1. MySQL 8.x Community Ection 설치

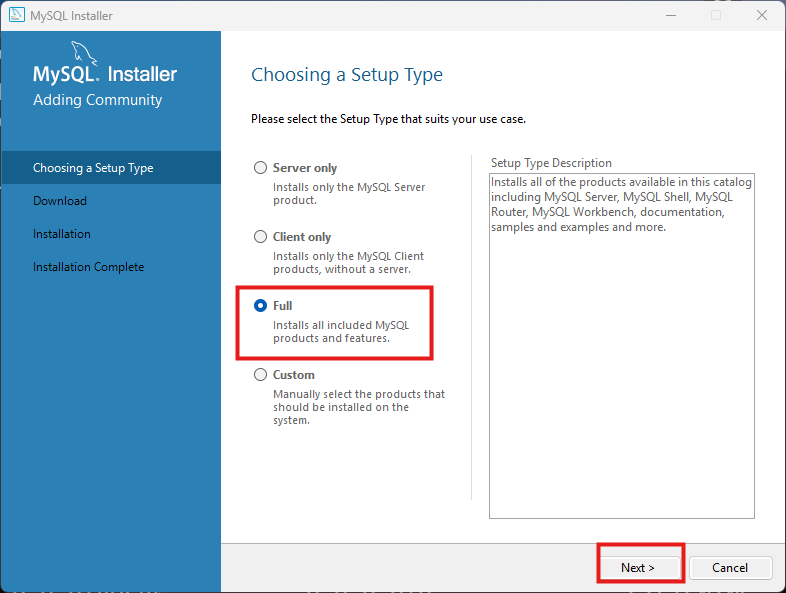
1. **MySQL 8.x Community Ection 설치**

**TCP : 3306**

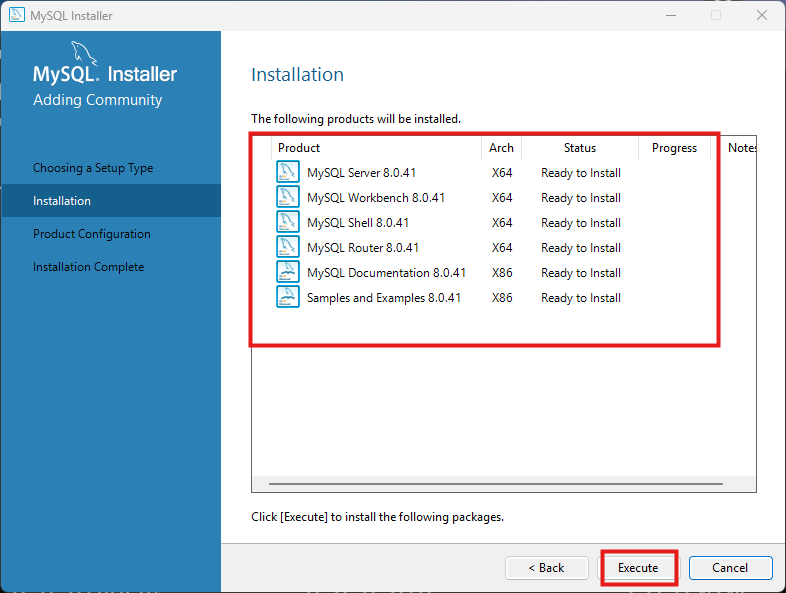
**MySQL Server 🡸 MySQL Server**

**WorkBanch 🡸 MySQL Server를 원격에서 관리하는 관리툴**

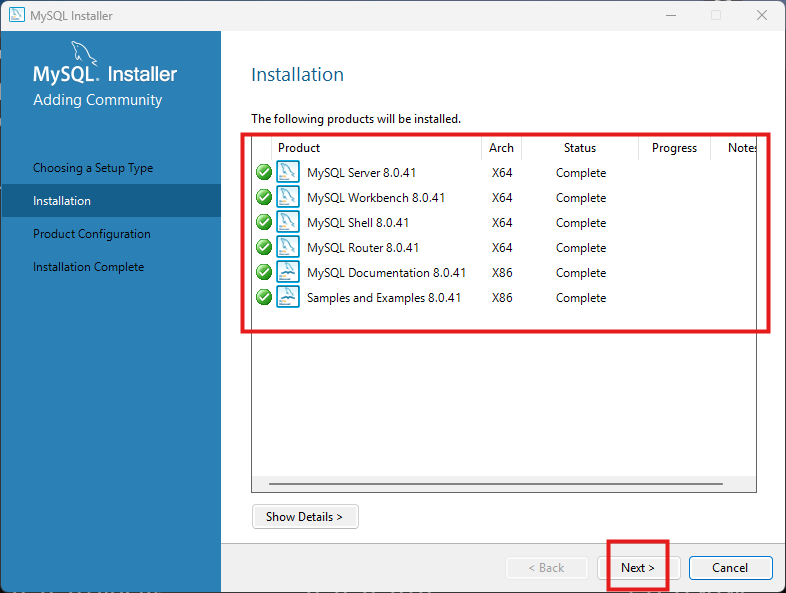
1. **mysql-installer-community-8.0.41.0.msi 실행 [아래 내용이 모두 포함 됨]   
   - MySQL Server  
   - WorkBanch**
2. **설치 타입 - [Full] 선택 후 [Next] 클릭**



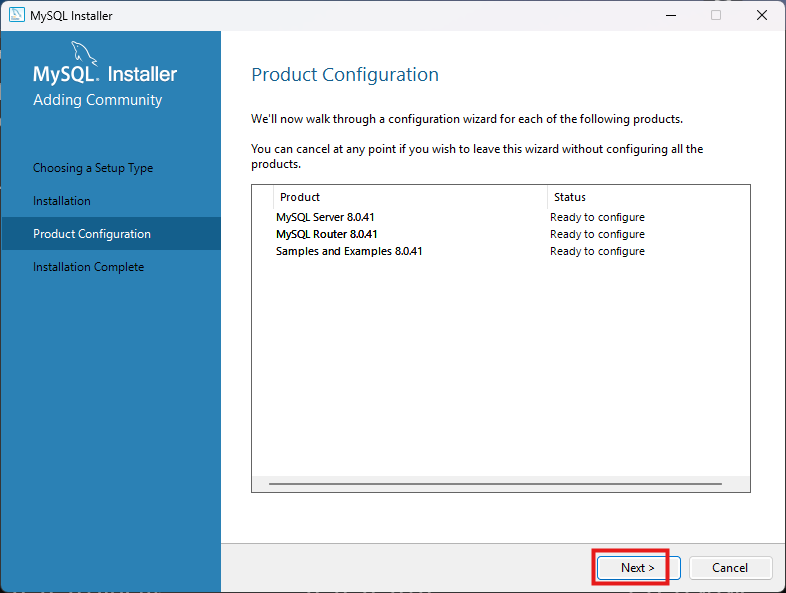
1. **설치 항목 - [Excute] 클릭**



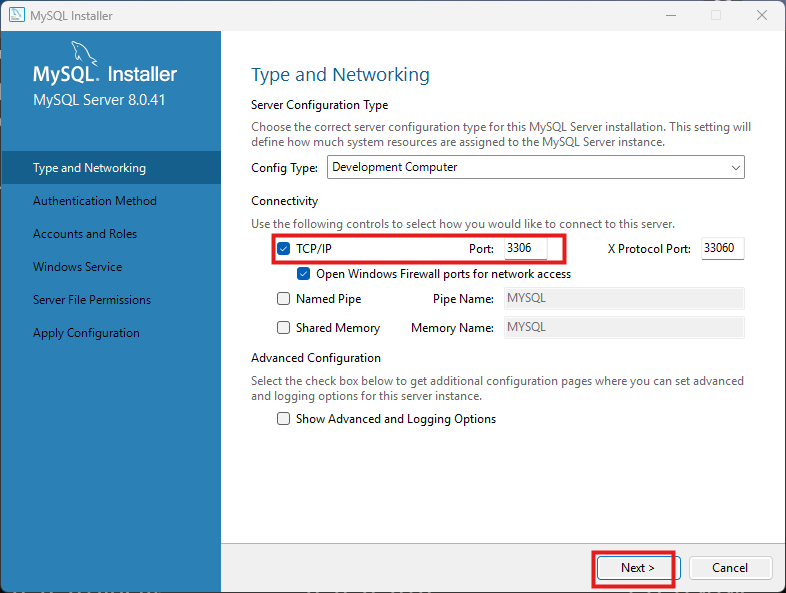
1. **설치 완료 - 왼쪽 [초록색 아이콘] 확인후 [Next] 클릭**



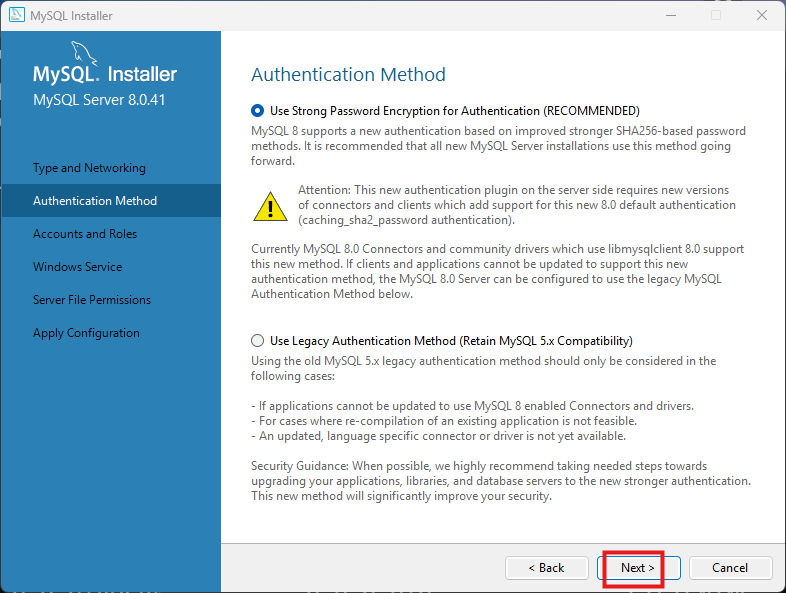
1. **설치 완료 후 MySQL 제품 구성 - [Next] 클릭**



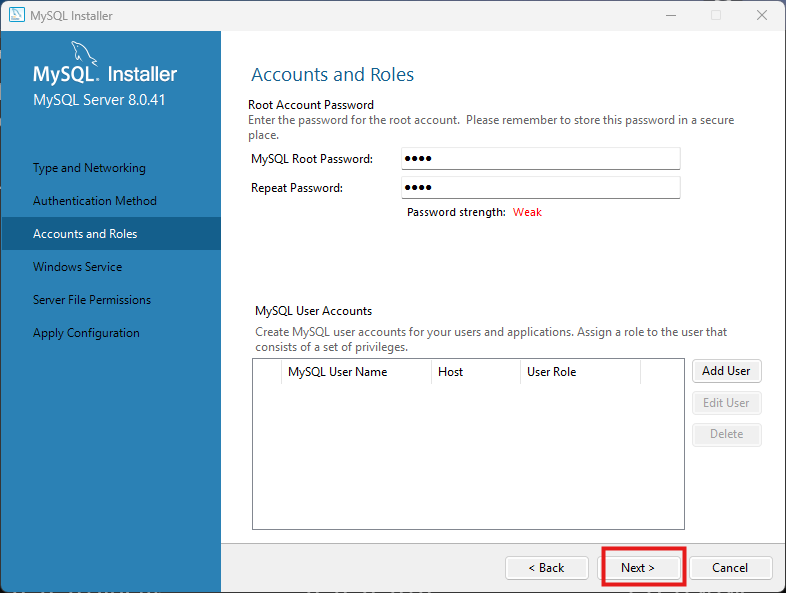
1. **서비스 포트 [ TCP 3306 포트] 확인후 - [Next] 클릭**



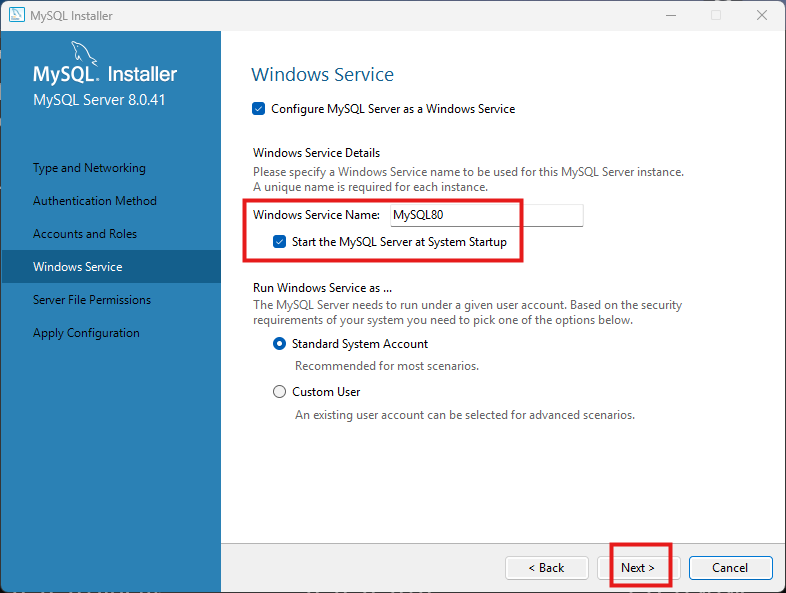
1. **인증 메소드 - [Next] 클릭**



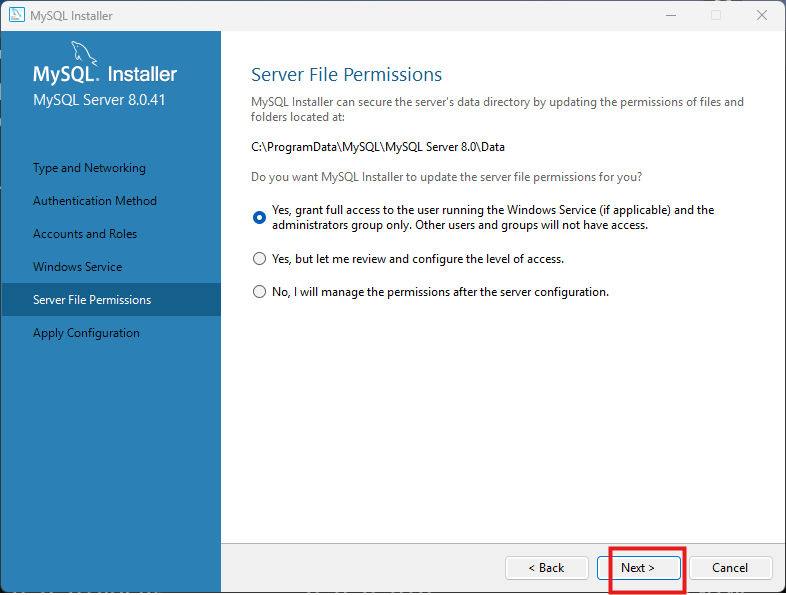
1. **root [최고관리자] 계정 패스워드 부여 : 1234 [주의]**



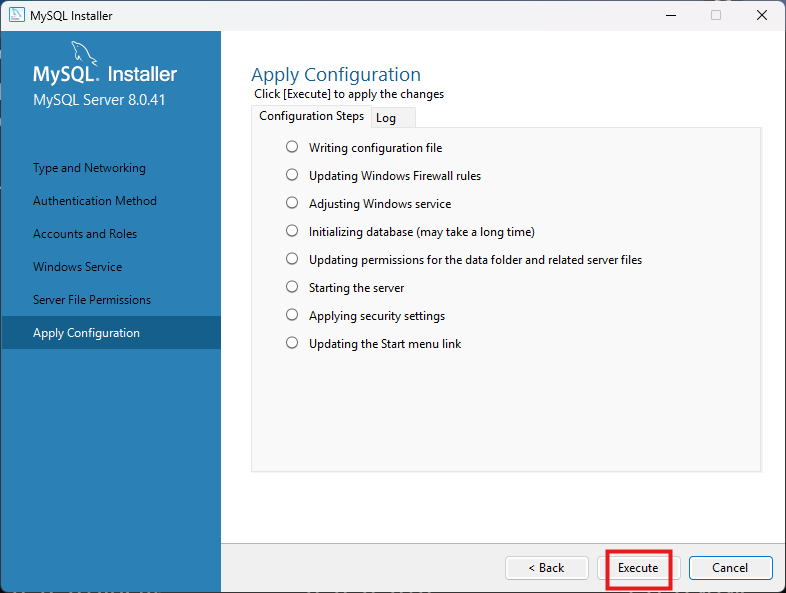
1. **서비스 이름 [MySQL80] 확인 - [Next] 클릭**



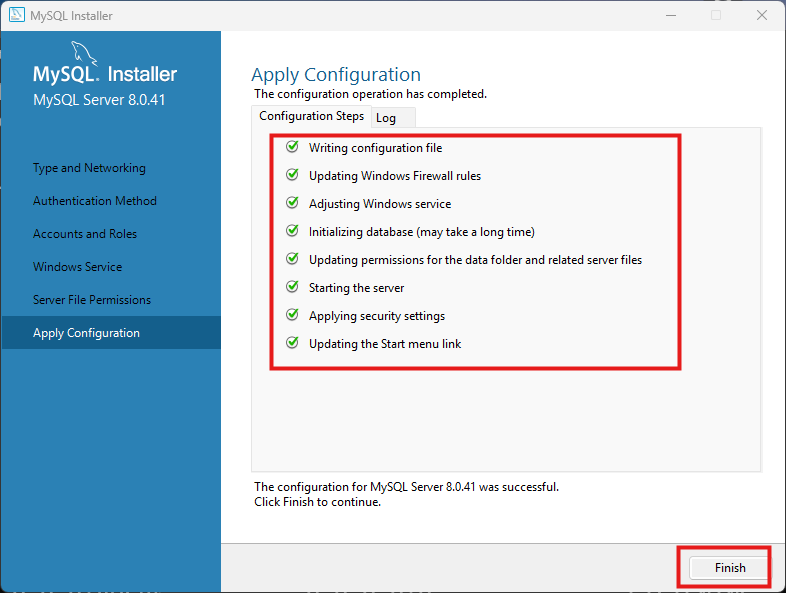
1. **서버 파일 퍼미션 - [Next] 클릭**



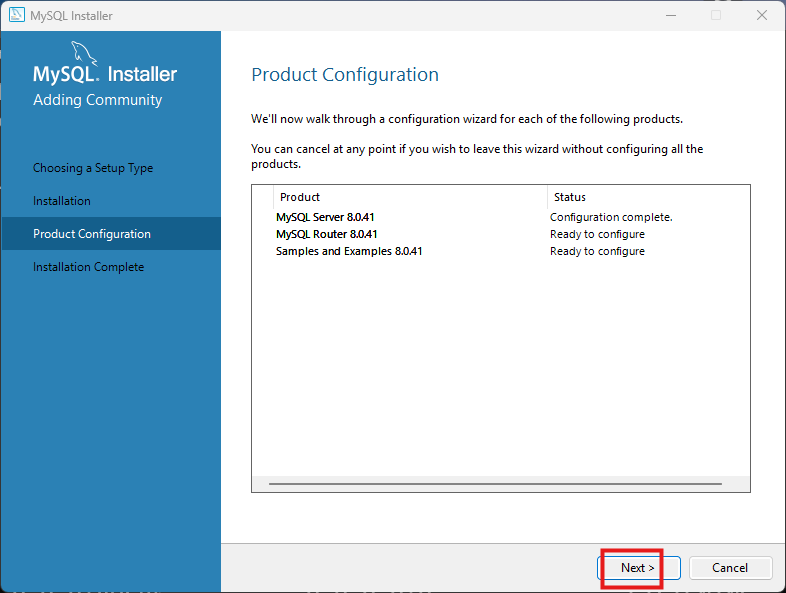
1. **구성 적용 - [Execute] 클릭**



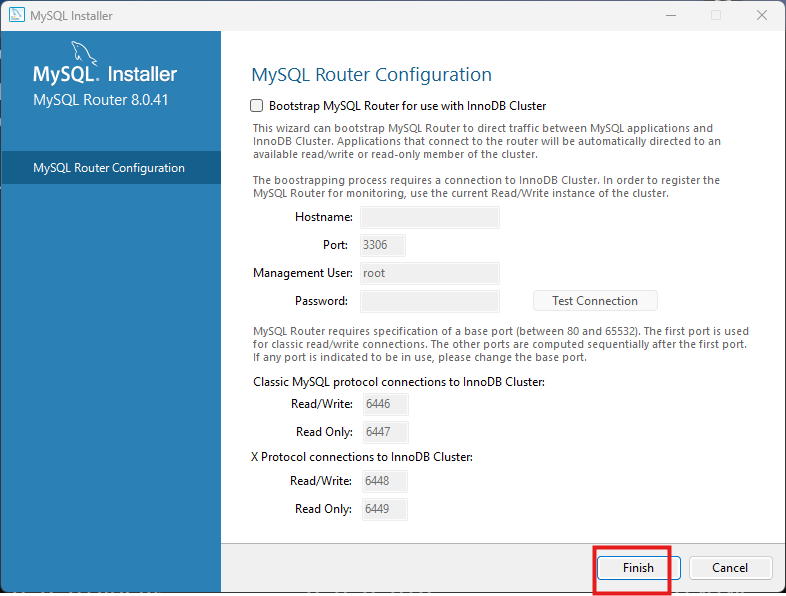
1. **구성 적용 확인 - [Finish] 클릭**



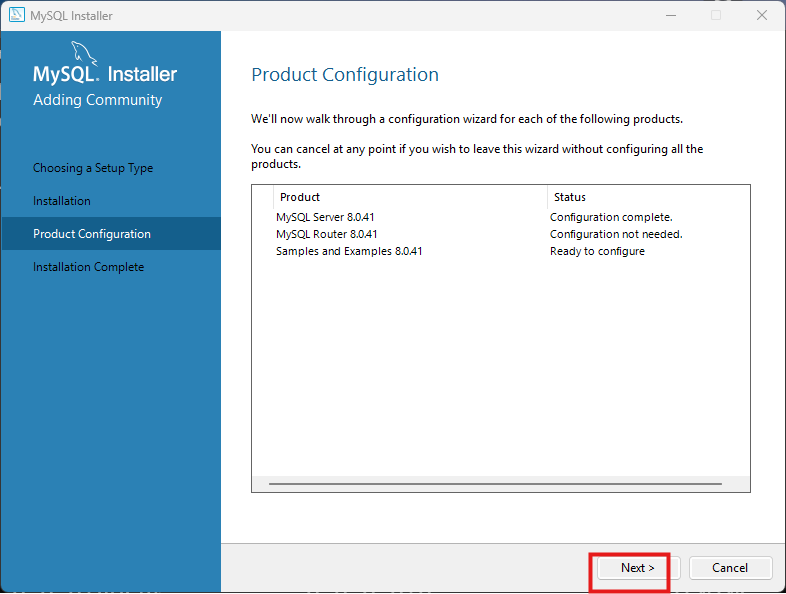
1. **제품 구성 - [Next] 클릭**



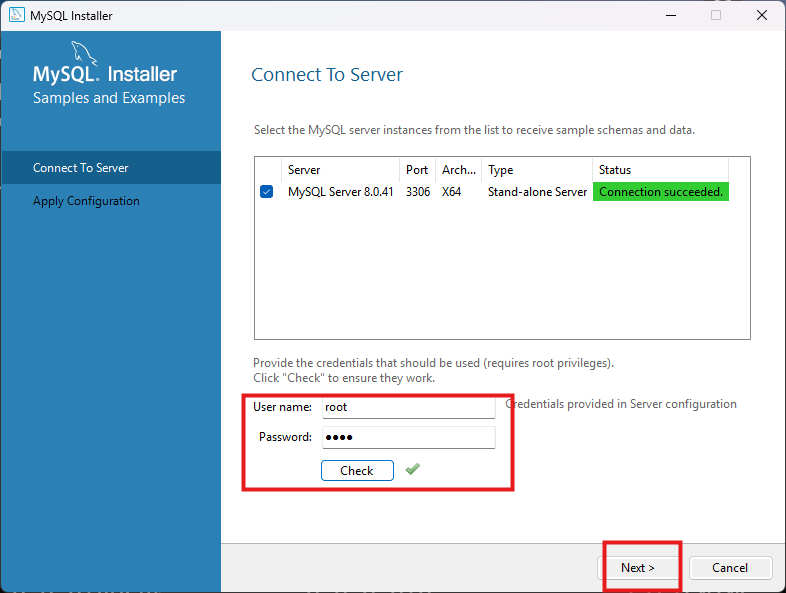
1. **서비스 포트**



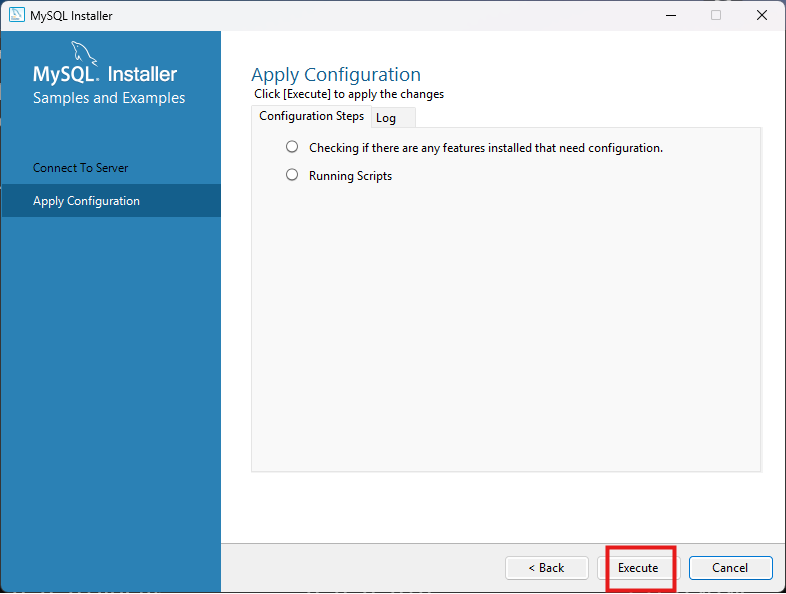
1. **서비스 포트**



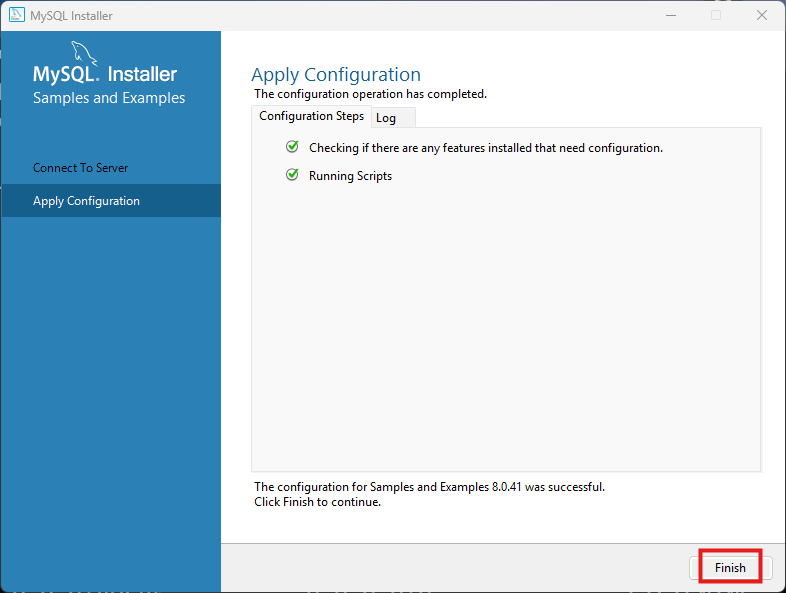
1. **서비스 포트**



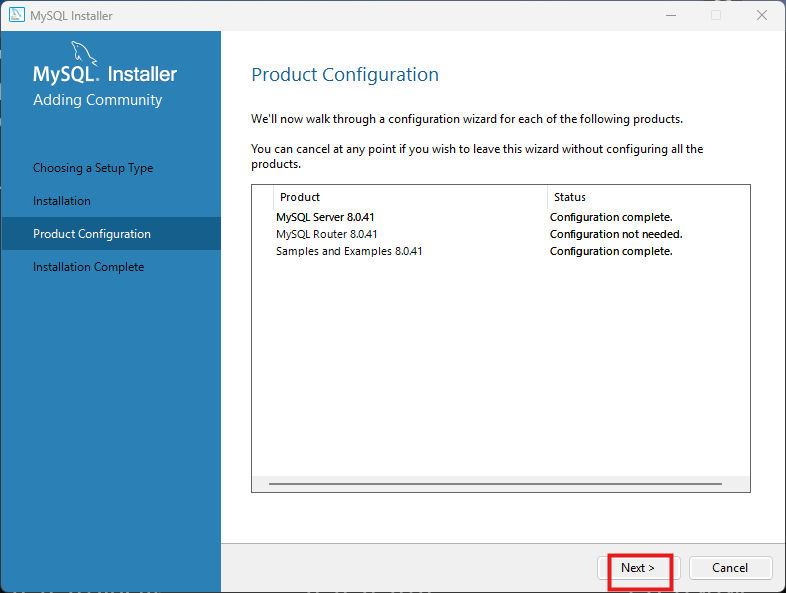
1. **서비스 포트**



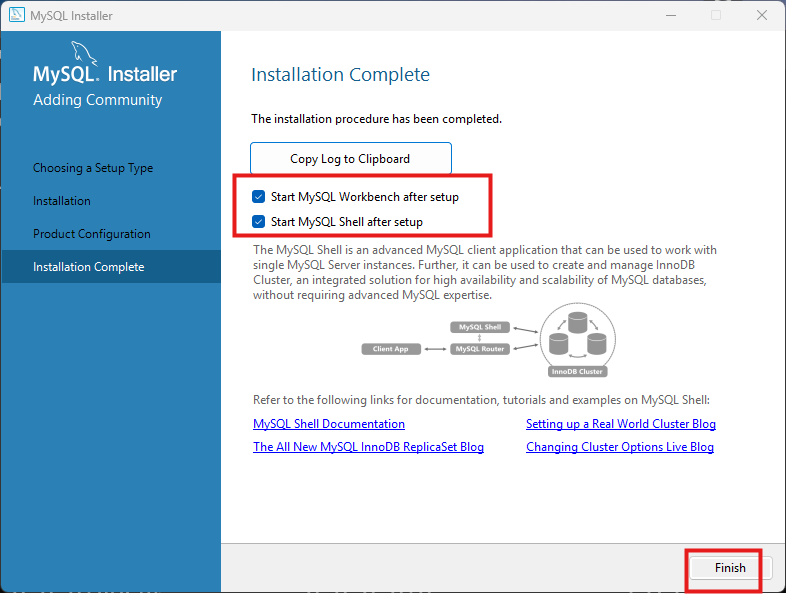
1. **서비스 포트**



1. **서비스 포트**

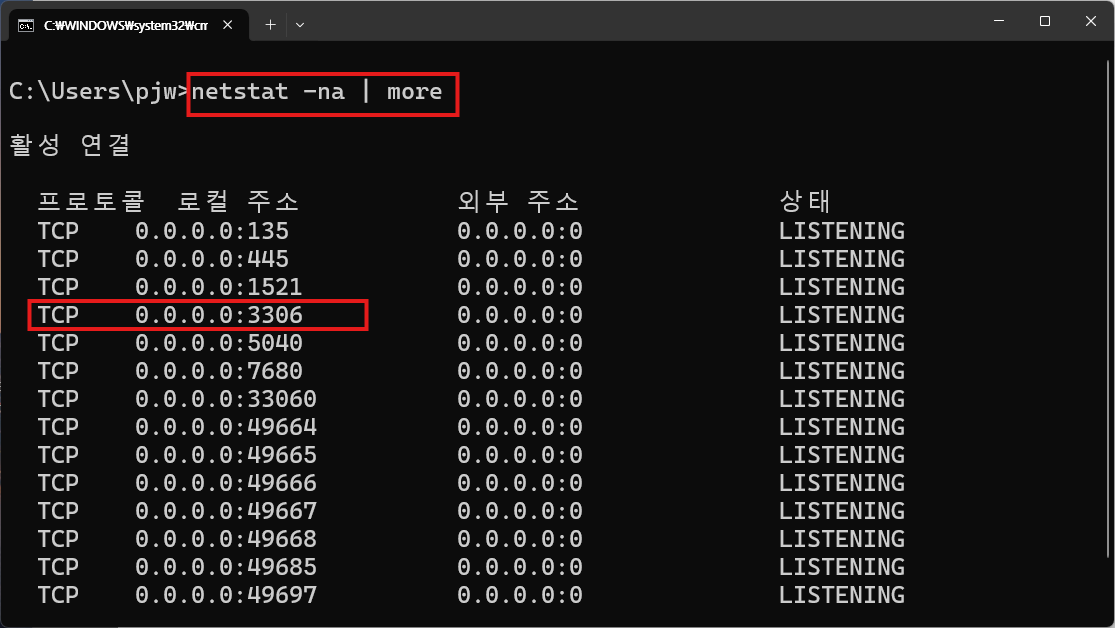


1. **서비스 포트**

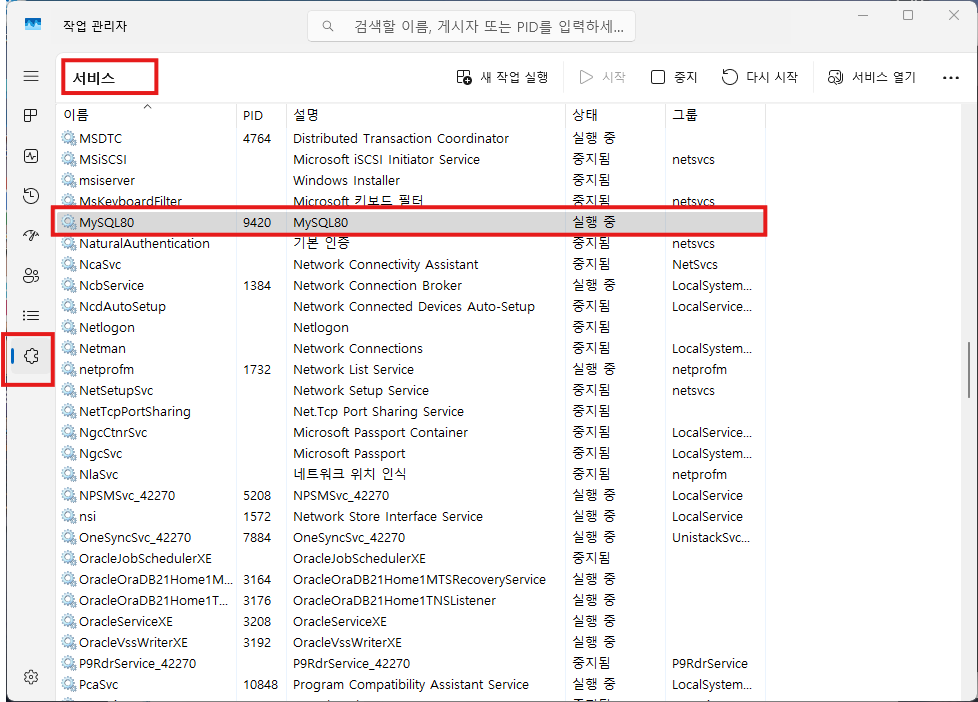


1. **서비스 포트**

**MySQL 설치 확인**

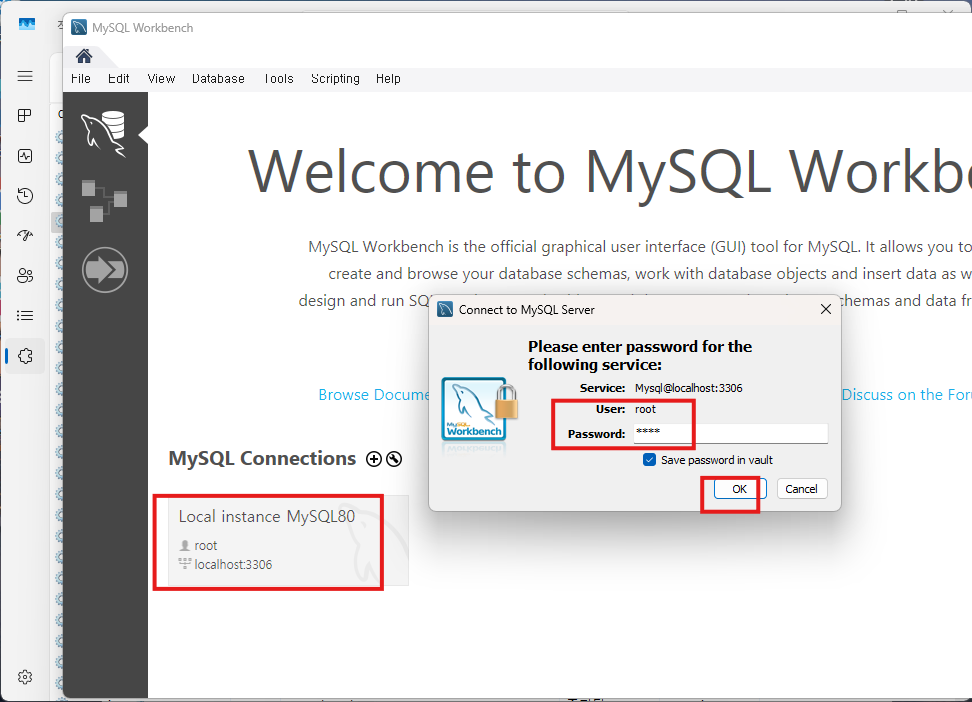


1. **서비스 포트**



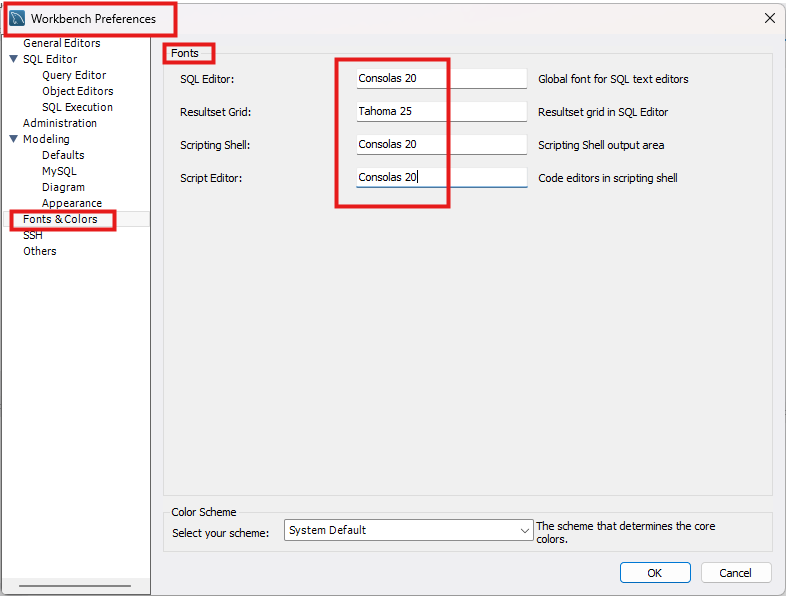
1. **서비스 포트**

**MySQL Workbench 에서 원격의 MySQL Server 에 접속**



1. **서비스 포트**

글자 크기 조정



| **-- 한줄 주석 : 주석 뒤에 반드시 공백 필요**  **/\***  **여러줄 주석**  **\*/**  **-- MySQL Server 에 존재하는 데이타 베이스 확인**  **show databases ;**  **-- DataBase 생성 (테이블, 뷰, 계정, 인덱스, 저장프로시져, 함수, 트리거)**  **create database mydb;**  **-- DataBase 삭제**  **drop database mydb;**  **-- DataBase 접속[사용] , database 의 테이블을 볼려면 db에 접속**  **use mydb;**  **use sakila; -- MySQL 예제 DB**  **-- 테이블의 정보 확인 하기 <== 반드시 해당 DB에 접속후 사용**  **show tables;**  **-- 테이블의 레코드[값을] 출력 , address 테이블의 정보를 출력**  **select \* from address;**  **/\***  **용어 정리**  **- DataBase : SQL Server 하위의 객체 ,**  **테이블, 인덱스, 뷰, 저장프로시져, 함수, 트리거, 계정 을 포함**  **- 테이블 : 실제 데이타가 저장 되어있는 객체 ,**  **- 컬럼[필드] : 테이블에서 열을 컬럼 또는 필드 라고 호칭 , city, name**  **- 레코드 : 모든 컬럼에 한라인으로 들어간 값을 레코드 라고 호칭**  **행으로 들어간 한라인의 값**  **- 인덱스 : 테이블에 컬럼에 부여되는 객체, 컬럼의 검색을 빠르게 하기 위해 생성**  **- 뷰 : 1. 테이블의 특정 컬럼을 숨길수 있다. [ 보안 ]**  **2. 복잡한 조인 구문을 단순하게 처리 함. [ 효율성 ]**    **\*/**  **-- 자신의 db에 접속**  **use mydb;**  **show tables;**  **-- 테이블 생성**  **create table member (**  **id int not null,**  **name varchar(50) not null ,**  **phone varchar(50) null ,**  **addr varchar(100) null**  **);**  **-- 테이블의 정보를 출력**  **desc member ;**  **-- 테이블의 값을 출력**  **select \* from member;**  **-- 테이블의 값을 입력 : insert <== 컬럼명을 명시후 저장 )**  **insert into member ( id, name, phone, addr)**  **values ( 1, '홍길동', '010-1111-2222', '서울');**  **insert into member (id, name)**  **values (2, '강감찬');**  **-- 오류 발생 : id, name컬럼은 반드시 값이 들어가야 한다.**  **insert into member (id, phone)**  **values ( 3, '010-2222-3333');**  **-- 위의 오류 수정**  **insert into member (id, phone, name )**  **values (4, '010-3333-4444', '세종대왕');**  **-- 테이블 삭제 : drop table member ;**  **drop table member ;**  **오후에 상담 : 14:00 ~ 17:00**  **Host OS에 Oracle 설치**  **Host OS에 MySQL 설치** |
| --- |

실습용 Sample Table 생성

**실습용 Sample Table 생성**

| **DROP TABLE IF EXISTS EMPLOYEE;**  **DROP TABLE IF EXISTS DEPARTMENT;**  **DROP TABLE IF EXISTS SALGRADE;**  **CREATE TABLE DEPARTMENT (**  **DNO INT PRIMARY KEY,**  **DNAME VARCHAR(14),**  **LOC VARCHAR(13)**  **);**  **CREATE TABLE EMPLOYEE (**  **ENO INT PRIMARY KEY,**  **ENAME VARCHAR(10),**  **JOB VARCHAR(9),**  **MANAGER INT,**  **HIREDATE DATE,**  **SALARY DECIMAL(7,2),**  **COMMISSION DECIMAL(7,2),**  **DNO INT,**  **FOREIGN KEY (DNO) REFERENCES DEPARTMENT(DNO)**  **);**  **CREATE TABLE SALGRADE (**  **GRADE INT,**  **LOSAL INT,**  **HISAL INT**  **);**  **INSERT INTO DEPARTMENT VALUES (10,'ACCOUNTING','NEW YORK');**  **INSERT INTO DEPARTMENT VALUES (20,'RESEARCH','DALLAS');**  **INSERT INTO DEPARTMENT VALUES (30,'SALES','CHICAGO');**  **INSERT INTO DEPARTMENT VALUES (40,'OPERATIONS','BOSTON');**  **INSERT INTO EMPLOYEE VALUES**  **(7369,'SMITH','CLERK', 7902,'1980-12-17',800,NULL,20),**  **(7499,'ALLEN','SALESMAN', 7698,'1981-02-20',1600,300,30),**  **(7521,'WARD','SALESMAN', 7698,'1981-02-22',1250,500,30),**  **(7566,'JONES','MANAGER', 7839,'1981-04-02',2975,NULL,20),**  **(7654,'MARTIN','SALESMAN',7698,'1981-09-28',1250,1400,30),**  **(7698,'BLAKE','MANAGER', 7839,'1981-05-01',2850,NULL,30),**  **(7782,'CLARK','MANAGER', 7839,'1981-06-09',2450,NULL,10),**  **(7788,'SCOTT','ANALYST', 7566,'1987-07-13',3000,NULL,20),**  **(7839,'KING','PRESIDENT', NULL,'1981-11-17',5000,NULL,10),**  **(7844,'TURNER','SALESMAN',7698,'1981-09-08',1500,0,30),**  **(7876,'ADAMS','CLERK', 7788,'1987-07-13',1100,NULL,20),**  **(7900,'JAMES','CLERK', 7698,'1981-12-03',950,NULL,30),**  **(7902,'FORD','ANALYST', 7566,'1981-12-03',3000,NULL,20),**  **(7934,'MILLER','CLERK', 7782,'1982-01-23',1300,NULL,10);**  **INSERT INTO SALGRADE VALUES**  **(1, 700, 1200),**  **(2,1201,1400),**  **(3,1401,2000),**  **(4,2001,3000),**  **(5,3001,9999);** |
| --- |