# 4. 실습 SELECT

# 1. SQL 분류 와 SELECT

| 분 류 | 종 류 |
|---|---|
| QUERY | SELECT(데이터 조회) |
| DML(Data Manipulation Language) | INSERT(데이터 입력), UPDATE(데이터 수정), DELETE(데이터 삭제) |
| TCL(Transaction Control Language) | COMMIT(트랜잭션 저장),ROLLBACK(트랜잭션 취소), SAVEPOINT(트랜잭션 임시 저장점) |
| DDL(Data Definition Language) | CREATE(데이터베이스 Object 생성), ALTER(데이타베이스 Object 변경), DROP(데이타베이스 Object 삭제) |
| DCL(Data Control Language) | GRANT(권한 부여),REVOKE(권한 취소) |

Google →
    oracle select syntax
        → SELECT(클릭)
            → SYNTAX(클릭)

**query_block::=**
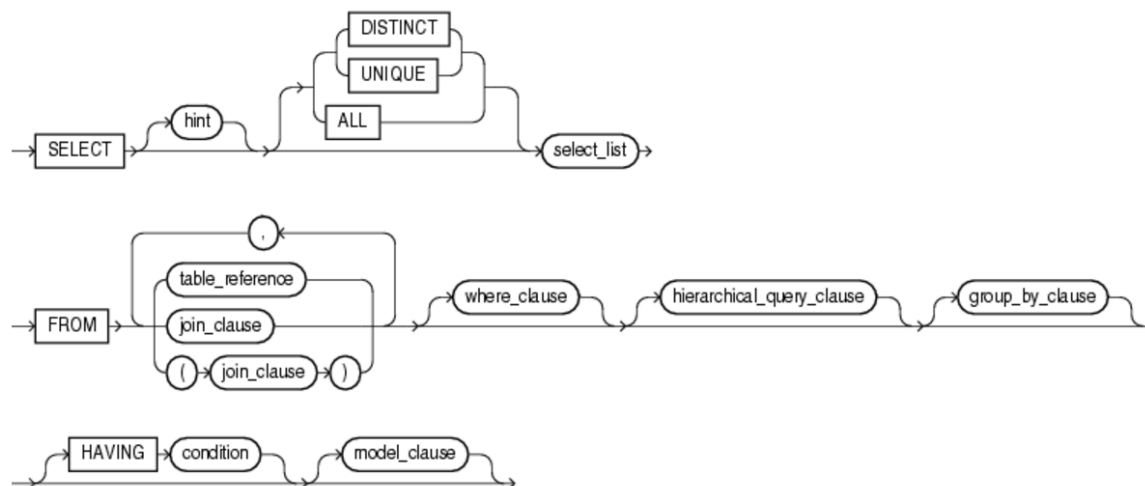


Description of the illustration query_block.gif

# ● 2. SELECT 실습

1) 실습환경
2) SELECT LIST
3) WHERE
4) ORDER BY
5) DISTINCT , CASE
6) GROUP BY , HAVING
7) NULL
8) DECODE, CASE
9) ROWNUM
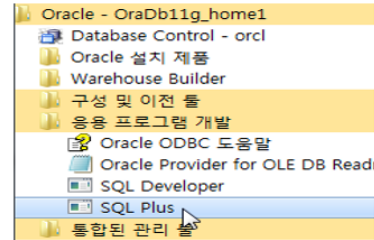10) AND OR NOT
11) SQL 연산자
12) 함수

# ● 2-1 실습 환경

**① SQL*Plus**

오랫동안 지원해왔던 개발도구로 11g버전 부터는 DOS모드의 SQL*Plus만 지원한다.
[참고] 10g 버전에서는 Window GUI상에서 동작하는 sqlplusw 와 DOS모드에서 동작하는
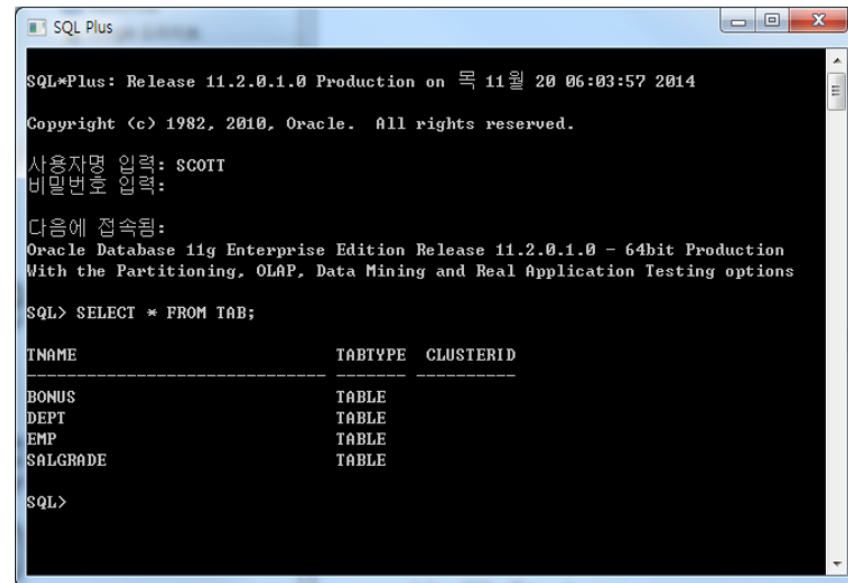　　　 sqlplus 를 제공 한다.

(1) [시작]버튼 클릭
(2) [Oralce-OraDb 11g_home1] 클릭
(3) [응용프로그램 개발] 클릭
(4) [SQL Plus] 클릭



(5) 사용자명/비밀번호 :scott/tiger
　 [주의]ID는 대소문자 구분하지 않고 비밀번호는 대소문자 구분 한다.

(6) SELECT * FROM TAB;
　 – [그림 2-1-1]의 결과화면 처럼
　　 4개 행(레코드)가 나타나는지
　　 확인한다.

(7) EXIT;
　 – SQL Plus 종료 명령어



```
SQL*Plus: Release 11.2.0.1.0 Production on 목 11월 20 06:03:57 2014

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

사용자명 입력: SCOTT
비밀번호 입력:

다음에 접속됨:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SELECT * FROM TAB;

TNAME                          TABTYPE  CLUSTERID
------------------------------ -------- ----------
BONUS                          TABLE
DEPT                           TABLE
EMP                            TABLE
SALGRADE                       TABLE

SQL>
```
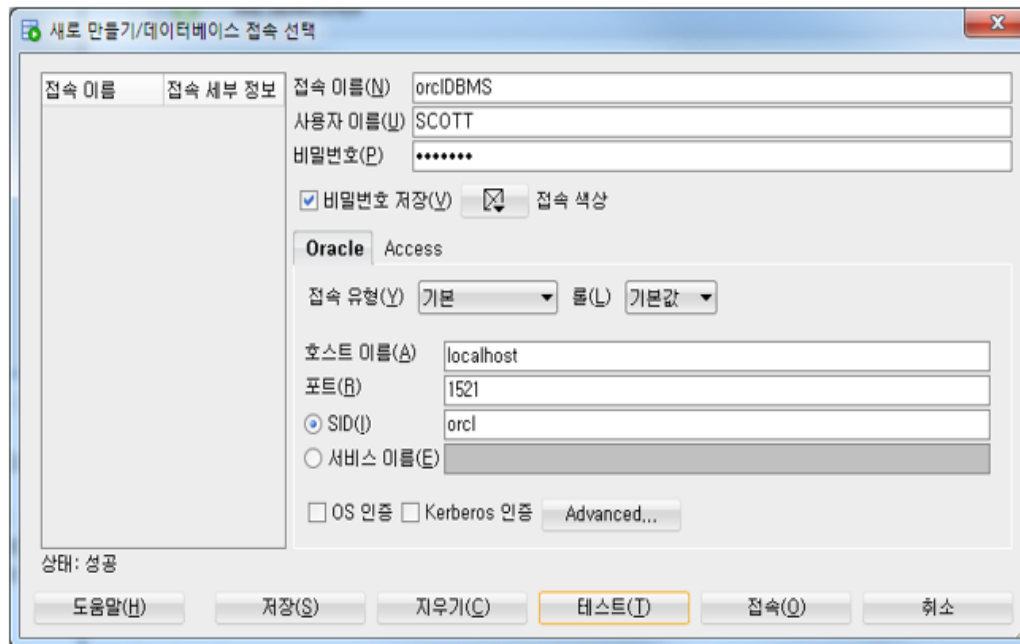
# ● **2-1 실습 환경**

## **② SQL Developer**

### Java 기반 **GUI IDE(Integrated Development Environment:통합개발환경)를 갖춘 SQL 개발툴**

(1) SQL-Developer 접속 환경설정

  ① 왼쪽상단 [클릭]

  ② 접속이름:orclDBMS , 사용자이름:SCOTT, 비밀번호:tiger , 비밀번호 저장 Checkbox 선택
    호스트이름:localhost , 포트:1521 , SID: ????? 입력

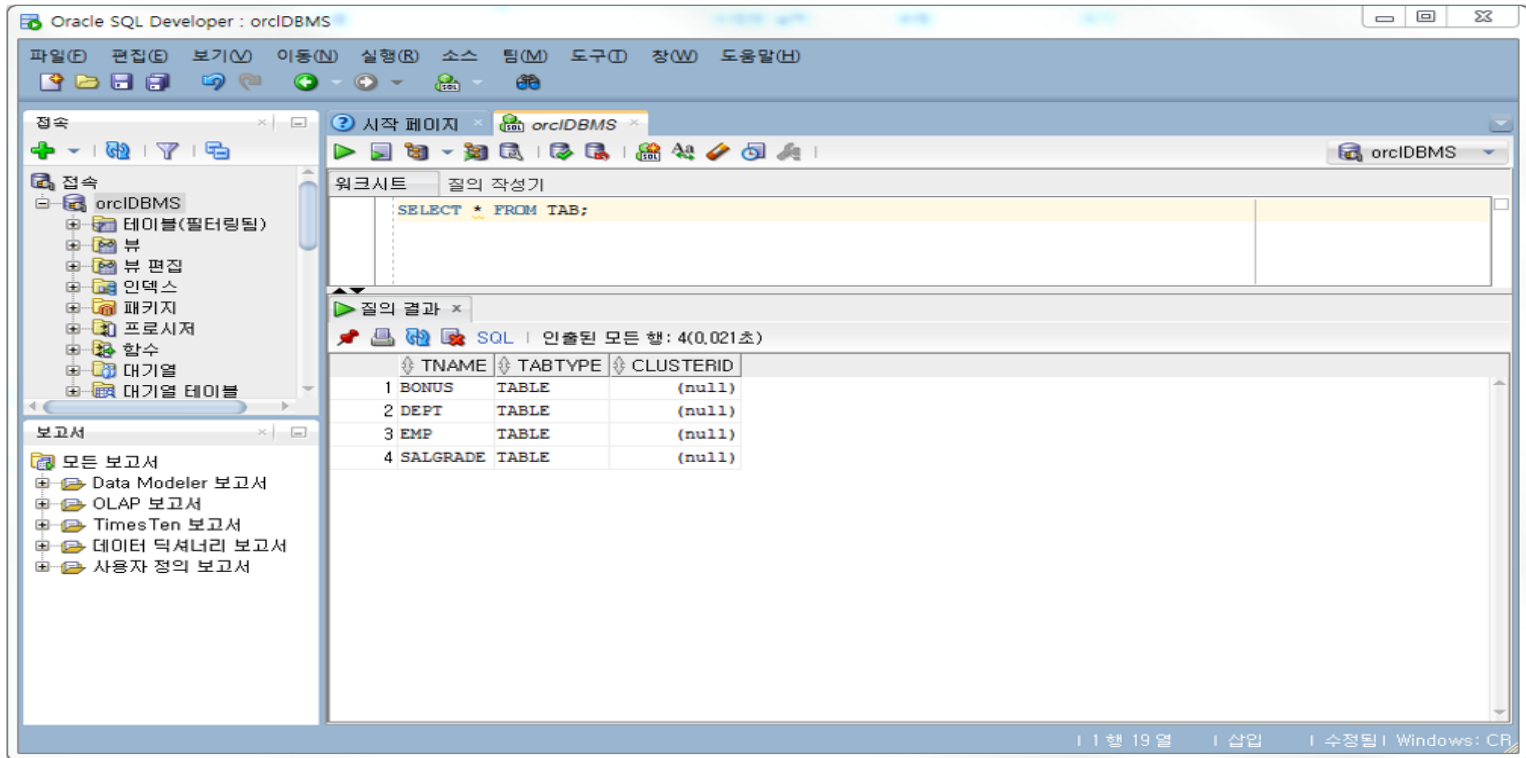  ③ [테스트] 클릭 ④ 상태:[성공] 메시지 확인 ⑤ [저장] 클릭

● **2-1 실습 환경**

(2) 왼쪽 상단 [orclDBMS]을 클릭하여 DBMS에 접속한다.
    SELECT * FOM TAB; 입력후 2가지 방법으로 SQL을 실행후 4개 행(레코드)을 확인한다.

    ① ▶ 클릭 , ② CTRL + Enter Key



(3) ALT + F4로 SQL-Developer를 종료한다.

# 2-2 SELECT LIST

## ▣ SELECT LIST

의미: SELECT 와 FROM사이를 SELECT LIST
역할: 원하는 COLUMN(즉 속성)만 조회(Projection)

① SELECT * FROM EMP;

② SELECT EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO FROM EMP;
③ SELECT EMPNO, ENAME, JOB, SAL FROM EMP;
④ SELECT SAL, JOB, EMPNO, ENAME FROM EMP;
⑤ SELECT EMPNO, EMPNO, EMPNO, ENAME, JOB, SAL FROM EMP;
⑥ SELECT EMPNO, SAL, 8, 'ORACLE' FROM EMP;

* 컬럼의 조회순서를 임의로 변경이 가능
* 하나의 컬럼을 여러 번 조회 가능
* 8: 숫자 데이터, 'ORACLE' :문자 데이터
   존재하지 않는 컬럼을 만들어서 조회 가능

| | EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
|---|---|---|---|---|---|---|---|---|
| 1 | 7369 | SMITH | CLERK | 7902 | 80/12/17 | 800 | (null) | 20 |
| 2 | 7499 | ALLEN | SALESMAN | 7698 | 81/02/20 | 1600 | 300 | 30 |
| 3 | 7521 | WARD | SALESMAN | 7698 | 81/02/22 | 1250 | 500 | 30 |
| 4 | 7566 | JONES | MANAGER | 7839 | 81/04/02 | 2975 | (null) | 20 |
| 5 | 7654 | MARTIN | SALESMAN | 7698 | 81/09/28 | 1250 | 1400 | 30 |
| 6 | 7698 | BLAKE | MANAGER | 7839 | 81/05/01 | 2850 | (null) | 30 |
| 7 | 7782 | CLARK | MANAGER | 7839 | 81/06/09 | 2450 | (null) | 10 |
| 8 | 7788 | SCOTT | ANALYST | 7566 | 87/04/19 | 3000 | (null) | 20 |
| 9 | 7839 | KING | PRESIDENT | (null) | 81/11/17 | 5000 | (null) | 10 |
| 10 | 7844 | TURNER | SALESMAN | 7698 | 81/09/08 | 1500 | 0 | 30 |
| 11 | 7876 | ADAMS | CLERK | 7788 | 87/05/23 | 1100 | (null) | 20 |
| 12 | 7900 | JAMES | CLERK | 7698 | 81/12/03 | 950 | (null) | 30 |
| 13 | 7902 | FORD | ANALYST | 7566 | 81/12/03 | 3000 | (null) | 20 |
| 14 | 7934 | MILLER | CLERK | 7782 | 82/01/23 | 1300 | (null) | 10 |

# 2-2 SELECT LIST

**▣ 산술 연산자**
* SQL 문장안에서 사칙연산( *, /, +, - )의 의미는?
* CLIENT/SERVER적 관점에서는?
* NUMBER 와 DATE 자료형에 적용한다.
* 연산자 우선순위 : 수학과 동일하다 ???

    ① ( ) ② NOT ③ 비교연산자 , SQL연산자 ④ AND ⑤ OR ⑥ 산술연산자(?) *, / , + , -

① SELECT ENAME,SAL,SAL*12,COMM,COMM+300 FROM EMP;

② SELECT SAL, SAL+300*12,(SAL+300)*12 FROM EMP;

**▣ Alias 종류: Table Alias, Column Alias**
Table Alias용도: 일반적으로는 SELF JOIN 시에 사용
    ex) 사용방법: SELECT E.EMPNO,E.ENAME FROM EMP E;

Column Alias 용도: Column Heading(컬럼 레이블 변경?)을 의미있는 다른 이름으로 재정의
    ① COLUMN의 의미 명료성  ②DB 프로그램 개발시 Code의 명료성(계산식에서 유용)

③ SELECT  ENAME,SAL+200 bonus,SAL*12 as annual_SAL, COMM, COMM+300 "Special Bonus" FROM EMP;

④ SELECT  ENAME, COMM+300  AS  "Special Bonus"  FROM EMP;

⑤ SELECT  ENAME, COMM+300  보너스  FROM EMP;

## 2-2 SELECT  LIST

### ▣ (문자열) 합성 연산자

① SELECT  ENAME||JOB  FROM EMP;

② SELECT DNAME||' 부서는 '||LOC||' 지역에 위치합니다.' as LOC FROM DEPT;

③ SELECT  ENAME,SAL,SAL*100, SAL||'00',to_char(SAL)||'00'  FROM EMP;

### ▣ DUAL
정의: Dummy Table
　　 sys user 소유, 1개의 row, 1개의 column을 가진 작고 가벼운 테이블,
용도:실제 테이블로부터 데이터를 가져오는 것이 아닌  function, calculation을 수행하기 위해서 사용

④ Desc  dual     // desc dual   , DESC DUAL
　 SELECT * FROM dual;

⑤ SELECT sysdate FROM dual;

⑥ SELECT 143475*127363, to_char(143475*127363,'999,999,999,999') FROM dual;

# ● 2-3 WHERE

**WHERE**

　　　　의미:조건절
　　　　　역할:원하는 ROW(레코드)만 조회(Selection)

① SELECT * FROM EMP WHERE DEPTNO = 10;

② SELECT EMPNO,ENAME,JOB,SAL  FROM EMP WHERE SAL > 2000;

③ SELECT SALGRADE,SAL,EMPNO,ENAME FROM EMP  WHERE  DEPTNO = 10 AND SAL > 2000;

　　SELECT DEPTNO,SAL,EMPNO,ENAME FROM EMP  WHERE  DEPTNO = 10 AND SAL > 2000;

　　SELECT DEPTNO,SAL,EMPNO,ENAME FROM EMP  WHERE  DEPTNO = 10 OR SAL > 2000;

④ SELECT DEPTNO,ENAME,JOB FROM EMP WHERE JOB = 'manager';

⑤ SELECT DEPTNO,ENAME,JOB FROM EMP WHERE 1=1;

⑥ SELECT DEPTNO,ENAME,JOB FROM EMP WHERE 1=2;

| | EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
|---|---|---|---|---|---|---|---|---|
| 1 | 7369 | SMITH | CLERK | 7902 | 80/12/17 | 800 | (null) | 20 |
| 2 | 7499 | ALLEN | SALESMAN | 7698 | 81/02/20 | 1600 | 300 | 30 |
| 3 | 7521 | WARD | SALESMAN | 7698 | 81/02/22 | 1250 | 500 | 30 |
| 4 | 7566 | JONES | MANAGER | 7839 | 81/04/02 | 2975 | (null) | 20 |
| 5 | 7654 | MARTIN | SALESMAN | 7698 | 81/09/28 | 1250 | 1400 | 30 |
| 6 | 7698 | BLAKE | MANAGER | 7839 | 81/05/01 | 2850 | (null) | 30 |
| 7 | 7782 | CLARK | MANAGER | 7839 | 81/06/09 | 2450 | (null) | 10 |
| 8 | 7788 | SCOTT | ANALYST | 7566 | 87/04/19 | 3000 | (null) | 20 |
| 9 | 7839 | KING | PRESIDENT | (null) | 81/11/17 | 5000 | (null) | 10 |
| 10 | 7844 | TURNER | SALESMAN | 7698 | 81/09/08 | 1500 | 0 | 30 |
| 11 | 7876 | ADAMS | CLERK | 7788 | 87/05/23 | 1100 | (null) | 20 |
| 12 | 7900 | JAMES | CLERK | 7698 | 81/12/03 | 950 | (null) | 30 |
| 13 | 7902 | FORD | ANALYST | 7566 | 81/12/03 | 3000 | (null) | 20 |
| 14 | 7934 | MILLER | CLERK | 7782 | 82/01/23 | 1300 | (null) | 10 |

# 2-4 ORDER BY

**[역할]** 정렬(SORTING)
**[기준]** 정렬시 값 비교기준

숫자 ~ 작은수/큰수        **EX)** 123 < 456
문자 ~ 알파벳 순서(ASCII)      **EX)** 'SCOTT' < 'T'
날짜 ~ 숫자와 동일         **EX)** '2003/11/16' > '19990916'
NULL ~ 가장 큰 값으로 간주

**[방향]**     ASC ~ 오름차순(ASCENDING ORDER) , DEFAULT
            DESC ~ 내림차순(DESCENDING ORDER)

SELECT DEPTNO,ENAME,SAL,HIREDATE FROM EMP ORDER BY ENAME;
SELECT DEPTNO,ENAME,SAL,HIREDATE FROM EMP ORDER BY SAL;
SELECT DEPTNO,ENAME,SAL,HIREDATE FROM EMP ORDER BY SAL ASC;
SELECT DEPTNO,ENAME,SAL,HIREDATE FROM EMP ORDER BY SAL DESC;
SELECT DEPTNO,JOB,ENAME FROM EMP ORDER BY DEPTNO;
SELECT DEPTNO,JOB,ENAME FROM EMP ORDER BY DEPTNO,JOB;
SELECT DEPTNO,JOB,ENAME FROM EMP ORDER BY DEPTNO,JOB desc;

| | EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
|----|-------|--------|-----------|--------|----------|------|--------|--------|
| 1 | 7369 | SMITH | CLERK | 7902 | 80/12/17 | 800 | (null) | 20 |
| 2 | 7499 | ALLEN | SALESMAN | 7698 | 81/02/20 | 1600 | 300 | 30 |
| 3 | 7521 | WARD | SALESMAN | 7698 | 81/02/22 | 1250 | 500 | 30 |
| 4 | 7566 | JONES | MANAGER | 7839 | 81/04/02 | 2975 | (null) | 20 |
| 5 | 7654 | MARTIN | SALESMAN | 7698 | 81/09/28 | 1250 | 1400 | 30 |
| 6 | 7698 | BLAKE | MANAGER | 7839 | 81/05/01 | 2850 | (null) | 30 |
| 7 | 7782 | CLARK | MANAGER | 7839 | 81/06/09 | 2450 | (null) | 10 |
| 8 | 7788 | SCOTT | ANALYST | 7566 | 87/04/19 | 3000 | (null) | 20 |
| 9 | 7839 | KING | PRESIDENT | (null) | 81/11/17 | 5000 | (null) | 10 |
| 10 | 7844 | TURNER | SALESMAN | 7698 | 81/09/08 | 1500 | 0 | 30 |
| 11 | 7876 | ADAMS | CLERK | 7788 | 87/05/23 | 1100 | (null) | 20 |
| 12 | 7900 | JAMES | CLERK | 7698 | 81/12/03 | 950 | (null) | 30 |
| 13 | 7902 | FORD | ANALYST | 7566 | 81/12/03 | 3000 | (null) | 20 |
| 14 | 7934 | MILLER | CLERK | 7782 | 82/01/23 | 1300 | (null) | 10 |

# ● 2-5 DISTINCT , FUNCTION

**DISTINCT [역할] : 중복된 데이터를 필터링하여 조회(SELECT)**

① SELECT JOB FROM EMP;

② SELECT DISTINCT JOB FROM EMP;

③ SELECT DISTINCT DEPTNO, JOB FROM EMP;

④ SELECT DISTINCT DEPTNO, JOB FROM EMP  ORDER BY DEPTNO;

| | DEPTNO | JOB |
|---|---|---|
| 1 | 20 | CLERK |
| 2 | 30 | SALESMAN |
| 3 | 20 | MANAGER |
| 4 | 30 | CLERK |
| 5 | 10 | PRESIDENT |
| 6 | 30 | MANAGER |
| 7 | 10 | CLERK |
| 8 | 10 | MANAGER |
| 9 | 20 | ANALYST |

**FUNCTION [역할] : 입력값(Input Data)을 기반으로 특정 연산을 수행하여**
**결과값(Output Data) 값을 리턴(Return)하는 단위 기능 모듈(Module)**

SELECT ENAME,LENGTH(ENAME),UPPER(ENAME),LOWER(ENAME) FROM EMP;

⑤ SELECT COUNT(*),COUNT(SAL),SUM(SAL),AVG(SAL),MIN(SAL),MAX(SAL) FROM EMP;

| | | |
|---|---|---|
| ORACLE 제공 함수 | 단일행 함수 | 문자 함수 |
| | | 숫자 함수 |
| | | 날짜 함수 |
| | | 형변환 함수 |
| | | 기타 함수 |
| | 그룹행 함수 | |
| 사용자 정의 함수 | PL/SQL로 사용자가 작성 | |

## ● 2-6  GROUP BY , HAVING

**GROUP BY  [역할] :  기준 컬럼으로 데이터를 그룹핑**

① SELECT   DEPTNO,COUNT(*)  FROM SCOTT.EMP  GROUP BY DEPTNO;
SELECT   DEPTNO,AVG(SAL) ,SUM(SAL) FROM EMP  GROUP BY DEPTNO;
SELECT   DEPTNO,AVG(SAL) ,SUM(SAL) FROM EMP
GROUP   BY DEPTNO
ORDER   BY DEPTNO;

| | DEPTNO | AVG(SAL) | SUM(SAL) |
|---|---|---|---|
| 1 | 30 | 1566.6666... | 9400 |
| 2 | 20 | 2175 | 10875 |
| 3 | 10 | 2916.6666... | 8750 |

② SELECT   DEPTNO,ROUND(AVG(SAL),1) ,SUM(SAL) FROM EMP
GROUP   BY DEPTNO
ORDER   BY DEPTNO;

③ [과제]  부서별 최대 ,최소 급여를 조회 하십시요

**HAVING  [역할] :  GROUP BY 결과 데이터에 대한 조건절**

⑤ SELECT   DEPTNO,ROUND(AVG(SAL),1) ,SUM(SAL) FROM
EMP
GROUP   BY DEPTNO
HAVING   SUM(SAL) >= 9000
ORDER   BY DEPTNO DESC;

| | DEPTNO | AVG(SAL) | SUM(SAL) |
|---|---|---|---|
| 1 | 30 | 1566.6666... | 9400 |
| 2 | 20 | 2175 | 10875 |
| 3 | 10 | 2916.6666... | 8750 |

## ● 2-7 NULL

**[정의]** 값이 정의되지 않은, 존재하지 않는(unassigned) → 데이터가 없는(unavailable, unknown,inapplicable)
<u>not the same as zero or a blank space</u>

**[주의]**

① 연산불가 ② 비교불가 ③ 적용불가

**[함수]** ① Single Row함수 ② Group Row 함수 ③ Null 무시함수

## ● 2-7  NULL

**① 연산불가(NULL 값 연산)**
SELECT 300+400, 300+NULL, 300/NULL FROM dual;                    // NULL 연산 결과는?
SELECT ENAME,SAL,COMM,COMM + SAL*0.3 as bonus FROM EMP;    // 실수하기 쉬운...해결책은....?

**② 비교불가(NULL 값 비교)**
SELECT ENAME,SAL,COMM FROM EMP WHERE SAL  > 1000;            // null이 없는 column과 비교
SELECT ENAME,SAL,COMM FROM EMP WHERE COMM > -1;              // null이 있는 column과 비교
SELECT ENAME,SAL,COMM FROM EMP WHERE COMM = null;

SELECT ENAME,SAL,COMM FROM EMP WHERE COMM is null;
SELECT ENAME,SAL,COMM FROM EMP WHERE COMM is not null;

**③ 적용불가(NULL값 함수에 적용)**
SELECT ENAME,length(ENAME),COMM,length(COMM) FROM EMP;.

**④ NULL 제어 함수**
SELECT  nvl(COMM,-1), decode(COMM,null,-999,COMM)  FROM EMP

[과제]  커미션(COMM)을 받는 사원수, 커미션 평균을 조회 하십시요

# 2-8 DECODE, CASE

① SELECT DEPTNO,ENAME,SAL,
              DECODE(DEPTNO,10,0.1,20,0.2,30,0.3,0.5) * SAL AS BONUS
   FROM EMP ORDER BY DEPTNO;

② SELECT DEPTNO,ENAME,SAL,
              DECODE(GREATEST(SAL,5000),SAL,'HIGH',
                       DECODE(GREATEST(SAL,2500),SAL,'MID','LOW'))
   FROM EMP
   ORDER BY DEPTNO;                                   -- SAL 대신 COMM으로 대체해서 실습해볼것

③ SELECT DEPTNO,ENAME,SAL,
                    CASE      WHEN SAL >= 5000 THEN 'HIGH'
                              WHEN SAL >= 2500 THEN 'MID'
                              WHEN SAL < 2500 THEN 'LOW'
                              ELSE
                                    'UNKNOWN'
                    END
   FROM EMP
   ORDER BY DEPTNO;

④ **NULL 제어 함수**
   SELECT nvl(COMM,-1), decode(COMM,null,-999,COMM) FROM EMP

## ● 2-8 DECODE, CASE

```
→ 노가다 SQL Coding
if($con_faxno == "c" && $con_telno == "p") {
  $loofInsertSql = "insert into $addr_u (ADDRESS_SEQ,USERID,NAME,EMAIL,".
            " P_HTELNO,P_TELNO,C_TELNO,P_FAXNO,C_FAXNO,CON_TELNO, CON_FAXNO)".
            " value (ADDRESS_SEQ.NEXTVAL,'$addid','$name','$email',".
            " $htelno,$telno,$faxno)";    }
  if($con_faxno == "p" && $con_telno == "c") {
  $loofInsertSql = "insert into $addr_u (ADDRESS_SEQ,USERID,NAME,EMAIL,".
            " P_HTELNO,P_FAXNO,C_TELNO,CON_TELNO, CON_FAXNO)".
            " value (ADDRESS_SEQ.NEXTVAL,'$addid','$name','$email',".
            " $htelno,$telno,$faxno)";  }
  if($con_faxno == "c" && $con_telno == "c") {
  $loofInsertSql = "insert into $addr_u (ADDRESS_SEQ,USERID,NAME,EMAIL,".
            " P_HTELNO,C_TELNO,C_FAXNO,CON_TELNO, CON_FAXNO)".
            " value (ADDRESS_SEQ.NEXTVAL,'$addid','$name','$email',".
            " $htelno,$telno,$faxno)";  }
  if($con_faxno == "p" && $con_telno == "p") {
  $loofInsertSql = "insert into $addr_u (ADDRESS_SEQ,USERID,NAME,EMAIL,".
            " P_HTELNO,P_TELNO,P_FAXNO,CON_TELNO, CON_FAXNO)".
            " value (ADDRESS_SEQ.NEXTVAL,'$addid','$name','$email',".
            " $htelno,$telno,$faxno,'$','$con_faxno')";  }
  echo $loofInsertSql;
  $db->execute($loofInsertSql);
}
```

```
→ Decode를 사용한 SQL coding
$loofInsertSql = "insert into $addr_u (ADDRESS_SEQ,USERID,NAME,EMAIL,".
            " P_HTELNO,P_TELNO,C_TELNO,P_FAXNO,C_FAXNO,CON_TELNO, CON_FAXNO)".
            " values (ADDRESS_SEQ.NEXTVAL,'$addid','$name','$email',".
            " '$htelno',decode('$con_telno','p',decode('$telno','N',null,'$telno'),null),".
            " decode('$con_telno','c',decode('$telno','N',null,'$telno'),null),".
            " decode('$con_faxno','p',decode('$telno','N',null,'$faxno'),null),".
            "
decode('$con_faxno','c',decode('$telno','N',null,'$faxno'),null),'$con_telno','$con_faxno')";
```

16

# ● 2-9 ROWNUM

■ ROWNUM

① SELECT ROWNUM,ENAME,DEPTNO,SAL FROM EMP;

② SELECT ROWNUM,ENAME,DEPTNO,SAL FROM EMP ORDER BY DEPTNO,SAL;

③ SELECT ROWNUM,ENAME,DEPTNO,SAL FROM EMP WHERE  DEPTNO IN (10,20) ORDER BY DEPTNO,SAL;
 ** 실행순서 ROWNUM or ORDER BY?

④ SELECT ENAME,DEPTNO,SAL FROM EMP WHERE ROWNUM = 1;
⑤ SELECT ENAME,DEPTNO,SAL FROM EMP WHERE ROWNUM = 5;
⑥ SELECT ENAME,DEPTNO,SAL FROM EMP WHERE ROWNUM > 5;

⑦ SELECT ENAME,DEPTNO,SAL FROM EMP WHERE ROWNUM <= 5;
⑧ SELECT ENAME,DEPTNO,SAL FROM EMP WHERE ROWNUM < 5;

과제
      - SubQuery를 공부한후 최상위 급여자 5명을 조회하는 SQL문을 작성하시요

# ● 2-10 논리연산자 AND OR NOT

① SELECT ENAME,JOB,SAL,DEPTNO FROM EMP WHERE DEPTNO = 10 AND SAL > 2000;
② SELECT ENAME,JOB,SAL,DEPTNO FROM EMP WHERE DEPTNO = 10 OR  SAL > 2000;
③ SELECT ENAME,JOB,SAL,DEPTNO FROM EMP WHERE SAL > 2000  OR  SAL > 2000;
④ SELECT ENAME,JOB,SAL,DEPTNO FROM EMP
    WHERE  DEPTNO = 10 AND SAL > 2000 OR JOB = 'CLERK';
⑤ SELECT ENAME,JOB,SAL,DEPTNO FROM EMP
    WHERE  (DEPTNO = 10 AND SAL > 2000) OR JOB = 'CLERK';
⑥ SELECT ENAME,JOB,SAL,DEPTNO FROM EMP
    WHERE  DEPTNO = 10 AND  (SAL > 2000 OR JOB = 'CLERK');

⑦ SELECT  EMPNO,ENAME, JOB,SAL,DEPTNO  FROM EMP  WHERE  DEPTNO = 10 AND SAL > 2000;
⑧ SELECT ENAME,JOB,SAL FROM EMP WHERE SAL > 2000;
⑨ SELECT ENAME,JOB,SAL FROM EMP WHERE JOB  != 'CLERK';
⑩ SELECT ENAME,JOB,SAL FROM EMP WHERE JOB NOT IN('CLERK','MANAGER');

⑪ SELECT ENAME,JOB,SAL FROM EMP WHERE SAL > 2000  OR JOB = 'CLERK';
⑫ SELECT ENAME,JOB,SAL FROM EMP WHERE SAL > 2000  AND JOB = 'CLERK';
⑬ SELECT ENAME,JOB,SAL FROM EMP WHERE SAL > 2000  OR SAL  > 2000;

⑭ SELECT ENAME,JOB,SAL FROM EMP WHERE SAL > 2000 or JOB = 'CLERK' and DEPTNO >= 20;   -- 우선순위?
⑮ SELECT ENAME,JOB,SAL FROM EMP WHERE SAL > 2000 or ( JOB = 'CLERK' and DEPTNO >= 20 );
    SELECT ENAME,JOB,SAL  FROM EMP WHERE ( SAL > 2000 or JOB = 'CLERK') and DEPTNO >= 20;

# 2-11 SQL 연산자

■ BETWEEN
① SELECT EMPNO,ENAME,SAL FROM EMP WHERE SAL between 1000 and 2000;          -- 숫자,범위
② SELECT EMPNO,ENAME,SAL FROM EMP WHERE SAL >= 1000 and SAL <= 2000;          -- 차이점?
③ SELECT EMPNO,ENAME,HIREDATE,SAL FROM EMP WHERE SAL between 2000 and 1000;        -- 이유는?

④ SELECT EMPNO,ENAME,HIREDATE,SAL FROM EMP WHERE ENAME  between 'C' and 'K';          -- 문자
⑤ SELECT EMPNO,HIREDATE,SAL FROM EMP WHERE HIREDATE between '81/02/20' and '82/12/09';  -- 날짜

⑥ SELECT ENAME,HIREDATE FROM EMP
   WHERE HIREDATE between to_date('81/02/20','yy/mm/dd') and to_date('82/12/09','yy/mm/dd');

■ IN
⑦ SELECT EMPNO,ENAME,JOB FROM EMP WHERE EMPNO in (7369,7521,7654);                -- 숫자
⑧ SELECT EMPNO,ENAME,JOB FROM EMP WHERE EMPNO = 7369 or EMPNO = 7521 or EMPNO=7654;

⑨ SELECT EMPNO,ENAME,JOB FROM EMP WHERE JOB in ('clerk','manager');                -- 문자
⑩ SELECT EMPNO,ENAME,HIREDATE FROM EMP WHERE HIREDATE in ('81/05/01','81/02/20')    -- 날짜
⑪ SELECT EMPNO,ENAME,JOB,DEPTNO FROM EMP
   WHERE (JOB,DEPTNO) in (('MANAGER',20),('CLERK',20));

# ● 2-11 SQL 연산자

■ LIKE
정의: 문자 패턴 매칭 연산자, 정확한 값을 몰라도 찾을수 있는  EX) 김서방 찾기 !!
　　　% : 0개 이상의 모든문자
　　　_ : 1개의 모든문자, 위치가의미를  가짐.

① SELECT ENAME FROM EMP WHERE ENAME like 'A%';　　　　　　　　　　-- pattern matching
② SELECT ENAME FROM EMP WHERE ENAME like '_A%';
③ SELECT ENAME FROM EMP WHERE ENAME like '%L%E%';
④ SELECT ENAME FROM EMP WHERE ENAME like '%LE%';
⑤ SELECT ENAME FROM EMP WHERE ENAME like '%A%';
⑥ SELECT ENAME FROM EMP WHERE ENAME NOT like '%A%';

⑦ SELECT ENAME,HIREDATE FROM EMP WHERE HIREDATE like '81%';　　　--? 날짜
⑧ SELECT ENAME,SAL FROM EMP WHERE SAL like 2%;　　　　　　　　-- ? 숫자
⑨ SELECT ENAME,SAL FROM EMP WHERE SAL like '2%';　　　　　　　-- ? 숫자 like 문자

# ● 2-12 함수

◼ **ORACLE DEFINED FUNCTION**

　　　① **SINGLE ROW FUNCTION**
　　　　　　**문자함수**
　　　　　　**숫자함수**
　　　　　　**날짜함수**
　　　　　　**변환함수(DATA TYPE CONVERSION)**
　　　　　　**기타함수**

　　　② **GROUP ROW FUNCTION**

◼ **USER DEFINED FUNCTION (by PL/SQL)**

## ● 2-12 함수

■ SINGLE ROW FUNCTION

<mark>Single Row function</mark> 정의    : <mark>1개의 row에 적용되고 1row당 1개의 결과를 return한다</mark>.

　　　　　사용되는 위치  : select list , where, order by,group by

① SELECT ENAME,EMPNO,SAL,COMM FROM EMP;                                  -- 14건
② SELECT ENAME,LOWER(ENAME),UPPER(LOWER(ENAME)),LENGTH(ENAME),    -- 14건
　　　ABS(SAL-EMPNO),COMM
　 FROM EMP;

③ SELECT ENAME,substr(ENAME,1,3) FROM EMP
　 WHERE  HIREDATE between to_date('81/01/01','RR/MM/DD') and to_date('82/12/31','RR/MM/DD')
　 ORDER  BY by length(ENAME);

■ GROUP ROW FUNCTION

<mark>Group Row function</mark> 정의: <mark>N(여러개) 개의 row에 적용되고 그룹당 1개의 결과를 return한다</mark>.

④ SELECT AVG(SAL),SUM(SAL),SUM(COMM),COUNT(*) FROM EMP;                -- 1건

⑤ SELECT DEPTNO,COUNT(*),SUM(SAL),AVG(SAL)   FROM EMP                  -- GROUPING
　 GROUP   BY DEPTNO;

⑥ SELECT DEPTNO,JOB,COUNT(*),SUM(SAL),AVG(SAL) FROM EMP
　 GROUP   BY DEPTNO,JOB;

● **2-12 함수 – 단일행 함수**

■ 문자 함수
 ① SELECT ENAME, lower(ENAME) ,upper(ENAME), initcap(ENAME) FROM EMP;

 ② SELECT ENAME, substr(ENAME,1,3), substr(ENAME,4), substr(ENAME,-3,2) FROM EMP;

 ③ SELECT ENAME, instr(ENAME,'A'), instr(ENAME,'A',2), instr(ENAME,'A',1,2) FROM EMP;    --문자위치

 ④ SELECT ENAME, rpad(ENAME,10,'*'), lpad(ENAME,10,'+') FROM EMP;

 ⑤ SELECT ENAME,REPLACE(ENAME,'S','s') FROM EMP;

 ⑥ SELECT ENAME, concat(ENAME,JOB), ENAME||JOB FROM EMP;

 ⑦ SELECT ltrim(' 대한민국 '), rtrim(' 대한민국 '), trim(' 대한민국 ') FROM dual;

 ⑧ SELECT length('abcd'), substr('abcd',2,2), length('대한민국'), substr('대한민국',2,2) FROM dual;

 ⑨ SELECT lengthb('abcd'), substrb('abcd',2,2), lengthb('대한민국'), substrb('대한민국',2,2) FROM dual;

 ⑩ SELECT length('abcd'), vsize('abcd'), length('대한민국'), vsize('대한민국') FROM dual;

# 2-12 함수 – 단일행 함수

■ 숫자 함수
① SELECT round(45.923,2), round(45.923,1), round(45.923,0), round(45.923), round(45.923,-1) FROM  dual;
② SELECT trunc(45.923,2), trunc(45.923,1), trunc(45.923,0), trunc(45.923), trunc(45.923,-1)  FROM  dual;
③ SELECT mod(100,3), mod(100,2) FROM dual;                                    -- 나머지
④ SELECT ENAME,SAL,SAL*0.053 as tax,round(SAL*0.053,0) as r_tax FROM EMP;        --급여의5.3%세금,원단위
⑤ SELECT  CEIL(-45.594),CEIL(-45.294),CEIL(45.294),
                ROUND(-45.594),ROUND(-45.294),ROUND(45.594)          FROM DUAL;  -- 절대값연산
⑥ SELECT  FLOOR(45.245),FLOOR(-45.245),FLOOR(45.545),FLOOR(-45.545) FROM DUAL;


■ DATE type
DATE DATA TYPE은 연산이 가능하다.
⑦ SELECT sysdate,sysdate + 7, sysdate –2, sysdate + 1/24 FROM dual;                          -- sysdate + 1/24 ?
⑧ SELECT deptno,ename, trunc(sysdate – hiredate) as work_day FROM emp ORDER BY deptno,work_day DESC;


  DATE는 날짜와 시간 정보를 가지고 있다.
⑨ SELECT ename,sysdate,hiredate FROM   emp;                        -- 왜 날짜만 보이고 시간은 보이지 않는가?


⑩ SELECT ename,to_char(sysdate,'YYYY-MM-DD:HH24:MI:SS'),to_char(hiredate,'YYYY-MM-DD:HH24:MI:SS')
   FROM   emp;


⑪ ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY-MM-DD:HH24:MI:SS';
   SELECT ename,sysdate,hiredate FROM   emp;
   SELECT SYSDATE FROM DUAL;

# 2-12 함수 – 단일행 함수

■ 날짜 함수

① SELECT HIREDATE,months_between(sysdate,HIREDATE),months_between(HIREDATE,sysdate) FROM EMP;
  -- 찌꺼기 일자 처리: trunc(months_between(sysdate,HIREDATE))

② SELECT sysdate, add_months(sysdate,3), add_months(sysdate,-1) FROM dual;

③ SELECT sysdate, last_day(sysdate), next_day(sysdate,'일요일'), next_day(sysdate,2) FROM dual;

④ SELECT sysdate,round(sysdate,'YEAR'),round(sysdate,'MONTH'),round(sysdate,'DAY'),round(sysdate)
  FROM   dual;
  -- 06월30일전/후 , 15일전/후 ,수요일 전/후 ,  전은 = 의 의미를 포함하고 있다.
  -- 11시59분 일때는 오늘 12:00은 내일...

⑤ SELECT sysdate,trunc(sysdate,'YEAR'),trunc(sysdate,'MONTH'),trunc(sysdate,'DAY'),trunc(sysdate)
  FROM   dual;

⑥ SELECT
        to_char(sysdate,'MM"월"DD"일" ') as mmdd1,
        to_char(sysdate,'MM')||'월'||to_char(sysdate,'DD')||'일' as mmdd2
  FROM dual;

⑦ SELECT
        EXTRACT(YEAR FROM SYSDATE),  EXTRACT(MONTH FROM SYSDATE),EXTRACT(DAY FROM SYSDATE)
  FROM DUAL;

■ 변환 함수
　명시적,암시적(성능문제 & 오류 발생가능성)
　FORMAT IS CASE SENSITIVE

① SELECT  SYSDATE,  TO_CHAR(SYSDATE,'YEAR'),TO_CHAR(SYSDATE,'Year'),
　　　　　　　　　 TO_CHAR(SYSDATE,'YYYY'),TO_CHAR(SYSDATE,'YY')   FROM DUAL;

② SELECT  TO_CHAR(SYSDATE,'MONTH'),TO_CHAR(SYSDATE,'MON'),　　　　　 -- MONTH:9자, MON:3자
　　　　 TO_CHAR(SYSDATE,'Mon'),TO_CHAR(SYSDATE,'mon'),
　　　　　TO_CHAR(SYSDATE,'MM'),TO_CHAR(SYSDATE,'mm')  FROM DUAL;

③ SELECT SYSDATE,
　　　　 TO_CHAR(SYSDATE,'DAY'), TO_CHAR(SYSDATE,'Day'),TO_CHAR(SYSDATE,'DY'), TO_CHAR(SYSDATE,'dy'),
　　　　 TO_CHAR(SYSDATE,'DD'),TO_CHAR(SYSDATE,'dd')   FROM  DUAL;　　 -- ??

④ SELECT 123456, TO_CHAR(123456,'999999'), LENGTH(TO_CHAR(123456,'999999')),
　　　　 LENGTH(TO_CHAR(123456,'fm999999'))　　　　　　　　 -- fm: remove padded blanks or
　 FROM  DUAL;

⑤ SELECT TO_CHAR(12345*123.45,'999,999.99'),TO_CHAR(12345*123.45,'99,999,999.99') FROM DUAL; --??
⑥ SELECT  TO_CHAR(SAL,'$999,999'),　　　　 REPLACE(TO_CHAR(SAL,'$999,999'),' ','?'),
　　　　 TO_CHAR(SAL,'L999,999'),　　　　 TO_CHAR(SAL,'999,999L'),　　　　 TO_CHAR(SAL,'fm999,999L')
　 FROM EMP;

① SELECT MIN(ENAME),MAX(ENAME),MIN(SAL),MAX(SAL),MIN(HIREDATE),MAX(HIREDATE) FROM   EMP;
   -- ENAME:문자, SAL:숫자,HIREDATE:날짜

② SELECT   COUNT(*),COUNT(EMPNO),COUNT(MGR),COUNT(COMM)   FROM   EMP;

③ SELECT   COUNT(JOB),COUNT(ALL JOB),COUNT(DISTINCT JOB),SUM(SAL),SUM(DISTINCT SAL)
   FROM     EMP;

④ SELECT   COUNT(*), SUM(COMM), SUM(COMM)/COUNT(*),AVG(COMM),SUM(COMM)/COUNT(COMM)
   FROM     EMP;

⑤ 그룹행함수 와 NULL  그리고 NVL , 효율적인 계산방안은 ?
   SELECT   SAL,COMM   FROM EMP;
   SELECT   SUM(NVL(COMM,0)) AS SUM_COMM1,
            SUM(COMM)             AS SUM_COMM2,
            NVL(SUM(COMM),0) AS SUM_COMM3
   FROM EMP