  restart: always
  ports:
    - 3001:80
```

docker-compose-green.yml

```
version: "3.8"
services:
  backend:
    build:
      context: ./backend
      dockerfile: Dockerfile.prod
    restart: always
    ports:
      - 8082:8080
  frontend:
    build:
      context: ./frontend
      dockerfile: Dockerfile
    restart: always
    ports:
      - 3002:80
```

service-url-front.inc

```
set $service_url_front http://127.0.0.1:3001;
```

service-url-back.inc

```
set $service_url_back http://127.0.0.1:8081;
```

환경 변수

Jenkins String Parameter

```
MYSQL_HOST : www.whiskeywiki.shop
MYSQL_PASSWORD : b101
MYSQL_PROD_PORT : 3306
MYSQL_USER : root
REDIS_HOST : www.whiskeywiki.shop
REDIS_PROD_PORT : 6379
AI_PROD_URL : http://ai.whiskeywiki.shop
```

.env.production

```
REACT_APP_KAKAOMAP_API_KEY=b17c4e53f55672ce6e6b6644da6ecc84
```

특이사항

Jenkins Execute shell

```
sudo cp /home/ubuntu/env/.env.production /var/lib/jenkins/workspace/prod-ci-cd/frontend
sed -i "s/\${MYSQL_HOST}/${MYSQL_HOST}/" "${WORKSPACE}/backend/src/main/resources/application-pr
sed -i "s/\${MYSQL_PROD_PORT}/${MYSQL_PROD_PORT}/" "${WORKSPACE}/backend/src/main/resources/app
sed -i "s/\${MYSQL_USER}/${MYSQL_USER}/" "${WORKSPACE}/backend/src/main/resources/application-pr
sed -i "s/\${MYSQL_PASSWORD}/${MYSQL_PASSWORD}/" "${WORKSPACE}/backend/src/main/resources/applic
sed -i "s/\${REDIS_HOST}/${REDIS_HOST}/" "${WORKSPACE}/backend/src/main/resources/application-pr
sed -i "s/\${REDIS_PROD_PORT}/${REDIS_PROD_PORT}/" "${WORKSPACE}/backend/src/main/resources/app
# sed -i "s/\${AI_PROD_URL}/${AI_PROD_URL}/" "${WORKSPACE}/backend/src/main/resources/applicatio

echo 'jenkinsbuild started...'
pwd
cd backend
chmod +x gradlew
./gradlew clean
./gradlew bootJar
cd ..
./deploy.sh
```

DB 접속 정보

- MySQL 호스트 : www.whiskeywiki.shop:3306
- MySQL 유저 : root
- MySQL 패스워드 : b101
- Redis 호스트 : www.whiskeywiki.shop:6379

배포 순서

1. ufw allow로 필요한 포트번호 전부 열어주기
2. Docker 설치
3. Docker-compose 설치
4. Nginx 설치
5. Jenkins 설치
6. /etc/nginx/sites-available/whiskeywiki.conf에 리버스 프록시 작성
7. /etc/nginx/conf.d/service-url-front.inc 및 service-url-back.inc 작성
8. Jenkins에서 새로운 Item 추가, Gitlab 연동 및 웹훅 설정 (String parameter 추가)
9. /var/lib/jenkins/workspace/Item이름 경로에 docker-compose.blue.yml, docker-compose.green.yml, .deploy.sh 작성
10. /var/lib/jenkins/workspace/Item이름/frontend 경로에 .env.production 추가
11. Jenkins에서 파라미터와 함께 build

2. DB 덤프파일

schema.sql

```
-- CREATE DATABASE IF NOT EXISTS whiskeywiki;

-- use whiskeywiki;
```

```

CREATE TABLE IF NOT EXISTS users (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    login_id VARCHAR(255) NOT NULL,
    password VARCHAR(255) NOT NULL,
    nickname VARCHAR(255) DEFAULT '',
    address VARCHAR(255) DEFAULT '',
    gender VARCHAR(255) DEFAULT '',
    age INT DEFAULT 0,
    latitude DECIMAL(13, 10),
    longitude DECIMAL(13, 10),
    region VARCHAR(255) DEFAULT '',
    city VARCHAR(255) DEFAULT '',
    village VARCHAR(255) DEFAULT ''
);

CREATE TABLE IF NOT EXISTS cocktail (
    cocktail_id INT AUTO_INCREMENT PRIMARY KEY,
    recipe VARCHAR(255),
    cocktail_name VARCHAR(255),
    cocktail_name_en VARCHAR(255),
    detail VARCHAR(255)
);

CREATE TABLE IF NOT EXISTS favorite (
    favorite_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    cocktail_id INT,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (cocktail_id) REFERENCES cocktail(cocktail_id)
);

CREATE TABLE IF NOT EXISTS whiskey (
    whiskey_id INT AUTO_INCREMENT PRIMARY KEY,
    whiskey_name_kr VARCHAR(255),
    whiskey_name_en VARCHAR(255),
    whiskey_flavor TEXT,
    abv DOUBLE,
    price INT,
    detail VARCHAR(255)
);

CREATE TABLE IF NOT EXISTS own_whiskey (
    own_whiskey_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    whiskey_id INT,
    is_empty TINYINT DEFAULT 0,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (whiskey_id) REFERENCES whiskey(whiskey_id)
);

CREATE TABLE IF NOT EXISTS ingredient (
    ingredient_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255)
);

```

```

CREATE TABLE IF NOT EXISTS cocktail_ingredient (
    cocktail_ingredient_id INT AUTO_INCREMENT PRIMARY KEY,
    cocktail_id INT,
    ingredient_id INT,
    FOREIGN KEY (cocktail_id) REFERENCES cocktail(cocktail_id),
    FOREIGN KEY (ingredient_id) REFERENCES ingredient(ingredient_id)
);

CREATE TABLE IF NOT EXISTS base (
    base_id INT AUTO_INCREMENT PRIMARY KEY,
    cocktail_id INT,
    whiskey_id INT,
    FOREIGN KEY (cocktail_id) REFERENCES cocktail(cocktail_id),
    FOREIGN KEY (whiskey_id) REFERENCES whiskey(whiskey_id)
);

CREATE TABLE IF NOT EXISTS review (
    review_id INT AUTO_INCREMENT PRIMARY KEY,
    whiskey_id INT,
    user_id INT,
    review_rating VARCHAR(255),
    content VARCHAR(255),
    created_date DATETIME,
    FOREIGN KEY (whiskey_id) REFERENCES whiskey(whiskey_id),
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);

CREATE TABLE IF NOT EXISTS chatrooms (
    chatroom_id INT AUTO_INCREMENT PRIMARY KEY,
    create_time DATETIME,
    edit_time DATETIME,
    last_chat VARCHAR(255)
);

CREATE TABLE IF NOT EXISTS user_chatrooms (
    user_chatroom_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    chatroom_id INT,
    exited TINYINT,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (chatroom_id) REFERENCES chatrooms(chatroom_id)
);

CREATE TABLE IF NOT EXISTS chats (
    chat_id INT AUTO_INCREMENT PRIMARY KEY,
    chatroom_id INT,
    user_id INT,
    is_read BOOLEAN,
    chat_visible BOOLEAN,
    message VARCHAR(255),
    date_time DATETIME,
    FOREIGN KEY (chatroom_id) REFERENCES user_chatrooms(chatroom_id),
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);

```

data.sql

```
INSERT INTO users (login_id, password, nickname, address, gender, age, latitude, longitude, region)
VALUES
  ('admin', 'admin', '관리자', '하늘', '관리자', 46, 36.355372, 127.298412, '대전광역시'),
  ('1234', '1234', 'ssafy', '삼성화재 연수원1', '남성', 20, 36.355359, 127.298385, '대전광역시'),
  ('asdf', '1234', '싸피', '삼성화재 연수원2', '여성', 45, 36.355212, 127.298924, '대전광역시'),
  ('whduswn1234', 'whduswn1234', '위스키좋아', '한밭대 운동장', '남성', 22, 36.353544, 127.301025, '대전광역시'),
  ('dlagustmd1234', 'dlagustmd1234', '칵테일좋아', '한밭대 정문', '남성', 31, 36.350993, 127.298499, '대전광역시'),
  ('whekagus1234', 'wheakgus1234', '숙취없어', '도로명 주소 봉명동 kfc', '여성', 25, 36.354348, 127.340370, '대전광역시'),
  ('dlwldms1234', 'dlwldms1234', '바텐더v', '도로명주소 버거킹', '남성', 35, 36.354208, 127.342057, '대전광역시'),
  ('wkdtpdud1234', 'wkdtpdud1234', '닉네임짓기어렵다', '도로명주소 유성온천역', '여성', 21, 36.353735, 127.340370, '대전광역시'),
  ('fosejadjadl', '1234', '안녕하세요', '도로명주소 고향촌 그 어딘가', '남성', 53, 36.353850, 127.340370, '대전광역시'),
  ('rkskekfkakqk', '1234', '마티니짱', '도로명 주소 흥인 근처', '여성', 41, 36.354386, 127.340370, '대전광역시'),
  ('tpwhdeodhkdW', '1234', '말리부 덕후', '도로명주소 하이테이블 근처', '남성', 35, 36.354749, 127.340370, '대전광역시'),
  ('sksmsdlagustmd', '1234', '잭다니엘은 신이야', '도로명 주소 봉명동 그 어딘가', '여성', 37, 36.353834, 127.340370, '대전광역시'),
  ('sksmswhekagus', '1234', '싸피 술꾼', '도로명 주소 아 몰라 어딘가에 있겠지', '여성', 32, 36.353553, 127.340370, '대전광역시'),
  ('sksmswhduswn', '1234', '알쓰그자체', '도로명 주소 봉명동 어딘가에 있겠지', '여성', 25, 36.353298, 127.340370, '대전광역시'),
  ('sksmsdlwldms', '1234', '아몰라닉네임따위', '도로명 주소 주소 짓기 참 힘드네', '남성', 21, 36.353052, 127.340370, '대전광역시');
```

```
INSERT INTO whiskey (whiskey_name_kr, whiskey_name_en, whiskey_flavor, abv, price, detail) VALUES
  ('앱솔루트', 'Absolute', 'NONE FLAVOR', 40, 35000, '앱솔루트 보드카'는 스웨덴의 400년 양조 전통이 낳은 최고급 보드카입니다. ),
  ('바카디', 'Bacardi', 'SWEET', 40, 27000, 'Bacardi는 쿠바에서 시작된 세계적으로 유명한 백 럼 브랜드입니다. ),
  ('베일리스', 'Baileys', 'SWEET', 17, 45000, 'Baileys는 크림과 아이리쉬 위스키를 베이스로 하는 아일랜드 크림 리큐르입니다. ),
  ('발렌타인 파이니스트', 'Ballantines_Finest', 'SMOKY & VANILLA', 23, 43000, '발렌타인 파이니스트는 부드럽고 달콤한 맛을 가진 리큐르입니다. ),
  ('비피터', 'Beefeater', 'SPICY & JUNIPER', 47, 42000, '영국에서 만들어진 고전적인 진으로, 로즈마리, 삽화, 레몬을 함유하고 있습니다. ),
  ('봄베이 사파이어', 'Bombay_Sapphire', 'SPICY & EARTHY', 47, 45000, '보테니컬이 풍부한 영국의 프리미엄 진입니다. ),
  ('캡틴 모건', 'Captain_Morgan', 'SWEET & RICH', 35, 32000, '캐리비안 럼의 대표적인 브랜드 중 하나로, 바닐라와 오렌지 향이 특징입니다. ),
  ('고든', 'Gordons', 'DRY & FRESH', 43, 29000, '국에서 가장 유명한 진 브랜드 중 하나로, 로즈마리, 오렌지 등의 향이 있습니다. ),
  ('하바나 클럽', 'Havana_Club', 'FRUITY & HERBAL', 40, 68000, '쿠바에서 생산되는 럼의 대표적인 브랜드 중 하나로, 바닐라와 오렌지 향이 있습니다. ),
  ('잭다니엘 허니', 'Jack-Daniels honey', 'SWEET & SMOOTH', 40, 70000, '잭 다니엘 브랜드의 한 제품으로, 허니를 첨가한 달콤한 맛을 자랑합니다. ),
  ('잭다니엘 No7', 'Jack-Daniels no.7', 'FRUITY & VANILLA', 40, 58000, '잭 다니엘 위스키는 테네시 주 프랭클린 카운티에서 생산됩니다. ),
  ('예거마이스터', 'Jagermeister', 'SPICY & HERBAL', 35, 49000, '쿠바에서 생산되는 럼의 대표적인 브랜드 중 하나로, 바닐라와 오렌지 향이 있습니다. ),
  ('짐빔', 'Jim_Beam', 'SWEET & FRUITY', 43, 50000, '짐빔은 보편적으로 버번(버번 주 정통 위스키)의 조제 방식을 따릅니다. ),
  ('조니워커 블랙', 'Johnie-Walker black', 'VANILLA & SWEET', 40, 60000, '조니 워커 브랜드의 프리미엄 라인에 속하는 블랙 위스키입니다. ),
  ('조니워커 블루', 'Johnie-Walker blue', 'VANILLA & SWEET', 56, 180000, '조니 워커 브랜드의 최고급 위스키입니다. ),
  ('조니워커 레드', 'Johnie-Walker red', 'VANILLA & SWEET', 40, 44000, '조니 워커 브랜드의 입문용 위스키 중 하나로, 바닐라와 오렌지 향이 있습니다. ),
  ('호세쿠엘보', 'Jose_Cuervo', 'SWEET & BITTER', 40, 60000, '멕시코의 대표적인 테킬라로, 대부분은 아군(블랑코)입니다. ),
  ('칼루아', 'Kahlua', 'SWEET & COFFEE', 16, 39000, '커피맛이 진한 달달한 리큐르입니다. 단독으로 마시진 않고 여타 음료에 첨가하여 마시기도 합니다. ),
  ('말리부', 'Malibu', 'SWEET & FRUITY', 21, 33000, '말리부는 코코넛 플레이버를 가진 라임 주를 만든 것으로 잘 알려져 있습니다. ),
  ('마티니 비앙코', 'Martini_Bianco', 'CRISP & ELEGANT', 18, 30000, '마티니 비앙코(Martini Bianco)는 이탈리아에서 생산되는 화이트 마티니입니다. ),
  ('스카치블루', 'Scotch_Blue', 'SMOOTH & COMPLEX', 40, 60000, '제조 과정 중 특별한 옥수수의 사용으로 유명한 스카치 블루 위스키입니다. ),
  ('스카이', 'Skyy', 'NEUTRAL & RICH', 40, 39000, '순수한 보드카로, 부드럽고 깔끔한 맛을 가지고 있습니다. 이 보드카는 스웨덴에서 생산됩니다. ),
  ('스미노프', 'Smirnoff', 'NEUTRAL & GRAIN', 37, 35000, '스미노프는 전 세계에서 가장 유명한 보드카 브랜드 중 하나로, 스웨덴에서 생산됩니다. ),
  ('트리플 섹', 'Triple_Sec', 'SWEET & ORANGE', 40, 35000, '트리플 섹의 특징은 40도 전후의 높은 도수에 무색투명한 맛을 지니고 있습니다. ),
  ('블루 큐라소', 'Blue_Curacao', 'SWEET & ORANGE', 20, 29000, '오렌지 리큐르 중의 한 종류로, 보통 트리플 섹에 사용됩니다. ),
  ('피치트리', 'Peach_Tree', 'SWEET & PEACH', 20, 32000, '고급스러운 복숭아 맛을 즐길 수 있는 리큐르로, 피치트리 리큐르는 과일과 꽃의 향이 어우러지며, 달콤한 맛을 자랑합니다. ),
  ('발렌타인 17년산', 'Ballantines 17', 'SMOKY ', 43, 200000, '발렌타인 17년산은 스코틀랜드의 발렌타인 디스티러리가 생산한 17년숙성된 싱글 몰트 스카치 위스키입니다. ),
  ('발렌타인 21년산', 'Ballantines 21', 'SMOKY', 40, 290000, '발렌타인 21년산은 과일과 꽃의 향이 어우러지며, 달콤한 맛을 자랑합니다. ),
  ('발렌타인 30년산', 'Ballantines 30', 'SMOKY', 40, 560000, '이 위스키는 최소 30년간 숙성되어 특유의 깊고 복잡한 맛을 자랑합니다. ');
```

```
INSERT INTO cocktail(cocktail_name, cocktail_name_en, recipe, detail)
VALUES
  ('앱솔루트 토닉', 'Absolute_Tonic', '하이볼 글래스에 얼음을 채운다. 앱솔루트와 토닉워터를 3:10 비율로 잔에 붓는다. '),
  ('보드카 크랜베리', 'Vodka_Cranberry', '하이볼 글래스에 얼음을 채운다. 앱솔루트 보드카와 크랜베리 주스를 1:1 비율로 붓는다. '),
  ('바카디 칵테일', 'Barcadi_Cocktail', '바카디 45ml와 라임즙 20ml과 그레나딘 시럽 15ml을 셰이커에 넣어주고 얼음과 함께 흔든다. ');
```


('스카이 다이빙', 'Sky_Diving', '바카디 30ml와 블루큐라소 20ml와 라임즙 10ml를 셰이커에 넣어주고 셰이킹 하
('베일리스 밀크', 'Baileys_Milk', '베일리스와 우유를 1:3 비율로 잔에 넣는다.', ' 달콤하고 부드러우며 베일리스
('B-52', 'B-52', '위스키 잔에 깔루아 20ml, 베일리스 20ml, 트리플섹 20ml을 천천히 벽을 따라 넣는다', 'B-52
('얼그레이 하이볼', 'Earl_Gray_Highball', '발렌타인 파이니스트 30ml, 탄산수 약 120ml, 얼그레이 시럽을 20ml
('위스키 사워', 'Whiksey_Sour', '발렌타인 파이니스트 60ml, 레몬 즙 30ml, 시럽 15ml, 얼음을 셰이커에 넣고
('김렛', 'Gimlet', '비피터 45ml, 라임즙 15m, 설탕 1tsp을 셰이커에 넣고 셰이킹한 후 잔에 따른다.', '새콤을
('오렌지 블라썸', 'Orange_Blossom', '비피터 45ml, 오렌지 즙 45ml, 설탕 시럽 2tsp을 셰이커에 넣고 셰이킹한
('스페니쉬 진토닉', 'Spanish_Gin_Tonic', '봄베이 사파이어와 토닉워터를 1:2 비율로 얼음이 담긴 잔에 넣는다.
('블루버드', 'Blue_Bird', '얼음이 담긴 셰이커에 봄베이 사파이어 40ml, 블루 큐라소 5ml, 트리플 섹 10ml, 러
('온더 락', 'On_The_Rocks', '얼음이 담긴 잔에 캡틴 모건을 따라 마십니다.', '가장 기본적인 마시는 방법입니다
('럼 토닉', 'Rum_Tonic', '얼음이 담긴 잔에 캡틴 모건과 토닉 워터를 1:3비율로 따르뒤 슬라이스된 라임을 넣어
('진 토닉', 'Gin_Tonic', '얼음이 담긴 잔에 고든과 토닉워터를 1:3 또는 1:4 비율로 넣은뒤 레몬즙을 2tsp 넣어
('오렌지 블라썸', 'Orange_Blossom', '고든과 오렌지 주스를 1:2 비율로 섞어줍니다. 기호에 따라 오렌지 껍질로
('모히또', 'Mojito', '하이볼 글라스에 민트 몇 가지와 설탕 2tsp를 넣는다. 라임즙 20ml를 넣고 민트잎과 섞어준
('다이키리', 'DaiQuiri', '얼음과 하바나 클럽 60ml와 라임즙 20ml과 설탕 2tsp을 셰이커에 넣어 셰이킹 한뒤 잔
('허니 위스키 사워', 'Honey_Whiksey_Sour', '잭 다니엘 허니 60ml, 레몬주스 30ml, 시럽 30ml 을 셰이커에
('잭 허니 하이볼', 'Jack_Honey_Highball', '잭다니엘 허니와 하이볼을 1:3의 비율로 얼음이 담긴 잔에 넣는다.
('잭콕', 'Jack_And_Coke', '잭다니엘 No.7과 콜라를 1:4의 비율로 얼음이 담긴 잔에 넣는다.', '잭다니엘 No.7
('잭 하이볼', 'Jack_Highball', '잭다니엘 No.7과 토닉워터를 1:3의 비율로 얼음이 담긴 잔에 넣는다.', '잭콕과
('예거 밤', 'Jager_Bomb', '예거마이스터와 에너지드링크를 1:3비율로 섞어 얼음이 담긴 잔에 넣는다', '예거밤은
('예거 콕', 'Jager_Coke', '예거마이스터와 콜라를 1:3~4 비율로 섞어 얼음이 담긴 잔에 넣는다.', '예거 콕은 예
('짐빔 하이볼', 'Jim_Beam_Highball', '짐빔과 토닉워터를 1:4 비율로 얼음이 담긴 잔에 넣는다.', '짐빔을 먹는
('짐빔 콕', 'Jim_Beam_Coke', '짐빔과 콜라를 1:2~3 비율로 얼음이 담긴 잔에 넣는다.', '짐빔 콕 칵테일은 위스
('얼그레이 하이볼', 'Earl_Gray_Highball', '조니워커 블랙 30ml와 탄산수 150ml를 얼음이 담긴잔에 넣는다. 얼
('조니워커 블랙 앤 소다', 'Johnnie_Walker_Black_And_Soda', '조니워커 블랙 25ml와 탄산수 75ml를 얼음이 담
('얼그레이 하이볼', 'Earl_Gray_Highball', '조니워커 블루 30ml와 탄산수 150ml를 얼음이 담긴잔에 넣는다. 얼
('조니워커 블루 앤 소다', 'Johnnie_Walker_Black_And_Soda', '조니워커 블루 25ml와 탄산수 75ml를 얼음이 담
('얼그레이 하이볼', 'Earl_Gray_Highball', '조니워커 레드 30ml와 탄산수 150ml를 얼음이 담긴잔에 넣는다. 얼
('조니워커 레드 앤 소다', 'Johnnie_Walker_Black_And_Soda', '조니워커 레드 25ml와 탄산수 75ml를 얼음이 담
('마타도르', 'Matador', '호세쿠엘보 30ml, 파인애플 주스 45ml, 레몬에이드 15ml를 셰이커에 넣어 섞은뒤 얼음
('데킬라 선라이즈', 'Tequila_Sunrise', '호세쿠엘보 45ml, 오렌지 주스 90ml, 그레나딘 시럽 15ml를 셰이커에
('에스프레소 마티니', 'Espresso_Matini', '에스프레소 30mL, 깔루아 30mL, 애플루트 30ml 를 셰이커에 넣고
('깔루아 밀크', 'Kahlua_Milk', '깔루아 30mL 우유75mL를 잔에 넣어 완성합니다.', '달달하고 맛있는 칵테일입니
('피치 코코', 'Pitch_Coco', '말리부 30ml, 피치트리 30ml, 오렌지주스 60ml, 크랜베리 주스 60ml를 셰이커에
('블루 하와이안', 'Blue_Hawaian', '말리부 30ml, 바카디 30ml, 블루큐라소 15ml, 파인애플주스 75ml, 레몬즙
('비앙코 브리즈', 'Bianco_Breeze', '마티니 비앙코 45ml, 자몽주스 50ml, 크랜베리 주스 100ml와 얼음을 셰이
('마티니', 'Martini', '봄베이 사파이어(진)과 마티니 비앙코를 1:6의 비유로 섞어 잔에 넣습니다. 기호에 따라 올
('스카치 하이볼', 'Scotch_Highball', '스카치 블루와 토닉워터를 1:3 비율로 섞어 얼음이 담긴 잔에 넣습니다.'
('존 콜린스', 'John_Collins', '스카치 블루 60ml와 레몬즙 20ml, 설탕 1tsp를 셰이커에 넣어 셰이킹 한뒤 얼음
('화이트 러시안', 'White_Russian', '스카이 30ml, 깔루아 30ml, 우유 30ml, 생크림 10ml를 얼음이 담긴 잔에
('코스모폴리탄', 'Cosmopolitan', '스카이 30ml, 트리플섹 15ml, 라임주스 15ml, 크랜베리주스 15ml를 셰이커에
('화이트 러시안', 'White_Russian', '스미노프 30ml, 깔루아 30ml, 우유 30ml, 생크림 10ml를 얼음이 담긴 잔
('코스모폴리탄', 'Cosmopolitan', '스미노프 30ml, 트리플섹 15ml, 라임주스 15ml, 크랜베리주스 15ml를 셰이
('말리부 콕', 'Malibu_Coke', '말리부와 콜라를 1:4 비율로 얼음이 담긴 잔에 넣는다.', '말리부는 코코넛 향이 특
('말리부 베이브리즈', 'Malibu_Bay_Breeze', '말리부와 크랜베리 주스, 파인애플 주스를 1:1:1 비율로 섞은뒤 얼
('말리부 모히또', 'Malibu_Mojito', '잔에 민트를 적당히 넣어준 뒤 작은 절구로 찜어줍니다. 라임주스 15ml와 말
('말리부 팔로마', 'Malibu_Paloma', '물기를 잔 입구에 묻힙니다. 마른 휴지에 소금을 적당히 부어준 후 잔을 뒤집
('더 말리부', 'The_Malibu', '얼음을 담은 잔에 말리부 30ml를 넣고 잔 끝까지 레몬에이드를 붓는다. 원하는 비주

```
INSERT INTO own_whiskey(user_id, whiskey_id, is_empty)
values
    (1, 1, 1), (1, 2, 1), (1, 3, 1), (1, 4, 1), (1, 5, 1), (1, 6, 1), (1, 7, 1), (1, 8, 1), (1, 9, 1), (1, 10, 1), (1, 11, 1), (1, 12, 1),
    (1, 13, 1), (1, 14, 1), (1, 15, 1), (1, 16, 1), (1, 17, 1), (1, 18, 1), (1, 19, 1), (1, 20, 1), (1, 21, 1), (1, 22, 1), (1, 23, 1),
    (2, 1, 1), (2, 2, 1), (2, 3, 0),
    (3, 2, 1), (3, 3, 1), (3, 5, 1), (3, 7, 1), (3, 9, 1), (3, 10, 1), (3, 11, 1), (3, 13, 1), (3, 14, 1), (3, 15, 1), (3, 16, 1), (3, 17, 1),
    (4, 2, 1), (4, 3, 0), (4, 4, 1), (4, 5, 1), (4, 6, 0), (4, 7, 1), (4, 8, 1), (4, 10, 1), (4, 12, 1), (4, 13, 1), (4, 14, 1), (4, 15, 1),
    (5, 1, 0), (5, 2, 1), (5, 3, 1), (5, 4, 1), (5, 5, 1), (5, 6, 1), (5, 7, 0), (5, 9, 1), (5, 10, 1), (5, 11, 1), (5, 12, 1), (5, 13, 1),
```

```
(6, 2, 0),(6, 3, 1),(6, 4, 0),(6, 5, 0),(6, 6, 1),(6, 7, 1),(6, 8, 1),(6, 9, 1),(6, 11, 1),
(7, 2, 1),(7, 3, 1),(7, 4, 1),(7, 5, 1),(7, 7, 1),(7, 8, 0),(7, 25, 0),(7, 26, 1),
(8, 1, 1),(8, 2, 1),(8, 3, 1),(8, 4, 1),(8, 5, 1),(8, 6, 1),(8, 7, 1),(8, 8, 1),(8, 9, 1),(8, 10, 1),
(9, 1, 1),(9, 2, 1),(9, 3, 0),(9, 5, 1),(9, 7, 1),(9, 8, 1),(9, 9, 1),(9, 10, 0),(9, 11, 1),
(10, 1, 0),(10, 2, 1),(10, 3, 1),(10, 4, 1),(10, 5, 1),(10, 6, 1),(10, 7, 1),(10, 8, 1),(10, 9, 1),
(11, 1, 1),(11, 2, 1),(11, 26, 0),
(12, 19, 1),(12, 20, 1),(12, 21, 0),(12, 22, 1),(12, 23, 1),(12, 24, 1),(12, 25, 1),(12, 26, 1),
(13, 26, 1),
(14, 1, 1),(14, 24, 1),
(15, 1, 1),(15, 2, 0),(15, 3, 1),(15, 4, 1),(15, 5, 0),(15, 6, 1),(15, 7, 1),(15, 8, 1),(15, 9, 1),
```

```
INSERT INTO base(cocktail_id, whiskey_id)
values (1,1), (2, 1),(3, 2),(4, 2), (4, 25),(5, 3),(6, 3), (6, 18), (6, 24),
(7, 4),(8,4),(9,5),(10,5),(11,6),(12,6),(12,25),(12,24),(13,7),(14,7),
(15,8),(16,8),(17,9),(18,9),(19,10),(20,10),(21,11),(22,11),
(23,12),(24,12),(25,13),(26,13),(27,14),(28,14),(29,15),(30,15),
(31,16),(32,16),(33,17),(34,17),(35,1),(35,18),(36,18),(37,19),
(37,26),(38,19),(38,2),(38,25),(39,20),(40,20),(40,6),(41,21),(42,21),
(43,22),(43,18),(44,22),(44,24),(45,23),(45,18),(46,23),(46,24),
(47,19),(48,19),(49,19),(50,19),(51,19);
```

```
INSERT INTO ingredient(name)
values ('레몬'), ('레몬즙'), ('라임'), ('라임즙'), ('민트'),('시나몬 스틱'),('크랜베리 주스'),('그레나딘 시럽'),
```

```
INSERT INTO cocktail_ingredient(cocktail_id, ingredient_id)
values(1,3), (1,18), (2, 7),(2, 3),(3, 4),(3,8),(4,4),(5,14),(7,19),(7,10),(8,2),(8,12),(9,4),(9,12),
(10,12), (11,18),(11,3),(11,6),(12,2),(14,18),(14,3),(15,18),(15,2),(16,13),(17,19),(17,3),
(18,4), (18,24), (19,2), (19,12), (19,1), (19,26), (20,18), (21,25), (22,18), (23,20), (24,20),
(26,25), (27,19), (27,10), (27,1), (28,19), (28,3), (29,19), (29,10), (29,1), (30,19), (30,12),
(31,10), (31,12),(32,19), (32,3), (33,16), (33,15), (34,13), (34,8), (34,9), (34,26), (35,13),
(37,13), (37,7), (38,16), (39,22), (39,7), (39,21), (40,27), (41,18), (42,2), (42,24), (42,18),
(43,23), (44,4), (44,7), (45,14), (45,23), (46,4), (46,7), (48,7), (48,16), (49,4), (49,5),
```

```
INSERT INTO favorite(user_id, cocktail_id)
values (1,1),(1,2),(1,4),
(2,3),(2,4),
(3,5),
(4, 22), (4, 27), (4, 30), (4, 35), (4, 41),
(5, 6), (5, 12), (5, 15), (5, 29), (5, 34), (5, 40),
(6, 8), (6, 14), (6, 18), (6, 21), (6, 24), (6, 26), (6, 31), (6, 36), (6, 42), (6, 45),
(7, 5), (7, 38), (7, 43), (7, 48),
(8, 6), (8, 10), (8, 13), (8, 17), (8, 21), (8, 24), (8, 28), (8, 33), (8, 39), (8, 44),
(9, 4), (9, 9), (9, 13), (9, 17), (9, 21), (9, 25), (9, 30), (9, 35), (9, 41), (9, 46),
(10, 7), (10, 11), (10, 14), (10, 18), (10, 31), (10, 37), (10, 42), (10, 47),
(11, 43), (11, 50),
(12, 3), (12, 7), (12, 11), (12, 15), (12, 19), (12, 23), (12, 28), (12, 32), (12, 36), (12, 37),
(13, 2), (13, 6), (13, 9),(13, 27), (13, 31), (13, 35), (13, 40), (13, 45),
(14, 4), (14, 7), (14, 11), (14, 14), (14, 18), (14, 21), (14, 26), (14, 30), (14, 34), (14, 35),
(15, 23), (15, 27), (15, 32), (15, 37), (15, 42), (15, 47);
```

```
INSERT INTO review(whiskey_id, user_id, review_rating, content, created_date)
values
(1, 5, 'Five', '정말 맛있는 위스키예요.', '2024-03-28 09:45:22.123456'),
(1, 13, 'Four', '꽤 괜찮은 위스키입니다.', '2024-03-27 14:32:11.987654'),
(1, 11, 'Five', '맛있어요!', '2024-03-26 17:21:59.234567'),
(1, 9, 'Five', '진짜 최고예요!', '2024-03-25 10:12:34.345678'),
(1, 1, 'Five', '완벽한 위스키입니다.', '2024-03-24 12:54:33.876543'),
(2, 3, 'Five', '맛있어요!', '2024-03-23 11:23:45.567890'),
```

(2, 12, 'Five', '진짜 최고예요!', '2024-03-22 15:43:21.987654'),
 (2, 2, 'Four', '조금 더 부드러웠으면 좋겠어요.', '2024-03-21 13:32:44.765432'),
 (2, 8, 'Five', '정말 맛있는 위스키예요.', '2024-03-20 10:09:12.543210'),
 (2, 6, 'Five', '꼭 추천하고 싶어요.', '2024-03-19 09:08:07.654321'),
 (3, 4, 'Five', '너무 맛있어요.', '2024-03-20 13:01:46.925709'),
 (3, 8, 'Five', '역시 짐빔이 짱이지.', '2024-03-22 13:01:46.925709'),
 (3, 7, 'Five', '짐빔 칵테일이 진짜 맛있지.', '2024-02-15 13:01:46.925709'),
 (3, 10, 'Four', '좀 더 부드러웠으면 좋겠어요.', '2024-03-21 13:01:46.925709'),
 (3, 14, 'Five', '정말 좋은 맛이에요.', '2024-03-19 13:01:46.925709'),
 (4, 15, 'Three', '정말 최고예요.', '2024-03-20 13:01:46.925709'),
 (4, 9, 'Three', '맛있어요!', '2024-03-22 13:01:46.925709'),
 (4, 8, 'Three', '조금 더 부드러웠으면 좋겠어요.', '2024-03-23 13:01:46.925709'),
 (4, 7, 'Three', '완벽한 맛이에요.', '2024-03-24 13:01:46.925709'),
 (4, 6, 'Three', '맛있는 위스키!', '2024-03-25 13:01:46.925709'),
 (5, 10, 'Three', '맛있어요!', '2024-03-26 13:01:46.925709'),
 (5, 9, 'Three', '정말 최고예요.', '2024-03-27 13:01:46.925709'),
 (5, 8, 'Four', '부드럽지만 더 강한 향이었으면 좋겠어요.', '2024-03-28 13:01:46.925709'),
 (5, 7, 'Five', '정말 좋은 맛이에요.', '2024-03-29 13:01:46.925709'),
 (5, 1, 'Five', '최고예요!', '2024-03-30 13:01:46.925709'),
 (6, 1, 'Five', '정말 좋은 맛이에요.', '2024-03-31 13:01:46.925709'),
 (6, 2, 'Five', '맛있어요!', '2024-04-01 13:01:46.925709'),
 (6, 3, 'Four', '좀 더 부드러웠으면 좋겠어요.', '2024-04-02 13:01:46.925709'),
 (6, 4, 'Five', '정말 최고예요.', '2024-04-03 13:01:46.925709'),
 (6, 5, 'Five', '최고의 위스키!', '2024-04-04 13:01:46.925709'),
 (7, 6, 'Five', '맛있어요!', '2024-04-05 13:01:46.925709'),
 (7, 7, 'Five', '정말 좋은 맛이에요.', '2024-04-06 13:01:46.925709'),
 (7, 8, 'Five', '맛있어요!', '2024-04-07 13:01:46.925709'),
 (7, 9, 'Four', '부드럽지만 더 강한 향이었으면 좋겠어요.', '2024-04-08 13:01:46.925709'),
 (7, 10, 'Five', '최고의 위스키!', '2024-04-09 13:01:46.925709'),
 (8, 11, 'Five', '맛있어요!', '2024-04-10 13:01:46.925709'),
 (8, 12, 'Three', '정말 좋은 맛이에요.', '2024-04-11 13:01:46.925709'),
 (8, 13, 'Four', '좀 더 부드러웠으면 좋겠어요.', '2024-04-12 13:01:46.925709'),
 (8, 14, 'Five', '맛있어요!', '2024-04-13 13:01:46.925709'),
 (8, 15, 'Five', '정말 최고예요.', '2024-04-14 13:01:46.925709'),
 (9, 1, 'Five', '정말 최고예요.', '2024-04-15 13:01:46.925709'),
 (9, 2, 'Five', '맛있어요!', '2024-04-16 13:01:46.925709'),
 (9, 3, 'Three', '부드럽지만 더 강한 향이었으면 좋겠어요.', '2024-04-17 13:01:46.925709'),
 (9, 4, 'Five', '최고의 위스키!', '2024-04-18 13:01:46.925709'),
 (9, 5, 'Five', '맛있어요!', '2024-04-19 13:01:46.925709'),
 (10, 6, 'Five', '맛있어요!', '2024-04-20 13:01:46.925709'),
 (10, 7, 'Five', '정말 최고예요.', '2024-04-21 13:01:46.925709'),
 (10, 8, 'Five', '맛있어요!', '2024-04-22 13:01:46.925709'),
 (10, 9, 'Four', '부드럽지만 더 강한 향이었으면 좋겠어요.', '2024-04-23 13:01:46.925709'),
 (10, 10, 'Five', '최고의 위스키!', '2024-04-24 13:01:46.925709'),
 (11, 11, 'Three', '맛있어요!', '2024-04-25 13:01:46.925709'),
 (11, 12, 'Five', '정말 좋은 맛이에요.', '2024-04-26 13:01:46.925709'),
 (11, 13, 'Four', '좀 더 부드러웠으면 좋겠어요.', '2024-04-27 13:01:46.925709'),
 (11, 14, 'Five', '맛있어요!', '2024-04-28 13:01:46.925709'),
 (11, 15, 'Five', '정말 최고예요.', '2024-04-29 13:01:46.925709'),
 (12, 1, 'Five', '맛있어요!', '2024-04-30 13:01:46.925709'),
 (12, 2, 'Five', '정말 좋은 맛이에요.', '2024-05-01 13:01:46.925709'),
 (12, 3, 'Four', '좀 더 부드러웠으면 좋겠어요.', '2024-05-02 13:01:46.925709'),
 (12, 4, 'Five', '맛있어요!', '2024-05-03 13:01:46.925709'),
 (12, 5, 'Three', '정말 최고예요.', '2024-05-04 13:01:46.925709'),
 (13, 6, 'Five', '맛있어요!', '2024-05-05 13:01:46.925709'),
 (13, 7, 'Five', '정말 최고예요.', '2024-05-06 13:01:46.925709'),
 (13, 8, 'Five', '맛있어요!', '2024-05-07 13:01:46.925709'),
 (13, 9, 'Four', '부드럽지만 더 강한 향이었으면 좋겠어요.', '2024-05-08 13:01:46.925709'),

(13, 10, 'Three', '최고의 위스키!', '2024-05-09 13:01:46.925709'),
(14, 11, 'Five', '맛있어요!', '2024-05-10 13:01:46.925709'),
(14, 12, 'Five', '정말 좋은 맛이에요.', '2024-05-11 13:01:46.925709'),
(14, 13, 'Four', '좀 더 부드러웠으면 좋겠어요.', '2024-05-12 13:01:46.925709'),
(14, 14, 'Five', '맛있어요!', '2024-05-13 13:01:46.925709'),
(14, 15, 'Five', '정말 최고예요.', '2024-05-14 13:01:46.925709'),
(15, 1, 'Five', '맛있어요!', '2024-05-15 13:01:46.925709'),
(15, 2, 'Five', '정말 좋은 맛이에요.', '2024-05-16 13:01:46.925709'),
(15, 3, 'Four', '좀 더 부드러웠으면 좋겠어요.', '2024-05-17 13:01:46.925709'),
(15, 4, 'Five', '최고의 위스키!', '2024-05-18 13:01:46.925709'),
(15, 5, 'Five', '맛있어요!', '2024-05-19 13:01:46.925709'),
(16, 6, 'Three', '맛있어요!', '2024-05-20 13:01:46.925709'),
(16, 7, 'Three', '정말 좋은 맛이에요.', '2024-05-21 13:01:46.925709'),
(16, 8, 'Three', '맛있어요!', '2024-05-22 13:01:46.925709'),
(16, 9, 'Four', '부드럽지만 더 강한 향이었으면 좋겠어요.', '2024-05-23 13:01:46.925709'),
(16, 10, 'Five', '최고의 위스키!', '2024-05-24 13:01:46.925709'),
(17, 11, 'Five', '맛있어요!', '2024-05-25 13:01:46.925709'),
(17, 12, 'Five', '정말 좋은 맛이에요.', '2024-05-26 13:01:46.925709'),
(17, 13, 'Four', '좀 더 부드러웠으면 좋겠어요.', '2024-05-27 13:01:46.925709'),
(17, 14, 'Five', '맛있어요!', '2024-05-28 13:01:46.925709'),
(17, 15, 'Five', '정말 최고예요.', '2024-05-29 13:01:46.925709'),
(18, 1, 'Five', '맛있어요!', '2024-05-30 13:01:46.925709'),
(18, 2, 'Five', '정말 좋은 맛이에요.', '2024-05-31 13:01:46.925709'),
(18, 3, 'Four', '좀 더 부드러웠으면 좋겠어요.', '2024-06-01 13:01:46.925709'),
(18, 4, 'Five', '최고의 위스키!', '2024-06-02 13:01:46.925709'),
(18, 5, 'Five', '맛있어요!', '2024-06-03 13:01:46.925709'),
(19, 6, 'Three', '맛있어요!', '2024-06-04 13:01:46.925709'),
(19, 7, 'Three', '정말 좋은 맛이에요.', '2024-06-05 13:01:46.925709'),
(19, 8, 'Three', '맛있어요!', '2024-06-06 13:01:46.925709'),
(19, 9, 'Four', '부드럽지만 더 강한 향이었으면 좋겠어요.', '2024-06-07 13:01:46.925709'),
(19, 10, 'Five', '최고의 위스키!', '2024-06-08 13:01:46.925709'),
(20, 11, 'Five', '맛있어요!', '2024-06-09 13:01:46.925709'),
(20, 12, 'Five', '정말 좋은 맛이에요.', '2024-06-10 13:01:46.925709'),
(20, 13, 'Four', '좀 더 부드러웠으면 좋겠어요.', '2024-06-11 13:01:46.925709'),
(20, 14, 'Five', '맛있어요!', '2024-06-12 13:01:46.925709'),
(20, 15, 'Five', '정말 최고예요.', '2024-06-13 13:01:46.925709'),
(21, 1, 'Five', '맛있어요!', '2024-06-14 13:01:46.925709'),
(21, 2, 'Five', '정말 좋은 맛이에요.', '2024-06-15 13:01:46.925709'),
(21, 3, 'Four', '좀 더 부드러웠으면 좋겠어요.', '2024-06-16 13:01:46.925709'),
(21, 4, 'Five', '맛있어요!', '2024-06-17 13:01:46.925709'),
(21, 5, 'Five', '정말 최고예요.', '2024-06-18 13:01:46.925709'),
(22, 6, 'Five', '맛있어요!', '2024-06-19 13:01:46.925709'),
(22, 7, 'Five', '정말 좋은 맛이에요.', '2024-06-20 13:01:46.925709'),
(22, 8, 'Four', '좀 더 부드러웠으면 좋겠어요.', '2024-06-21 13:01:46.925709'),
(22, 9, 'Five', '맛있어요!', '2024-06-22 13:01:46.925709'),
(22, 10, 'Five', '정말 최고예요.', '2024-06-23 13:01:46.925709'),
(23, 11, 'Five', '맛있어요!', '2024-06-24 13:01:46.925709'),
(23, 12, 'Three', '정말 좋은 맛이에요.', '2024-06-25 13:01:46.925709'),
(23, 13, 'Five', '맛있어요!', '2024-06-26 13:01:46.925709'),
(23, 14, 'Four', '부드럽지만 더 강한 향이었으면 좋겠어요.', '2024-06-27 13:01:46.925709'),
(23, 15, 'Five', '최고의 위스키!', '2024-06-28 13:01:46.925709'),
(24, 1, 'Five', '맛있어요!', '2024-06-29 13:01:46.925709'),
(24, 2, 'Five', '정말 좋은 맛이에요.', '2024-06-30 13:01:46.925709'),
(24, 3, 'Four', '좀 더 부드러웠으면 좋겠어요.', '2024-07-01 13:01:46.925709'),
(24, 4, 'Five', '맛있어요!', '2024-07-02 13:01:46.925709'),
(24, 5, 'Five', '정말 최고예요.', '2024-07-03 13:01:46.925709'),
(25, 6, 'Five', '맛있어요!', '2024-07-04 13:01:46.925709'),
(25, 7, 'Five', '정말 좋은 맛이에요.', '2024-07-05 13:01:46.925709'),


```
(25, 8, 'Five', '맛있어요!', '2024-07-06 13:01:46.925709'),  
(25, 9, 'Five', '부드럽지만 더 강한 향이었으면 좋겠어요.', '2024-07-07 13:01:46.925709'),  
(25, 10, 'Five', '최고의 위스키!', '2024-07-08 13:01:46.925709');
```