



# 02강.oracle

- 테이블 검색

# 1

## 테이블 검색

---

1. Select \* from tab;

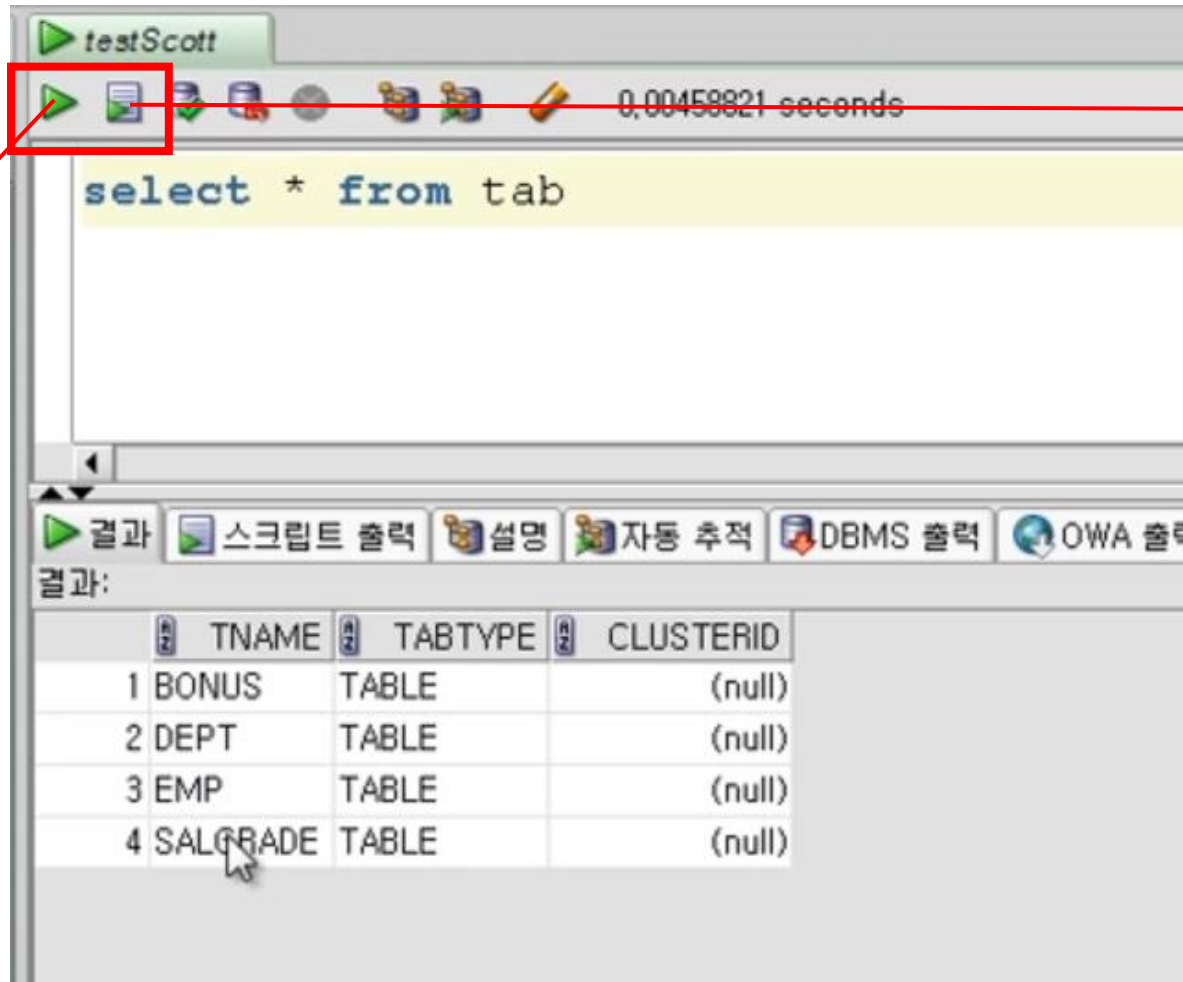
현재 사용자가 가지고 있는 테이블을 모두 보여줌.

<b>예</b>	<b>SELECT * FROM TAB;</b>
----------	---------------------------

# 1 테이블 검색

Select \* from tab;

단축키 **F9**  
- 테이블 형태로 보여짐.



단축키 **F5**  
- 콘솔 형태로 보여짐.

# 1 테이블 구조를 살펴보기 위한 DESC

Desc emp;

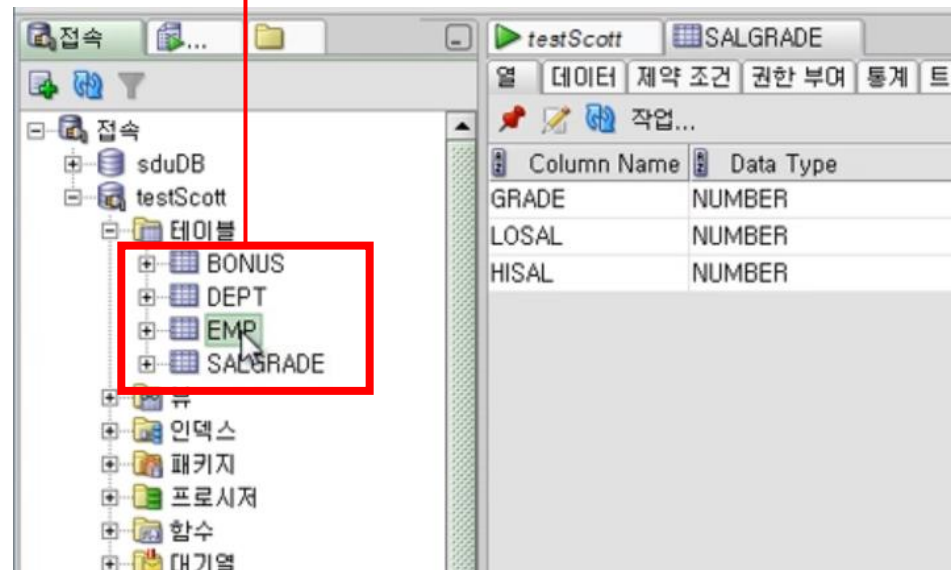
각 필드의 유형을 알아볼 수 있음.

- ❖ 테이블에서 데이터를 조회하기 위해서는 테이블의 구조를 알아야 합니다. 테이블의 구조를 확인하기 위한 명령어로는 DESCRIBE가 있습니다.

형식	DESC[RIBE] 테이블명
----	-----------------

- DESC 명령어는 테이블의 컬럼 이름, 데이터 형, 길이와 NULL 허용 유무 등과 같은 특정 테이블의 정보를 알려줍니다.

클릭시 테이블 유형을  
볼 수 있음



# 1 오라클 데이터 형

number

## ❖ NUMBER

- NUMBER 데이터 형은 숫자 데이터를 저장하기 위해서 제공됩니다.

형식

NUMBER(precision, scale)

- precision은 소수점을 포함한 전체 자리수를 의미하며 scale은 소수점 이하 자리수를 지정합니다.
- scale을 생략한 채 precision만 지정하면 소수점 이하는 반올림되어 정수 값만 저장됩니다.
- precision과 scale을 모두 생략하면 입력한 데이터 값만큼 공간이 할당됩니다.

# 1 오라클 데이터 형

date

## ❖ DATE

- DATE는 세기, 년, 월, 일, 시간, 분, 초의 날짜 및 시간 데이터를 저장하기 위한 데이터 형입니다.
- 이렇듯 날짜 타입 안에는 세기, 년, 월, 일, 시, 분, 초, 요일 등 여러 가지 정보가 들어 있지만 별다른 설정이 없으면 년, 월, 일만 출력합니다.
- 기본 날짜 형식은 "YY/MM/DD" 형식으로 “년/월/일” 로 출력됩니다.
- 2005년 12월 14일은 “05/12/14” 로 출력됩니다.

# 1 오라클 데이터 형

char

## ❖ CHAR

- 문자 데이터를 저장하기 위한 자료형으로 CHAR가 있습니다. CHAR는 고정 길이 문자 데이터를 저장합니다.
- 입력된 자료의 길이와는 상관없이 정해진 길이만큼 저장 영역 차지하며 최소 크기는 1입니다.
- 주소를 저장하기 위해서 address 란 컬럼을 생성하되 저장될 데이터의 최대 크기를 고려해서 CHAR(20)이라고 주었고, 'seoul' 이란 데이터를 저장하였다고 합시다.

address

s	e	o	u	l															
---	---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

- CHAR는 주어진 크기만큼 저장공간이 할당되므로 편차가 심한 데이터를 입력할 경우 위의 예와 같이 저장공간의 낭비를 초래합니다.

# 1 오라클 데이터 형

varchar2

## ❖ VARCHAR2

- VARCHAR2 데이터 형은 가변적인 길이의 문자열을 저장하기 위해서 제공합니다.
- 이번에는 주소를 저장하기 위해서 address 란 컬럼의 데이터형을 VARCHAR2(20)이라고 설정하고, 'seoul' 이란 데이터를 저장하였다고 합시다.

address

s	e	o	u	l
---	---	---	---	---

- VARCHAR2는 저장되는 데이터에 의해서 저장공간이 할당되므로 메모리 낭비를 줄일 수 있습니다.



## 1. 가변 문자형 데이터타입 [ VARCHAR2 vs VARCHAR ]

오라클 : VARCHAR2

MySQL : VARCHAR

## 2. 숫자형 데이터타입 [NUMBER vs INT]

오라클 : NUMBER

MySQL : INT

## 3. 날짜형 데이터타입 [SYSDATE vs DATETIME]

오라클 : SYSDATE

MySQL : DATETIME

# 1 데이터를 조회하기 위한 **select** 문

❖ **SELECT 문은 데이터를 조회하기 위한 SQL 명령어입니다.**

형식

```
SELECT [DISTINCT] {*, column[Alias], ...}  
FROM table_name;
```

- SQL 명령어는 하나의 문장으로 구성되어야 하는데 여러 개의 절이 모여서 문장이 되는 것이고 이러한 문장들은 반드시 세미콜론(;)으로 마쳐야 합니다.
- SELECT 문은 반드시 SELECT와 FROM 이라는 2개의 키워드로 구성되어야 합니다.
- SELECT절은 출력하고자 하는 컬럼 이름을 기술합니다.
- 특정 컬럼 이름 대신 \* 를 기술할 수 있는데, \* 는 테이블 내의 모든 컬럼을 출력하고자 할 경우 사용합니다.
- FROM절 다음에는 조회하고자 하는 테이블 이름을 기술합니다.
- SQL 문에서 사용하는 명령어들은 대문자와 소문자를 구분하지 않는다는 특징이 있습니다.

# 1 부서 테이블 검색

Select \* from dept;

❖ 다음은 부서 테이블의 내용을 살펴보기 위한 쿼리문입니다.

예	<b>SELECT *</b> <b>FROM dept;</b>
---	--------------------------------------

# 1 부서 테이블 검색

Select \* from dept;

❖ 다음은 부서 테이블의 내용을 살펴보기 위한 쿼리문입니다.

예	<pre>SELECT * FROM dept;</pre>
---	------------------------------------

Select empno,ename from dept;

❖ 사원번호에 해당되는 컬럼 이름은 empno이고 사원명에 해당되는 컬럼 이름은 ename입니다. empno와 ename을 출력하기 위해서는 출력하고자 하는 순서대로 기술하되 컬럼과 컬럼 사이에 콤마를 기술합니다.

예	<pre>select empno, ename from emp;</pre>
---	--

# 1 부서 테이블 검색

실습1.

사원의 이름과 급여와 입사일자만을 출력하는 SQL 문을 작성해 보시오.  
<힌트> 사원 정보가 저장된 테이블의 이름은 EMP이고, 사원이름 칼럼은 ENAME, 급여 칼럼은 SAL, 입사일자 칼럼은 HIREDATE입니다.

# 1 연산자 사용

Select \* from dept;

❖ 다음은 부서 테이블의 내용을 살펴보기 위한 쿼리문입니다.

예	<pre>SELECT * FROM dept;</pre>
---	------------------------------------

Select empno,ename from dept;

❖ 직원번호에 해당되는 컬럼 이름은 empno이고 직원명에 해당되는 컬럼 이름은 ename입니다. empno와 ename을 출력하기 위해서는 출력하고자 하는 순서대로 기술하되 컬럼과 컬럼 사이에 콤마를 기술합니다.

예	<pre>select empno, ename from emp;</pre>
---	--



THANK YOU