

sql 독학 강의 # select를 잘 이용하는 방법(1), 2편 -sTricky

sTricky 2020. 3. 19. 13:46

sql 독학 강의 # select를 잘 이용하는 방법(1), 2편



컨텐츠 index

1. SQL select 문장을 사용하기 위한 용어 정리
2. DESC 사용하기

3. 원하는 데이터만 select 하기
4. 표현식 (Expression) 사용하기
5. 컬럼명 대신 별칭(alias)을 사용하기
6. DISTINCT로 중복된 값을 제외하고 출력하기
7. 연결 연산자 함수로 칼럼 값을 붙여서 출력하기

안녕하세요.

오늘부터 본격적으로 SQL 독학 강의를 시작합니다.

이번 시간에 다루어볼 내용은 **select**입니다.

SQL에서 가장 기본이 되는 내용이 바로 이 **select**인데요, 아마 SQL이 뭔지 잘 모르시는 분들도 **select**라는 말은 들어보거나 직접 보신 분들이 많이 계실 겁니다.

축구에서는 볼 트래핑, 패스 이런 것들이 기본이 되면서도, 가장 오랜 기간 연습을 해야 하는 것처럼 SQL을 다루는 데 있어서도 이 **select**가 기본이 되며, 앞으로도 가장 많이 다루게 되실 겁니다.

다른 명령어들이 많이 있습니다. **create, insert, update, alter, delete, rename, commit** 등등... 수많은 데이터베이스 SQL 명령어들이 있지만 이 모든 건 **select**를 잘하기 위해서 있는 겁니다. 더 올라가서 생각해보면 데이터베이스 모델링을 하고 그 위로 더 올라가면 모델링을 하기 위해서 업무분석을 하고, 요건 정의를 하는데, 이게 결국은 모두 **select**를 잘하기 위해서 라고 생각합니다.

그만큼 중요한 부분이니 잘 따라 와주시길 바랍니다.

우선, 강의를 시작함에 앞서 모두들 개인 연습용 mysql을 설치하여 주시기 바랍니다.

mysql을 설치하는 방법은 구글이나 네이버에 **윈도 mysql 설치, 리눅스 mysql 설치** 등의 검색을 해보시면 너무나도 많이 잘~~ 나와 있는 블로그들이 많으니 해당 블로그나 웹사이트를 참고하셔서 설치를 완료하시고, 지금부터 시작하는 SQL 기본 강의를 봐주시기 바랍니다.

내 컴퓨터 윈도우 OS에 설치하셔서 사용하셔도 되고, 리눅스 서버의 경우, AWS를 이용하시거나 네이버 클라우드 플랫폼 등을 이용하셔서 그 안에 나만의 데이터베이스를 구축하실 수 있습니다.

그럼 mysql의 설치를 완료하셨거나, 이미 가지고 계신 분들은 아래 내용을 따라와 주시면 됩니다.

시작하겠습니다.

#앞으로 SQL강의를 진행하는 동안, 이 글을 보시는 대상을 SQL을 거의 처음 접하시는 분들로 설정하고 진행하겠습니다. 참고 부탁드립니다.

#지난 독학 강의 보기#

2020/03/17 - [Database/sql 강의] - sql 공부 강의# 개념 파악 및 공부법 안내 1편 -sTricky.

	sql 공부 강의# 개념 파악 및 공부법 안내 1편 -sTricky stricky.tistory.com
--	--

1. SQL select 문장을 사용하기 위한 용어 정리

select 문장을 배우기 위해서는 알고 있어야 할 몇 가지 용어들이 있습니다.

우리가 흔히 알고 있는 표라는 것이 있습니다. 이것은 앞으로 우리가 배울 SQL이 조회하게 되는 데이터베이스의 테이블과 매우 흡사한 구조를 가지고 있습니다.

우리의 일상생활에서 표의 구조를 이야기하는 용어와 데이터베이스 테이블의 구조를 이야기할 때 쓰는 용어는 다릅니다. 아래 그림을 보면 이해가 되실 겁니다.

NAME	DEPT_CD	PHONE	ADDRESS

생활용어

NAME	DEPT_CD	PHONE	ADDRESS

데이터베이스 용어

일상생활에서 표는 행과 열로 표현하지만, 데이터베이스에서는 row와 column으로 표현하게 됩니다.

이렇게 표와 유사한 형태를 지닌 데이터베이스의 테이블에서 데이터를 가지고 오는 방법은 크게 두 가지로 나뉩니다. 하나는 원하는 column만 가지고 오는 방법이 있으며, 다른 하나는 원하는 row만 가지고 오는 방법입니다. 우리는 이 두가지 방법은 각각 **Projection**과 **Selection**이라고 지칭합니다.

아래와 같은 테이블과 데이터가 있다고 합시다.

NAME	DEPT_CD	PHONE	ADDRESS
홍길동	A	01023456789	조선 한양읍
손흥민	A	0112345434	영국 런던
박찬호	C	01023433456	충남 공주
김유신	D	0187766645	신라 경주
박나래	D	0192929384	서울특별시 영등포구
강감찬	E	01023432123	고려

위 테이블에서 Projection , Selection의 방법으로 데이터를 가지고 오면 아래 그림과 같은 결과가 나오게 됩니다.

NAME	DEPT_CD	PHONE	ADDRESS
홍길동	A	01023456789	조선 한양읍
손흥민	A	0112345434	영국 런던
박찬호	C	01023433456	충남 공주
김유신	D	0187766645	신라 경주
박나래	D	0192929384	서울특별시 영등포구
강감찬	E	01023432123	고려

NAME	DEPT_CD	PHONE	ADDRESS
홍길동	A	01023456789	조선 한양읍
손흥민	A	0112345434	영국 런던

Selection

NAME	PHONE
홍길동	01023456789
손흥민	0112345434
박찬호	01023433456
김유신	0187766645
박나래	0192929384
강감찬	01023432123

Projection

자, 이제 기본적인 것을 알았으니, 본격적으로 select의 사용법에 대해 배워보겠습니다.

2. DESC 사용하기

DESC 명령은 특정 테이블에 어떤 칼럼이 있는지 조회하는 명령어입니다. Describe의 약어이며, 아래와 같이 사용합니다.

```
MariaDB [address_svc_m]> desc juso_info_m;
```

Field	Type	Null	Key	Default	Extra
mgmt_num	varchar(26)	NO	PRI	NULL	
road_nm_cd	varchar(12)	NO		NULL	
umd_seq	varchar(2)	NO		NULL	
basement_yn	varchar(1)	YES		NULL	
build_main_num	int(5)	YES		NULL	
build_sub_num	int(5)	YES		NULL	
base_area_num	varchar(5)	YES		NULL	
update_type_cd	varchar(2)	YES		NULL	
anncmt_date	varchar(8)	YES		NULL	
orgn_road_nm_addr	varchar(100)	YES		NULL	
get_detail_addr_yn	varchar(1)	YES		NULL	
inf_full_jb_addr	varchar(300)	YES		NULL	
inf_full_road_addr	varchar(300)	YES	MUL	NULL	
inf_latitude	varchar(30)	YES		NULL	
inf_longitude	varchar(30)	YES		NULL	
tst_lcnnumber	varchar(20)	YES	MUL	NULL	
aud_last_update	datetime	YES		NULL	
inf_flag	varchar(1)	YES	MUL	NULL	
tst_la	varchar(1)	YES		NULL	
tst_long	varchar(1)	YES		NULL	
tst_lc_check	varchar(1)	YES	MUL	NULL	
inf_ip_yearmonth	varchar(6)	YES		NULL	
inf_full_jb_addr_2	varchar(300)	YES		NULL	
inf_full_road_addr_2	varchar(300)	YES		NULL	

24 rows in set (0.003 sec)

위의 그림은 사용법을 설명드린 것이고, 우리도 직접 실습을 해봐야겠죠.

설치한 mysql 데이터베이스에 접속하는 방법은 위의 그림과 같이 cui 기반의 커맨드 창을 열어서 접속을 해도 되고, sqlyog이나 [workbench](#) 같은 무료 툴을 이용해서 접속하시면 됩니다.

[workbench](#)는 아래 링크로 접속하셔서 다운로드하여서 설치하시면 됩니다.

<https://dev.mysql.com/downloads/workbench/>

	<p>MySQL :: Download MySQL Workbench</p> <p>dev.mysql.com</p>
--	---

workbench 툴로 자신이 설치한 데이터베이스에 접속을 합니다. SQL 기본 강의에서는 툴을 사용하여 강의를 진행하도록 하겠습니다.

#필요하신 분들이 많으시면 mysql 설치 및 workbench 설치, 설정법 등을 따로 포스팅하겠습니다. 우선은 네이버나 구글에서 검색하셔서 설정을 마치시길 바랍니다.

자신의 데이터베이스에 workbench 툴로 접속을 하셨다면 먼저 데이터베이스를 하나 만들어야 합니다. 스키마라고도 부르는데, mysql이나 mariadb에서는 데이터베이스라고 표현합니다. 데이터베이스를 만드는 명령어는 아래와 같습니다.

```
create database kmong;
# kmong은 데이터베이스명 입니다. 자신이 원하시는걸로 만드시면 됩니다.
```

그리고, 자신이 생성한 데이터베이스에 접속하여 아래 SQL문을 실행하여 테이블을 하나 생성합니다.

참고로, 아직 create 문에 대해서 배우는 단계가 아니니 아래 create 문은 그냥 테이블을 만드는 명령어구나 생각하시고 실행하시고 지나가면 되겠습니다.

```
create table select_test
(
    name      varchar(50),
    dept_cd   varchar(1),
    phone     varchar(15),
    address   varchar(100)
) character set utf8;
```

#가장 끝에 있는 character set utf8 명령은, 테이블에 한글 데이터가 들어갈 수 있도록 캐릭터 셋을 잡아주는 명령어입니다. 이 글을 보시는 분들이 어떤 캐릭터셋으로 데이터베이스를 설치 했는지 알수 없기에 해당 명령어를 삽입했습니다. 만약, mysql을 설치할때 데이터베이스단에 캐릭터셋을 한글 사용이 가능하게끔 설정했다면 해당 명령어는 제외하고 실행하셔도 무관합니다.

그리고 아래와 같이 DESC 명령을 통해서 테이블 칼럼 목록을 확인합니다.

```
desc select_test;
```

	Field	Type	Null	Key	Default	Extra
1	name	varchar(50)	YES		<null>	
2	dept_cd	varchar(1)	YES		<null>	
3	phone	varchar(15)	YES		<null>	
4	address	varchar(100)	YES		<null>	

select_test라는 테이블의 칼럼 목록이 조회되는 것을 확인할 수 있습니다.

이젠 이 테이블은 select 명령어를 통해서 조회해보도록 하겠습니다.

3. 원하는 데이터만 select 하기

select 명령어로 조회하는 방법은 아래와 같습니다.

```
select * from kmong.select_test;
```

```
select * from select_test;
```

Output kmong.select_test

0 rows Tx: Auto DB

name dept_cd phone address

아직 한건의 데이터도 들어 있지 않아, 칼럼명만 표시되어 있는 것을 확인할 수 있습니다. 수업 진행을 위하여 여기에 데이터를 insert 하겠습니다.

```
INSERT INTO kmong.select_test (name, dept_cd, phone, address) VALUES ('홍길동', 'A', '01023456789',
'조선 한양읍');
INSERT INTO kmong.select_test (name, dept_cd, phone, address) VALUES ('손흥민', 'A', '0112345434',
'영국 런던');
INSERT INTO kmong.select_test (name, dept_cd, phone, address) VALUES ('박찬호', 'C', '01023433456',
'충남 공주');
INSERT INTO kmong.select_test (name, dept_cd, phone, address) VALUES ('김유신', 'D', '0187766645',
'신라 경주');
INSERT INTO kmong.select_test (name, dept_cd, phone, address) VALUES ('박나래', 'D', '0192929384',
'서울특별시 영등포구');
INSERT INTO kmong.select_test (name, dept_cd, phone, address) VALUES ('강감찬', 'E', '01023432123',
'고려');
```

위에 있는 insert 문을 본인의 데이터베이스에서 실행하여 데이터를 넣어줍니다. 각 insert문의 select_test 앞에 있는 kmong은 제가 지금 사용하고 있는 데이터베이스명입니다. 스키마라고도 표현하는데, 이 kmong을 위에서 본인이 만든 데이터베이스명으로 변경하시면 데이터가 잘 들어가는 것을 확인하실 수 있습니다.

이렇게 테이블에 데이터를 넣고 다시 select를 해보도록 하겠습니다.


```
select * from select_test;
```

Output kmong.select_test X				
< < 6 rows > > ↺ + - Tx: Auto ▾ DB ✓ ↶ ■ DDL				
	name	dept_cd	phone	address
1	홍길동	A	01023456789	조선 한양읍
2	손흥민	A	0112345434	영국 런던
3	박찬호	C	01023433456	충남 공주
4	김유신	D	0187766645	신라 경주
5	박나래	D	0192929384	서울특별시 영등포구
6	강감찬	E	01023432123	고려

위에서 insert 한 데이터가 잘 나오는 것을 확인할 수 있습니다.

위 그림에서 보면 `select_test`라는 테이블에는 `name`, `dept_cd`, `phone`, `address` 이렇게 4개의 컬럼이 존재하는 걸 알 수 있습니다.

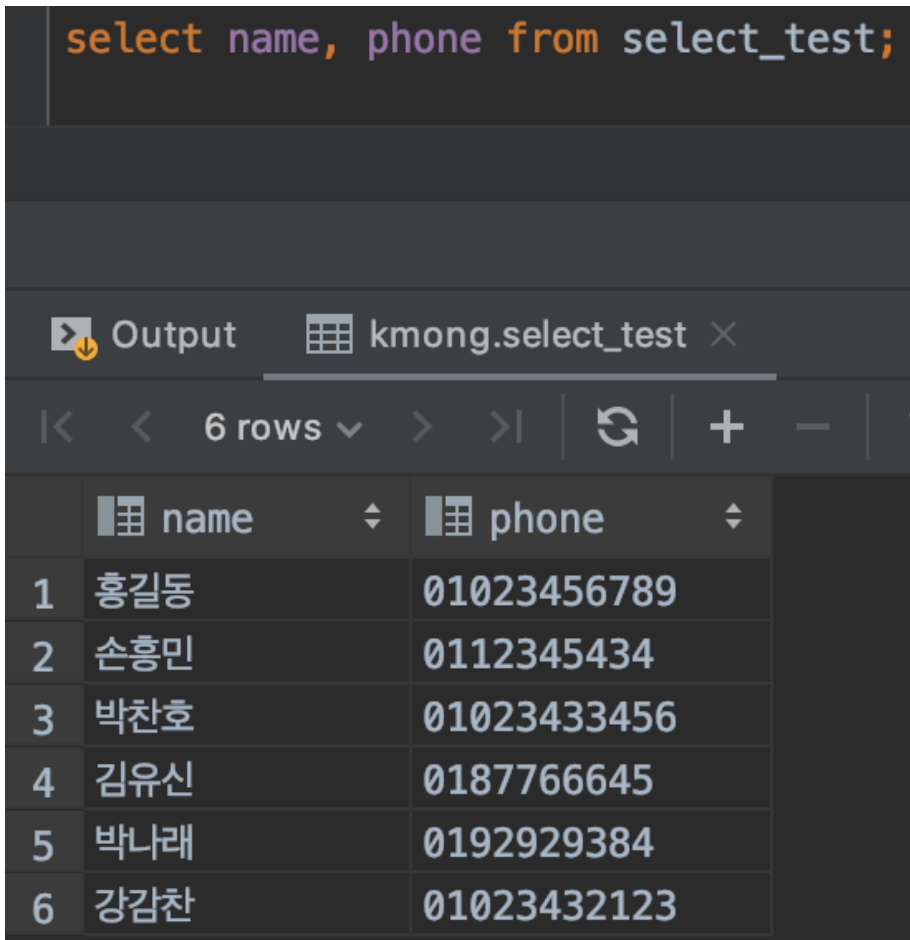
```
select * from kmong.select_test;
```

위 select 명령어 뒤에 있는 `*`은 우리가 **아스타**라고 부르기도 하는데, 어떤 의미를 가질까요? `select_test`라는 테이블의 **모든 컬럼을 다 출력하라는** 의미입니다.

그렇다면 위에서 알아봤던 대로 원하는 컬럼만 출력하는 Projection을 하려면 어떻게 하면 될까요? 아래와 같이 select 명령어를 좀 바꿔서 실행해보겠습니다.

```
select name, phone from select_test;
```

```
select name, phone from select_test;
```



	name	phone
1	홍길동	01023456789
2	손흥민	0112345434
3	박찬호	01023433456
4	김유신	0187766645
5	박나래	0192929384
6	강감찬	01023432123

이렇게 명령어를 좀 변경했더니 위 그림처럼 데이터 출력이 바뀌었습니다. select 다음에 * (아스타) 자리에 **name, phone**이라는 칼럼명이 들어가니 **name, phone** 칼럼의 데이터만 표시가 됩니다.

이번엔 **where**이라는 키워드에 대해서 알아보겠습니다.

□**where**는 조건절이라고 부르기도 합니다. 위에서 알아봤던 selection에 대한 내용인데, 어떤 테이블에 많은 row들 가운데 특정한 조건을 입력하여 조건에 해당하는 데이터만 가지고 오라는 명령어입니다. 아래와 같이 select 문을 작성해서 데이터베이스에서 실행해 보겠습니다.

```
select * from kmong.select_test
where dept_cd = 'A';
```

```
select * from kmong.select_test
where dept_cd = 'A';
```

select_test

Output kmong.select_test

2 rows

	name	dept_cd	phone	address
1	홍길동	A	01023456789	조선 한양읍
2	손흥민	A	0112345434	영국 런던

where 조건절에 dept_cd = 'A'라고 입력했더니 6개의 row 중에 dept_cd가 'A'인 두 row만 출력이 되었습니다. where 키워드가 어떤 의미인지 파악이 되셨나요? 이것이 위에서 그림으로 표현해 드렸던 selection입니다.

위에서 언급한 내용을 정리해보겠습니다. select 명령어 바로 뒤에 출력을 원하는 칼럼명을 쓰면 그 칼럼만 출력을 하라는 의미가 됩니다. 역시 같은 의미로 우리가 그냥 썼던 from 뒤 자리는 테이블명이 오는데, 여기에 다른 테이블 명을 쓰면 그 테이블에 해당하는 데이터를 출력하라는 의미가 되는 것입니다. 그리고 where 뒤에 오는 내용은 어떤 row를 걸러서 출력을 할지를 결정하게 되는 것입니다.

select
from
where

출력할 컬럼명
테이블명
출력할 데이터 조건

키워드

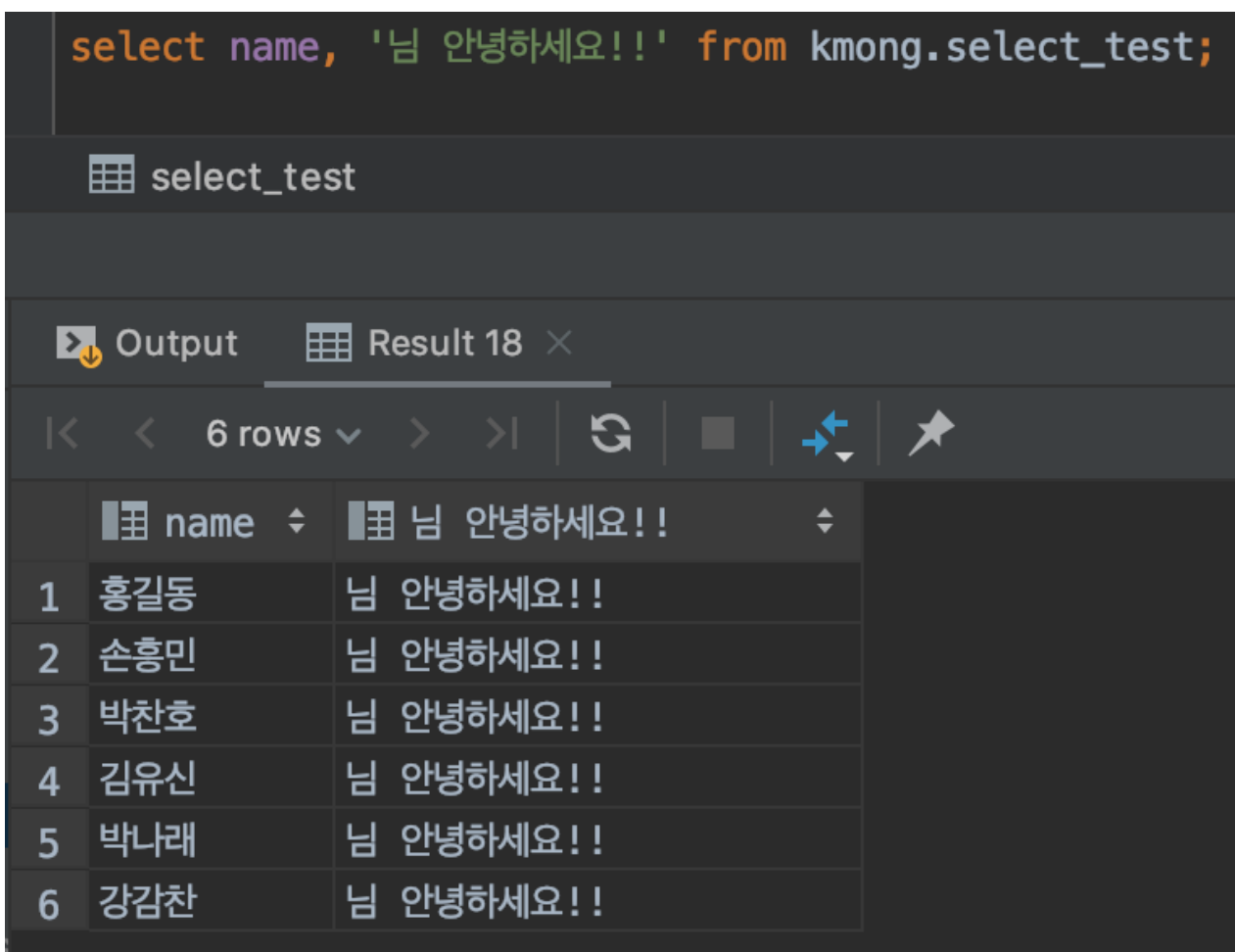
우리는 **select, from, where**를 키워드라고 부릅니다. 이는 mysql이나 oracle 등의 DBMS에서 미리 정해놓은 단어라는 걸 알고 있으면 됩니다. 위 그림과 같은 모습이 우리가 앞으로 공부할 select 명령어의 가장 기본적인 모습이라 생각하면 됩니다.

4. 표현식 (Expression) 사용하기

표현식 (Expression) 이란 칼럼의 데이터 외에 다른 문자열이나 내용을 출력하고 싶을 때 사용할 수 있습니다.

먼저 예문을 보겠습니다.

```
select name, '님 안녕하세요!!' from kmong.select_test;
```



The screenshot shows a SQL IDE interface. At the top, the query `select name, '님 안녕하세요!!' from kmong.select_test;` is entered. Below the query editor, the table `select_test` is selected. The output window shows the results of the query, which are 6 rows. The first column is `name` and the second column is `님 안녕하세요!!`.

	name	님 안녕하세요!!
1	홍길동	님 안녕하세요!!
2	손흥민	님 안녕하세요!!
3	박찬호	님 안녕하세요!!
4	김유신	님 안녕하세요!!
5	박나래	님 안녕하세요!!
6	강감찬	님 안녕하세요!!

위 SQL을 한번 살펴보겠습니다. from 절에 있는 `kmong.select_test`라는 테이블에서 `name`이라는 칼럼만 가지고 와서 출력을 했고, 그 뒤에 `'님 안녕하세요!!'`라는 문자열을 붙였습니다. 이렇게 실행을 하니까 `name` 칼럼에 있는 6개의 데이터가 다 나오고, 그 우측에 `'님 안녕하세요!!'`라는 칼럼이 하나 생기면서 모든 이름 옆에 `'님 안녕하세요!!'`라는 표현식, 리터럴(literal) 상수가 출력되고 있는 것을 확인할 수 있습니다. 추후 SQL을 잘 작성하게 되면 업무에 따라서 이 표현식 기능을 유용하게 사용할 수 있습니다.

5. 컬럼명 대신 별칭(alias)을 사용하기

위에서 표현식을 출력한 예문 화면의 컬럼명을 다시 한번 확인해보겠습니다. name, 님 안녕하세요!!라고 컬럼명이 표시되어 있습니다.

	name	님 안녕하세요!!
1	홍길동	님 안녕하세요!!
2	손흥민	님 안녕하세요!!
3	박찬호	님 안녕하세요!!
4	김유신	님 안녕하세요!!
5	박나래	님 안녕하세요!!
6	강감찬	님 안녕하세요!!

이것을 별칭(alias)으로 표현하는 방법을 알아보겠습니다. select 명령은 아래와 같이 실행하시면 됩니다.(두 가지 방법이 있습니다. 결과는 같습니다)

```
select name 이름, '님 안녕하세요!!' 인사문구 from kmong.select_test;
select name as 이름, '님 안녕하세요!!' as 인사문구 from kmong.select_test;
```

컬럼명 또는 표현식 구문 뒤에 한 칸 공백을 넣고 별칭으로 쓰고 싶은 단어를 넣습니다. 그리고 컬럼 사이의 구분을 해주는 ','를 넣어주면 해당 단어가 아래 그림처럼 별칭으로 나타납니다.

또 다른 방법은 컬럼명, 또는 표현식 구문 뒤에 'as'를 입력하고 별칭을 입력하는 방법입니다. 두 방법이 차이는 'as'입니다.

전 후자를 더 추천합니다. 어차피 기능적으로 같고, 결과도 같지만, 'as'를 넣은 SQL이 훨씬 가독성이 좋기 때문입니다. 인라인 뷰나 여러가지 서브쿼리가 들어가는 복잡한 SQL을 작성 할때 가독성이 매우 중요하기 때문 입니다.

```
select name 이름, '님 안녕하세요!!' 인사문구 from kmong.select_test;
select name as 이름, '님 안녕하세요!!' as 인사문구 from kmong.select_test;
```

select_test > address

Output Result 19 ×

6 rows

	이름	인사문구
1	홍길동	님 안녕하세요!!
2	손흥민	님 안녕하세요!!
3	박찬호	님 안녕하세요!!
4	김유신	님 안녕하세요!!
5	박나래	님 안녕하세요!!
6	강감찬	님 안녕하세요!!

6. DISTINCT로 중복된 값을 제외하고 출력하기

우선 아래 select 명령을 실행합니다.

```
select dept_cd from kmong.select_test;
```

```
select dept_cd from kmong.select_test;
```

	dept_cd
1	A
2	A
3	C
4	D
5	D
6	E

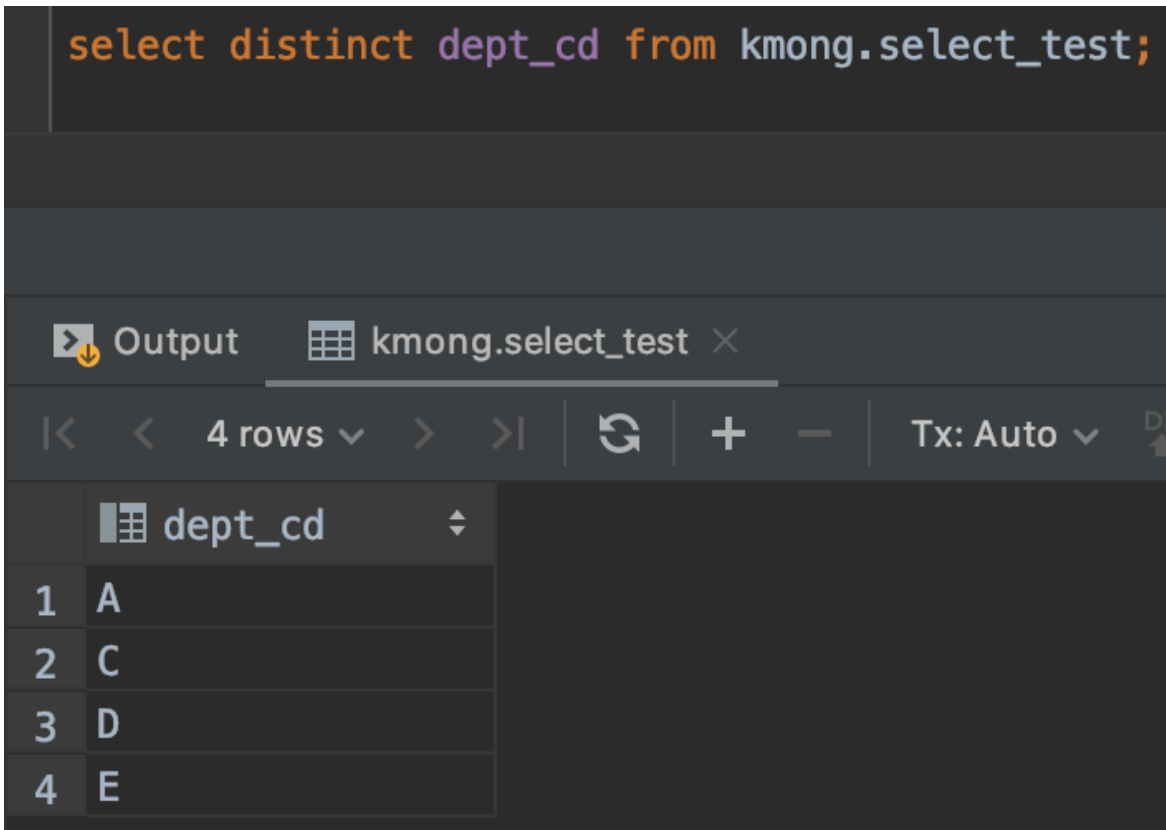
kmong.select_test 테이블에서 dept_cd 칼럼만 출력을 했습니다. dept_cd는 보통 '부서 코드'라고 통칭되는 컬럼명입니다. 만일, 누군가 여기에 있는 부서 코드의 리스트를 보고 싶다면 위와 같이 출력하면 안 되겠죠. A와 D가 중복된 값이 있기 때문입니다.

중복된 값을 제외하는 select 명령문은 아래와 같습니다.

```
select distinct dept_cd from kmong.select_test;
```

눈치채셨나요? 바로 **distinct** 명령이 중복된 값을 제외하고 출력하라는 명령입니다. 결과를 확인해 보겠습니다.

```
select distinct dept_cd from kmong.select_test;
```



	dept_cd
1	A
2	C
3	D
4	E

위와 같이 중복된 데이터가 있던 A와 D의 중복 값이 제외되고 출력된 것을 확인할 수 있습니다.

7. 연결 연산자 함수로 칼럼 값을 붙여서 출력하기

여러 칼럼의 값이나, 문자열로 된 표현식의 값을 붙여서 하나의 column으로 표현할 수 있습니다.

예를 들어서, '홍길동의 부서 코드는 A입니다.' 이런 형식의 데이터를 만들어 출력할 수 있습니다.

아래의 select 명령어를 실행해 볼게요.

```
select concat(name, '의 부서코드는 ', dept_cd, ' 입니다.') from kmong.select_test;
```



```
select concat(name, '의 부서코드는 ', dept_cd, ' 입니다.') from kmong.select_test;
```

Output concat(name, '의 부서코드는...t_cd, ' 입니다.):varchar ×

6 rows

`concat(name, '의 부서코드는 ', dept_cd, ' 입니다. '`

1	Select All 코드는 A 입니다.
2	손흥민의 부서코드는 A 입니다.
3	박찬호의 부서코드는 C 입니다.
4	김유신의 부서코드는 D 입니다.
5	박나래의 부서코드는 D 입니다.
6	강감찬의 부서코드는 E 입니다.

위와 같은 결과가 출력됩니다.

concat이라는 문자열 또는 칼럼의 값을 연결해주는 함수를 사용한 것입니다.

사실 oracle에서는 위와 같이 함수를 사용하지 않고 연결 연산자 만으로도 위와 같이 표현할 수 있습니다.

```
select name||'의 부서코드는 '||dept_cd||' 입니다.' from kmong.select_test;
```

이렇게 말이죠.

서두에서 말씀드린 부분이 이런 부분입니다. 어떤 한 RDBMS의 SQL을 잘 다루게 되면 다른 RDBMS에서도 역시 비슷한 기능을 가진 명령어를 찾아서 똑같이, 내지는 흡사하게 표현할 수 있다는 것입니다.

위에서 보여드린 것과 같이 mysql과 oracle 각각의 RDBMS에서 똑같이 결과를 만들어 낼 수 있습니다.

그럼 여기서 마지막으로 '손흥민'이라는 이름을 가진 사람의 row만 concat 문자열 연결 함수를 사용하여 위와 같이 출력해 볼까요?

```
select concat(name, '의 부서코드는 ', dept_cd, ' 입니다.')
from kmong.select_test
where name = '손흥민'
```

```
select concat(name, '의 부서코드는 ', dept_cd, ' 입니다. ')
from kmong.select_test
where name = '손흥민';
```

Output concat(name, '의 부서코드는...t_cd, ' 입니다.'):varchar ×

1 row ▾ ↺ ■ ↻ ⚡

	`concat(name, '의 부서코드는 ', dept_cd, ' 입니다. ')`
1	손흥민의 부서코드는 A 입니다.

이렇게 where 절을 사용하여 '손흥민'이라는 사람의 데이터만 출력해봤습니다.

참, SQL을 작성할 때 줄 바꿈은 보기 좋게 마음대로 사용하셔도 좋습니다. 단, 위에서 말씀 드리 키워드 부분에서 중간에 줄 바꿈을 한다면, 테이블명, 컬럼명 중간에서 줄 바꿈만 안 하시면 됩니다.