

데이터베이스(world) SQL 실습

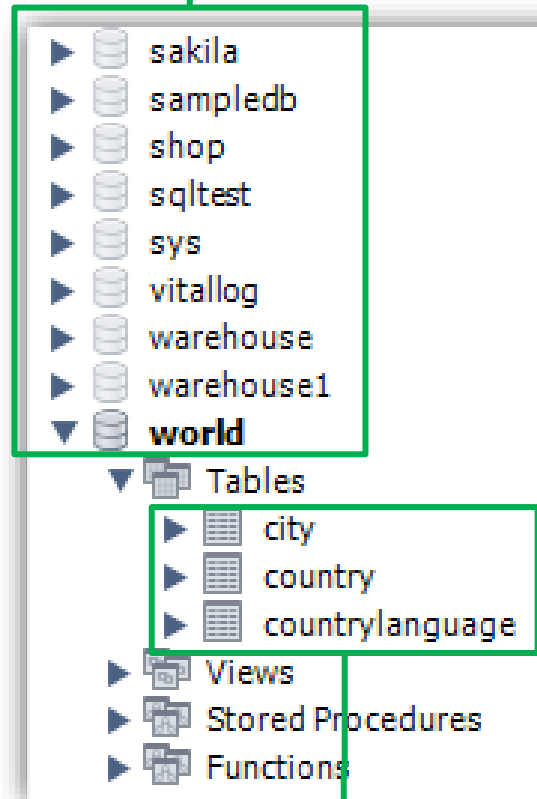
지능물류빅데이터연구소 이상현

데이터베이스란?

- 데이터베이스 (Database)
 - 디지털화되고 구조화된 데이터를 테이블이라고 하고
 - 이러한 테이블들의 모음을 데이터베이스라 한다.
- 데이터베이스관리시스템 (DBMS: DataBase Management System)
 - 데이터베이스를 관리하고 사용자와 상호 작용하는 소프트웨어
 - MySQL, Oracle, MSSQL, PostgreSQL, MongoDB, H2, Sqlite 등
- SQL (Structured Query Language)
 - 데이터베이스와 사용자 간의 상호 작용에 사용되는 표준 언어

데이터베이스란?

데이터베이스 목록



테이블 목록

MySQL Workbench interface showing the Schemas list and the SQL execution results.

SQL 실행 창

선택한 테이블 정보

SQL 실행 결과 창

#	Time	Action	Message	Duration / Fetch
39	09:46:31	SELECT Name, Population, SurfaceArea, Population / SurfaceArea AS PopulationDensity FROM country WHERE (Population / SurfaceArea > 100)	91 row(s) returned	0.000 sec / 0.000 sec
40	09:49:04	SELECT Name FROM country WHE	1 row(s) returned	0.000 sec / 0.000 sec
41	09:50:26	SELECT Name FROM country WHE	1 row(s) returned	0.000 sec / 0.000 sec
42	09:50:30	SELECT Name FROM country WHERE HeadOfState is null LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

SQL문 종류

- DDL (Data Definition Language)
 - 데이터베이스 구조를 정의
 - **CREATE**(테이블 생성), **ALTER**(테이블 수정), **DROP**(테이블 삭제)
- DML (Data Manipulation Language)
 - 테이블에 저장된 데이터를 조작
 - **SELECT**(조회), **INSERT**(입력), **UPDATE**(수정), **DELETE**(삭제)
- DCL (Data Control Language)
 - 데이터베이스에 대한 접근 권한 관리
 - **GRANT**(접근권한 부여), **REVOKE**(권한 제거)

SELECT문 - 기본 문법

- 기본 문법

SELECT [* DISTINCT]	속성이름(들)	➔ 무엇을
FROM	테이블이름(들)	➔ 어디에서
[WHERE	검색조건(들)]	➔ 어떤
[GROUP BY	속성이름(들)]	
[HAVING	검색조건(들)]	
[ORDER BY	속성이름 [ASC DESC]];	

- SQL문은 항상 세미콜론(;)으로 문장을 종료

SELECT문 - '*'

- CITY 테이블에 저장된 데이터들을 출력하세요.

USE world; ➔ world 데이터베이스를 사용하겠다고 선언

SELECT [무엇을]

FROM [어디에서];

SELECT * ➔ * : ALL

FROM CITY;

- 실습

- COUNTRY 테이블에 저장된 데이터들을 출력하세요.
- COUNTRYLANGUAGE 테이블에 저장된 데이터들을 출력하세요.

SELECT문 - Columns

- CITY 테이블에서 Column(필드, 속성)명 Name, Population을 출력하세요.

```
SELECT Name, Population
```

```
FROM CITY;
```

- 실습

- COUNTRY 테이블에서 Column명이 Code, Name, Continent, LocalName, GNP를 출력하세요.
- COUNTRYLANGUAGE 테이블에서 Column명이 CountryCode, Language, Percentage를 출력하세요.

SELECT문 - Where

- COUNTRY 테이블에서 code가 'KOR'인 데이터를 찾아서 출력하세요.

```
SELECT *
```

```
FROM COUNTRY
```

```
WHERE code='KOR';
```

- 검색 조건
 - =(equal), <>(not equal), <(lt), >(gt), <=(le), >=(ge)
 - like : 문자열 비교
 - % : 0개 이상의 임의의 문자와 일치 → code like 'K%'
 - _ : 정확히 하나의 문자가 일치 → code like 'K_'

SELECT문 - Where + AND(OR)

- COUNTRYLANGUAGE에서 code가 'KOR', IsOfficial이 'T'인 데이터를 찾아서 출력하세요.

```
SELECT *
```

```
FROM COUNTRYLANGUAGE
```

```
WHERE CountryCode='KOR'
```

```
AND isofficial='T';
```

- 검색 조건
 - AND : 모든 조건이 참인 데이터만 검색
 - OR : 조건들 중에서 하나라도 참이면 검색

SELECT 실습

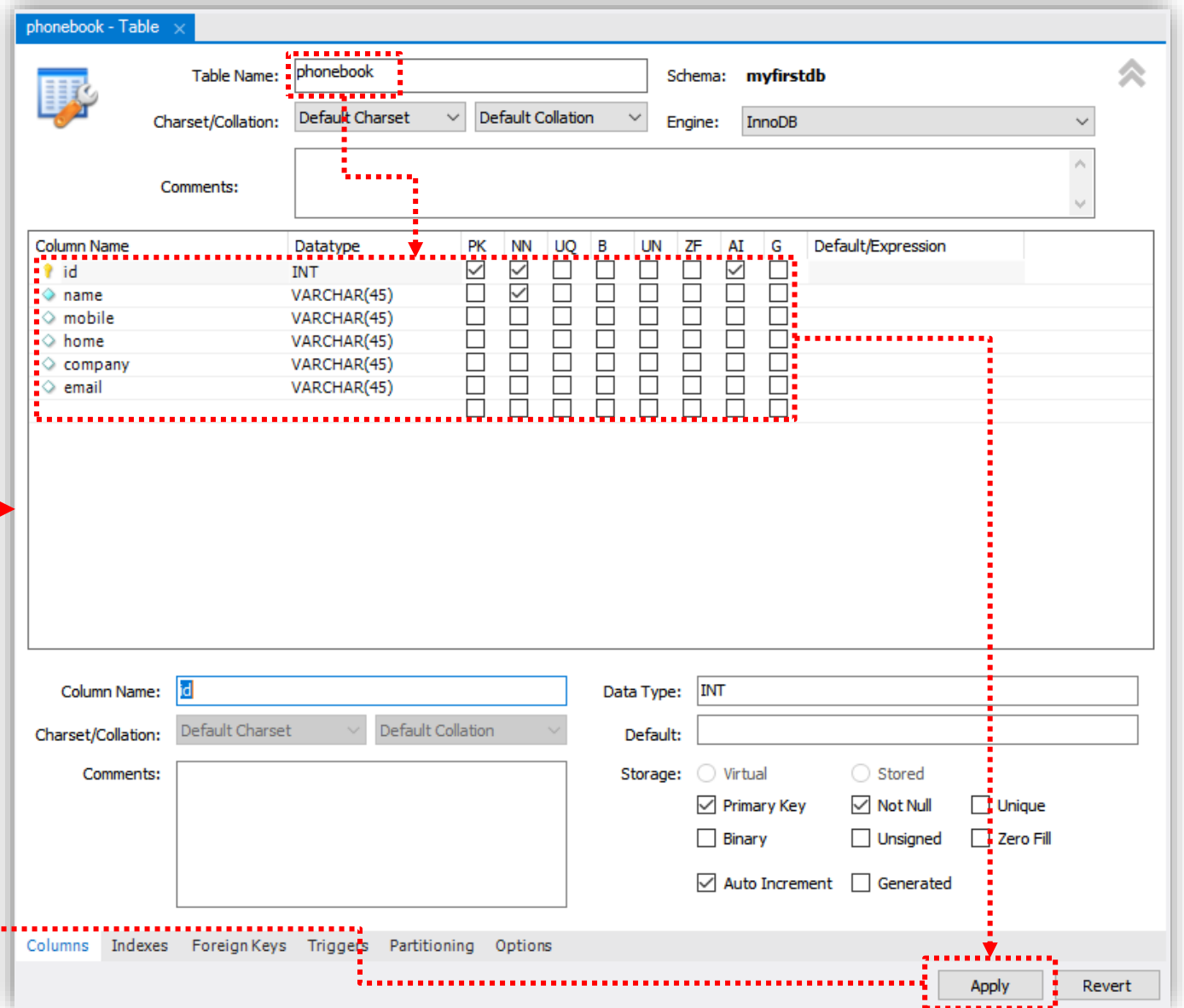
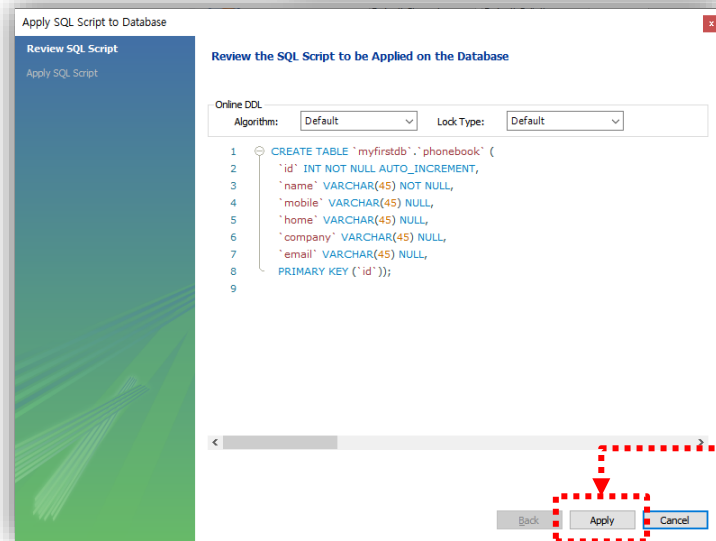
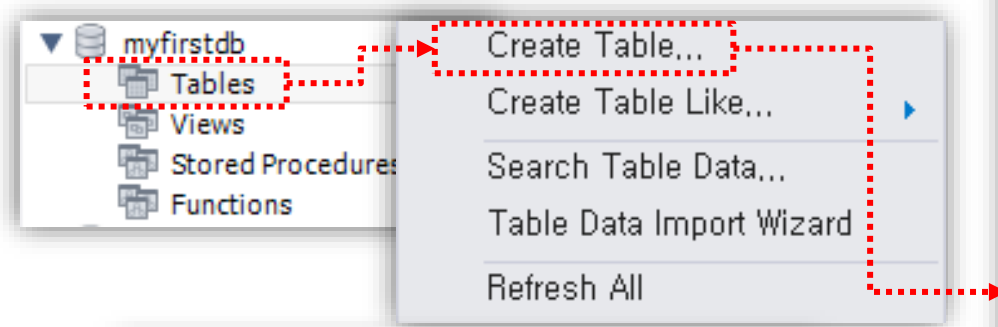
1. 인구가 500,000명 이상인 모든 도시를 검색하세요.
2. 중국 도시의 이름과 인구를 검색하세요.
3. 아시아 대륙('Asia')에 위치한 국가를 검색하세요. (Continent)
4. 국가 넓이가 10,000² km 미만인 국가의 이름과 면적을 검색하세요.
5. 기대 수명이 60 이하인 국가의 이름과 기대 수명을 검색하세요. (LifeExpectancy)
6. 미국 주이름이 'California'인 도시를 검색하세요. (District)
7. 인구가 100,000명에서 1,000,000명 사이인 인도의 도시를 찾아보세요.
8. 정부 형태가 공화국('Republic')인 국가를 찾아서 대륙명과 국가 이름을 검색하세요. (GovernmentForm);
9. 'Kyonggi' 지역에 있는 모든 도시를 검색하세요. (District)
10. 국가 공식 언어가 'Spanish'인 국가를 검색하세요. (CountryCode를 출력하되 조인이 가능하면 국가명을 출력)
11. 인구가 5천만 명이 넘는 모든 국가를 검색해서 국가명과 인구수를 출력하세요.
12. 국가 테이블에서 LocalName에 'Republic'이 포함된 국가 이름을 검색하세요.
13. CountryLanguage에서 사용자가 90% 이상인 언어의 국가 코드와 비율을 검색하세요.
14. City 테이블에서 이름이 문자 's'로 시작하는 도시를 검색하세요.
15. 인구 밀도가 평방 킬로미터당 100명을 초과하는 모든 국가를 검색하세요. (Population/SurfaceArea)
16. 국가 코드가 3자리이고 'c'로 시작하는 국가를 검색하세요.
17. 인구가 10,000명 미만인 도시 이름을 검색하세요.
18. GNP가 1백만 이상인 국가의 국가 이름을 검색하세요.
19. 도시 이름에 'land'가 포함된 모든 도시를 검색하세요.
20. 국가 테이블에서 HeadOfState가 지정되지 않은 국가 이름을 검색하세요.

데이터베이스/테이블 생성

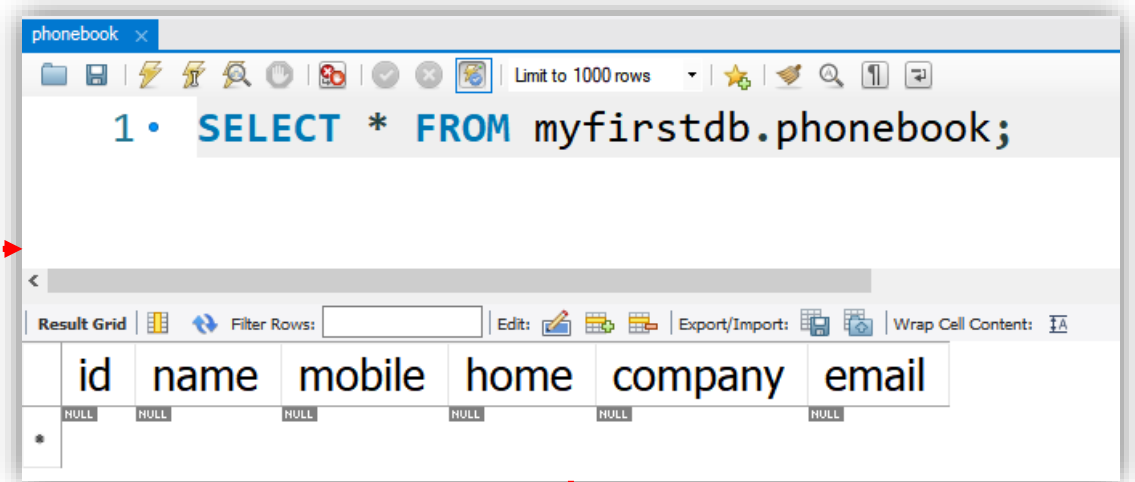
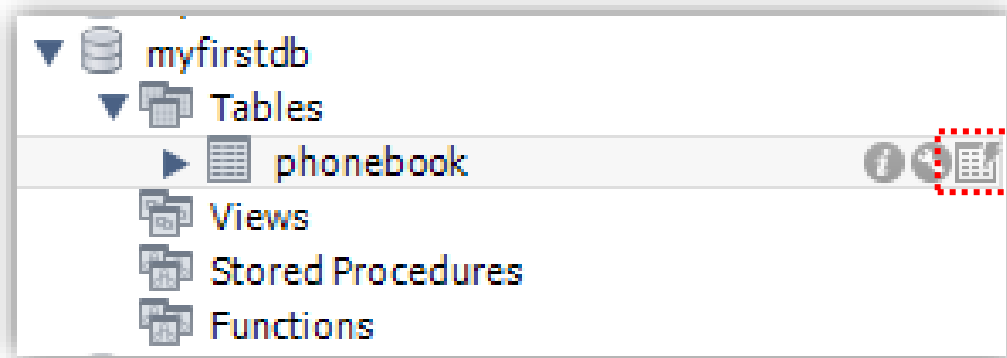
- 데이터베이스 생성

- CREATE DATABASE MYFIRSTDB;

- 테이블 생성



데이터 입력 (insert)



	id	name	mobile	home	company	email
...	NULL	홍길동	010-1111-2222	051-123-4567	051-123-1234	hongkd@korea.com
*	NULL	NULL	NULL	NULL	NULL	NULL

apply

```
INSERT INTO `myfirstdb`.`phonebook` (`name`, `mobile`, `home`, `company`, `email`)
VALUES ('홍길동', '010-1111-2222', '051-123-4567', '051-123-1234', 'hongkd@korea.com');
```

데이터 입력 (insert)

- INSERT 문법

```
INSERT INTO [어느 테이블에](어느 필드에)  
VALUES (어떤 값을);
```

```
INSERT INTO phonebook (name, mobile, email)  
VALUES ('홍이동', '010-2222-3333', '2dhong@naver.com');
```

➔ 전화번호 테이블에 INSERT 구문을 이용해서 9건 이상 입력하세요.

데이터 수정 (update)

	id	name	mobile	home	company	email
...	1	홍길동	010-1111-2222	051-123-4567	051-123-1234	hongkd@korea.com
*						

	id	name	mobile	home	company	email
...	1	홍길동	010-1111-9999	051-123-4567	051-123-12...	hongkd@korea.com
*						

apply

```
UPDATE `myfirstdb`.`phonebook` SET `mobile` = '010-1111-9999' WHERE (`id` = '1');
```

데이터 수정 (update)

- update 문법

UPDATE [어느 테이블에]

SET [필드1=값1, 필드2=값2]

WHERE [어떤 조건의 데이터를 대상으로];

➔ primary key가 아니면 safe mode인 경우 에러
(edit-preference-Sql Editor-safe updates)

UPDATE phonebook

SET home='051-999-9999', company='051-999-9998'

WHERE id=5;

➔ 전화번호 테이블에 UPDATE 구문을 이용해서 데이터 수정 실습

데이터 삭제 (delete)

	id	name	mobile	home	company	email
	1	홍길동	010-1111-9999	051-123-4567	051-123-1234	hongkd@korea.com

삭제하고자 하는 데이터에서 마우스 우클릭

apply

```
DELETE FROM `myfirstdb`.`phonebook` WHERE (`id` = '1');
```

- Open Value in Editor
- Set Field to NULL
- Mark Field Value as Function/Literal
- Delete Row(s)
- Load Value From File...
- Save Value To File...
- Copy Row
- Copy Row (with names)
- Copy Row (unquoted)
- Copy Row (with names, unquoted)
- Copy Row (with names, tab separated)
- Copy Row (tab separated)
- Copy Field
- Copy Field (unquoted)
- Paste Row
- Capitalize Text
- lowercase Text
- UPPERCASE Text

데이터 삭제 (delete)

- update 문법

```
DELETE FROM [어느 테이블에]  
WHERE [어떤 조건의 데이터를 대상으로];
```

```
DELETE FROM phonebook  
WHERE id=5;
```

➔ 전화번호 테이블에 DELETE 구문을 이용해서 데이터 삭제 실습