

★ 용어

- ① TABLE:관계형 데이터베이스에서 데이터 저장하는 기본적인 구조.  
즉 데이터를 저장하는 그릇 역할.  
테이블은 열(COLUMN)과 행(ROW)로 구성된 2차원 구조
- ② ROW(=Record)
- ③ COLUMN(=Field, Attribute)
- ④ COLUMN HEADING

EMPNO	ENAME	JOB	DEPTNO	HIREDATE	SAL
7369	SMITH	CLERK	20	1980-12-17	800
7499	ALLEN	SALESMAN	30	1981-02-20	1600
7521	WARD	SALESMAN	30	1981-02-22	1250
7566	JONES	MANAGER	20	1981-04-02	2975
7654	MARTIN	SALESMAN	30	1981-09-28	1250
7698	BLAKE	MANAGER	30	1981-05-01	2850
7782	CLARK	MANAGER	10	1981-06-09	2450
7788	SCOTT	ANALYST	20	1987-04-19	3000
7839	KING	PRESIDENT	10	1981-11-17	5000
7844	TURNER	SALESMAN	30	1981-09-08	1500
7876	ADAMS	CLERK	20	1987-05-23	1100
7900	JAMES	CLERK	30	1981-12-03	950
7902	FORD	ANALYST	20	1981-12-03	3000
7934	MILLER	CLERK	10	1982-01-23	1300

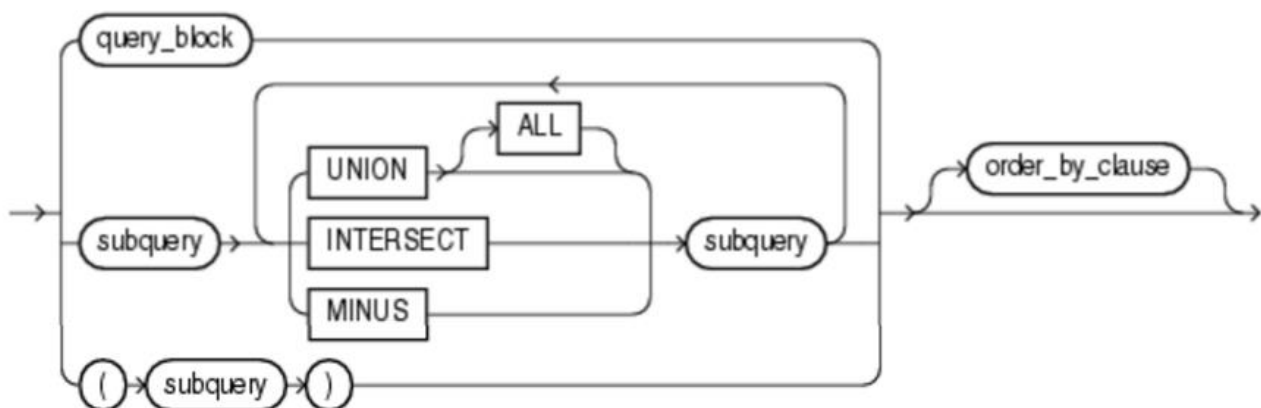
	A	B	C	D	E	F	G	H
	EMPNO	ENAME	JOB	DEPTNO	HIREDATE	SAL	COMM	DEPTNO
1								
2	7369	SMITH	CLERK	7902	1980-12-17	800		20
3	7499	ALLEN	SALESMAN	7698	1981-02-20	1600	300	30
4	7521	WARD	SALESMAN	7698	1981-02-22	1250	500	30
5	7566	JONES	MANAGER	7839	1981-04-02	2975		20
6	7654	MARTIN	SALESMAN	7698	1981-09-28	1250	1400	30
7	7698	BLAKE	MANAGER	7839	1981-05-01	2850		30
8	7782	CLARK	MANAGER	7839	1981-06-09	2450		10
9	7788	SCOTT	ANALYST	7566	1987-04-19	3000		20
10	7839	KING	PRESIDENT		1981-11-17	5000		10
11	7844	TURNER	SALESMAN	7698	1981-09-08	1500	0	30
12	7876	ADAMS	CLERK	7788	1987-05-23	1100		20
13	7900	JAMES	CLERK	7698	1981-12-03	950		30
14	7902	FORD	ANALYST	7566	1981-12-03	3000		20
15	7934	MILLER	CLERK	7782	1982-01-23	1300		10

- SELECT SYNTAX Diagram

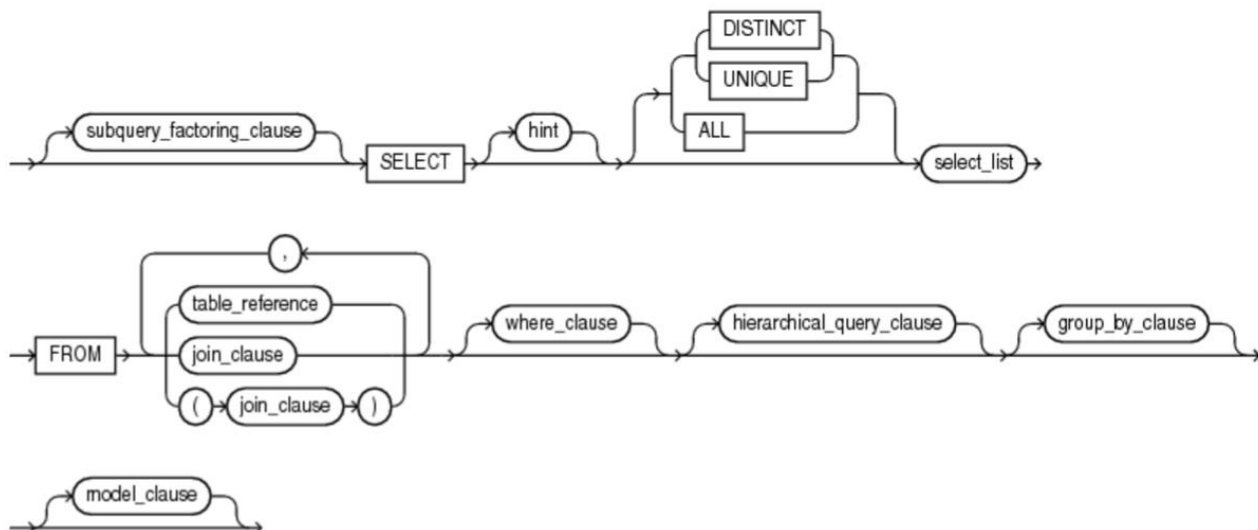
***select::=***



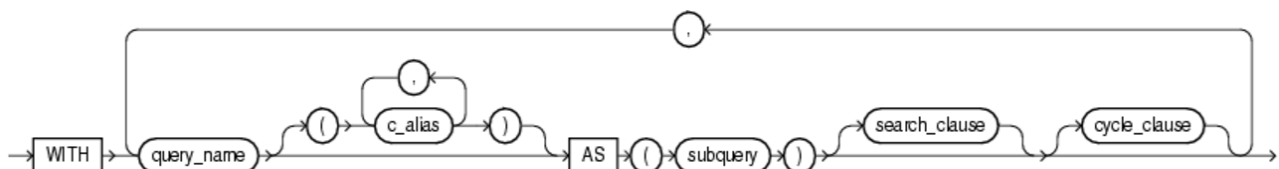
***subquery::=***



***query\_block::=***



***subquery\_factoring\_clause::=***



SELECT

- ① SELECT \* FROM EMP;  
 ② SELECT EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO FROM EMP;

\* 8개 COLUMN, 전체 ROW  
 좋은 방식의 SQL문은?  
 컬럼 구분자 ,(콤마)

\* COLMN HEADING의 정렬 방향 : 문자와날짜 ~ \_\_\_\_\_정렬,  
 숫자 ~ \_\_\_\_\_정렬 ,  
 기본적으로 대문자로 표기

[질문] 컬럼의 데이터의 정렬 방향은 ?

SELECT LIST

[설명] SELECT LIST 의미: SELECT 와 FROM사이를 SELECT LIST 라고 명명.  
 역할: 원하는 COLUMN(즉 속성)만 조회(Projection)

- 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 데이터유형  
 존재하지 않는 컬럼을 만들어서 조회 가능

WHERE 조건절

[설명] WHERE 의미:조건절  
 역할:원하는 ROW(레코드)만 조회(Selection)

[설명] 비교연산자 종류 : = , > , < , >= , <= , <> , != , ^=  
 사칙(산술) 연산자 종류 : \*, / , + , -

- ⑦ SELECT \* FROM EMP WHERE DEPTNO = 10;  
 ⑧ SELECT EMPNO,ENAME,JOB,SAL FROM EMP WHERE SAL > 2000;  
 ⑨ SELECT EMPNO ,ENAME,JOB,SAL FROM EMP WHERE SAL > 2000;  
 ⑩ SELECT EMPNO,ENAME, JOB,SAL,DEPTNO FROM EMP WHERE DEPTNO = 10 AND SAL > 2000;  
 ⑪ SELECT DEPTNO,ENAME,JOB FROM EMP WHERE JOB = 'manager';  
 ⑫ SELECT DEPTNO,ENAME,JOB FROM EMP WHERE 1=1;  
 ⑬ SELECT DEPTNO,ENAME,JOB FROM EMP WHERE 1=2;  
 ⑭ SELECT ENAME, SAL, SAL\*12,COMM,COMM+300 FROM EMP;  
 ⑮ SELECT SAL, SAL+300\*12, (SAL+300)\*12 FROM EMP; // 연산자 우선순위?

[질문] SQL 문장안에서 사칙연산( \*, /, +, - )의 의미는?  
 CLIENT/SERVER적 관점에서는?  
 연산자 우선순위 ?  
 ( ) > NOT > 비교연산자 > SQL연산자 > AND > OR > \*, /, +, -

[요구] NUMBER 와 DATE 자료형에 사칙연산 적용 하는 SQL 예제 작성  
 SELECT List 와 WHERE절에 사칙 연산 사용할수 있는 SQL예제 작성

## COLUMN ALIAS

[설명] Alias 종류 : Table Alias, Column Alias

Table Alias 용도: 일반적으로는 SELF JOIN 시에 사용  
 사용방법: ① 공백문자 ex) SELECT E.EMPNO,E.ENAME FROM EMP E;

Column Alias 용도: Column Heading(컬럼 레이블 변경)을 의미 있는 다른 이름으로 재정의  
 ① COLUMN의 의미 명료성 ②DB 프로그램개발시 Code의 명료성(계산식에서 유용)

사용방법:

- ① 공백문자
- ② AS or as (ANSI 표준)
- ③ “ ” :특수문자, 공백문자, 대소문자 구분이 필요한 경우

- ① SELECT ENAME,SAL+200 bonus,SAL\*12 as annual\_SAL, COMM, COMM+300 "Special Bonus" FROM EMP;
- ② SELECT ENAME, COMM+300 AS "Special Bonus" FROM EMP;
- ③ SELECT ENAME, COMM+300 보너스 FROM EMP;

## (문자열) 결합(합성) 연산자

[설명] 문자열 합성(결합) 연산자

- ④ SELECT ENAME, JOB FROM EMP; // column의 개수, heading은?
- ⑤ SELECT ENAME||JOB FROM EMP; // column의 개수, heading은? 용도는?
- ⑥ .....
- ⑦SELECT DNAME||' 부서는 '||LOC||' 지역에 위치합니다.' as LOC FROM DEPT;
- ⑧SELECT ENAME||' is a '||JOB FROM EMP;
- ⑨SELECT ENAME||' 's JOB is '||JOB FROM EMP; // ex) scott's JOB is clerk
- ⑩SELECT ENAME||' 's JOB is '||JOB as JOB\_list FROM EMP;
- ⑪SELECT EMPNO,ENAME,HIREDATE FROM EMP; // Data type:숫자,문자,날짜
- ⑫SELECT ENAME,SAL,SAL\*100, SAL||'00',to\_char(SAL)||'00' FROM EMP; // Data type Conversion  
 // Implicit vs Explicit Conversion  
 // 장,단점?..

[요구] EMP 테이블에서 사번,이름,직무,급여,부서번호 데이터 사이에 구분자 , 를 삽입하는 예제 SQL 작성  
 SELECT \* FROM TAB ; 의 용도를 찾아 설명 하십시오

## DUAL

[설명] Dummy Table

SYS user 소유, 1개의 Row 와 1개의 Column을 가진 작고 가벼운 테이블,

[용도] 실제 테이블로부터 데이터를 가져오는 것이 아닌 function 또는 calculation을 수행하기 위한 테이블

① Desc dual

DESCRIBE dual

② SELECT \* FROM dual;

③ SELECT sysdate FROM dual;

\* DBMS가 설치되어있는 SERVER의 현재 날짜와 시간을 return 하는 함수  
시간아는 방법: 114 , 손목시계

④ SELECT 143475\*127363, to\_char(143475\*127363,'999,999,999,999') FROM dual;

[요구] SYS 계정이란?

현재 시간,분,초,1/100초 까지 표현하는 SQL 작성

현재 시간,분,초,1/1000초 까지 표현하는 SQL 작성

## NULL

[정의] 사전적의미: 값이 정의되지 않은, 존재하지 않는 -> 현재 데이터를 입력하지 못한 경우

unavailable, unassigned, unknown,inapplicable

not the same as zero or a blank space

/\* ASCII 코드 0->48, ' ' ->32 , null-> 00 \*/

모든 데이터 유형에 null값이 적용된다.

[주의사항]

① 연산불가

② 비교불가

③ 적용불가

[함수와와의 관계] 1. Single Row함수 2. Group Row 함수 3. Null 무시함수

\*\* 연산불가(NULL 값 연산) \*\*

⑤ SELECT 300+400, 300/0 FROM dual;

// divided by zero ?

⑥ SELECT 300+400, 300+NULL, 300/NULL FROM dual;

// NULL과 관계된 연산 결과는?

⑦ SELECT ENAME,SAL,COMM,COMM + SAL\*0.3 as bonus FROM EMP;

// 실수하기 쉬운...해결책은...?

\*\* 비교불가(NULL 값 비교) \*\*

⑧ SELECT EMPNO,SAL,COMM FROM EMP;

// COMM Column을 관찰

⑨ SELECT ENAME,SAL,COMM FROM EMP WHERE SAL > 1000;

// null이 없는 column과 비교

⑩ SELECT ENAME,SAL,COMM FROM EMP WHERE COMM > -1;

// null이 있는 column과 비교

⑪ SELECT ENAME,SAL,COMM FROM EMP WHERE COMM = null;

⑫ SELECT ENAME,SAL,COMM FROM EMP WHERE COMM <> null;

// !=, ^=, <>, not =

⑬ SELECT ENAME,SAL,COMM FROM EMP WHERE COMM is null;

⑭ SELECT ENAME,SAL,COMM FROM EMP WHERE COMM is not null;

\*\* 적용불가(NULL값 함수에 적용)

- ① SELECT ENAME,length(ENAME),COMM,length(COMM) FROM EMP;.
- ② SELECT SAL-EMPNO,abs(SAL-EMPNO),abs(SAL-COMM)+100 FROM EMP;

\*\* NULL무시하는 함수들

- ③ SELECT concat(ENAME||' is ',COMM),nvl(COMM,-1),decode(COMM,null,-999,COMM) FROM EMP

[요구] ⑦번 SQL의 문제점을 해결 하는 SQL작성 하십시오

아래 5개의 SQL에 대해 설명 하십시오

```
SELECT COUNT(*) FROM SCOTT.EMP;

SELECT COUNT(*) FROM SCOTT.EMP WHERE 1 = 1;

SELECT COUNT(*) FROM SCOTT.EMP WHERE EMPNO = EMPNO;

SELECT COUNT(*) FROM SCOTT.EMP WHERE MGR = MGR;

SELECT COUNT(*) FROM SCOTT.EMP WHERE COMM = COMM;
```

아래 1개의 SQL에 대해 설명 하십시오

```
SELECT SAL,COMM,NVL(COMM,SAL),nvl2(COMM,SAL,0), NULLIF(JOB,'MANAGER') FROM emp;
```

ORDER BY

[정의]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)

- ④ SELECT ENAME,HIREDATE FROM EMP;
- ⑤ SELECT ENAME,HIREDATE FROM EMP ORDER BY ENAME ; // 정렬방향?
- ⑥ SELECT ENAME,HIREDATE FROM EMP ORDER BY ENAME asc; // SQL문의 가독성
- ⑦ SELECT ENAME,HIREDATE FROM EMP ORDER BY ENAME desc;
- ⑨ SELECT ENAME,SAL,HIREDATE FROM EMP ORDER BY ENAME; // column name
- ⑩ SELECT ENAME,SAL,HIREDATE FROM EMP ORDER BY 2; // column position
- ⑪ SELECT ENAME,SAL\*12 as annual\_SAL FROM EMP ORDER BY annual\_SAL; // column alias
- ⑫ SELECT EMPNO,ENAME,COMM,JOB FROM EMP ORDER BY COMM \* 12; // expression
- ⑬ SELECT DEPTNO,JOB,ENAME FROM EMP ORDER BY DEPTNO;
- ⑭ SELECT DEPTNO,JOB,ENAME FROM EMP ORDER BY DEPTNO,JOB; // 조합의 순서쌍
- ⑮ SELECT DEPTNO,JOB,ENAME FROM EMP ORDER BY DEPTNO,JOB desc; // 차이점은?

[요구] 아래의 SQL은 선택하지 않은 column을 기준으로 정렬이 가능한가?

```
SELECT ENAME,HIREDATE FROM EMP ORDER BY SAL desc;
```

ORDER BY시 NULL의 위치는? NULL은 가장 큰값인가?

```
SELECT EMPNO,COMM FROM EMP ORDER BY COMM asc;
SELECT EMPNO,COMM FROM EMP ORDER BY COMM desc;
```

DISTINCT

[용도] 중복제거 (??)

- |  |  |
|--|--|
| <p>① SELECT JOB FROM EMP;<br/>SELECT ALL JOB FROM EMP;</p> <p>② SELECT UNIQUE JOB FROM EMP;<br/>③ SELECT DISTINCT JOB FROM EMP;</p> <p>④ SELECT DISTINCT DEPTNO, JOB FROM EMP;<br/>⑤ SELECT JOB FROM EMP ORDER BY JOB;</p> <p>⑥ SELECT DISTINCT JOB, DISTINCT DEPTNO FROM EMP;<br/>⑦ SELECT JOB, DISTINCT DEPTNO FROM EMP;</p> <p>** NULL과 DISTINCT</p> <p>⑧ SELECT COMM FROM EMP WHERE COMM IS NOT NULL;<br/>SELECT DISTINCT COMM FROM EMP;</p> <p>⑨ [SQL*PLUS 에서 실습]<br/>SELECT DISTINCT COMM FROM EMP;<br/>SET FEEDBACK ON<br/>L or I<br/>/</p> | <p>// Row 개수 만큼,<br/>// [ALL]</p> <p>// 직업(직군) 종류<br/>// 차이점? ANSI ?</p> <p>// 중복 데이터(?)<br/>// distinct의 원리</p> <p>// 범위?<br/>// 위치?</p> <p>// 4 ROWS 인가 ?<br/>// 5 ROWS ??</p> <p>// 5 ROWS ??<br/>// SQL*PLUS 명령어<br/>// L은 축약된?<br/>// Run or Send ?</p> |
|--|--|

[요구]

- ③ 실행 결과가 Oracle 9i 에서는 정렬되어 나타나지만 Oracle 10g이후 버전부터는 정렬된 결과가 나타나지 않는다. 이유를 찾아 설명 하십시오

다음의 집합 연산자 UNION, UNION ALL, INTERSECT, MINUS를 공부한후 각각의 예제 SQL을 만든후 결과가 왜 정렬되는지 설명 하십시오.

- ⑥⑦ SQL이 왜 에러가 나는지 2페이지의 Select Syntax Diagram을 보고 설명 하십시오

요구사항.....
캔!!!!!!!!!!

Interactive SQL 과 Embded SQL를 설명 하고 각각의 사용예를 찾아서 기록하고 해석 하십시오

그x의 PHP 노가다 코딩

No	코딩 사례
①	<pre> if(\$con_faxno == "c" &amp;&amp; \$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,\$stelno,\$faxno)"; }  if(\$con_faxno == "p" &amp;&amp; \$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,\$stelno,\$faxno)"; }  if(\$con_faxno == "c" &amp;&amp; \$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,\$stelno,\$faxno)"; }  if(\$con_faxno == "p" &amp;&amp; \$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,\$stelno,\$faxno','\$','\$con_faxno')"; }  echo \$loofInsertSql;  \$db-&gt;execute(\$loofInsertSql); } </pre>
②	<pre> \$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('\$stelno','N',null,'\$stelno'),null),".     " decode('\$con_telno','c',decode('\$stelno','N',null,'\$stelno'),null),".     " decode('\$con_faxno','p',decode('\$stelno','N',null,'\$faxno'),null),".     " decode('\$con_faxno','c',decode('\$stelno','N',null,'\$faxno'),null),'\$con_telno',     '\$con_faxno')"; </pre>



DECODE, CASE
--------------

[정의] 조건절(IF) 연산자

- ① SELECT DEPTNO, ENAME, DECODE(DEPTNO,10,'ACCOUNTING',20,'RESEARCH',30,'SALES','ETC')  
FROM EMP  
ORDER BY DEPTNO;
- ② SELECT COMM, DECODE(COMM,NULL,-99,COMM) FROM EMP;
- ③ SELECT GREATEST(3000,1500,2100,5000),LEAST(3000,1500,2100,5000) FROM DUAL ;  
  
SELECT DEPTNO, ENAME, SAL,  
DECODE(GREATEST(SAL,5000),SAL,'HIGH',  
DECODE(GREATEST(SAL,2500),SAL,'MID','LOW'))  
FROM EMP  
ORDER BY DEPTNO;
- ④ SELECT DEPTNO, ENAME, SAL,COMM, // NULL ??  
DECODE(GREATEST(COMM,5000),COMM,'HIGH',  
DECODE(GREATEST(COMM,2500),COMM,'MID','LOW'))  
FROM EMP  
ORDER BY DEPTNO;
- ⑤ SELECT DEPTNO, ENAME, SAL,  
CASE WHEN SAL >= 5000 THEN 'HIGH'  
WHEN SAL >= 2500 THEN 'MID'  
WHEN SAL < 2500 THEN 'LOW'  
ELSE  
'UNKNOWN'  
END // alias  
FROM EMP  
ORDER BY DEPTNO;  
  
SELECT DEPTNO, ENAME, SAL,  
CASE WHEN SAL >= 5000 THEN 'HIGH'  
WHEN SAL >= 2500 THEN 'MID'  
WHEN SAL BETWEEN 300 AND 2500 THEN 'LOW'  
ELSE  
'UNKNOWN'  
END  
FROM EMP  
ORDER BY DEPTNO;  
  
// ELSE에서 NULL ??  
SELECT DEPTNO, ENAME, SAL,COMM,  
CASE WHEN COMM >= 5000 THEN 'HIGH'  
WHEN COMM >= 2500 THEN 'MID'  
WHEN COMM BETWEEN 300 AND 2500 THEN 'LOW'  
ELSE  
'UNKNOWN'  
END  
FROM EMP  
ORDER BY DEPTNO;
- ⑥ SELECT DEPTNO,  
CASE DEPTNO  
WHEN 10 THEN 'Accounting'  
WHEN 20 THEN 'Research'  
WHEN 30 THEN 'Sales'  
WHEN 40 THEN 'Operations'  
ELSE 'Unknown'

```

        END as DNAME
FROM EMP
ORDER BY DEPTNO;

```

// 여러 개의 WHEN 절에서 조건이 중첩되는 경우 처리방식

```

⑦ SELECT SAL, CASE WHEN SAL >= 1000 THEN 1
                   WHEN SAL >= 2000 THEN 2
                   WHEN SAL >= 3000 THEN 3
                   WHEN SAL >= 4000 THEN 4
                   WHEN SAL >= 5000 THEN 5
                   ELSE 0
                   END AS SAL_CHK
FROM EMP
ORDER BY SAL;

```

```

⑧ SELECT SAL, CASE WHEN SAL >= 5000 THEN 5
                   WHEN SAL >= 4000 THEN 4
                   WHEN SAL >= 3000 THEN 3
                   WHEN SAL >= 2000 THEN 2
                   WHEN SAL >= 1000 THEN 1
                   ELSE 0
                   END AS SAL_CHK
FROM EMP
ORDER BY SAL;

```

```

⑨ NULLIF 함수 시뮬레이션
SELECT JOB,
       NULLIF(JOB,'MANAGER'),
       CASE
           WHEN JOB = 'MANAGER' THEN NULL
           ELSE JOB
       END AS NULLIF_SIM
FROM EMP;

```

[요구]

- \* 부서별 차등 보너스 계산 SQL을 작성 하십시오
  - 10번 부서 급여의 0.3% , 20번부서 급여의 20%, 30번 부서 급여의 10%, 나머지 모든 부서 1%
  - 부서 번호, 이름,직무,급여,보너스
  - 부서별, 최고 보너스 순으로 정렬
  - 소수점 절삭

\* DECODE 와 CASE이 차이점 정리후 발표

\*④ SQL에서 NULL값에 대한 처리를 하십시오

ROWNUM

// ROWNUM은 각각의 ROW에 부여된 절대적인 숫자인가?

- ① SELECT ROWNUM,ENAME,DEPTNO,SAL FROM EMP;  
 ② SELECT ROWNUM,ENAME,DEPTNO,SAL FROM EMP ORDER BY DEPTNO,SAL;

// 실행순서? SELECT , ROWNUM, WHERE ,ORDER BY

- ③ SELECT ROWNUM,ENAME,DEPTNO,SAL FROM EMP WHERE DEPTNO IN (10,20) ORDER BY DEPTNO,SAL;

// 조건절에서 ROWNUM 사용시 주의사항

- ④ SELECT ENAME,DEPTNO,SAL FROM EMP WHERE ROWNUM = 1; // O  
 SELECT ENAME,DEPTNO,SAL FROM EMP WHERE ROWNUM = 5; // X  
 SELECT ENAME,DEPTNO,SAL FROM EMP WHERE ROWNUM > 5; // X  
 SELECT ENAME,DEPTNO,SAL FROM EMP WHERE ROWNUM <= 5; // O  
 SELECT ENAME,DEPTNO,SAL FROM EMP WHERE ROWNUM < 5; // O

[요구] 캔!!!!

\* 아래의 SQL을 참조하여 최상위 급여자 5명을 조회하는 SQL문을 작성 하십시오

```
SELECT  DEPTNO,ENAME,SAL
FROM    EMP
WHERE   ROWNUM <= 5
ORDER  BY SAL DESC ;
```

\* Pseudo Column에 대한 정의를 설명하고 사용예제 SQL을 작성 하십시오

논리 연산자 (AND OR NOT)

// 논리 연산자를 통해 여러 조건 사용

- ⑤ SELECT ENAME,JOB,SAL,DEPTNO FROM EMP WHERE DEPTNO = 10 AND SAL > 2000;

- ⑥SELECT ENAME,JOB,SAL,DEPTNO FROM EMP WHERE DEPTNO = 10 OR SAL > 2000;  
 SELECT ENAME,JOB,SAL,DEPTNO FROM EMP WHERE SAL > 2000 OR SAL > 2000;  
 SELECT ENAME,JOB,SAL,DEPTNO FROM EMP WHERE SAL > 1000 OR SAL > 2000;

// 연산자 우선순위 , 좋은 방식의 SQL? , OPTIMIZER는 누구를 좋아 하는지

- ⑦SELECT ENAME,JOB,SAL,DEPTNO FROM EMP  
 WHERE DEPTNO = 10 AND SAL > 2000 OR JOB = 'CLERK';

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 );

- ⑧SELECT ENAME,JOB,SAL,DEPTNO FROM EMP  
 WHERE (DEPTNO = 10 AND SAL > 2000) OR JOB = 'CLERK';

SELECT ENAME,JOB,SAL,DEPTNO FROM EMP  
 WHERE DEPTNO = 10 AND (SAL > 2000 OR JOB = 'CLERK');

- ⑨SELECT ENAME,JOB,SAL FROM EMP WHERE JOB != 'CLERK';  
 SELECT ENAME,JOB,SAL FROM EMP WHERE JOB NOT IN('CLERK','MANAGER');

- ⑩SELECT ENAME,JOB,SAL FROM EMP WHERE SAL > 2000 OR JOB = 'CLERK';  
 SELECT ENAME,JOB,SAL FROM EMP WHERE SAL > 2000 AND JOB = 'CLERK';

[요구]

- \* OR 연산시 중복되는 ROW는 어떻게 처리되는지 아래의 SQL을 실행한후 결과를 설명 하십시오  
DEPTNO = 20 OR JOB = 'CLERK' 를 둘다 만족하는 ROW는 2번 출력 되는가 ?

```
SELECT DEPTNO,JOB,ENAME FROM EMP WHERE DEPTNO = 20;           // 5 Rows
SELECT DEPTNO,JOB,ENAME FROM EMP WHERE JOB = 'CLERK';         // 4 Rows
SELECT DEPTNO,JOB,ENAME FROM EMP WHERE DEPTNO = 20 OR JOB = 'CLERK' ;      // 7 Rows
```

- \* 집합연산자(UNION, UNION ALL, MINUS, INTERSECT)중 하나를 사용하여 위 2개 SQL에서의 중복된 데이터를 찾으십시오

- \* UNION ALL 집합 연산자를 사용하여 OR 와 동일한 결과를 생성하는 SQL을 작성 하십시오

BETWEEN

[정의] 범위 연산자

- ① SELECT EMPNO,ENAME,SAL FROM EMP WHERE SAL BETWEEN 1000 AND 2000; // 숫자  
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; // ?  
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,SAL FROM EMP  
WHERE HIREDATE between to\_date('81/02/20','yy/mm/dd') and to\_date('82/12/09','yy/mm/dd'); // ?  
⑦ SELECT ENAME,HIREDATE,SAL FROM EMP  
WHERE HIREDATE between to\_date('81/02/20','rr/mm/dd') and to\_date('82/12/09','rr/mm/dd');

[요구]

- \* ⑥ SQL이 실행되지 않는 이유를 날짜 포맷 YY 와 RR의 차이점을 조사한후 설명 하십시오.

- \* SELECT \* FROM SALGRADE WHERE 3000 BETWEEN LOSAL AND HISAL; 이 결과 데이터를 설명 하십시오

IN

[정의] 리스트 연산자

- ⑧ SELECT EMPNO,ENAME,JOB FROM EMP WHERE EMPNO IN (7369,7521,7654); // 숫자  
⑨ SELECT EMPNO,ENAME,JOB FROM EMP WHERE EMPNO = 7369 or EMPNO = 7521 or EMPNO=7654;
- ⑩ SELECT EMPNO,ENAME,JOB FROM EMP WHERE JOB in ('clerk','manager'); // 문자  
⑪ SELECT EMPNO,ENAME,HIREDATE FROM EMP WHERE HIREDATE IN ('81/05/01','81/02/20'); // 날짜?
- ⑫ SELECT EMPNO,ENAME,JOB,DEPTNO FROM EMP  
WHERE (JOB,DEPTNO) in (('MANAGER',20),('CLERK',20)); // 다중컬럼리스트  
⑬ SELECT EMPNO,ENAME,JOB FROM EMP WHERE EMPNO IN (7369,7369,7654);

[요구]

- \* SELECT DEPTNO, ENAME, JOB FROM SCOTT.EMP  
WHERE DEPTNO IN (SELECT DEPTNO FROM SCOTT.DEPT WHERE LOC IN ('NEWYORK','CHICAGO'));
- ① 서브쿼리(Sub Query의 정의 설명) ② SCOTT.EMP 설명 ③ 정상적 결과가 나오도록 SQL 수정

### ANY(=SOME) , ALL

- ① SELECT ENAME, SAL FROM EMP WHERE SAL > (1300,2450,3000); --?  
 ② SELECT ENAME, SAL FROM EMP WHERE SAL > ANY (1300,2450,3000); // ? AND OR  
 ③ SELECT ENAME, SAL FROM EMP WHERE SAL > ALL (1300,2450,3000); // ? AND OR
- ④ SELECT ENAME, SAL FROM EMP WHERE SAL = ANY (1300,2450,3000);  
 ⑤ SELECT ENAME, SAL FROM EMP WHERE SAL = SOME (1300,2450,3000); // ANY = SOME  
 ⑥ SELECT ENAME, SAL FROM EMP WHERE SAL = ALL (1300,2450,3000);

### LIKE

[정의] 문자 패턴 매칭 연산자, 정확한 값을 몰라도 찾을수 있는  
 % : 0개 이상의 모든문자  
 \_ : 1개의 모든문자, 위치가의미를 가짐.

EX) 김서방 찾기 !!  
 // Wilcard 문자

- ⑦ SELECT ENAME FROM EMP WHERE ENAME like 'A%'; // pattern matching  
 ⑧ SELECT ENAME FROM EMP WHERE ENAME like '\_A%'; // \_, %  
 ⑨ SELECT ENAME FROM EMP WHERE ENAME like '%L%E%';  
 ⑩ SELECT ENAME FROM EMP WHERE ENAME like '%LE%';  
 ⑪ SELECT ENAME FROM EMP WHERE ENAME like '%A%';  
 ⑫ SELECT ENAME FROM EMP WHERE ENAME NOT like '%A%';
- ⑬ SELECT ENAME, HIREDATE FROM EMP WHERE HIREDATE like '81%'; // ? 날짜  
 ⑭ SELECT ENAME, SAL FROM EMP WHERE SAL like 2%; // ? 숫자  
 ⑮ SELECT ENAME, SAL FROM EMP WHERE SAL like '2%'; // ? 숫자 like 문자

[요구]

\* 이름에 \_ 가 들어가 있는 사원들을 전부 찾고 싶다.. 방법은 ?  
 -힌트 : escape option

함수
----

ORACLE 제공 함수	<u>단일행</u> 함수	문자 함수
		숫자 함수
		날짜 함수
		<u>형변환</u> 함수
		기타 함수
	<u>그룹행</u> 함수	
사용자 정의 함수	PL/SQL로 사용자가 작성	

단일행 함수 개요 TEST
----------------

Single Row function 정의 : 1개의 row에 적용되고 1 row당 1개의 결과를 return  
 함수가 사용되는 위치 : select list , where, order by , group by

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

그룹행 함수 개요 TEST
----------------

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

- ③ SELECT AVG(SAL),SUM(SAL),SUM(COMM),COUNT(\*) FROM EMP; // 1건
- ④ SELECT DEPTNO,COUNT(\*),SUM(SAL),AVG(SAL) FROM EMP // GROUPING  
 GROUP BY DEPTNO;
- ⑤ SELECT DEPTNO,JOB,COUNT(\*),SUM(SAL),AVG(SAL) FROM EMP  
 GROUP BY DEPTNO,JOB;

단일행 - 문자 함수
-------------

함수	설명
LOWER	소문자로 변환하는 함수
UPPER	대문자로 변환하는 함수
INITCAP	단어의 첫 글자만 대문자로 변환하는 함수
SUBSTR, SUBSTRB * B:Byte	부분 문자 추출 함수, 한글과 영어를 동일한 문자 단위로 연산부분 Byte 추출 함수, 한글 과 영어를 다르게 연산한다.
INSTR, INSTRB	문자열내의 특정 문자의 위치 값을 반환 함수. 문자열내의 특정 문자(BYTE 단위) 위치 값을 반환 함수.
LENGTH ,LENGTHB	문자열의 길이를 반환 함수. 문자(Byte 단위)의 길이를 반환 함수.
VSIZE	Byte 크기 반환 함수
RPAD, LPAD	입력된 문자를 입력된 길이만큼 쪽(LPAD),오른쪽(RPAD)를 채운 결과를 반환하는 함수
TRIM, LTRIM, RTRIM	잘라내고 남은 문자열을 반환 함수
CONCAT	문자열 합성한 결과를 반환 함수
REPLACE	문자 변환 결과를 반환 함수
ASCII	ASCII 코드값 반환 함수
CHR	ASCII 코드로 변환 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;
```

```
SELECT rpad(ENAME,10,' ')||' 's JOB is '||lpad(JOB,10,' ') as JOB_list FROM EMP;           // 공백문자
```

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;

```
SELECT trim('장' from '장발장'), ltrim('장발장','장'), rtrim('장발장','장') FROM dual;
```

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

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

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

## 단일행 - 숫자 함수

함 수	설 명
ROUND	반올림 함수 (기준 자리 수 기준)
TRUNC	절삭 함수 (기준 자리 수 기준)
MOD	나눗셈 결과 나머지 값 반환 함수
CEIL	입력된 값보다 큰 정수들 중 가장 작은 값 반환함수
FLOOR	입력된 값보다 작은 정수들 중 가장 큰 값 반환함수
ABS	절대 값 반환 함수

⑨ SELECT round(45.923,2), round(45.923,1), round(45.923,0), round(45.923), round(45.923,-1) FROM dual;

```

⑩ SELECT trunc(45.923,2), trunc(45.923,1), trunc(45.923,0), trunc(45.923), trunc(45.923,-1)      FROM
dual;
```

```
⑪ SELECT mod(100,3), mod(100,2) FROM dual;           // 나머지
```

⑫ SELECT ENAME,SAL,SAL\*0.053 as tax,round(SAL\*0.053,0) as r\_tax FROM EMP; --급여의5.3%세금,원단위

```
⑬ SELECT CEIL(-45.594),CEIL(-45.294),CEIL(45.294),
        ROUND(-45.594).ROUND(-45.294).ROUND(45.594) FROM DUAL; // 절대값연산
```

⑭ SELECT FLOOR(45.245),FLOOR(-45.245),FLOOR(45.545),FLOOR(-45.545) FROM DUAL;

DATE TYPE 실습

[중요] DATE TYPE은 연산이 가능하다.

- ① SELECT sysdate,sysdate + 7, sysdate -2, sysdate + 1/24 FROM dual; // sysdate + 1/24 ?  
 ② SELECT deptno,ename, trunc(sysdate - hiredate) as work\_day FROM emp ORDER BY deptno,work\_day DESC;

[중요] DATE는 날짜와 시간 정보를 가지고 있다.

[질문] //아래의 결과에서는 왜 날짜만 보이고 시간은 보이지 않는가?

SELECT ename,sysdate,hiredate FROM emp;

[답변] 시간정보가 보이지 않는 이유는 날짜를 표시하는 FORMAT 문제 입니다.

- ③ SELECT ename,to\_char(sysdate,'YYYY-MM-DD:HH24:MI:SS'),to\_char(hiredate,'YYYY-MM-DD:HH24:MI:SS')  
FROM emp;

- ④ ALTER SESSION SET NLS\_DATE\_FORMAT = 'YYYY-MM-DD:HH24:MI:SS';  
SELECT ename,sysdate,hiredate FROM emp;  
SELECT SYSDATE FROM DUAL;

- ⑤ ALTER SESSION 명령어의 실행범위 테스트

SQL\*PLUS를 실행

SQL1> ALTER SESSION SET NLS\_DATE\_FORMAT = 'YYYY-MM-DD:HH24:MI:SS';  
SELECT SYSDATE FROM DUAL;  
SELECT HIREDATE FROM EMP;

새로운 SQL\*PLUS를 실행

SQL2> SELECT SYSDATE FROM DUAL;  
SELECT HIREDATE FROM EMP;

[요구]

\* 세션(Session) 의미 및 역할 2~3줄 정리

\* ALTER SESSION 명령어의 의미

단일행 - 날짜 함수

함수	설명
<u>SYSDATE</u>	DBMS Server의 현재 날짜와 시간을 DATE 타입으로 반환하는 함수 * DBMS 내부에서 DATE 타입은 숫자 데이터와 동일한 방식으로 저장되며 연산이 가능하다.
MONTHS_BETWEEN	입력된 두 날짜 사이의 개월 수 반환 함수
ADD_MONTHS	개월 수 증가 함수
LAST_DAY	해당 월의 마지막 날짜를 반환하는 함수
NEXT_DAY	지정한 날짜의 다음 요일의 날짜를 반환하는 함수
ROUND	날짜 반올림 함수
TRUNC	날짜 절삭 함수
EXTRACT	날짜에서 년/월/일/시/분/초를 추출하는 함수

- ⑥ SELECT HIREDATE,months\_between(sysdate,HIREDATE),months\_between(HIREDATE,sysdate) FROM EMP;  
// 찌꺼기 일자 처리: trunc(months\_between(sysdate,HIREDATE))

- ⑦ SELECT sysdate, add\_months(sysdate,3), add\_months(sysdate,-1) FROM dual;

- ⑧ SELECT sysdate, last\_day(sysdate), next\_day(sysdate,'일요일'), next\_day(sysdate,1),next\_day(sysdate,2)



FROM dual;

⑨ 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...

⑩ SELECT sysdate,trunc(sysdate,'YEAR'),trunc(sysdate,'MONTH'),trunc(sysdate,'DAY'),trunc(sysdate)  
FROM dual;

⑪ SELECT  
to\_char(sysdate,'MM"월"DD"일"') as mmdd1,  
to\_char(sysdate,'MM')||'월'||to\_char(sysdate,'DD')||'일' as mmdd2  
FROM dual

⑫ SELECT  
EXTRACT(YEAR FROM SYSDATE),  
EXTRACT(MONTH FROM SYSDATE),  
EXTRACT(DAY FROM SYSDATE)  
FROM DUAL;

SELECT HIREDATE, EXTRACT(YEAR FROM HIREDATE) FROM EMP;

[요구]

\* EXTRACT 함수의 결과 데이터 타입이 무엇인지 오라클 매뉴얼을 검색하여 설명 하십시오

\*다음의 날짜 포맷을 조사하여 [암기후] 설명 하십시오

구분	날짜 포맷	의미
시간	PM	
	HH, HH12,HH24	
분	MI	
초	SS	
	SSSSS	
	FF[1..9]	
일	D , d	
	DD, DDD, dd,ddd	
	DAY, DY, day	
공백	FM	
월	MM,MON,MONTH,	
년	YYYY , YYY,YY,Y ,y	
	RRRR , RR	
	YEAR ,Year	

\* 아래 SQL을 참고하여 해당월의 마지막 법정 영업일자를 구하는 SQL을 작성 하십시오(법정 영업일은 월~금요일)  
SELECT TO\_CHAR(SYSDATE,'DDD'),TO\_CHAR(SYSDATE,'DD'),TO\_CHAR(SYSDATE,'D') FROM DUAL;  
SELECT LAST\_DAY(SYSDATE) FROM DUAL;

\* (확장)해당월의 마지막 법정 영업일자를 구하십시오( 법정 영업일은 월~금요일,법정 공휴일)  
➔ Stored Function 이용

\*안전하지 않은 PROGRAMMING방식을 해결 하십시오

실습: ALTER SESSION SET NLS\_DATE\_FORMAT = 'RR/MM/DD';  
SELECT \* FROM EMP WHERE HIREDATE LIKE '82%'; -- 검색이 된다.

ALTER SESSION SET NLS\_DATE\_FORMAT = 'DD-MON-RR';  
SELECT \* FROM EMP WHERE HIREDATE LIKE '82%'; -- 날짜포맷이 바뀌니 검색이 안된다.  
SELECT \* FROM EMP WHERE HIREDATE LIKE '%82'; -- 검색조건을 바꾸어야 검색된다.

단일행 - 변환 함수(TO\_CHAR)

[주의] 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  
 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;
  
- ⑦ SELECT 0012345600,TO\_CHAR(00123456,'999999999'),TO\_CHAR(00123456,'000000000') FROM DUAL;

그룹행 함수

- ⑧ SELECT MIN(ENAME),MAX(ENAME),MIN(SAL),MAX(SAL),MIN(HIREDATE),MAX(HIREDATE) FROM EMP;  
     // ENAME:문자, SAL:숫자,HIREDATE:날짜
  
- ⑨ SELECT COUNT(\*),COUNT(EMPNO),COUNT(MGR),COUNT(COMM) FROM EMP;
  
- ⑩ SELECT COUNT(JOB),COUNT(ALL JOB),COUNT(DISTINCT JOB),SUM(SAL),SUM(DISTINCT SAL)  
 FROM EMP;
  
- ⑪ SELECT COUNT(\*), SUM(COMM), SUM(COMM)/COUNT(\*),AVG(COMM),SUM(COMM)/COUNT(COMM)  
 FROM EMP;
  
- // 그룹행함수 와 NULL 그리고 NVL , 효율적인 계산방안은 ?
- ⑫ SELECT SAL,COMM FROM EMP;
  
- ⑬ SELECT SUM(NVL(COMM,0)) AS SUM\_COMM1,  
     SUM(COMM) AS SUM\_COMM2,  
     NVL(SUM(COMM),0) AS SUM\_COMM3  
 FROM EMP

## GROUP BY - 그룹핑

[주의] GROUP BY 실행 방식이 10g 부터 hash 방식으로 변경된후에는 정렬된 결과 집합이 되지 않는다.

```
// 부서단위로 그룹핑하여 결과 출력, ALIAS, 정렬(?), 소수점 이하(?)
① SELECT DEPTNO,COUNT(*),SUM(SAL),AVG(SAL) FROM EMP
   GROUP BY DEPTNO;
```

```
SELECT JOB, COUNT(*), AVG(SAL) FROM EMP
   GROUP BY JOB;
```

// 정렬된 결과를 얻기 위해서는

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

// NULL도 그룹핑 대상이 되는지?

```
SELECT COMM,COUNT(*) FROM EMP
   GROUP BY COMM;
```

\*\* COLUMN HEADING의 가독성을 높이고 급여 평균에서 소수점 이하 자리 처리하는 방법은 ?

```
② SELECT DEPTNO,
          COUNT(*) AS CNT_DEPT,
          SUM(SAL) AS SUM_SAL,
          ROUND(AVG(SAL),0) AS AVG_SAL
   FROM EMP
   GROUP BY DEPTNO
   ORDER BY DEPTNO,SUM_SAL;
```

[요구]

\* 아래의 SQL 실행 결과를 설명하고 문제를 해결 하십시오

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

\* SQL\*PLUS에서 ①을 실행시 소수점이하 5자리가 보이는 이유 와 SHOW,SET 명령어 역할을 설명 하십시오

참고: SHOW NUMWIDTH , SET NUMWIDTH

SQL DEVELOPER에서 ① 실행시 표현되는 소수점 이하자리수를 조절하는 방법을 설명 하십시오

\* 다음과 같은 결과를 생성하는 SQL문장을 작성하십시오. CAN!!!!

10번부서	20번부서	30번부서
3	5	6

## HAVING

[정의] GROUP BY 결과집합에 대한 조건절

```
③ SELECT DEPTNO,COUNT(*),SUM(SAL),AVG(SAL) FROM EMP GROUP BY DEPTNO;
```

```
④ SELECT DEPTNO,COUNT(*),SUM(SAL),AVG(SAL) FROM EMP
   GROUP BY DEPTNO HAVING SUM(SAL) > 9000;
```

```
⑤ SELECT DEPTNO,COUNT(*),SUM(SAL),AVG(SAL) FROM EMP
   GROUP BY DEPTNO HAVING DEPTNO in (10,20);
```

⑥ // 실행순서, WHERE과 HAVING 비교

```
SELECT DEPTNO,COUNT(*),SUM(SAL),AVG(SAL) FROM EMP
WHERE DEPTNO IN (10,20)
GROUP BY DEPTNO
HAVING SUM(SAL) > 9000
ORDER BY SUM(SAL);
```