*####2번째*

**create** **table** **if** **not** **exists** address\_book (

**no** **int** **unsigned** **not** **null** **auto\_increment**,

**name** **varchar**(10) **not** **null**,

tel **varchar**(14),

nickname **varchar**(20) **default** ‘별명’,

**primary** **key**(**no**)

) **auto\_increment**=10001 **default** **charseT**=**utf8**;

**USE** world;

**SHOW** **TABLES**;

**USE** ezen; *# 내가 온곳을 알려줌*

**DESC** address\_book; *# 내가 만든 테이블의 구조를 보여줌*

**CREATE** **TABLE** tmp (

id **INT** **PRIMARY** **KEY**,

**NAME** **VARCHAR**(10)

);

**SHOW** **TABLES**; 내가 만든거 보여줘

**DROP** **TABLE** tmp;

**SHOW** **TABLES**;

*#이거 지우고 다시 보여줘*

**RENAME** **TABLE** address\_book **TO** addr\_book;

**SHOW** **TABLES**;

*#내가 날른 쿼리들이 전부 아래 나온다*

*#왼쪽은*

**DESC** addr\_book;

**ALTER** **TABLE** addr\_book **ADD** gender **CHAR**(2) **NOT** **NULL**;

**DESC** addr\_book;

**ALTER** **TABLE** addr\_book **ADD** email **VARCHAR**(20) **NOT** **NULL** **AFTER** tel;

**DESC** addr\_book;

###추가

* 닉네임 지우기

**ALTER** **TABLE** addr\_book **DROP** nickname;

**DESC** addr\_book;

#drop은 지우기

* 컬럼명 변경 자료형 지우기

**ALTER** **TABLE** addr\_book **CHANGE** **NO** abid **INT**(8);

**DESC** addr\_book;

#프라이머리키값은 널값이 되면 안되서 살아잇ㅇ<ㅁ?

이름만 바꾸고 싶다고 해서 바꿀수 있느건아님

**이름해당되는 데이터 타입도 변경해야한다**

* 순서 변경하기

**ALTER** **TABLE** addr\_book **MODIFY** gender **CHAR**(2)**AFTER** **NAME**;

**DESC** addr\_book;

* 데이터 조작

내가 제대로 사용할라면 정식사용하려면 쿼리에서 사용해야한다.

**SELECT** \* **FROM** city;

**SELECT** \* **FROM** city **WHERE** COUNTrycode='kor';

#내가원하는 데이터 선택

Where 다음에 뭐뭐는 뭐뭐다

DB에서는 = 같다라는 표시

이거는 숫자에서 사용하는거

이거만 다름

**SELECT** \* **FROM** city **WHERE** COUNTrycode **LIKE** 'kor';

# 이것이 좋은 표기

#이것이 문자열에서 사용하는 표기 like

**SELECT** \* **FROM** city **WHERE** district **LIKE** 'kyonggi';

# 경기도 도시 찾기

**SELECT** \* **FROM** city **WHERE** district **LIKE** 'kyonggi' **AND** population > 500000;

#500,000만보다 큰 보시 찾기

#R에서 dplyr에서는 filter로 사용하는 것들

#이렇쿵 저러쿵 할수 잇겠구나~

**select** district **from** city **where** countrycode='kor';

도시에 있는 애들이 다 나옴

**select** **distinct** district **from** city **where** countrycode='kor';

#유니크 하게 distinct에 있는 애들만 발라내준다.

**DESC** ezen.addr\_book;

#다른 데이터를 사용할때는 다른 데이터베이스?? 를꺼내서 사용하면된다.

**\*/주석처리**

**##두개가 같은 값으로 나온다 %을 사용하자**

**SELECT** \* **FROM** city **WHERE** district **LIKE** 'Taejon' **OR**

district **LIKE** 'chungchongbuk' **OR** district **LIKE** 'chungchongnam';

#정말 다 대소문자 구분안함

**SELECT** \* **FROM** city **WHERE** district **LIKE** 'Taejon' **OR**

district **LIKE** 'chungchon%';

**##두개가 같은 값으로 나온다 %을 사용하자**

**SELECT** \* **FROM** city **WHERE** countrycode **LIKE** 'kor' **and** population > 1000000;

#백만이 넘는 도시

* 문제1, Id가 짝수인 놈만 나오게!

**어디서든 나머지를 구하는 연산 modulo 가 - %% 이다** 하지만 R은 % 하나이다

**SELECT** \* **FROM** city **WHERE** countrycode **LIKE** 'kor' **AND**

population > 1000000 **AND** id%2 = 1;

id값이 하나인아이를 구하려면

조건을 잘하는게 SQL 잘하는거시다!

**select** \* **from** city **WHERE** countrycode='kor' **and**

population **between** 1000000 **and** 2000000;

* 대 로 시작하는 것

**select** \* **from** city **WHERE** countrycode='kor' **and**

**NAME** **LIKE** 'tae%';

* R에서 사용하는 desc()

**select** \* **from** city **WHERE** district **LIKE** 'kyonggi'

**ORDER** **BY** population **desc**;

* 오름차순 내림차순 정리하는 것

**select** \* **from** city **WHERE** countrycode='kor'

**order** **by** district, population **DESC**;

R에서 여러함수를 사용함

SQL에서도 여러 함수를 사용ㅎ라수있따

**select** **MAX**(population) **from** city **where** countrycode='kor';

* 가장 많이 사는 도시의 숫자

**select** **Min**(population) **from** city **where** countrycode='kor';

* 가장 적게 사는 도시의 숫자
* 이

**select** **group\_concat**(**distinct** district) **from** city

**where** countrycode='kor';

**유니크하게 = 한번만 나오게**

* **Group\_By 테이블을 연산할 때 아주 유용하게 나오는 툴**

**select** district, **count**(\*) **from** city **where** countrycode='kor';

-한국에서 도시의 수를 알려주고

이건 group\_by할 때 확인할 때 하는 조회문 인데 결과값이 이상함

제대로 나오게 하려면group\_by사용해야한다.

**select** district, **count**(\*) **from** city **where** countrycode='kor'**group** **by** district;

광역시도별 도시수가 나온다.

그룹핑을 해서 처리를 하는건 Group\_by가 매우 편리하다

* **6개인 아이들을 찾아줘!**

**select** district, **count**(\*) **from** city **where** countrycode='kor'

**group** **by** district **having** **count**(\*)=6;

* **내림차순으로 보여줘(order by)**

**select** district, **count**(\*) **from** city **where** countrycode='kor'

**group** **by** district **having** **COUNT**(\*)>=6 **order** **BY** **COUNT**(\*) **desc**;

**# 이거의 뜻이 무엇일까!?도시가 50개가 넘는 국가를 표시해줘**

**select** countrycode, **count**(\*) **from** city

**group** **by** countrycode **having** **count**(\*) >=50;

# 도시를 내림차순으로 보야줘!

**select** countrycode, **count**(\*) **from** city **group** **by** countrycode

**having** **count**(\*) >=50 **order** **by** **count**(\*) **desc**;

데이터베이스에서 일반적으로 사용되는 쿼리는

테이블하나에서 사용되는 것 보다 여러 개의 테이블에서 가지고온다.

**select** city.**Name**, city.Population, country.**Name** **from** city

**inner** **join** country **on** city.CountryCode = country.**Code**

**where** city.Population > 7000000

**ORDER** **BY** city.Population **DESC**;

* 나라중 인구수가 7000000명이 넘는 곳을 보여줘

**select** city.**Name**, city.Population, country.**Name** **from** city

**inner** **join** country **on** city.CountryCode = country.**Code**

**where** city.Population > 7000000

**ORDER** **BY** city.Population **DESC** **LIMIT** 5;

* 나라중 인구수가 700000명이 넘는 곳을 5섯 곳(limit사용)만 보여줘
* R과 database 같은 함수들

조건(filter), grouping(group\_by) column선택(select), 순서(arrange),개수(head),함수(summarise)

한화이글스

Backno - int unsigned PK

Name - varchar(10), not null

Position - varchar(4)

Highschool - varchar(10)

Hand - varchar(10)

Insert into engles valrus(22, ‘이태양’, ‘투수’, ‘순천효천고’,’우투수’

Insert into engles valrus

Insert into engles(backno, name, position) values

(22, “이태양”,”투수”,”순천효천고”,”우투수”)

(36,”장민재”,”투수”,”대전고”,”우투수”)

(36,”박효신”,”타자”,”서울고”,”좌타자”),

(54,”우지호”,”타자”,”신촌고”,”우타자”),

(21,”남궁성”,”타자”,”도마고”,”우타자”),

(15,”엄대윤”,”타자”,”제일고”,”좌타자”),

(56,”김선우”,”타자”,”서울고”,”좌타자”),

(77,”김우형”,”투수”,”대신고”,”우투수”),

(56,”박세영”,”투수”,”갈마중”,”좌투수”),

(52,”최태규”,”투수”,”여수고”,”우투수”),

Selert \* from engles;

**CREATE** **TABLE** engles(

“Backno” **int** **UNSIGNED** **NOT** **NULL**, **PRIMARY** **KEY**,

“Name” **varchar**(10) **NOT** **NULL**,

“POSITION” **varchar**(4) **NOT** **NULL**,

“Highschool” **varchar**(10) **NOT** **NULL**,

“Hand” **VARCHAR**(10) **NOT** **NULL**,

)

**AS** **SELECT** \* **FROM** PLAYER;

###내가 만든 소스 테이블

**CREATE** **TABLE** eenglesngles

(

Backno **int** **PRIMARY** **KEY** **AUTO\_INCREMENT**,

**NAME** **VARCHAR**(10) **NOT** **NULL** **DEFAULT** “Null”,

**POSITION** **VARCHAR**(10) **NOT** **NULL**,

Highschool **VARCHAR**(10) **NOT** **NULL**,

Hand **VARCHAR**(10) **NOT** **NULL**

) **ENGINE**=**INNODB**;

**DESCRIBE** engles;