**실습과제1: DB Schema Design and DB Population**

■ 실습 기간: 2021. 11. 3.(수) ~ 2021. 11. 12.(금)

■ 제출 마감: 2021. 11. 12. (금) 23:59:59 까지

■ 제출물: 자신의 학번과 이름으로 이루어진 “학번\_이름\_과제번호.zip" 압축 파일을 제출 하여야 하며, 압축 파일은 아래 파일을 포함하여야 한다.

1. 데이터베이스의 백업파일: 학번\_이름\_backup.sql

1.1 실습 완료 후의 DB 백업파일

2. 캡쳐 파일: 학번\_이름\_일련번호.jpg(png 등 다른 파일 형식도 가능)

2.1 데이터베이스 생성 캡처의 경우 일련번호 1

2.2 테이블 생성 쿼리의 경우 일련번호 2

2.3 테이블 결과 화면: 일련번호 3

**[참고]**

- 화면 캡쳐는 반드시 현재 활성화된 윈도우가 아닌 전체 화면을 대상으로 캡처하고 오른쪽 하단에 현재 시간이 포함되어야 한다.

- 캡처파일들은 자신의 수행 과정, 특정 테이블, 결과가 누락되지 않도록 전부 보여줄 수 있는 화면이어야 한다.

- 각각의 캡처 화면이 많은 경우, 1-1, 1-2,..., 3-3... 식으로 일련번호를 지정한다.

- DB백업 명령어는 “mysqldump -u[사용자계정 = root] -p 데이터베이스명 > 저장될 파일명” 이다.

\* MySQL DBMS 설치에 관한 화면 캡쳐는 필요가 없으며, 이미 본인의 컴퓨터에 MySQL DBMS가 설치되어 있는 학생들은MySQL 버전이 8.0.27과 동일한지 체크하기 바람. 혹시 버전이 다르면 채점 시 제대로 동작하지 않을 수도 있기 때문임

■ 제출할 곳

스마트캠퍼스 과제제출 게시판

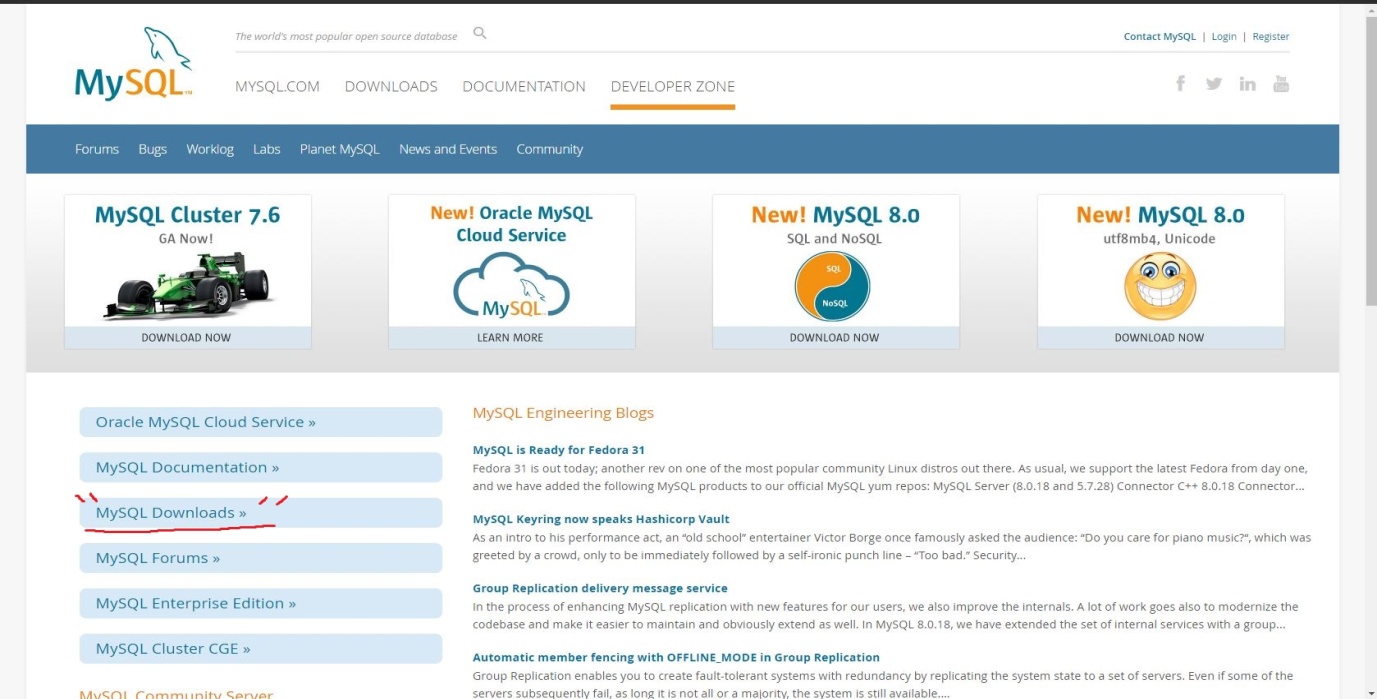
■ 과제 관련 질의

스마트캠퍼스 문의 게시판

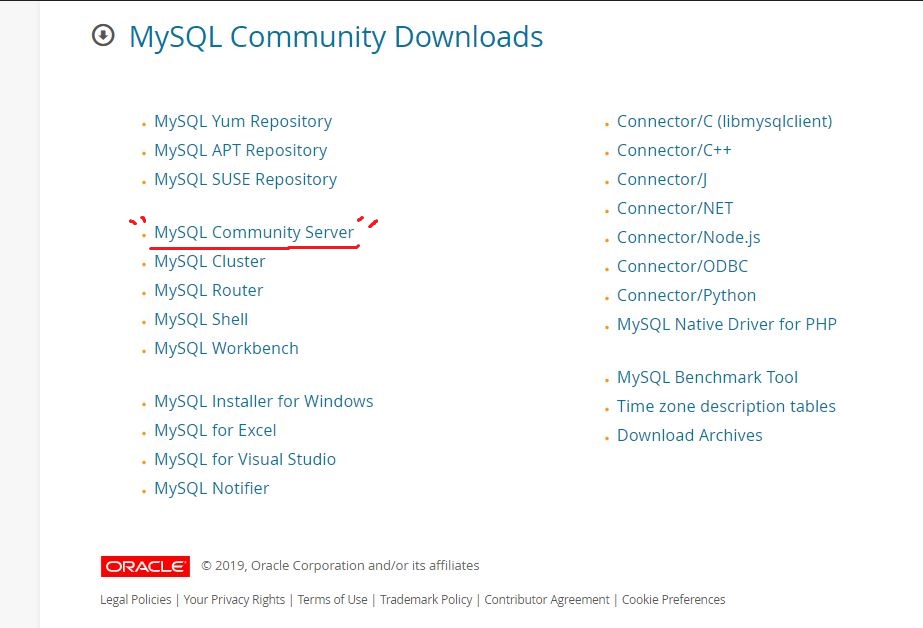
**1. MySQL 설치하기**

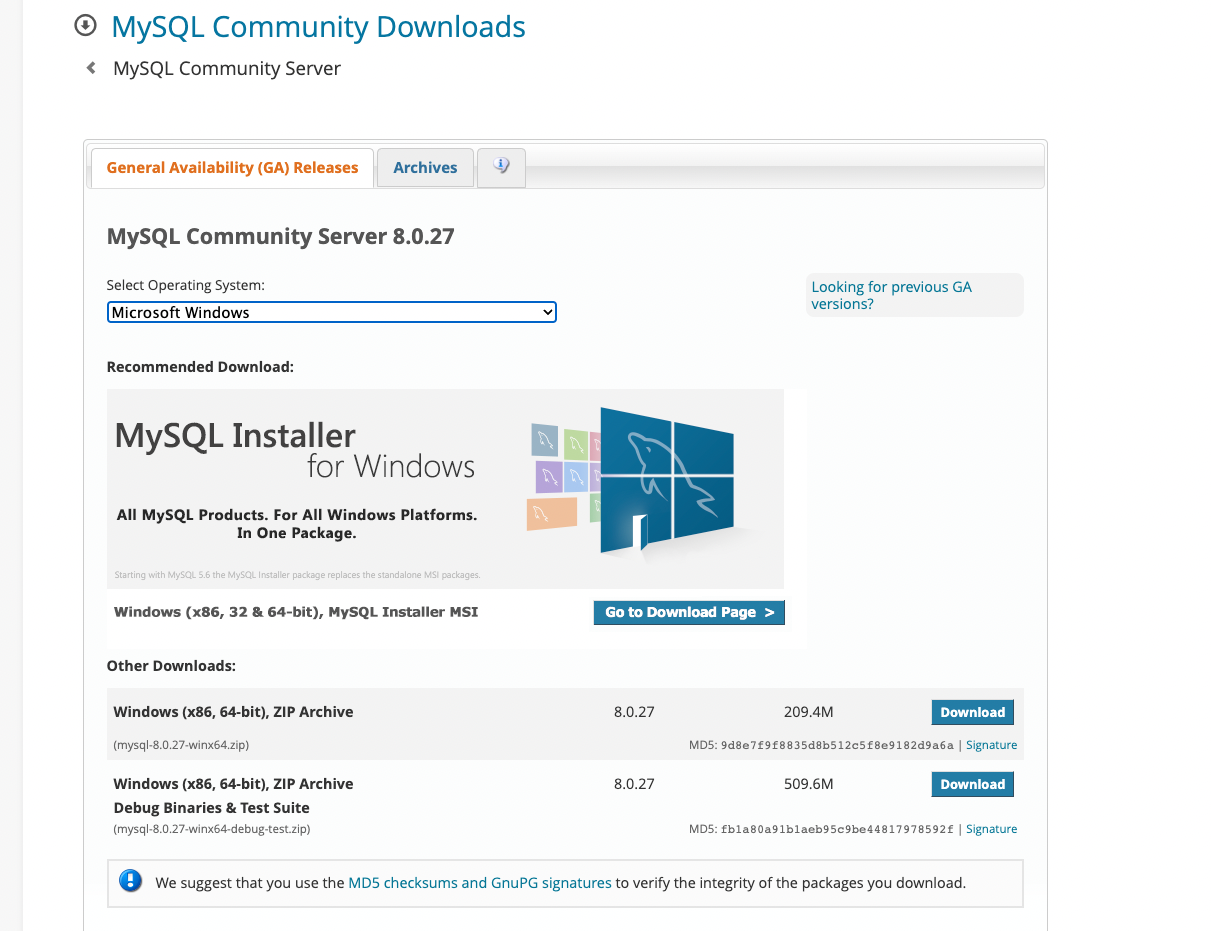
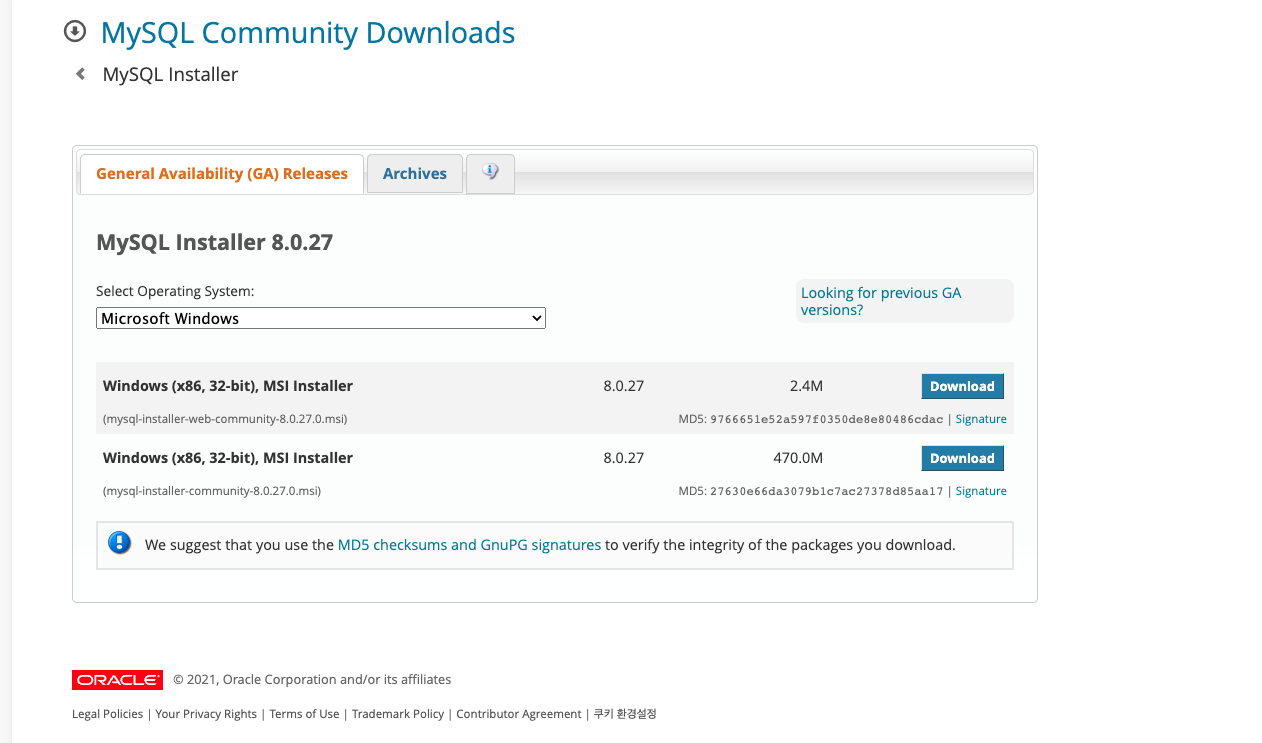
<http://www.mysql.com/>페이지에서 DEVELOPER ZONE 메뉴를 선택.

[MySQL Downlads](http://dev.mysql.com/downloads/mysql/)**클릭**

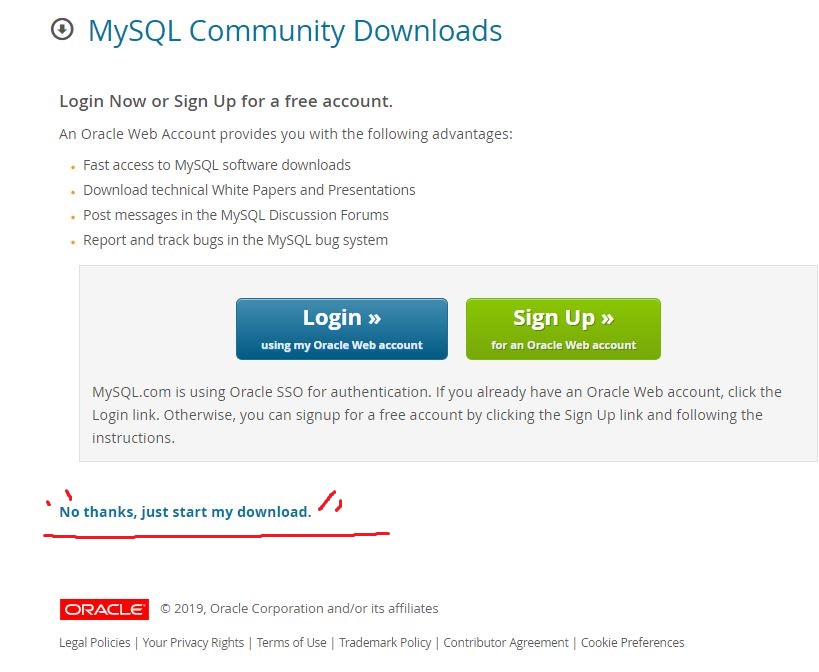


● Developer Zone > MySQL Community Server 클릭



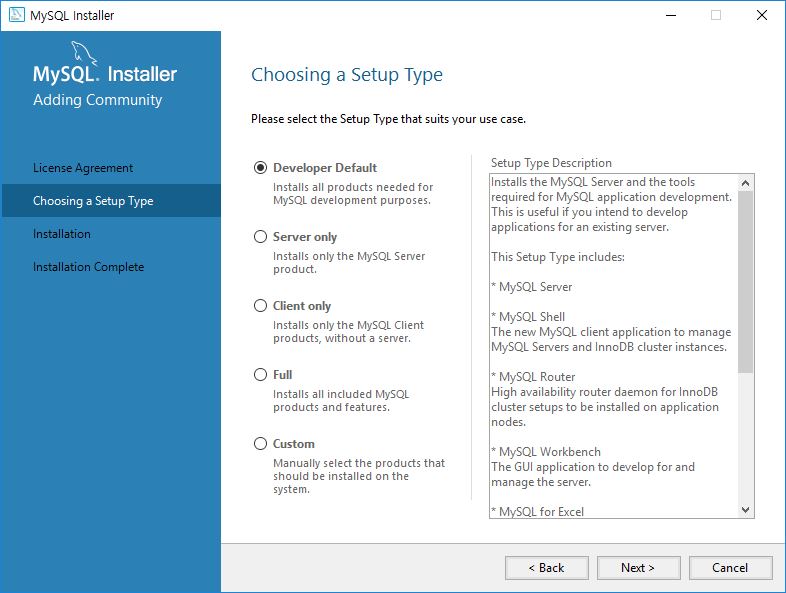
● Select Operating System: microsoft Windows > 중단에 Go to download page 클릭

● Windows (x86, 32-bit), MSI Installer란에 있는 Download 클릭

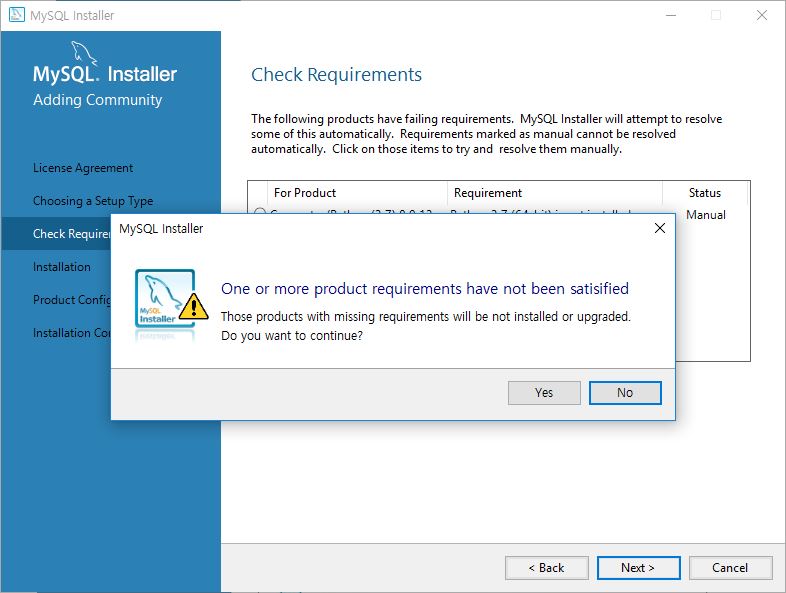


● 로그인과 회원가입 과정 필수 아님

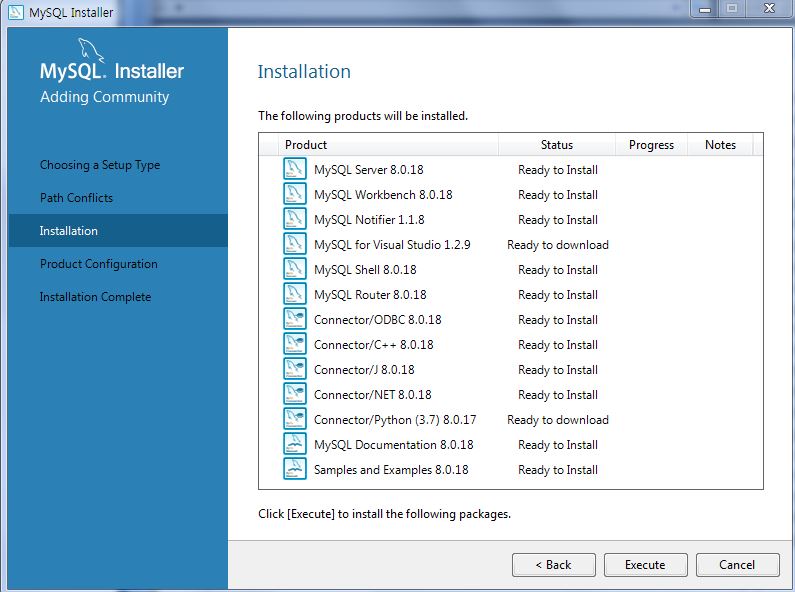
2. Installer 실행하기

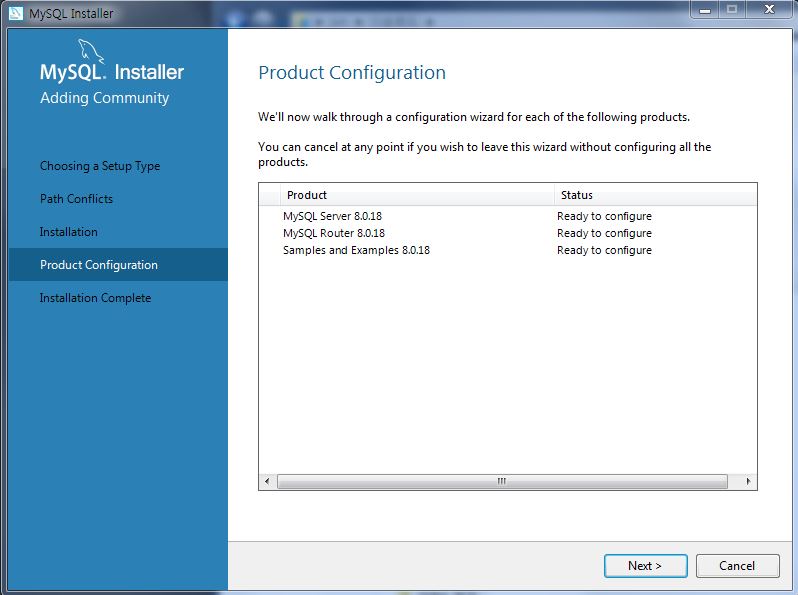


설정 변경 없이 Next 클릭

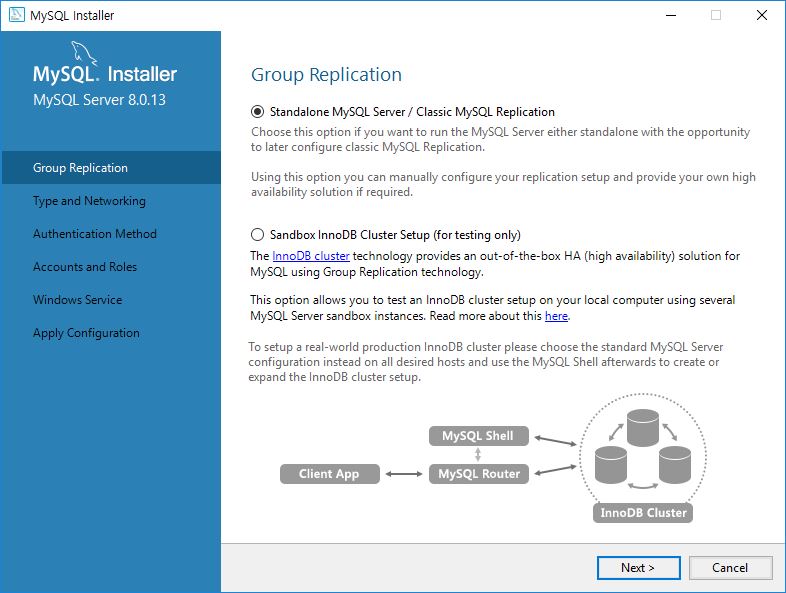


환경이나 설치된 프로그램에 따라 MySQL과 연동을 위한 추가 설치 제시  
무시하고 NEXT 클릭,경고문의 Yes 클릭 후NEXT 클릭

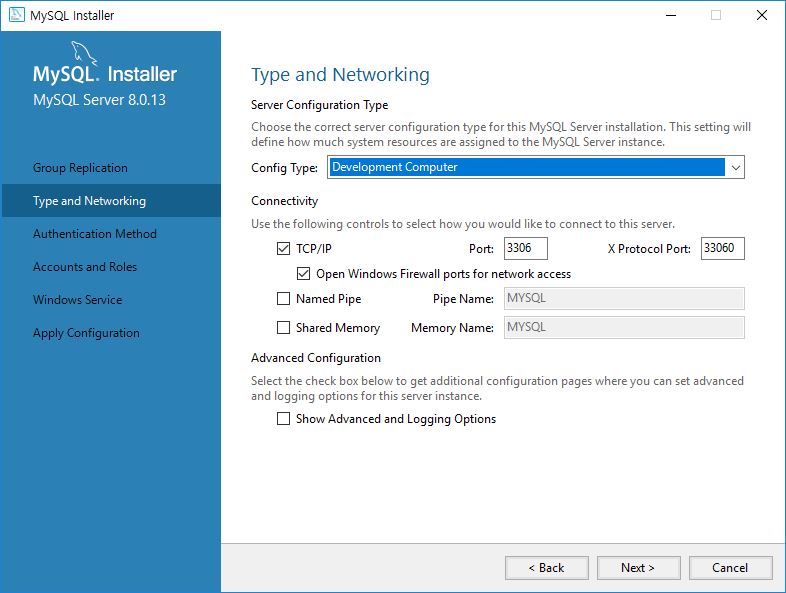


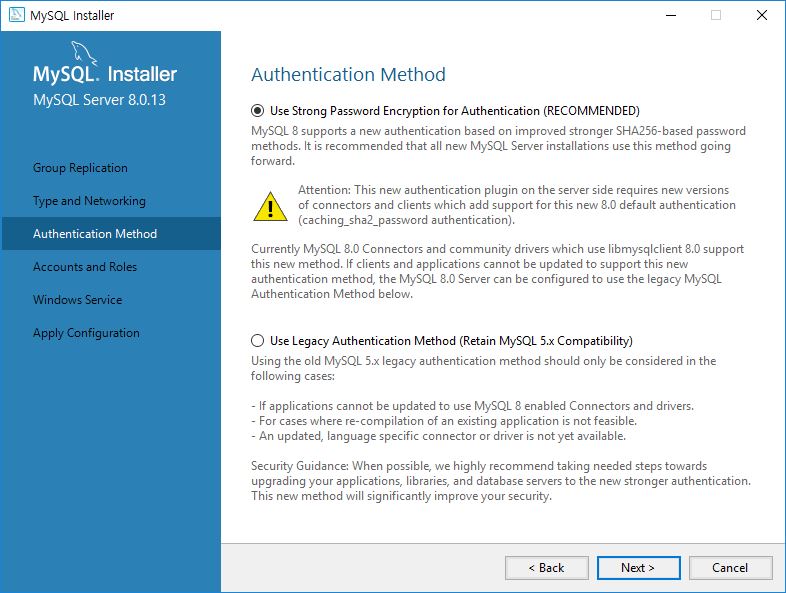
Execute 클릭-> 설치 시작

Next 클릭

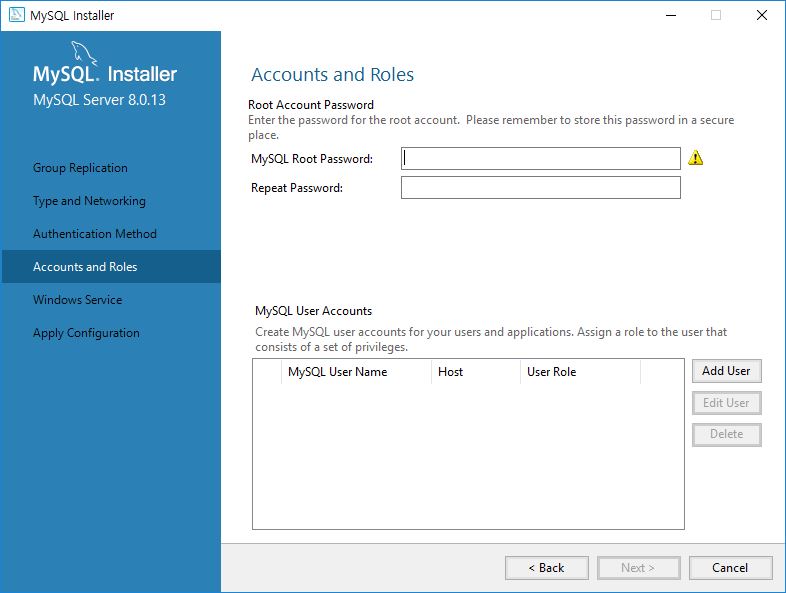


변경없이Next 클릭

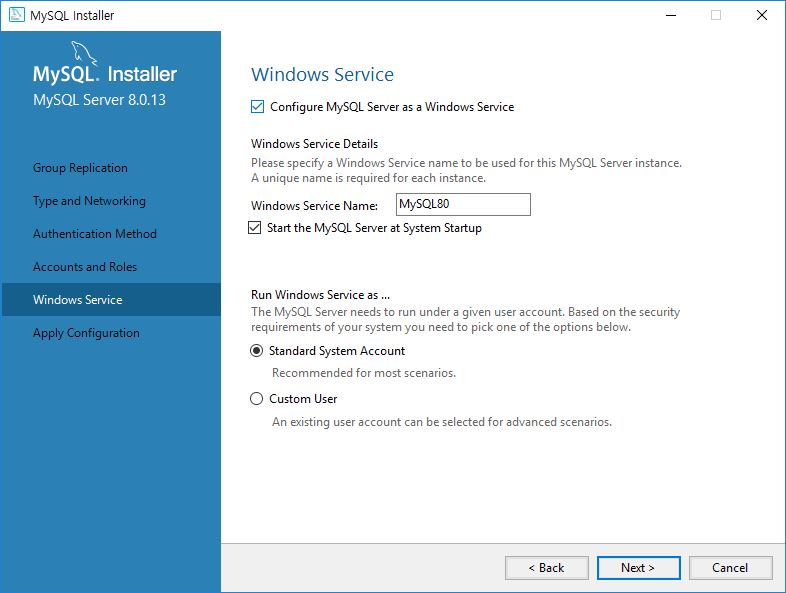


기본값으로 Next 클릭

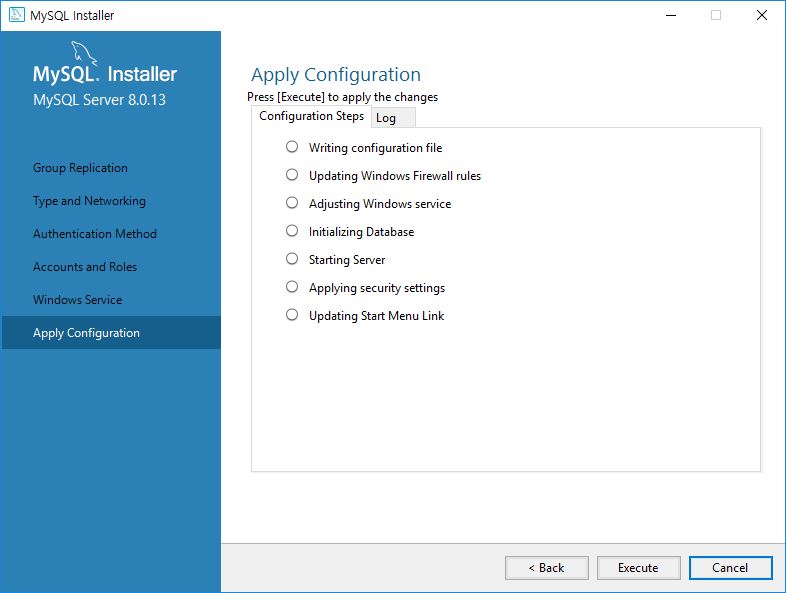
기본값으로 Next 클릭



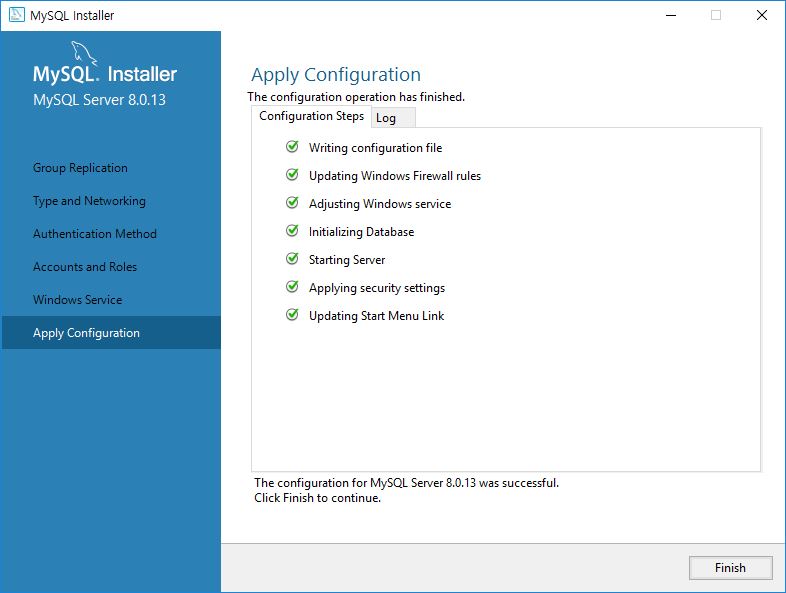
root 계정의 암호 설정 후 NEXT 클릭



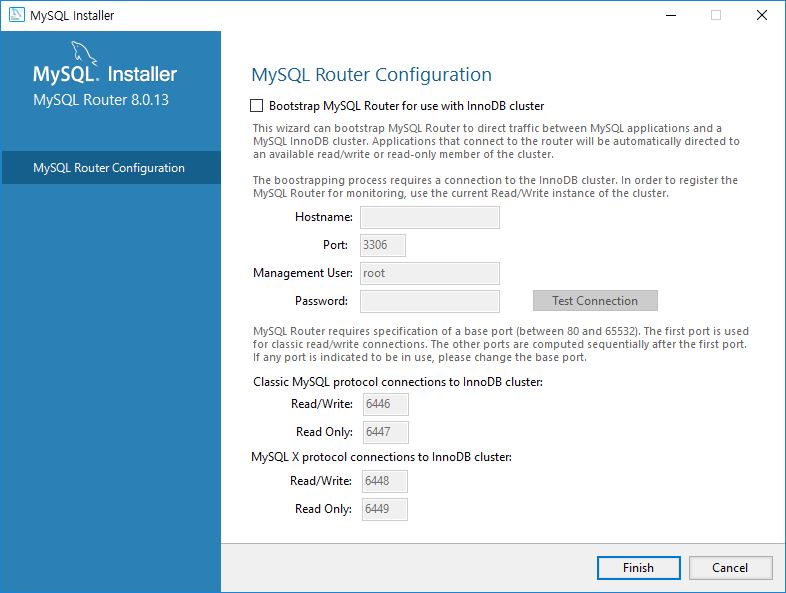
기본값으로 Next 클릭



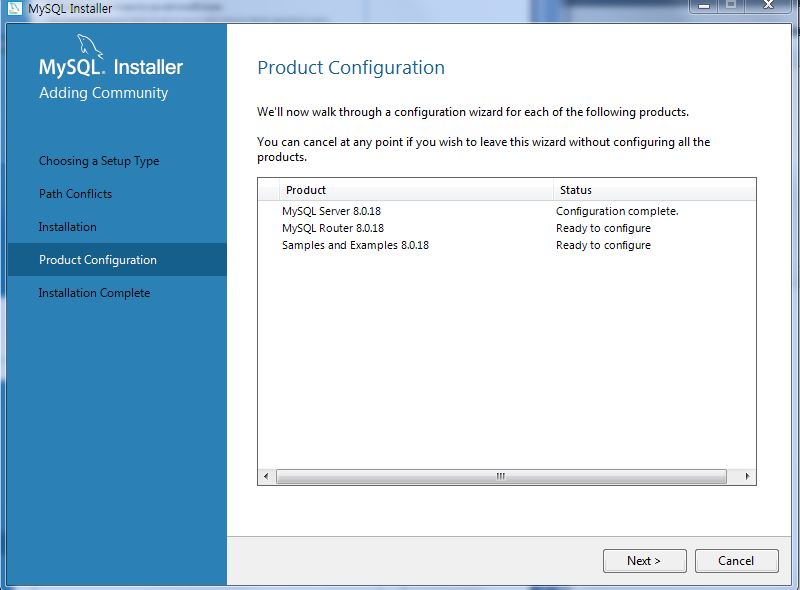
Execute 클릭

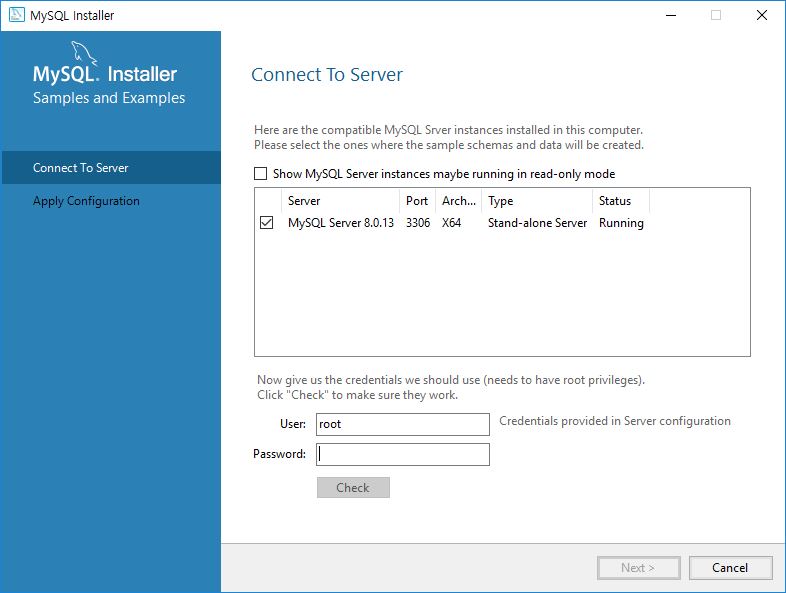


Finish 클릭

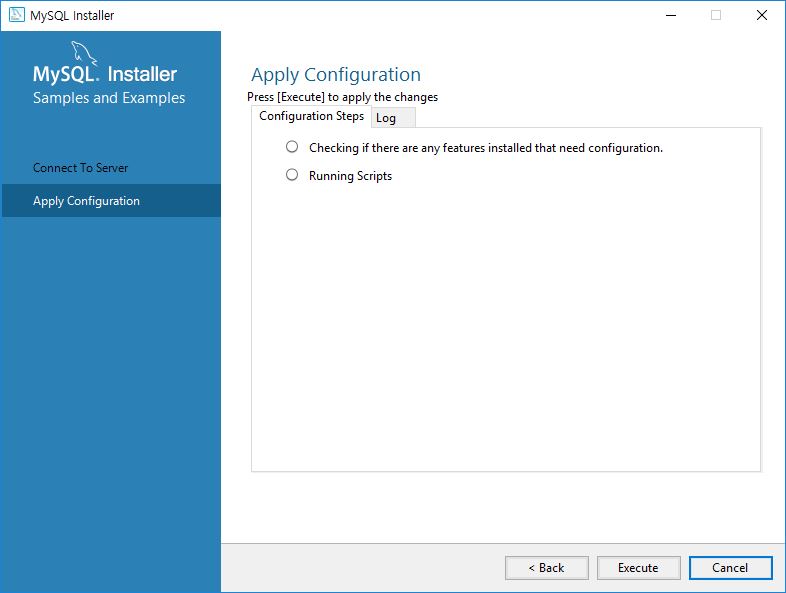


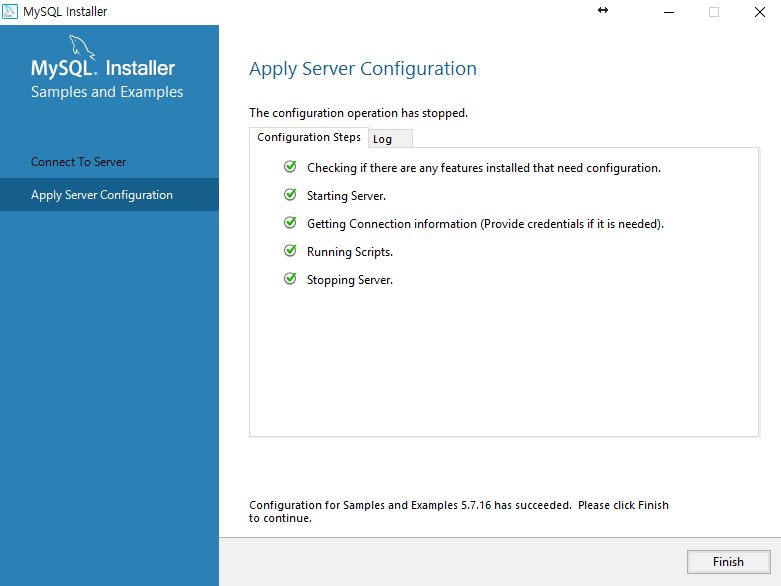
Check 후 Next클릭

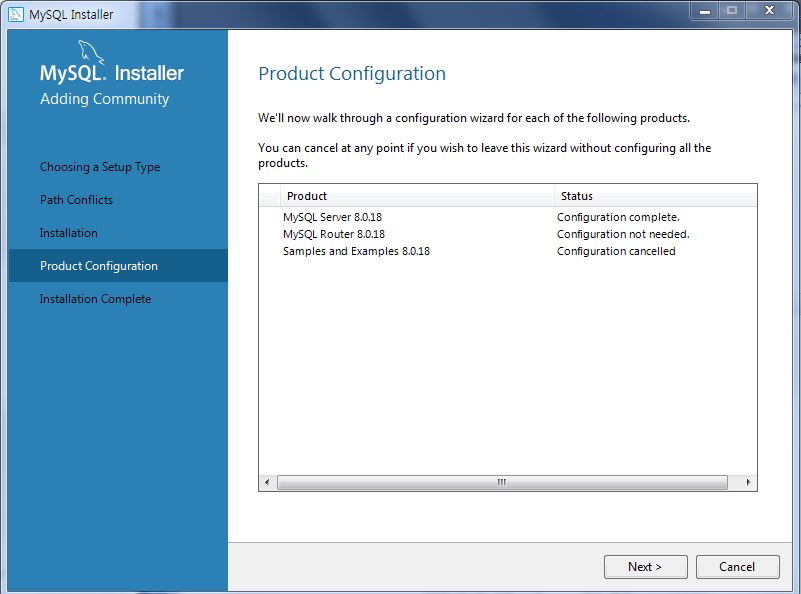


추가 설정 단계를 위해 Next 클릭

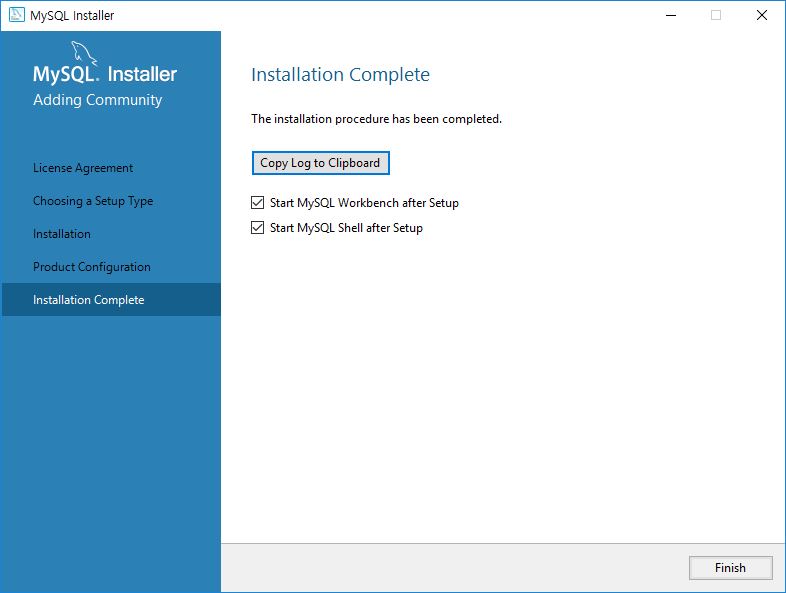
정상적인 서버 연결을 검사하기 위해 앞서 설정한 루트 계정으로 검사



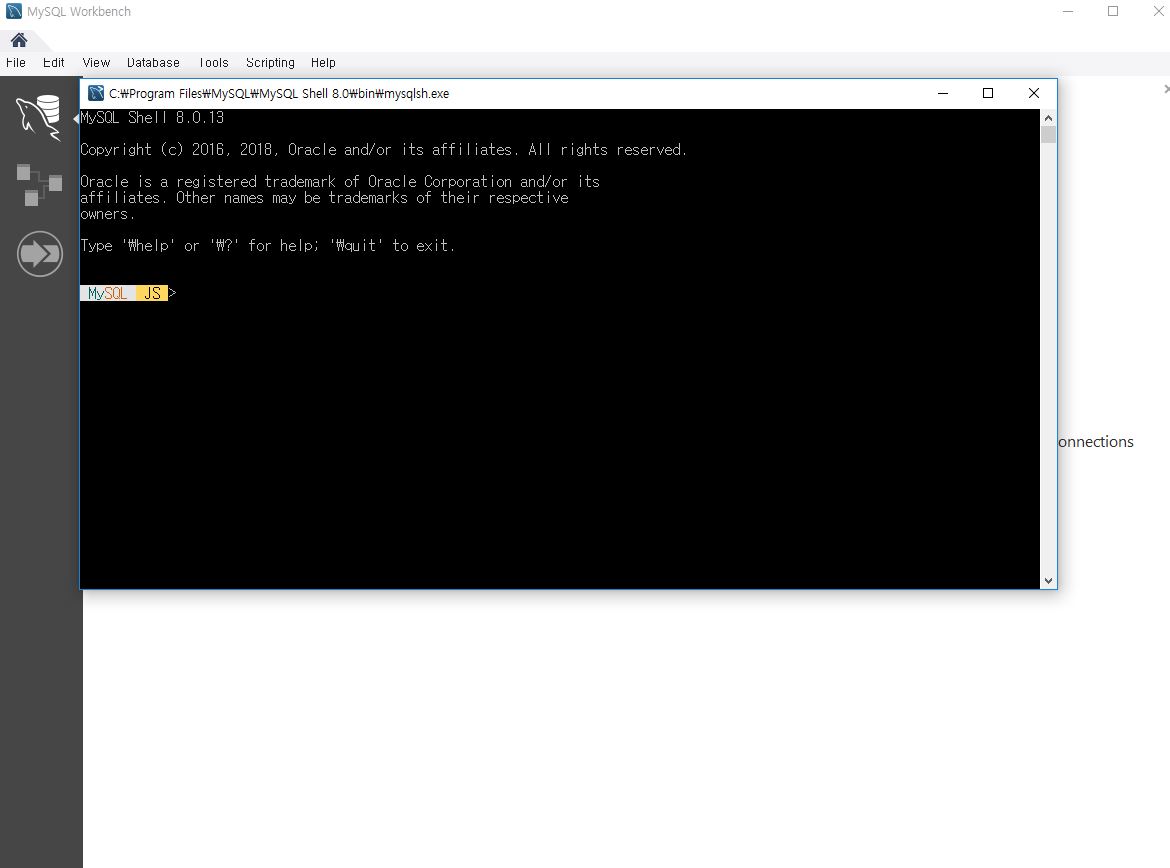
Execute 클릭

Finish 클릭

Status에 상관없이 Next클릭

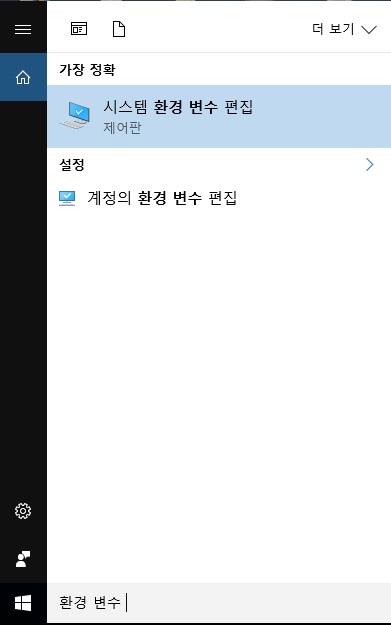
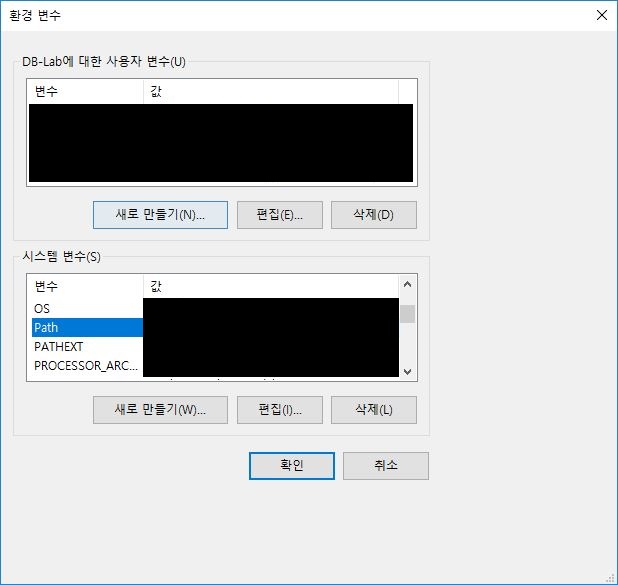


Finish 클릭



정상적으로 설치가 완료되면 쉘과 워크벤치가 실행된다.

**3 . 환경변수 설정과 실행 및 DB생성**

****

MySQL이 설치된 경로를 시스템 변수 Path에 추가한다. 즉, 편집을 눌러 뒷부분에‘C:\Program Files\MySQL\MySQL Server 8.0\bin’를 추가한다.

주의: 변수값을 잘못 변경하거나 기존 내용을 지울 경우 포맷을 해야 할지도 모른다. 사고를 방지하기 위해 변경 전 내용을 메모장 등에 백업하는 것을 추천한다.

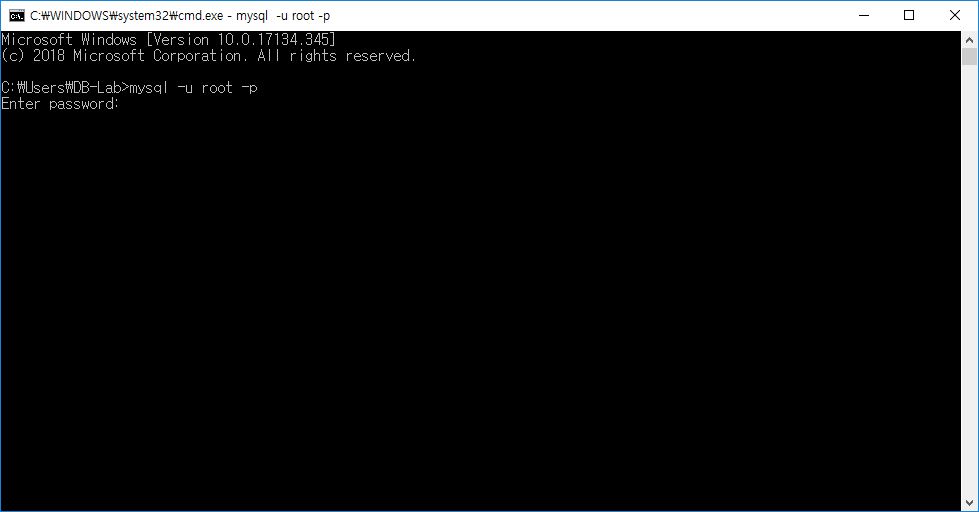
설치 및 환경설정이 끝났다면 MySQL이 제대로 실행되는지 확인한다.

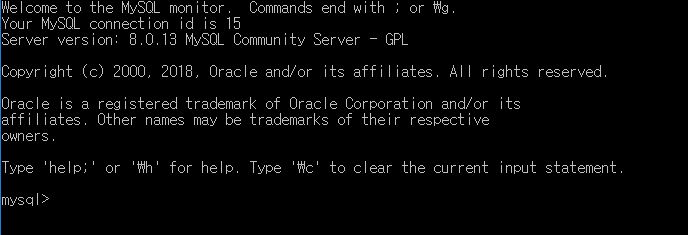
2-1. 도스창 실행

윈도우의 시작버튼 -> 실행 ->cmd명령창을 실행

2-2. MySQL 실행

윈도우 명령 프롬프트에서 mysqlsh–u root –p 명령어를 수행  
-u :유저명 입력



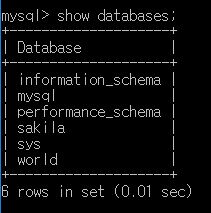
앞서 설정한 루트 계정 비밀번호로 접속  


정상적으로 접속된 화면

2-3. Database 확인

Show 명령어를 이용하여 MySQL에 기본적으로 생성 되어있는 데이터베이스를 확인한다

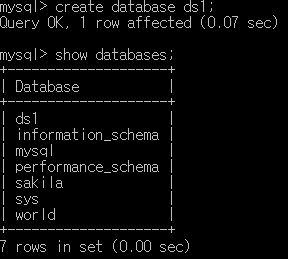
윈도우 명령 프롬프트에서 show databases; 명령어를 수행한다.



2-4. Database 생성

새로운 윈도 명령 프롬프트창에서 ‘create database ds1;’명령을 수행시킨다.

‘show databases;’로 생성된 ds1을 확인한다.

  
워크 벤치에서도 동일하게 작업 가능

**4. 데이터베이스 스키마 생성 및 자료 입력**

‘DB+본인 학번’을 데이터베이스 이름으로 갖는 데이터베이스를 구축한다.

예) 20060012의 학번을 갖는 학생이라면 Database Name은 DB20060012 가 되어야 한다. (참고로 생성 데이터베이스 이름은 제출용 백업파일명과 다를 수 있음)

반드시 각 테이블에 대한 제약조건을 준수하여 과제를 수행하여야 한다.

|  |  |  |
| --- | --- | --- |
| Customer | | |
| 컬럼 이름 | 데이터 타입 |  |
| customer\_id | varchar | Primary Key |
| customer\_name | varchar or char(크기 10 이상) |  |
| customer\_street | varchar or char(크기 10 이상) |  |
| customer\_city | varchar or char(크기 10 이상) |  |

* customer\_street: 간략한 주소
* customer\_city: 도시 이름 ex)김포, 서울, 성남, 대전, 부산 등

|  |  |  |
| --- | --- | --- |
| Loan | | |
| 컬럼 이름 | 데이터 타입 |  |
| loan\_number | varchar or char(크기 10 이상) | Primary Key |
| amount | int |  |

* amount: 1만원 단위. ex) 1천만원의 경우 1000

|  |  |  |
| --- | --- | --- |
| Account | | |
| 컬럼 이름 | 데이터 타입 |  |
| account\_number | varchar or char(크기 10 이상) | Primary Key |
| balance | int |  |

* balance: 1만원 단위. ex) 1천만원의 경우 1000

|  |  |  |
| --- | --- | --- |
| Branch | | |
| 컬럼 이름 | 데이터 타입 |  |
| branch\_name | varchar or char(크기 10 이상) | Primary Key |
| branch\_city | varchar or char(크기 10 이상) |  |
| assets | int |  |

* assets: 1만원 단위. ex) 1천만원의 경우 1000 / 칼럼 이름에 주의 assets임

|  |  |  |
| --- | --- | --- |
| Payment | | |
| 컬럼 이름 | 데이터 타입 |  |
| loan\_number | varchar or char(크기 10 이상) | Foreign Key: reference Loan |
| payment\_number | varchar or char(크기 10 이상) |  |
| payment\_amount | int |  |
| payment\_date | date |  |

* payment\_number: 대출 계좌에 대출금을 갚을 때 상환 순서를 기록함
* primary key: (loan\_number, payment\_number)
* loan\_number: Loan의 loan\_number와 타입 일치 시킬 것
* payment\_amount: 1만원 단위. ex) 1천만원의 경우 1000

|  |  |  |
| --- | --- | --- |
| Borrower | | |
| 컬럼 이름 | 데이터 타입 |  |
| customer\_id | varchar or char(크기 10 이상) | Foreign Key: reference Customer |
| loan\_number | varchar or char(크기 10 이상) | Foreign Key: reference Loan |

* customer\_id : Customer의 칼럼과 데이터 타입 일치 시킬 것
* loan\_number: Loan의 칼럼과 데이터 타입 일치 시킬 것

|  |  |  |
| --- | --- | --- |
| Depositor | | |
| 컬럼 이름 | 데이터 타입 |  |
| customer\_id | varchar or char(크기 10 이상) | Foreign Key: reference Customer |
| account\_number | varchar or char(크기 10 이상) | Foreign Key: reference Account |
| access\_date | date |  |

* customer\_id: Customer의 칼럼과 데이터 타입 일치 시킬 것
* account\_number: Account의 칼럼과 데이터 타입 일치 시킬 것

|  |  |  |
| --- | --- | --- |
| Loan\_Branch | | |
| 컬럼 이름 | 데이터 타입 |  |
| loan\_number | varchar or char(크기 10 이상) | Foreign Key: reference Loan |
| branch\_name | varchar or char(크기 10 이상) | Foreign Key: reference Branch |

* loan\_number: Loan의 칼럼과 데이터 타입 일치 시킬 것
* branch\_name: Branch의 칼럼과 데이터 타입 일치 시킬 것

\*\* 외래키의 경우 반드시 데이터 타입을 일치시켜 주어야 한다.

**<테이블 관련 제약 조건>**

- customer와 loan은 borrower의 관계를 가지며 이 관계는 m:n 이다.

- customer와 account는 depositor의 관계를 가지며 customer 와 account의 관계는 m:n depositor는 access\_date(날짜형) 속성을 추가적으로 가지고 있다.

- branch와 loan은 loan\_branch의 관계를 가지며 branch와 loan의 관계는 branch 입장에서 볼 때 1:n이다.

- loan은 여러 개의 payment를 가지며, 이때 대출 상환 순서를 payment number로 기록한다.

**<속성 관련 제약 조건>**

- 모든 테이블에는 반드시 기본 키(primary key)가 존재한다.

- 금전과 관련된 칼럼의 자료형은 정수 자료형을 사용한다.

- 테이블 간의 자연 조인(natural join)을 위해 필요한 칼럼들의 자료형은 같아야 한다.

- 한 계좌의 예금액은 0원 ~ 15억 원이다 (0<=balance<=150000). 새로운 튜플을 넣는 경우(insert)나 기존의 튜플을 수정하는 경우(update) 등 예금액의 범위 밖으로 나가게 되는 명령어는 실패되어야 한다.

**<튜플 관련 입력 조건>**

- 생성된 각 테이블에 각각 7개 이상의 적절한 튜플을 삽입한다. 지점 이름(branch\_name), 고객 이름 (customer\_name), 고객 주소(customer\_street), 고객 도시(customer\_city) 칼럼에 해당하는 값은 우리나라 실정에 맞는 실제 자료를 삽입한다. 이 중 고객 주소 칼럼에는 도시 이하의 주소(구, 동, 번지)를 입력하면 된다.

- 화폐의 단위는 만원이며, 자료 입력 시 만원 이하의 금액 단위는 생략한다. 즉, 한 고객이 1,000만원을 예금한 정보는 1000으로 입력된다.

- 이씨 성을 가진 고객은 세 명만 존재한다.

- 지점이 위치하고 있는 도시가 아닌 도시에 살고 있는 고객이 두 사람 이상 존재한다.

- 지점 도시(branch\_city)는 반드시 다섯 개 이상의 유일한 값이 존재하여야 한다. 또한, ‘대전’이라는 도시에는 반드시 지점이 존재하여야 한다.

- 지점 도시가 ‘서울’인 지점은 두 개 이상 존재하여야 한다.

- 지점 이름(branch\_name)은 반드시 ‘지점’ 접미사로 끝난다. 예를 들어 ‘명동지점’, ‘숭실대지점’과 같이 입력되어야 한다.

- 자산(assets)의 값은 만원 단위이며 최대값이 50억원(500000)을 넘지 않도록 한다.

- ‘서울’이 아닌 곳에 있는 지점 중에 고객에게 대출해 주지 않은 지점이 하나만 있도록 한다.

- ‘서울’에 있는 지점은 대출 금액의 합이 자산을 반드시 넘지 않도록 한다.

- ‘서울’이 아닌 곳에 있는 지점 중에서 대출하여 준 금액의 합이 자산을 넘는 지점이 두개가 있도록 한다.

- 최소한 하나의 지점에서 여러 번 대출이 이루어진 결과가 나타나야 한다.

- ‘서울’에 있는 모든 지점에서 대출한 고객이 생기도록 자료를 입력한다.

- 대출하지 않은 고객은 셋 이상 존재한다.

- 상환(payment)이 한번도 이루어지지 않은 대출이 나타나도록 자료를 입력한다.

- 최대 금액을 예금한 예금계좌가 반드시 한 개 존재하도록 자료를 입력한다.

- 대출하지 않은 고객 중에서 예금계좌를 두 개 가지고 있는 고객이 생기도록 자료를 입력한다.

- 대출한 고객 중에서 예금계좌를 갖지 않는 고객이 생기도록 자료를 입력한다.