

# Oracle Introduction to SQL

# 테이블 생성

---

SQL> select \* from tab;

- ◆ 부서 테이블 : dept Table
  - ◆ 사원 테이블 : emp Table
  - ◆ 호봉 테이블 : salgrade Table
-

## 부서 테이블 : dept Table

---

```
SQL> desc dept;
```

```
SQL> describe dept;
```

```
SQL> select *  
      from dept;
```

---

# 사원 테이블 : emp Table

---

```
SQL> desc emp;
```

```
SQL> select * from emp;
```

---

호봉 테이블 : salgrade Table

---

SQL> desc salgrade;

SQL> select \* from salgrade;

---

# SELECT 문장

---

## <Syntax>

```
SELECT [DISTINCT] {*, column [alias], ...}  
FROM   table_name  
[WHERE condition]  
[GROUP BY group_by_expression]  
[HAVING group_condition]  
[ORDER BY {column, expression} [ASC | DESC]];
```

[참고]

◆ 산술 표현식

: 데이터가 출력되는 방식을 수정하거나 계산을 수행하고자 할 때 사용  
그리고 SELECT 문장에서는 FROM 절을 제외한 SQL 문장의 절에서  
사용 가능

---

# 연산자의 종류

---

산술 연산자	( ), *, /, +, - 순으로 우선순위가 결정됨
비교 연산자	=, !=, <, > 등
논리 연산자	NOT, AND, OR
SQL 연산자	IN, BETWEEN, IS NULL, LIKE, EXISTS 등
결합 연산자	(스트링 값을 결합할 때 사용)
집합 연산자	UNION, UNION ALL, INTERSECT, MINUS

---

# Null 값의 처리

---

- ❑ Null 값은 0 이나 공백과는 다르다.
- ❑ 0은 숫자이며 공백은 문자이다.
- ❑ 열이 NOT NULL 로 정의되지 않았거나, 열이 생성될 때 Primary Key 로 정의되지 않았다면, 어떤 데이터형의 열은 null 값을 포함할 수 있다.

[예] EMP 테이블의 보너스(COMM) 열에서 오직 SALESMAN 만이 보너스를 받을 수 있다.

[참고]

- ◆ null 값을 포함한 산술 표현식의 결과는 null 이 된다.
  - ◆ 어떠한 datatype column 들이라도 null 값을 포함할 수 있다.
-



# NVL 함수

---

- 1) Null 값을 어떤 특정한 값(실제 값)으로 변환하는데 사용
- 2) 사용될 수 있는 데이터 타입은 날짜, 문자, 숫자이다.
- 3) NVL 함수를 사용할 때 전환되는 값의 데이터 타입을 일치 시켜야 한다.

<Syntax>

NVL(expr1, expr2)

expr1	Null 값을 포함하고 있는 Column 이나 표현식
expr2	Null 변환을 위한 목표 값

---

◆ 다양한 데이터형에 대한 NVL 변형

NUMBER	[예] NVL(comm, 0)
CHAR or VARCHAR2	[예] NVL(job, '업무없음')

---

[문제] EMP 테이블에서 사원이름, 급여, 보너스, 연봉을 출력  
하시오.

---

[문제점]

```
SQL> select  ename, sal, comm, sal*12+comm  
        from  emp;
```

---

# SELECT 문장

---

```
SQL> select  ename, sal, comm, sal*12+NVL(comm,0)
         from  emp;
```

---

# 열에 별칭(Alias) 부여

---

- ❑ 질의의 결과를 출력할 때 보통 열 Heading 으로 선택된 열 이름을 사용한다. 이 Heading 은 때로 사용자가 이해하기 어려운 경우가 있기 때문에 열 Heading 을 변경하여 질의 결과를 출력하면 보다 쉽게 사용자가 이해할 수 있다.
  - ❑ 열 별칭(Alias) 정의
    - 1) 열 Heading 이름을 변경
    - 2) 열 이름 바로 뒤에 사용하며 열 이름과 별칭 사이에 키워드 AS를 넣기도 한다.
    - 3) 공백이나 특수 문자 또는 대문자가 있으면 이중 인용 부호(" ")가 필요하다.
-

[예제] EMP 테이블에서 ENAME을 NAME으로 SAL을 SALARY로 출력하시오.

---

```
SQL> select ename AS name, sal salary  
       from emp;
```

---

[문제1] EMP 테이블에서 ENAME을 “성 명”으로 SAL을 “급 여”로  
출력하시오.

---

[문제2] EMP 테이블에서 ENAME을 Name으로 SAL\*12를  
Annual Salary 로 출력하시오.

---

---

[문제1]

```
SQL> select ename "성명", sal "급여"  
          from emp;
```

[문제2]

```
SQL> select ename "Name", sal*12 "Annual Salary"  
          from emp;
```

---

# SQL 연산자

---

연산자	설 명
BETWEEN a AND b	a와 b 사이에 있다.(a, b 값 포함) <b>작은값을 a로</b>
IN (list)	List 의 값 중 어느 하나와 일치한다.
LIKE	문자 형태와 일치한다.(%, _ 사용)
IS NULL	NULL 값을 가졌다.
NOT BETWEEN a AND b	a와 b 사이에 있지 않다.(a, b 값 포함하지 않음)
NOT IN (list)	List 의 값과 일치하지 않는다.
NOT LIKE	문자 형태와 일치하지 않는다.
IS NOT NULL	NULL 값을 갖지 않는다.

[예제] EMP 테이블에서 급여가 1300에서 1500 사이의 정보를  
성명, 담당업무, 급여, 부서번호를 출력하시오.

---



---

[방법1]

```
SQL> select  ename, job, sal, deptno  
        from  emp  
        where sal BETWEEN 1300 AND 1500;
```

---

[방법2]

```
SQL> select  ename, job, sal, deptno  
        from  emp  
        where sal >= 1300 AND sal <=1500;
```

---

[예제] EMP 테이블에서 사원번호가 7902,7788,7566 인 사원의 정보를 사원번호, 성명, 담당업무, 급여, 입사일자를 출력하시오.

---

---

```
SQL> select empno, ename, job, sal, hiredate  
      from emp  
      where empno IN (7902,7788,7566);
```

---

[예제] EMP 테이블에서 입사일자가 81 년도에 입사한 사원의 정보를 사원번호, 성명, 담당업무, 급여, 입사일자, 부서번호를 출력하시오.

---

---

```
SQL> select empno, ename, job, sal, hiredate, deptno  
       from emp  
       where hiredate LIKE '81%';
```

---

```
SQL> select empno, ename, job, sal, hiredate, deptno  
       from emp  
       where job LIKE 'SAL%';
```

---

[예제] EMP 테이블에서 보너스가 NULL 인 사원의 정보를  
사원번호, 성명, 담당업무, 급여, 입사일자, 부서번호를 출력  
하시오.

---

---

```
SQL> select empno, ename, job, sal, hiredate, deptno  
       from emp  
       where comm IS NULL;
```

---

# 논리 연산자

---

연산자	의 미
AND	양쪽 컴포넌트의 조건이 TRUE 이면 TRUE를 RETURN 한다.
OR	한쪽 컴포넌트의 조건만이 TRUE 이면 TRUE를 RETURN 한다.
NOT	이후의 조건이 FALSE 이면 TRUE를 RETURN 한다.

---



[예제] EMP 테이블에서 급여가 1100 이상이고 JOB이  
MANAGER 인 사원의 정보를 출력하시오.

---

---

```
SQL> select *  
      from emp  
      where sal >= 1100 AND job = 'MANAGER';
```

---

[예제] EMP 테이블에서 급여가 1100 이상 **이거나** JOB 이 Manager 인 사원의 정보를 출력하시오.

---

```
SQL> select *  
      from emp  
      where sal >= 1100 OR job = 'MANAGER';
```

---

[예제] EMP 테이블에서 JOB 이 MANAGER, CLERK, ANALYST 가 아닌 사원의 정보를 출력하시오.

---

```
SQL> select *  
      from emp  
      where job NOT IN ('MANAGER', 'CLERK', 'ANALYST');
```

---

## 다양한 정렬 방법(모두 같은 결과가 나옴)

---

SQL> select empno, ename, job, sal, sal\*12 annsal  
from emp  
order by **annsal**;                      ← 별칭으로

SQL> select empno, ename, job, sal, sal\*12 annsal  
from emp  
order by **sal\*12**;                      ← 표현식으로

SQL> select empno, ename, job, sal, sal\*12 annsal  
from emp  
order by **5**;                          ← 숫자로

---