

GIT (형상관리툴)

GitHub의 Git이라는 분산 버전 관리 시스템은 웹 기반 호스팅 서비스이다.

개발 프로젝트의 코드 변경 이력을 효율적으로 관리하고, 협업할 수 있는 장점으로 손꼽힌다.

프로젝트의 소스 코드 저장소를 호스팅하며, 코드 리뷰, 이슈 트래킹, 프로젝트 관리 기능을 통해

쉬운 코드 공유, 안정적인 코드 병합(Merge) 및 관리, 오픈 소스 기여 등을 가능하게 할수 있다.

원격 저장소가 필요한 이유 (한줄 요약)

프로젝트를 안전하게 백업하고 여러 사람과 협업하기 쉬워진다.

깃허브는 깃 기반의 대표적 원격 저장소 서비스로, 코드 업로드·공유·리뷰를 온라인에서 처리한다.

Git

- 분산형 버전 관리 시스템(VCS)
- 소스코드의 변경 이력을 추적하고, 여러 개발자가 동시에 작업할 수 있게 도와줌
- 로컬(내 PC)에서 동작 가능, 인터넷 없어도 버전 관리 가능
- 대표 기능: `commit`, `branch`, `merge`, `checkout` 등
- 예: `git init`, `git add`, `git commit` → 이건 전부 Git 명령어

GitHub

- Git을 기반으로 한 원격 저장소 서비스
- 인터넷에서 Git 저장소를 관리하고 협업할 수 있게 해주는 플랫폼
- 클라우드에 코드를 올려서 다른 사람과 공유하고 협업 가능
- 단순한 저장소 제공뿐 아니라 이슈 관리, PR(Pull Request), 코드 리뷰, 액션(자동화) 같은 기능 제공

- GitHub 말고도 비슷한 서비스로 GitLab, Bitbucket 등이 있음

비유로 설명

- Git = 코드 버전 관리 기술(도구)
- GitHub = Git 저장소를 올려두고 협업할 수 있는 온라인 공간(플랫폼)

정리하면

- Git은 도구이고
- GitHub는 **서비스(웹 기반 협업 플랫폼)**이다.

저장소 생성

깃허브 로그인 후 New Repository를 만든다.

Repository name을 정하고 공개 범위를 Public 또는 Private으로 선택한 뒤 Create Repository를 누른다.

로컬 프로젝트에 원격 연결

처음 한 번만 원격 주소를 등록한다.

```
git remote add origin https://github.com/계정명/리포지터리명.git  
git remote get-url origin  
git remote -v      # 연결 확인  
git remote show origin  # 상세 정보  
git remote rename 이전이름 새이름  
git remote rm origin   # 연결 제거
```

원격으로 업로드

현재 브랜치 커밋을 원격으로 올린다.

```
git push -u origin master
```

원격 변경사항 내려받기

원격과 로컬의 커밋이 달라지면 먼저 내려받아 동기화해야 한다.

푸시 오류가 나면 풀을 먼저 수행한다.

```
git pull origin master # = fetch + merge
```

충돌이 표시되면 파일에서 충돌 표식을 정리한 뒤 커밋한다.

처음 가져올 때는 클론

프로젝트에 처음 참여하거나 새 폴더에 전부 받아올 때 사용한다.

```
git clone https://github.com/유저네임/원격저장소이름.git
```

클론은 .git 이력까지 포함해 그대로 복제한다.

fetch로 미리 확인

병합 없이 원격 브랜치·커밋 정보만 갱신한다.

```
git fetch  
git status
```

협업 기본 흐름

작업 시작 전 fetch 또는 pull로 최신 상태를 맞춘다.

작업 단위를 커밋하고 필요하면 pull로 다시 동기화한다.

충돌이 나면 수동으로 해결 후 커밋하고 push 한다.

협업자 초대

Repository의 Settings → Manage access에서 invite a collaborator로 초대한다.

상대는 이메일 초대를 수락한 뒤 clone 또는 remote 설정으로 참여한다.

원격 연결 재설정

기존 원격을 제거한 뒤 새 주소로 다시 추가할 수 있다.

```
git remote remove origin
```

```
git remote add origin https://github.com/계정명/리포지터리명.git
```

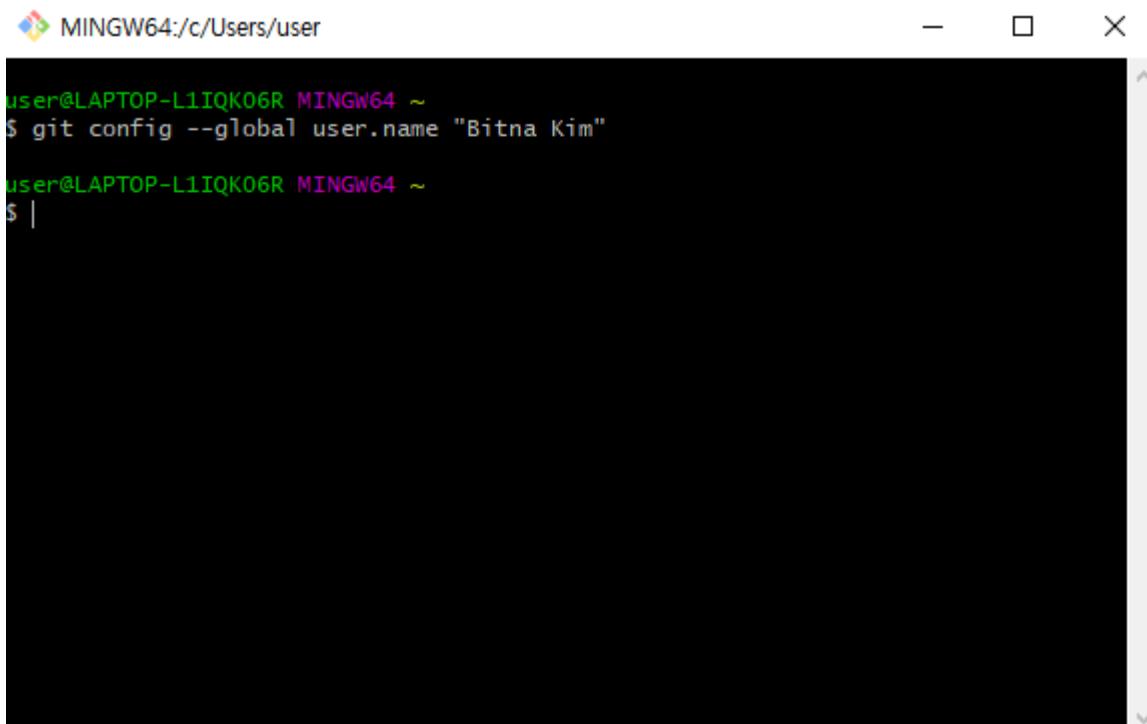
Git = 기능(버전 관리 도구)

Git Bash = Git을 실행할 수 있는 터미널 + 리눅스 스타일 쉘

git bash에서 환경설정

Step 1 : 유저이름 설정

```
git config --global user.name "acehs91"
```



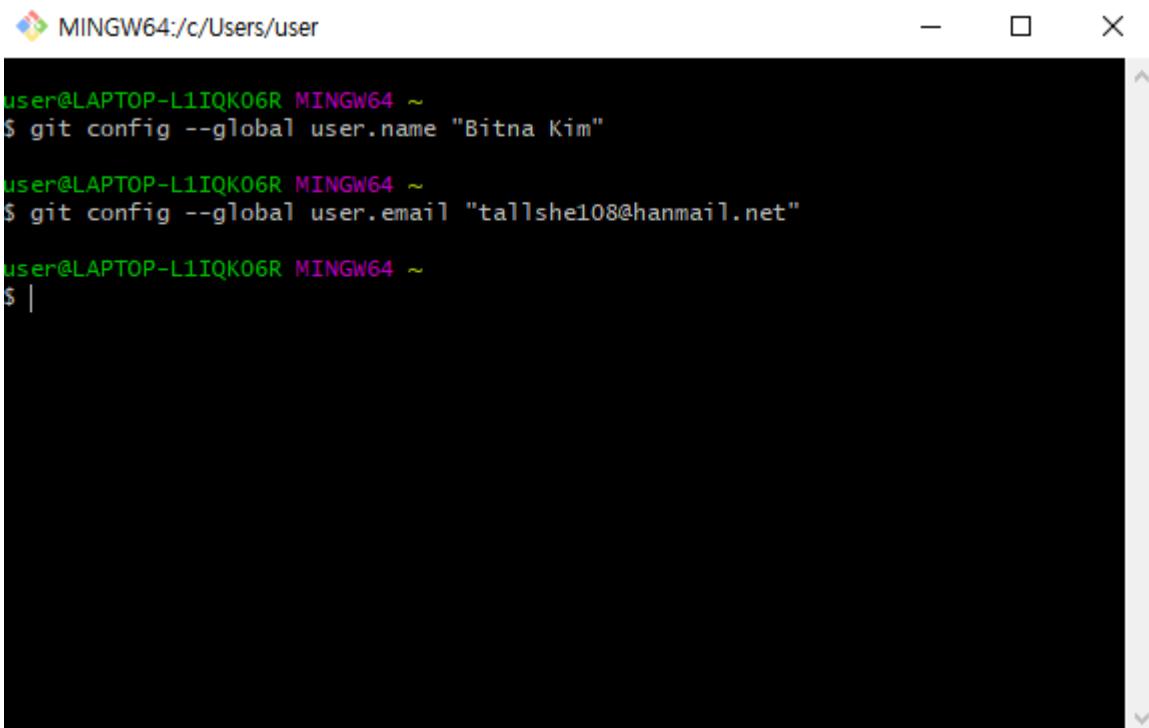
```
user@LAPTOP-L1IQK06R MINGW64 ~
$ git config --global user.name "Bitna Kim"

user@LAPTOP-L1IQK06R MINGW64 ~
$ |
```

Step 2 : 유저 이메일 설정하기

```
git config --global user.email "acehs91@naver.com"
```

Github가입시 사용한 이메일을 써주세요!



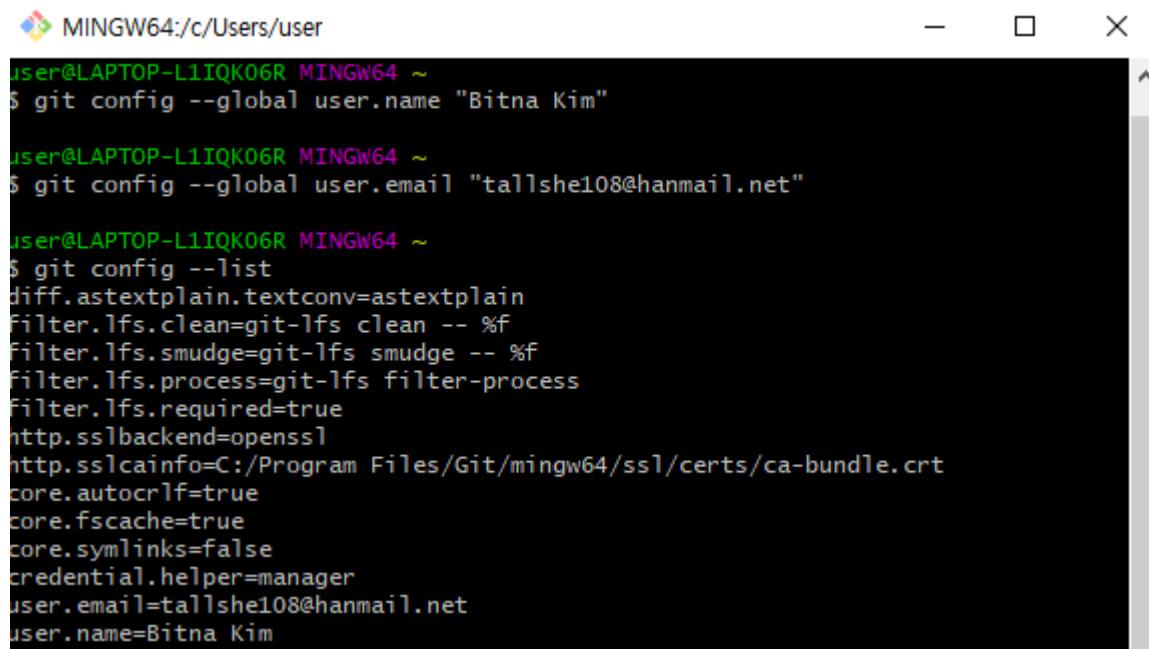
```
MINGW64:/c/Users/user
user@LAPTOP-L1IQK06R MINGW64 ~
$ git config --global user.name "Bitna Kim"

user@LAPTOP-L1IQK06R MINGW64 ~
$ git config --global user.email "tallshe108@hanmail.net"

user@LAPTOP-L1IQK06R MINGW64 ~
$ |
```

Step 3 : 정보 확인하기

git config --list



```
MINGW64:/c/Users/user
user@LAPTOP-L1IQK06R MINGW64 ~
$ git config --global user.name "Bitna Kim"

user@LAPTOP-L1IQK06R MINGW64 ~
$ git config --global user.email "tallshe108@hanmail.net"

user@LAPTOP-L1IQK06R MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
Filter.lfs.clean=git-lfs clean -- %f
Filter.lfs.smudge=git-lfs smudge -- %f
Filter.lfs.process=git-lfs filter-process
Filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
credential.helper=manager
user.email=tallshe108@hanmail.net
user.name=Bitna Kim
```

Github에 처음 코드 업로드하기 🎉

초기화

git init

추가할 파일 더하기

git add .

.(점) 은 모든 파일이라는 뜻, 선택적으로 올리고 싶으면 add뒤에 파일 이름 붙여주면 됨

(예. git add index.html)

상태 확인 (선택사항)

git status

히스토리 만들기

git commit -m "first commit"

- m 은 메세지의 준말로 뒤에 "" 안에 주고 싶은 히스토리 이름을 주면 됨
(즉, 굳이 first commit일 필요가 없다.)

Github repository랑 내 로컬 프로젝트랑 연결

git remote add origin <https://github.com/acehs91/firstproject.git>

잘 연결됐는지 확인 (선택사항)

git remote -v

내가 연결한 주소값이 잘 뜨면 성공! 

Github로 올리기

git push origin master

master 자리에는 branch이름이 들어가면 됨 branch이름이 main라하면

git push origin main 이라고 써야함

Github에 계속 업데이트 하는법 🧑

추가할 파일 더하기

git add .

히스토리 만들기

git commit -m "first commit"

Github로 올리기

git push origin master

내 컴퓨터에 소스코드를 업데이트를 하고 싶으면 이 세개의 스텝만 계속 반복하면 됨.

Github로 팀프로젝트 하는법 👤

Github에서 소스코드 다운로드 & git clone 주소 폴더이름 / 주소는 깃허브에서 들고와야함

폴더이름은 선택사항 (즉 없어도됨) 폴더이름을 줄경우에는 그 폴더가 새로 생성이 되면서 그 안에 코드들이 다운로드가 되고, 폴더이름을 안줄경우엔 깃허브 프로젝트 이름으로 폴더가 자동으로 생기고 그안에 코드들이 다운로드된다.

Github에서 내 브렌치(branch)만들기

git checkout -b 브렌치이름

내 브렌치에 소스코드 업데이트하기

git add .

git commit -m "first commit"

git push origin 브렌치이름

마스터 브렌치에 소스 가져오기(pull)

git pull origin master

pull을 하기전에는 기존에 소스코드들을 commit을 먼저 해놔야 한다.

브렌치끼리 이동하는 법

git checkout 브렌치이름

내가 내 브렌치에서 마스터 브렌치로 이동을 하고 싶거나

다른 브렌치로 이동하고싶으면 해당 명령어를 쓰면 된다.

📌 오라클 설치

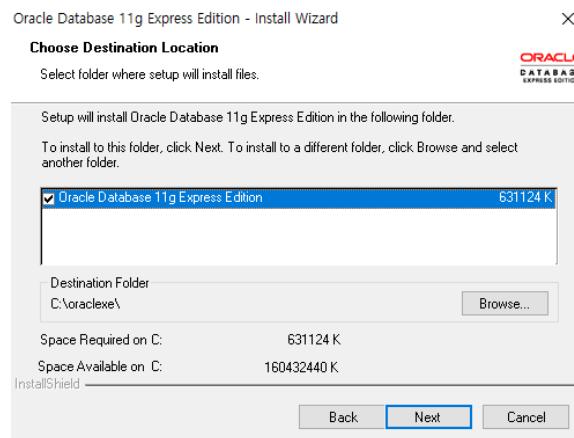
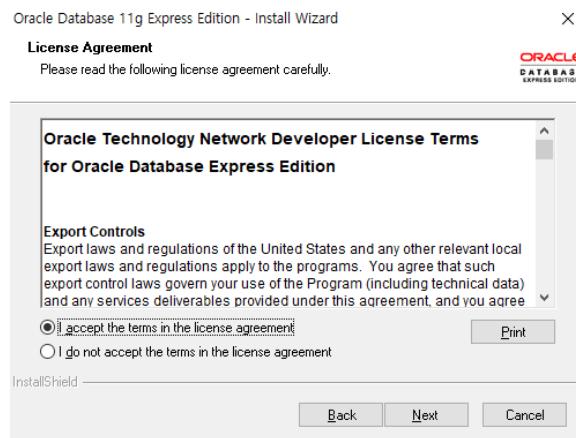
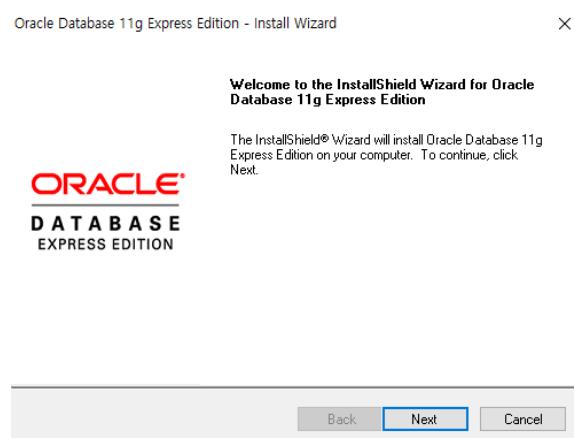
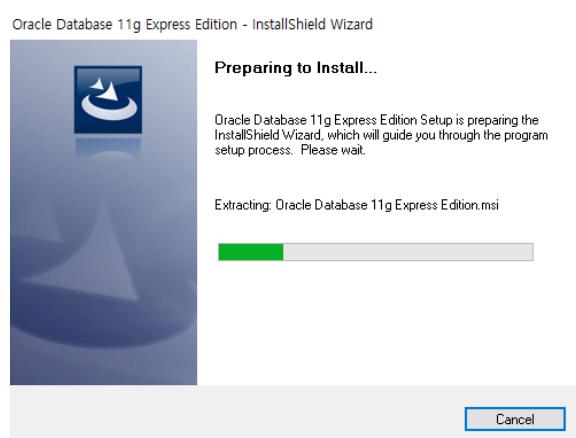
Oracle(RDBMS)은 인스턴스가 데이터베이스를 열어 데이터를 저장·관리하고,
SQL은 그 데이터에 작업을 지시하는 언어다.

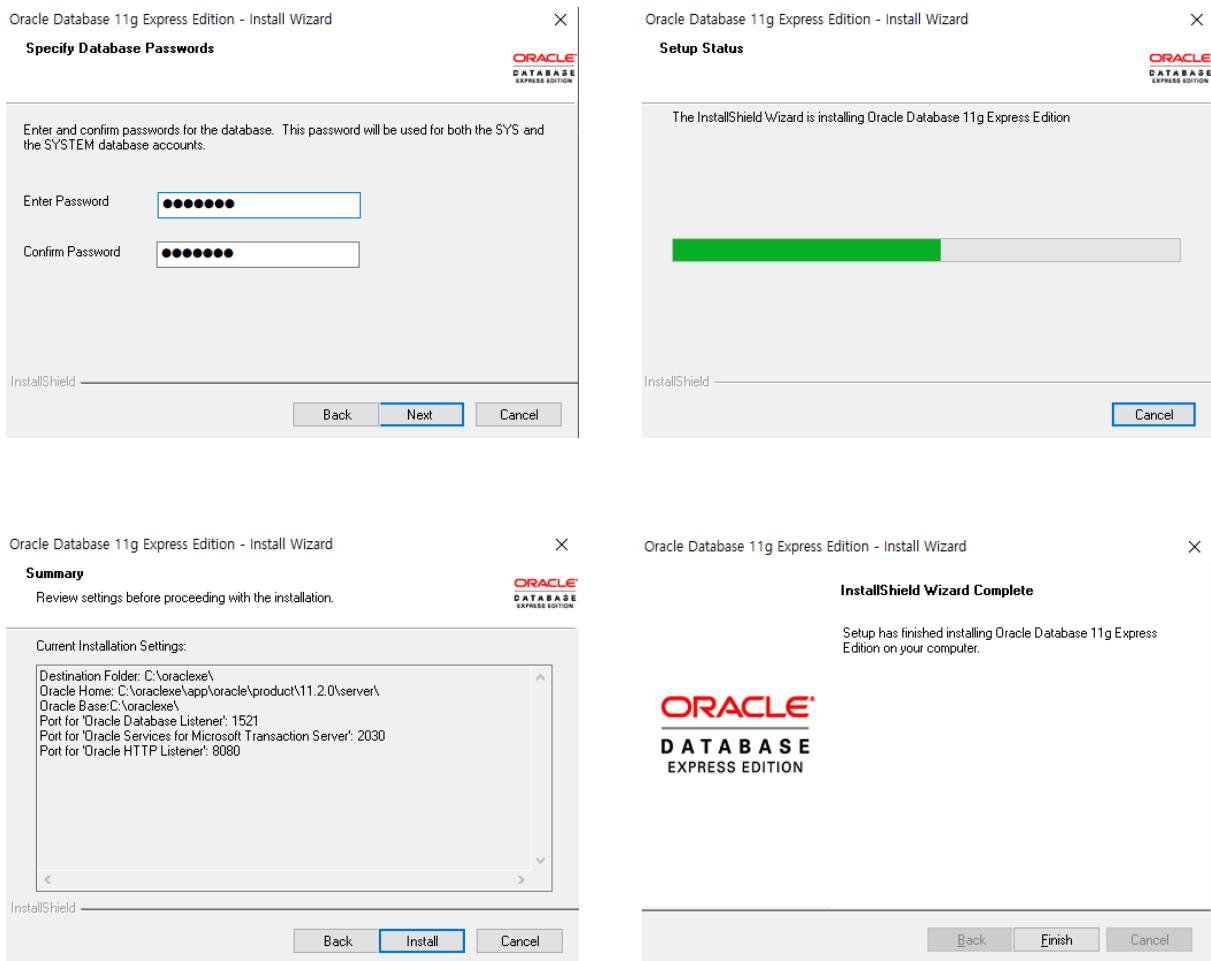
수업시간 오라클 버전 - OracleXE112_Win64

c:/SQL 폴더에 다운로드

<https://naver.me/x3Hf5aQq> - 1111

설치 과정





DB서버의 관리자(SYSTEM)계정 비밀번호 입력하고 'Next' 버튼'

PW : manager

CMD

명령어

(해석)

sqlplus / as sysdba

(사용자 hr 계정 잠금 해제 변경;) - 운영체제 인증으로 SYSDBA 최고권한 접속

```
관리자: 명령 프롬프트 - sqlplus / as sysdba
Microsoft Windows [Version 10.0.19045.6332]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>sqlplus / as sysdba

SQL*Plus: Release 11.2.0.2.0 Production on 금 10월 10 10:48:47 2025

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL> alter user hr account unlock;
User altered.

SQL>
```

alter user hr account unlock;

(사용자 hr 계정 잠금 해제 변경;)

alter user hr identified by hr;

(HR 계정 비밀번호를 'hr'로 설정)

conn hr/hr

(HR 계정으로 재접속)

select * from employees;

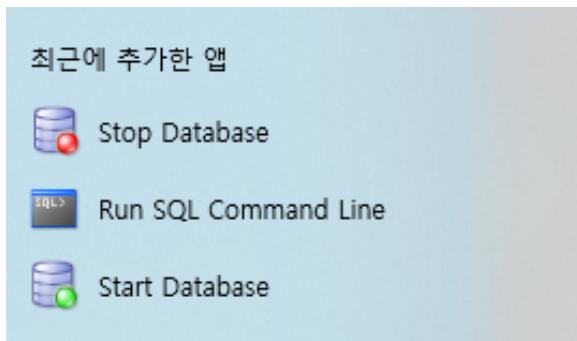
(employees 테이블의 모든 행·열 조회)

show user;

(현재 접속한 사용자 확인)

```
SQL> show user;
USER is "HR"
SQL> exit
Disconnected from Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
```

순서하고 **exit**로 나오고 창을 닫는다.



ORACLE® Oracle Database XE 11.2

Home Storage Sessions Parameters Application Express

Storage
View currently used storage. [Storage >](#)

Sessions
View current database sessions. [Sessions >](#)

Parameters
View initialization parameters. [Parameters >](#)

Application Express
Get started with Oracle Application Express. [Application Express >](#)

Links

- > Online Help
- > Learning Library
- > Oracle Technology Network
- > Oracle SQL Developer
- > Oracle Express Edition
- > Oracle Application Express
- > Oracle JDeveloper 11g
- > Pre-built Developer VIs

News OTN News

Language English Português (Brasil) 中文 (简体) 日本語

Oracle

Login

Username

Password [Login](#)

Login as a database user which has been granted the DBA database role (for example, SYSTEM).

system : manger

DB 비밀번호 : manager 로 설정

11g 버전 설치 : 교육용 무료 버전

오라클 실행 (CMD)

sqlplus / as sysdba : SQL 시작

alter user hr account unlock; : hr 계정 해제

alter user hr identified by hr; : hr 계정 비밀번호로 접속 가능 (비번을 hr 로 설정)

conn hr/hr : hr 계정으로 접속

select * from employees; : 연습용 테이블 보여주기

show user; : 접속한 계정 보여주기

conn system/manager : 시스템 계정으로 접속 (비번은 manager 인 상태)

exit : 접속 해제

오라클 실행 (윈도우 브라우저)

시작메뉴 → Get Started → Application Express → system 계정으로 로그인

Database User → Use Existing 설정

Database username → HR 로 설정 → Create Warkspace 실행

OracleDeveloper

새로 SQL - Developer 다운

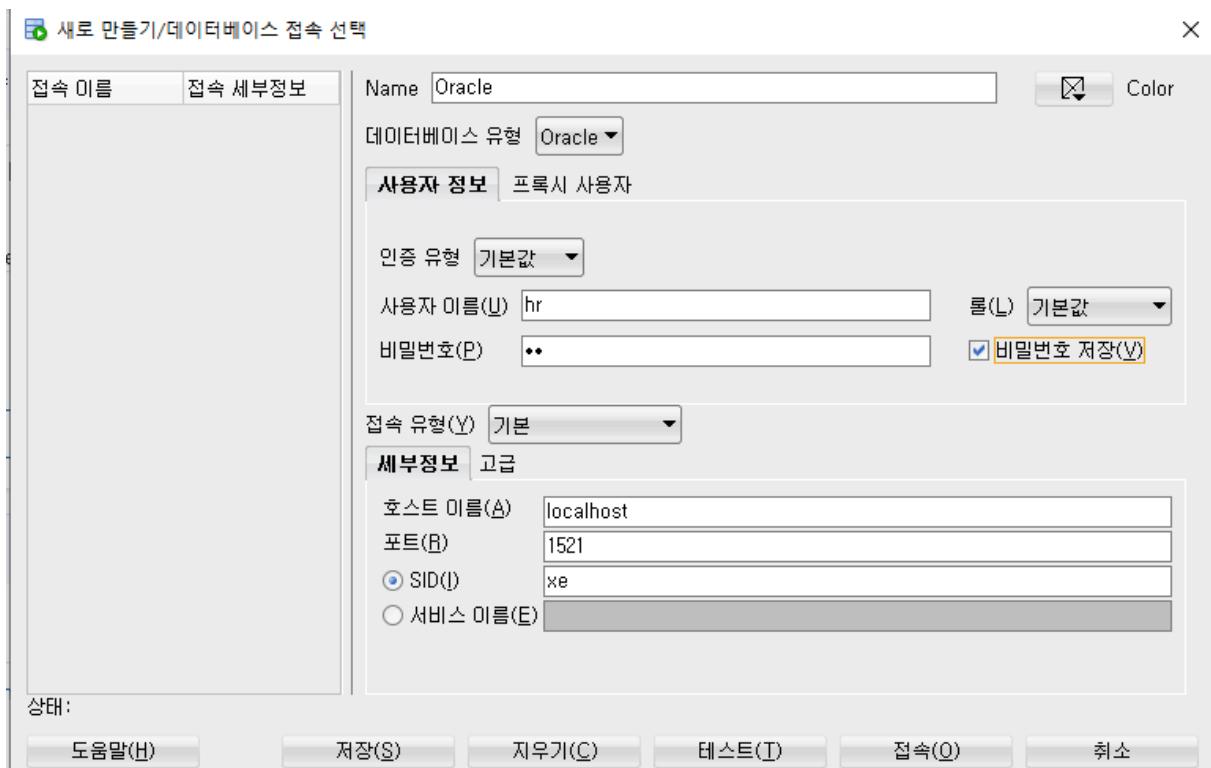
<https://naver.me/5jBeC7wV> - 1111

	icon	2021-07-23	오후 5:03	PNG 파일	2KB		
	sqldeveloper	2021-07-23	오후 5:22	응용 프로그램	89KB		
	sqldeveloper	2021-07-23	오후 5:03	Shell Script	1KB		

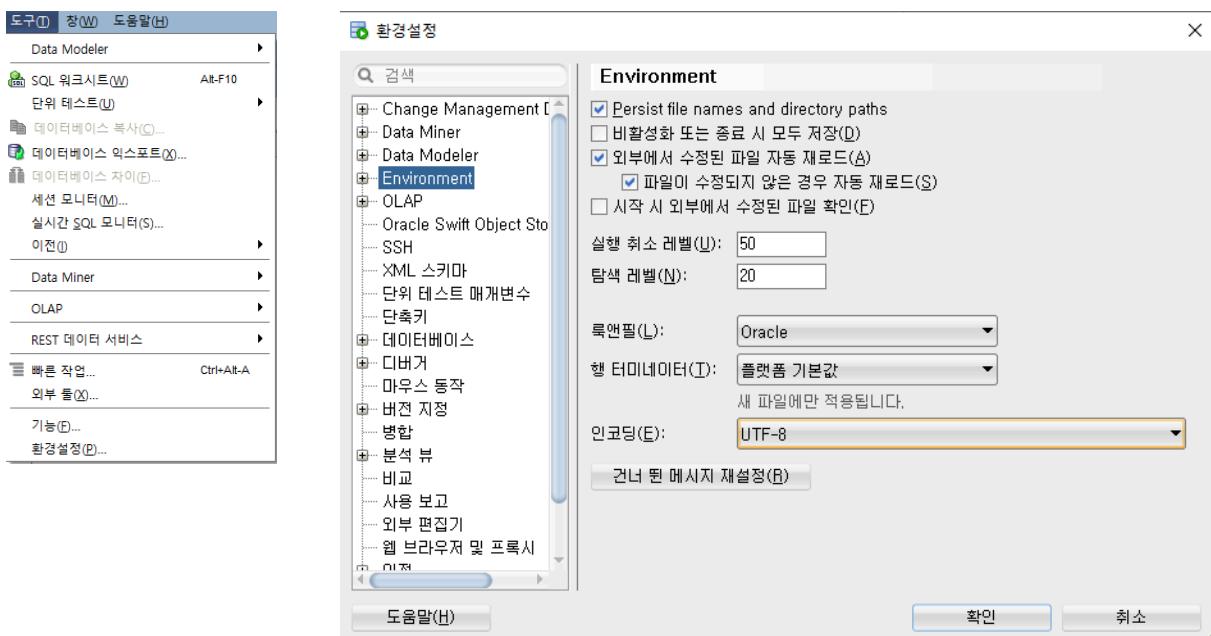
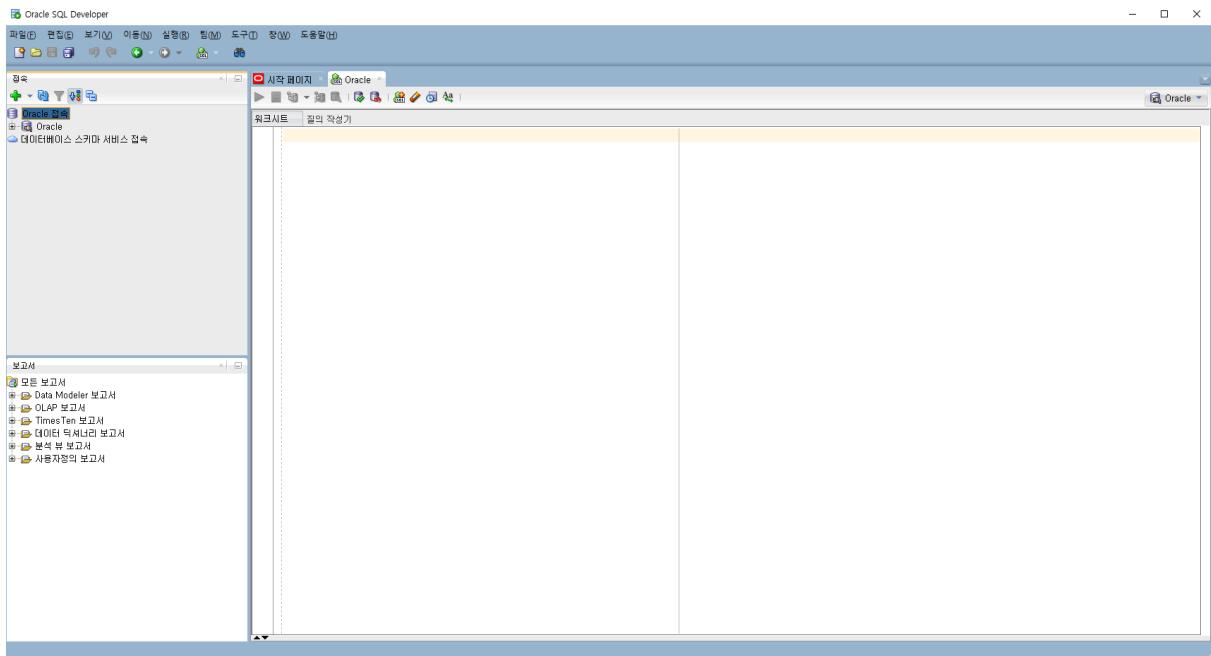
실행



Oracle - HR



접속 누르고 만들때 (반드시 비밀번호 저장 Check !!! : 잊어버리면 엄청 피곤해짐)



도구 - 환경설정 - UTF-8

CMD/ 윈도우 브라우저 방식을 대체하는 가장 대중적인 프로그램
새로 만들기

Oracle 접속 우클릭 → 새 접속

최상림_형상관리_수업정리 2Name : 프로젝트 이름

사용자/비번 : hr/hr

SID(서버아이디) : xe (11g버전 서버아이디)

테스트(테스트 성공 확인용) → 접속
도구 → 환경설정 → Environment → 인코딩 : UTF-8 로 설정