02. SELECT

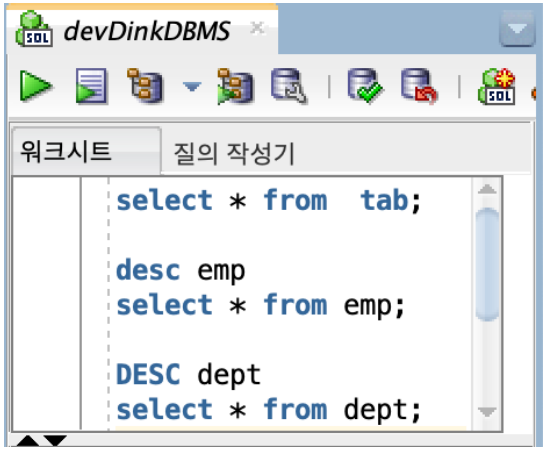
목차

0. SQL Developer

❑ Oracle 社에서 무료로 제공하는 Java 기반 GUI IDE(Integrated Development Environment: 통합개발환경)를 갖춘 SQL 개발(개발자,분석가) 및 관리툴(DBA : Database Administrator,Tuner)

- 유료제품: Toad, Orange

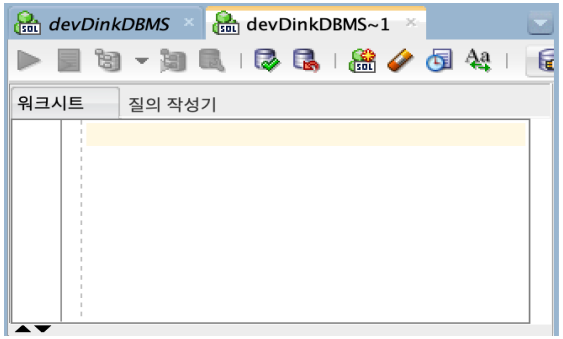
❑ SQL 워크시트에서 N개 SQL 작성



* 해당 SQL 차례로 선택(클릭)후 실행(ctrl + enter)

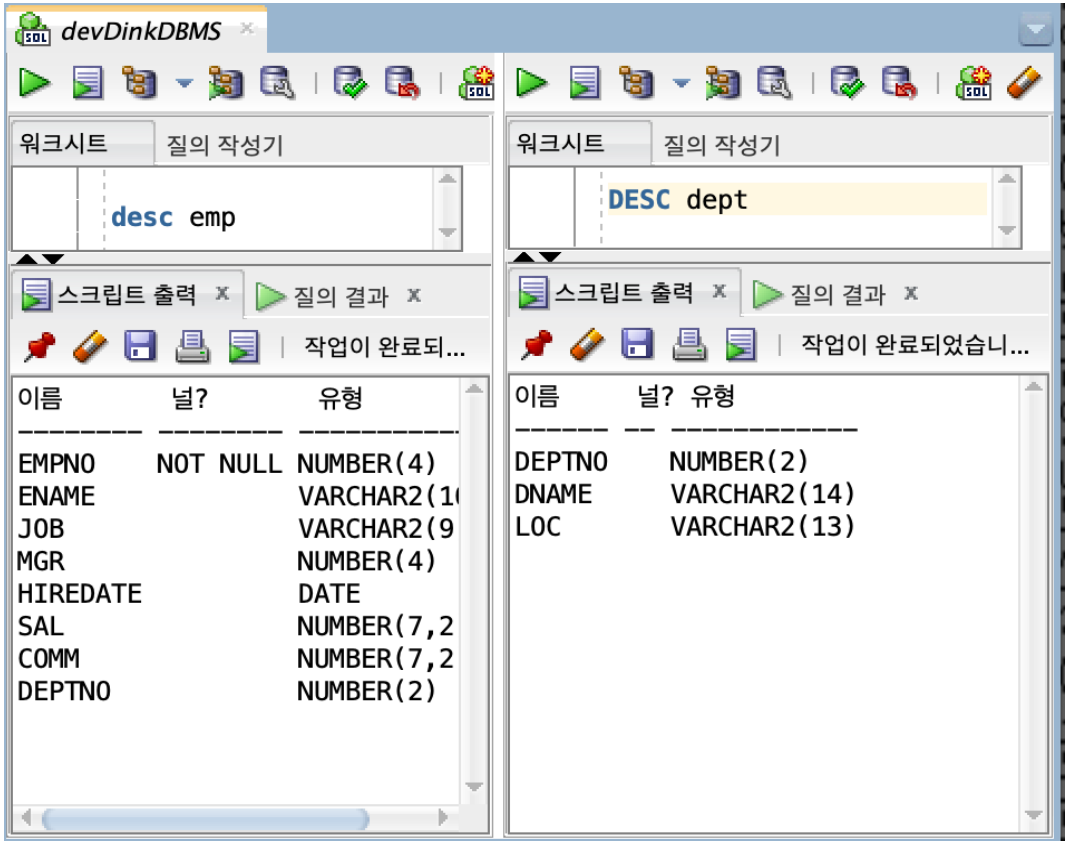
❑ N개 SQL 워크시트 생성

- 도구(T) > SQL 워크시트 > devDinkDBMS 선택



❑ SQL 워크시트 화면 N개 분할 (SQL공유, 실행결과 분할)

- 창(W) > Configure Window > 세로로 분할
- 기존창 : desc emp 실행 , 새로운창: DESC dept 실행



1. SQL 명령어 분류 와 SELECT

분류	대상	구문
Query	데이터	SELECT(조회)
DML(Data Manipulation Language)	데이터	INSERT(입력), UPDATE(수정), DELETE(삭제),MERGE(INSERT+UPDATE)
TCL(Transaction Control Language)	트랜잭션	COMMIT(저장), ROLLBACK(취소), SAVEPOINT(중간 저장점)
DDL(Data Definition Language)	Object	CREATE(생성), ALTER(변경), DROP(Object 삭제) , TRUNCATE(절삭)
DCL(Data Control Language)	권한	GRANT(부여), REVOKE(취소)

QUERY

① 질의어 : 데이터베이스에 저장된(테이블) 데이터를 조회 하는 명령어

② **SQL**

1. SQL 명령어 분류 와 SELECT

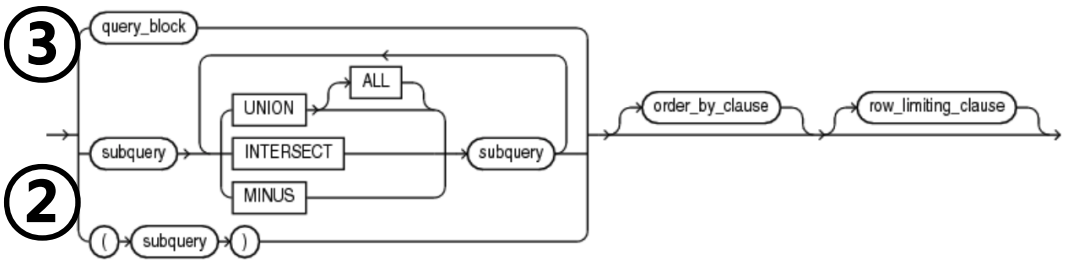
Google → **oracle select syntax 12c** → SELECT(클릭) → Syntax(클릭)

* oracle 21c

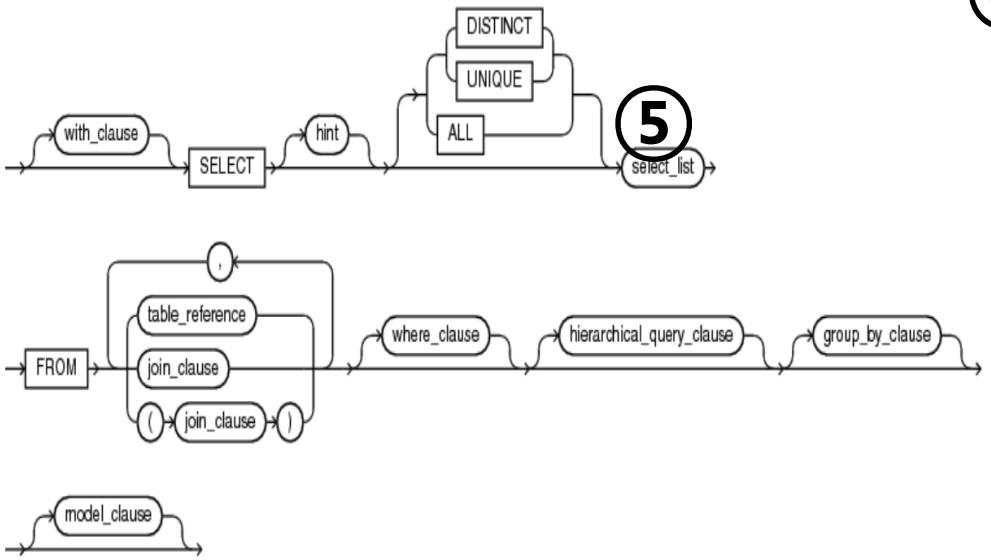
❑ SELECT Syntax Diagram

***select*::=**

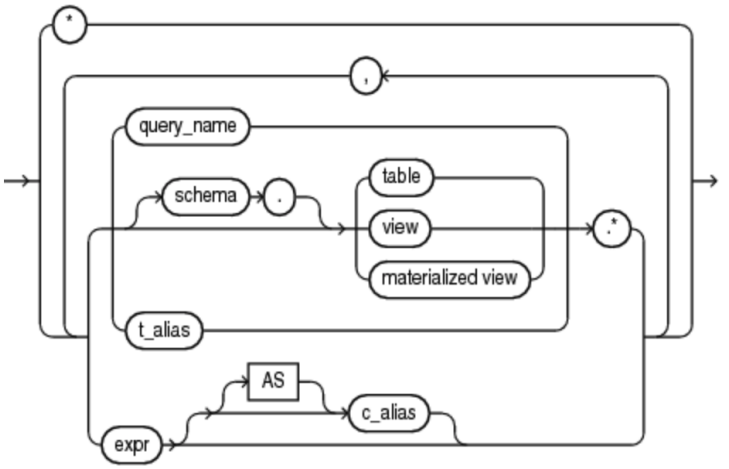
***subquery*::=**



4 ***query_block*::=**



6 ***select_list*::=**



2. SELECT LIST

❑ SELECT * FROM TAB;

* 내가 사용 가능한 모든 테이블 리스트를 조회하는 SQL

* 내가 ~ DBMS에 Connection(접속)한 사용자(Schema)
  사용가능한 ~ 소유한(Owner) 테이블 + 권한(Privilege)을 부여받은(granted) 테이블

❑ DESC[RIBE] EMP

* 테이블 구조를 조회하는 명령어

```
이름          널?       유형
--------- -------- ------------
EMPNO     NOT NULL NUMBER(4)
ENAME              VARCHAR2(10)
JOB                VARCHAR2(9)
MGR                NUMBER(4)
HIREDATE           DATE
SAL                NUMBER(7,2)
COMM               NUMBER(7,2)
DEPTNO             NUMBER(2)
```

① 컬럼 이름 , 테이블내에서 컬럼 정의 순서
② 컬럼 데이터 타입
   ex) VARCHAR2(문자),NUMBER(숫자),DATE(날짜)
③ 데이터 길이
④ NULL 허용여부

2. SELECT LIST

❑ SELECT LIST

* SELECT 와 FROM사이에 조회(Projection)를 원하는 COLUMN(들)을 지정

① SELECT EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO FROM EMP;
② SELECT * FROM EMP;
- EMP는 8개 컬럼으로 정의 , ① ② 같은 결과 조회, * 의미는? 조회순서는?
- 좋은 방식의 SQL문은?
- 컬럼 구분자 콤마(,) , SQL 문장 종결자(;)

③ SELECT EMPNO, ENAME, HIREDATE FROM EMP;
- 원하는 컬럼만 선택하여 조회
- 데이터 타입 : EMPNO(숫자), ENAME(문자), HIREDATE(날짜)
- 데이터 정렬 방향 : 문자 ( ) 정렬
  숫자/날짜 ( ) 정렬
- 컬럼 헤딩(Column Heading)
  * 대문자 표기, 데이터와 정렬 방향이 동일

EMPNO	ENAME	HIREDATE
7369	SMITH	80/12/17
7499	ALLEN	81/02/20
7521	WARD	81/02/22

④ SELECT SAL, JOB, EMPNO, ENAME FROM EMP;
- 컬럼 조회 순서 임의로 변경 가능

⑤ SELECT EMPNO, EMPNO, EMPNO, ENAME, JOB, SAL FROM EMP;
- 동일한 컬럼 여러번 조회 가능, 용도?

⑥ SELECT EMPNO, SAL, 2022, 'Happy New Year' FROM EMP;
- 홑따옴표(') = 단일인용부호 = single quotation mark
- 2022 : 숫자 데이터 , 'Happy New Year' : 문자 데이터
- 테이블내에 존재하지 않지만 SELECT시 임의적으로 데이터를 만들어서 조회 가능

2. SELECT LIST

❑ 산술 연산자

* **SQL 구문내에서 산술(사칙)연산(*, /, +, -)자 사용 가능**
* **CLIENT/SERVER적 관점 ?**
* NUMBER 와 **DATE 타입에 사용**
* 연산자 우선순위 :

( ) > NOT > 비교연산자 > SQL연산자 > **AND > OR** > 산술연산자

① SELECT ENAME,SAL, SAL*12, HIREDATE, HIREDATE-7, HIREDATE + 100 FROM EMP;

- SAL NUMBER TYPE, HIREDATE DATE TYPE

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

③ SELECT ENAME,COMM,COMM+300 FROM EMP;

- 기존 커미션(COMM)에 300을 더해서 보너스 지급, 원하는 결과가 안나오는 이유는?

❑ Alias (가명,별칭)

Table Alias: 일반적으로는 SELF JOIN 시에 사용

ex) SELECT E.EMPNO,E.ENAME FROM EMP E;

Column Alias : Column Heading(컬럼 레이블)을 의미있는 다른 이름으로 재정의

① COLUMN의 의미 명료성 ②**DB 프로그램 개발시 Code의 명료성 & 컬럼제어(계산식에서 유용)** ③ 대문자 표현

④ SELECT ENAME, SAL+12 ,SAL*12 as annual_salary FROM EMP;

⑤ SELECT ENAME, MGR Manager, SAL*12 as annual_SAL, COMM+300 "Special Bonus" FROM EMP;

- 컬럼 Alias 3가지 표현방식(공백문자, AS, "~") 중 가장 명료한 방식은? , "~"은 특수문자, 공백문자, 대소문자 구분이 필요한 경우 사용

⑥ SELECT ENAME, COMM+300 보너스 , COMM+300 AS "Special Bonus" FROM EMP;

2. SELECT LIST

❑ (문자열) 결합 연산자 (Concatenation operator)

① SELECT ENAME||JOB FROM EMP;
- | : 수직선 기호(Vertical bar), 파이프 문자 , OR 연산

② SELECT DNAME||' 부서는 '||LOC||' 지역에 위치합니다.' as LOC FROM DEPT;
- 명목형 자료를 문장으로

③ SELECT ename||''''s JOB is '||job as job_list FROM emp;
- 홑따옴표(') = 단일인용부호 = single quotation mark
- Oracle DBMS 문자 데이터 표현 : 'Happy New Year'
- SCOTT's JOB is ANALYST vs SCOTT"s JOB is ANALYST

④ SELECT sal, **sal*100**, **sal || '00'**, **to_char(sal)||'00'** FROM EMP;
- **sal||'00' : 숫자 타입과 문자 타입 결합, 결과 타입은?**
- Data type Conversion
  Implicit Conversion(암시적, 자동으로) ex) sal || '00'
  Explicit Conversion (명시적, 수동으로) ex) to_char(sal) || '00'
- 좋은방식의 SQL Coding은?

❑ DUAL

* sys 계정 소유의 **Dummy Table**로 실제 테이블로부터 데이터를 가져오는 것이 아닌 function, calculation을 수행하기 위해서 사용하는 작고 가벼운 테이블 (1 row, 1 column)

⑤ desc dual // describe dual , 사전적 의미 : 둘의(이중의) ?
SELECT * FROM DUAL;

⑥ SELECT **sysdate** FROM dual; // system의 현재 date(날짜와 시간)을 리턴하는 함수

⑦ SELECT 2025*12345, to_char(2025*12345,'999,999,999'), to_char(2025*12345,'999,999,999') as cal
FROM dual;

❑ 과제

1) EMP 테이블에서 사번,이름,직무,급여,부서번호 데이터 사이에 구분자 ,

를 삽입하는 예제 SQL 작성

* **CSV**(Comma-Separated Values) 파일에서 사용할수 있는 데이터 포맷 출력

2) SYS 계정, SYSTEM 계정 이란?

3) sysdate 함수에서 시간이 나타나지 않는 이유?

- 시스템의 현재 시간,분,초,1/100초 표현하는 SQL 작성
- 시스템의 현재 시간,분,초,1/1000초 표현하는 SQL 작성

3. WHERE

❑ WHERE

* 원하는 ROW(레코드)를 조회하는 조건절

① SELECT * FROM EMP WHERE DEPTNO = 10;
② SELECT DEPTNO, ENAME, SAL, JOB FROM EMP WHERE SAL > 2000;
③ SELECT DEPTNO, ENAME, SAL, JOB FROM EMP WHERE DEPTNO = 10 AND SAL > 2000;
④ SELECT DEPTNO, ENAME, SAL, JOB FROM EMP WHERE DEPTNO = 10 OR SAL > 2000;
⑤ SELECT DEPTNO, SAL, JOB FROM EMP WHERE DEPTNO = 10 AND SAL > 2000 OR JOB='MANAGER';
SELECT DEPTNO, SAL, JOB FROM EMP WHERE (DEPTNO = 10 AND SAL > 2000) OR JOB='MANAGER';
SELECT DEPTNO, SAL, JOB FROM EMP WHERE DEPTNO = 10 AND (SAL > 2000 OR JOB='MANAGER');
- 연산자 우선순위 ? , 좋은 방식의 SQL 코딩?

⑥ SELECT DEPTNO, ENAME, SAL, JOB FROM EMP WHERE JOB = 'manager';
- 이유는?

⑦ SELECT DEPTNO,ENAME,JOB FROM EMP WHERE 1=1;
⑧ SELECT DEPTNO,ENAME,JOB FROM EMP WHERE 1=2;

SELECT LIST

WHERE

	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

⑨ SELECT DEPTNO, ENAME, SAL, JOB FROM EMP
WHERE (DEPTNO,JOB,MGR) = ((10,'MANAGER',7839));

널널한 제비연

4. NULL

❑ NULL

* 사전적 의미: 값이 정의되지 않은, 존재하지 않는
  unavailable, unassigned, unknown, inapplicable

Data : **데이터가 존재(미입력)하지 않는 상태, 결측치(Missing Value)**

* not the same as zero or a blank space
  - ASCII 코드 0->48, ' ' ->32 , null-> 00 , - Oracle DBMS: NULL을 표현하기 위해 내부에 1 Byte 사용, Trailing NULL인 경우 저장하지 않는다.
* 모든 데이터 유형에 null 적용

① 연산불가(NULL 과 산술연산)

```
SELECT 300/0 FROM dual;                                  // "divisor is equal to zero"
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 비교 연산)

```
SELECT ENAME,SAL,COMM FROM EMP WHERE COMM > -1;           // null이 있는 column 비교연산
SELECT ENAME,SAL,COMM FROM EMP WHERE COMM = null;
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;
```

4. NULL

③ 제어불가(NULL을 함수에 적용불가)

```
SELECT ENAME, length(ENAME), COMM, length(COMM) FROM EMP;
SELECT sal, comm, abs(sal-comm)+300 FROM emp;
```

④ NULL 제어 함수

```
SELECT  COMM, NVL(COMM,0), DECODE(COMM, NULL, 0, COMM) AS NVL_SIMUL  FROM EMP;
SELECT  concat('Commission is ',COMM),  'Commission is '||COMM  FROM EMP;        -- NULL 무시
select    count(sal) as sal_cnt, count(comm) as comm_cnt, sum(comm) as comm_sum from emp;
```

- 단일행 함수(Single Row Function) ex) length, abs
- 그룹행 함수(Group Row Function) ex) count, sum

[과제]

1) 커미션 평균을 계산하는 SQL을 sum/count 함수 사용하는 방식과, avg 함수 사용하는 2가지 방식으로 작성

2) 다음 SQL 설명

```
SELECT SAL, COMM, NVL(COMM,SAL), NVL2(COMM,SAL,0), JOB, NULLIF(JOB,'MANAGER') FROM emp;
```

3) 다음 SQL 설명

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

4) 다음 SQL 결과 설명

```
SELECT  deptno,ename,job FROM emp  where deptno = NVL(NULL, deptno);
SELECT  deptno,ename,job FROM emp  where deptno = NVL(10, deptno);          -- decode ??
```

5. ORDER BY

[역할] 지정한 컬럼을 기준으로 데이터 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 ~ Oracle DBMS: 가장 큰 값으로 간주 , MS SQL-Server: 가장 작은값으로 간주

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

① SELECT ENAME, HIREDATE, SAL, COMM FROM EMP ORDER BY ENAME ; // 정렬방향? Default
② SELECT ENAME, HIREDATE, SAL, COMM FROM EMP ORDER BY ENAME asc; // SQL 가독성
③ SELECT ENAME, HIREDATE, SAL, COMM FROM EMP ORDER BY HIREDATE desc; // 날짜

④ SELECT ENAME, HIREDATE, SAL, COMM FROM EMP ORDER BY ENAME; // column name
⑤ SELECT ENAME, HIREDATE, SAL, COMM FROM EMP ORDER BY 2; // column position
⑥ SELECT ENAME, SAL*12 as 연봉 FROM EMP ORDER BY 연봉; // column alias
⑦ SELECT ENAME, HIREDATE, SAL, COMM FROM EMP ORDER BY COMM * 12; // expression, NULL 위치
SELECT ENAME, HIREDATE, SAL, COMM FROM EMP ORDER BY COMM * 12 NULLS FIRST;

⑧ SELECT DEPTNO,JOB,ENAME FROM EMP ORDER BY DEPTNO;
⑨ SELECT DEPTNO,JOB,ENAME FROM EMP ORDER BY DEPTNO,JOB; // 조합의 순서쌍
⑩ SELECT DEPTNO,JOB,ENAME FROM EMP ORDER BY DEPTNO,JOB desc; // 차이점은?

6. DISTINCT

❑ DISTINCT

* 중복된 데이터를 필터링하여 조회(SELECT)

- 조회시 사용하며 실제 데이터를 삭제 하지 않는다.

① SELECT JOB FROM EMP; // Row 개수 만큼,
② SELECT UNIQUE JOB FROM EMP; // 직군 종류, Oracle
③ SELECT DISTINCT JOB FROM EMP; // ANSI

④ SELECT DISTINCT DEPTNO, JOB FROM EMP; // 중복 데이터(?)
- 컬럼(들)의 조합의 중복 필터링

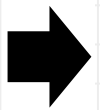
⑤ SELECT JOB FROM EMP ORDER BY JOB;
- distinct 연산 내부 알고리즘: Oracle 9i: Sort , 10g: Hash

⑥ SELECT DISTINCT JOB, DISTINCT DEPTNO FROM EMP; // 범위?
⑦ SELECT JOB, DISTINCT DEPTNO FROM EMP; // 위치?

⑧ SELECT COMM FROM EMP WHERE COMM IS NOT NULL; // 4 Rows
⑨ SELECT DISTINCT COMM FROM EMP; // 5 Rows ??
- ** NULL과 DISTINCT

①

JOB
CLERK
SALESMAN
SALESMAN
MANAGER
SALESMAN
MANAGER
MANAGER
ANALYST
PRESIDENT
SALESMAN
CLERK
CLERK
ANALYST
CLERK



②

JOB
CLERK
SALESMAN
PRESIDENT
MANAGER
ANALYST

❑ 과제

1) ③ SQL 실행 결과가 Oracle 9i 에서는 정렬되어 나타나지만 Oracle 10g이후 버전부터는 정렬된 결과가 나타나지 않는다. 이유를 찾아 설명

2) ⑥⑦ SQL이 왜 에러가 나는지 Select Syntax Diagram을 보고 설명

3) Interactive SQL 과 Embedded SQL를 설명 하고 각각의 사용예시를 찾아서 발표

7. SQL 연산자

❑ BETWEEN

* 범위 연산자
  - 하한값 과 상한값 사이의 범위 검색 연산자

```
① SELECT ENAME,SAL,HIREDATE FROM EMP WHERE SAL between 1000 and 2000;          // 숫자 타입
② SELECT ENAME,SAL,HIREDATE FROM EMP WHERE SAL >= 1000 and SAL <= 2000;        // 차이점?
③ SELECT ENAME,SAL,HIREDATE FROM EMP WHERE SAL between 2000 and 1000;          // 이유는?

④ SELECT ENAME,SAL,HIREDATE FROM EMP WHERE ENAME BETWEEN 'C' AND 'K';          // 문자 타입
⑤ SELECT ENAME,SAL,HIREDATE FROM EMP
   WHERE HIREDATE BETWEEN '81/02/20' AND '82/12/09';                           // 날짜 타입, 형변환

⑥ SELECT ENAME,SAL,HIREDATE FROM EMP
   WHERE HIREDATE BETWEEN to_date('81/02/20','yy/mm/dd') AND to_date('82/12/09','yy/mm/dd');
   // 날짜 타입, 명시적 형변환,  검색이 안되는 이유는 ?

⑦ SELECT ENAME,HIREDATE,
          TO_CHAR(HIREDATE,'YYYY/MM/DD'),
          TO_CHAR(HIREDATE,'YYYY/MM/DD HH24:MI:SS'),                           // Date에 시간정보확인
          TO_CHAR(HIREDATE,'RRRR/MM/DD HH24:MI:SS'),
          TO_CHAR(SYSDATE, 'YYYY/MM/DD HH24:MI:SS')                            // sysdate에 시간정보확인
   FROM EMP;
   SELECT ENAME,SAL,HIREDATE FROM EMP
   WHERE HIREDATE BETWEEN to_date('81/02/20','rr/mm/dd') AND to_date('82/12/09','yy/mm/dd');
```

7. SQL 연산자

❑ BETWEEN

* 범위 연산자
  - 하한값 과 상한값 사이의 범위 검색 연산자

```
① SELECT ENAME,SAL,HIREDATE FROM EMP WHERE SAL between 1000 and 2000;          // 숫자 타입
② SELECT ENAME,SAL,HIREDATE FROM EMP WHERE SAL >= 1000 and SAL <= 2000;        // 차이점?
③ SELECT ENAME,SAL,HIREDATE FROM EMP WHERE SAL between 2000 and 1000;          // 이유는?

④ SELECT ENAME,SAL,HIREDATE FROM EMP WHERE ENAME BETWEEN 'C' AND 'K';          // 문자 타입
⑤ SELECT ENAME,SAL,HIREDATE FROM EMP
   WHERE HIREDATE BETWEEN '81/02/20' AND '82/12/09';                           // 날짜 타입, 형변환

⑥ SELECT ENAME,SAL,HIREDATE FROM EMP
   WHERE HIREDATE BETWEEN to_date('81/02/20','yy/mm/dd') AND to_date('82/12/09','yy/mm/dd');
   // 날짜 타입, 명시적 형변환,  검색이 안되는 이유는 ?

⑦ SELECT ENAME,HIREDATE,
          TO_CHAR(HIREDATE,'YYYY/MM/DD'),
          TO_CHAR(HIREDATE,'YYYY/MM/DD HH24:MI:SS'),                           // Date에서 시간정보확인
          TO_CHAR(HIREDATE,'RRRR/MM/DD HH24:MI:SS'),
          TO_CHAR(SYSDATE, 'YYYY/MM/DD HH24:MI:SS')                            // sysdate에서 시간정보확인
   FROM EMP;
```

7. SQL 연산자

❑ BETWEEN

* 범위 연산자
  - 하한값 과 상한값 사이의 범위 검색 연산자

⑧
```
SELECT ENAME,SAL,HIREDATE FROM EMP
WHERE  HIREDATE BETWEEN to_date('1981/02/20','yyyy/mm/dd') AND to_date('1982/12/09','yyyy/mm/dd');
```

⑨
```
SELECT ENAME,SAL,HIREDATE FROM EMP
WHERE  HIREDATE BETWEEN to_date('81/02/20','rr/mm/dd') AND to_date('82/12/09','rr/mm/dd');
```

⑩
```
SELECT ENAME,SAL,HIREDATE FROM EMP
WHERE  HIREDATE BETWEEN to_date('2081/02/20','yyyy/mm/dd') AND to_date('2082/12/09','yyyy/mm/dd');
```

7. SQL 연산자

❑ LIKE

* 문자 패턴 매칭 연산자, 정확한 값을 몰라도 찾을수 있는 EX) 김서방 찾기 !!
* Wildcard 문자
  ① % : **0개 이상의 모든 문자** ② _ : **1개의 모든 문자**, 위치가 의미를 가짐.

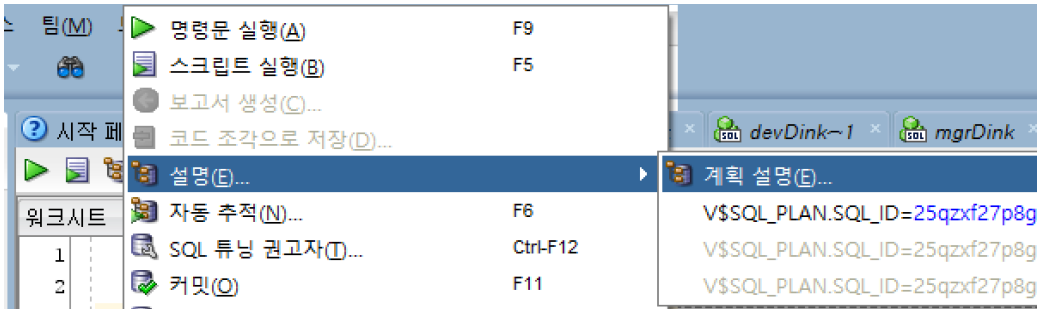
```
① SELECT ENAME FROM EMP WHERE ENAME like 'A%';                    // pattern matching
② SELECT ENAME FROM EMP WHERE ENAME like '_A%';
③ SELECT ENAME FROM EMP WHERE ENAME like '%L%E%';
④ SELECT ENAME FROM EMP WHERE ENAME like '%LE%';
⑤ SELECT ENAME FROM EMP WHERE ENAME like '%A%';
⑥ SELECT ENAME FROM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

⑦ SELECT ENAME,HIREDATE FROM EMP WHERE HIREDATE like '81%';       // 날짜 , 암시적 형변환
⑧ SELECT ENAME,SAL FROM EMP WHERE SAL like 2%;                    // 숫자 ,
⑨ SELECT ENAME,SAL FROM EMP WHERE SAL like '2%';                  // 암시적 형변환
⑩ SELECT ENAME,SAL FROM EMP WHERE TO_CHAR(SAL) like '2%';         // 명시적 형변환
```

7. SQL 연산자 – Data Type Conversion

❑ 실행계획 조회(Execution Plan)

① SELECT * FROM EMP WHERE SAL = '3000';

② SQL 선택 > 마우스 오른쪽 버튼 > 설명 > 계획설명 (Explain execution plan)



③[에러 발생시] 권한이 없는 경우 DBA 계정으로 SQL 워크시트 생성

- 도구(T) > SQL 워크시트 > mgrDinkDBMS 클릭
- grant select_any_catalog to scott; ## 권한부여
- 도구(T) > SQL 워크시트 > devDinkDBMS 클릭 ## 신규 session 생성부터 적용

④ ② 재실행 !! 암시적 형변환(Implicit Datatype Conversion)

OPERATION	OBJECT_NAME	OPTIONS
SELECT STATEMENT		
TABLE ACCESS	EMP	FULL
Filter Predicates		
SAL=3000		

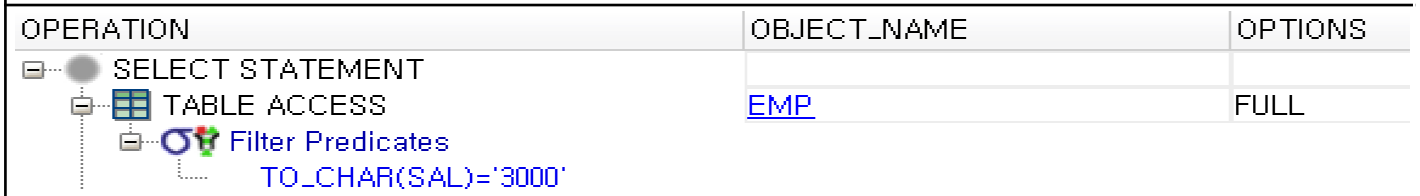
* 문자 < 숫자 : '3000' ➔ 3000 * 문자 < 숫자 < 날짜

7. SQL 연산자 – Data Type Conversion

실행계획 조회(Execution Plan)

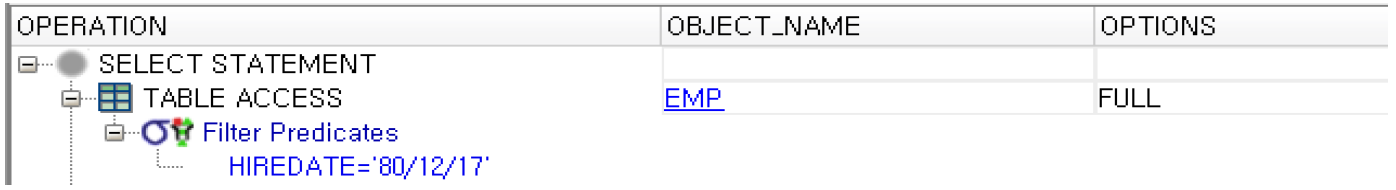
⑤ 명시적 형변환
숫자➔문자

SELECT * FROM EMP WHERE TO_CHAR(SAL) = '3000'; ## SQL선택 ➔ F10



⑥ 암시적 형변환
날짜 > 문자

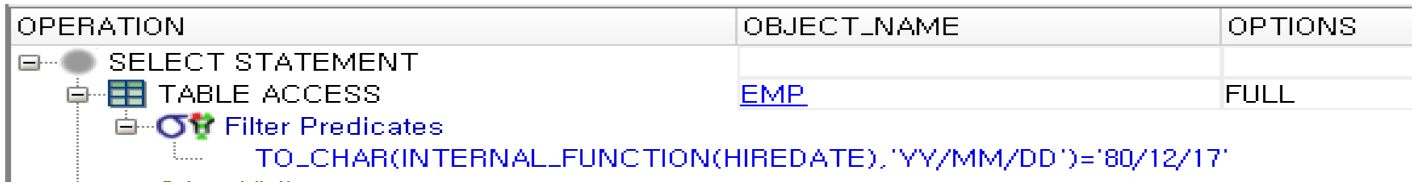
SELECT * FROM EMP WHERE HIREDATE = '80/12/17';



⑦ 명시적 형변환

날짜 ➔ 문자

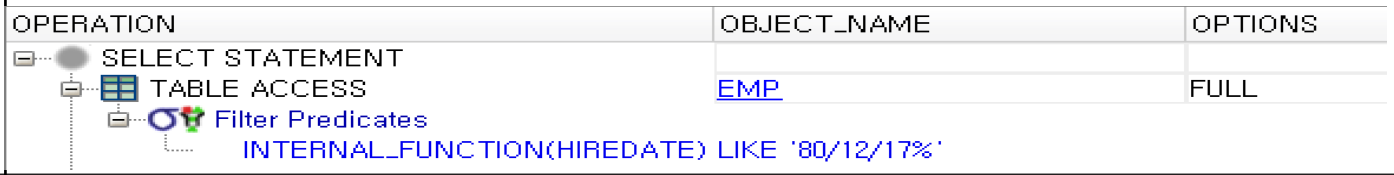
SELECT * FROM EMP WHERE TO_CHAR(HIREDATE,'YY/MM/DD') = '80/12/17'



⑧ 암시적 형변환

날짜 ➔ 문자

SELECT * FROM EMP WHERE HIREDATE LIKE '80/12/17%';



7. SQL 연산자

❑ IN

* 리스트 연산자

```
① SELECT EMPNO, JOB FROM EMP WHERE EMPNO IN (7369,7521,7654);                    // 숫자
② SELECT EMPNO, JOB FROM EMP WHERE EMPNO = 7369 or EMPNO = 7521 or EMPNO=7654;   // 차이점 ?
③ SELECT EMPNO,ENAME,JOB FROM EMP WHERE JOB IN ('clerk','manager');              // 문자
④ SELECT EMPNO,ENAME,HIREDATE FROM EMP WHERE HIREDATE IN ('81/05/01','81/02/20'); // 날짜?
⑤ SELECT EMPNO,ENAME,JOB,DEPTNO FROM EMP
   WHERE (JOB,DEPTNO) in (('MANAGER',20),('CLERK',20));                          // 다중컬럼리스트 , in (1..1000)
```

❑ ANY , ALL

* ANY(SOME) :리스트내에 조건을 만족하는 값이 1개 이상이면 결과 리턴 (OR)
* ALL :리스트내에 모든값이 조건을 만족해야 결과 리턴 (AND)

```
① SELECT ENAME, SAL FROM EMP WHERE SAL > (1500,2450,3000);          // Error ?? 단일값 비교 연산자
② SELECT ENAME, SAL FROM EMP WHERE SAL >   ANY(1500,2450,3000);     // OR  , SAL > 15000
③ SELECT ENAME, SAL FROM EMP WHERE SAL >= ALL(1500,2450,3000);      // AND  , SAL >= 3000

④ SELECT ENAME, SAL FROM EMP WHERE SAL = ANY (1500,2450,3000);
⑤ SELECT ENAME, SAL FROM EMP WHERE SAL = SOME (1500,2450,3000);     // ANY = SOME
   SELECT ENAME, SAL FROM EMP WHERE SAL IN (1500,2450,3000);
⑥ SELECT ENAME, SAL FROM EMP WHERE SAL = ALL (1300,2450,3000);      //  0 Rows가 검색되는 이유?
```

7. SQL 연산자

❑ 과제

1) SELECT * FROM SALGRADE WHERE 3000 BETWEEN LOSAL AND HISAL;
결과 데이터를 설명

2) 아래의 SQL 실행시 데이터 검색이 안되는 이유를 날짜 포맷 YY 와 RR의 차이점을 조사한 후 설명

```
SELECT ENAME,SAL,HIREDATE FROM EMP
WHERE HIREDATE BETWEEN to_date('81/02/20','yy/mm/dd') AND to_date('82/12/09','yy/mm/dd');
```

3) LIKE 연산자를 사용하여
ENAME에 '_' 가 들어가 있는 사원들을 찾는 SQL 작성
ENAME 컬럼의 데이터를 수정(UPDATE)하여 검색 결과 예시 출력
-힌트 : escape option

8. DECODE, CASE

Decode 사용 전/후

No	코딩 사례

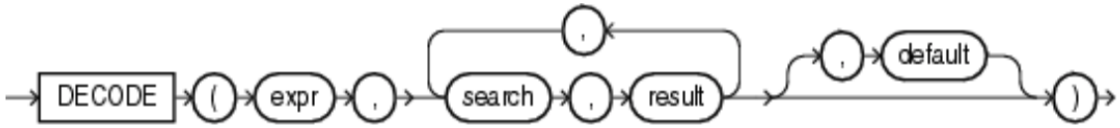
①	if($con_faxno == "c" && $con_telno == "p") {
	$looInsertSql = "insert into $addr_u (ADDRESS_SEQ,USERID,NAME,EMAIL,".
	" P_HTELNO,P_TELNO,C_TELNO,P_FAXNO,C_FAXNO,CON_TELNO, CON_FAXNO)".
	" value (ADDRESS_SEQ.NEXTVAL,'$addid','$name','$email',".
	" $htelno,$telno,$faxno)"; }
	if($con_faxno == "p" && $con_telno == "c") {
	$looInsertSql = "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") {
	$looInsertSql = "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") {
	$looInsertSql = "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 $looInsertSql;
	$db->execute($looInsertSql); }

②	$looInsertSql = "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')";

8. DECODE, CASE

❑ DECODE

* **조건절(IF ~ ELSE IF ~ELSE) 연산자**
* '=' 비교 연산자만 사용
* DECODE Syntax Diagram



① SELECT DEPTNO, ENAME, DECODE(DEPTNO,10,'ACCOUNTING',20,'RESEARCH',30,'SALES','ETC')
FROM EMP
ORDER BY DEPTNO;

```
if       deptno = 10 then 'ACCOUNTING'
else if  deptno =20 then 'RESEARCH'
else if  deptno =30 then 'SALES'
else                      'ETC'
```

② SELECT COMM, DECODE(COMM,NULL,0,COMM) FROM EMP;

③ SELECT GREATEST(3000,1500,2100,5000),LEAST(3000,1500,2100,5000) FROM DUAL ;
SELECT DEPTNO, ENAME, SAL,
DECODE(GREATEST(SAL,4800),SAL,'HIGH',DECODE(GREATEST(SAL,3000),SAL,'MID','LOW'))
FROM EMP
ORDER BY DEPTNO;

- '=' 연산자만 사용 가능한 decode를 greatest 함수와 사용하여 범위 연산 수행

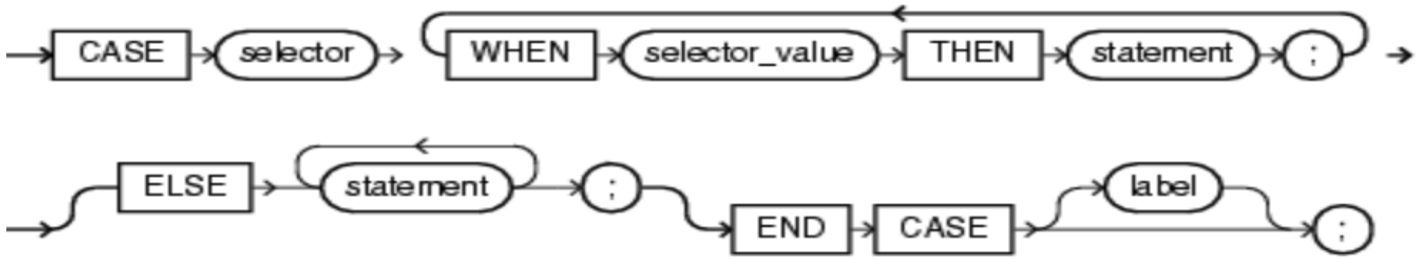
8. DECODE, CASE

❑ CASE

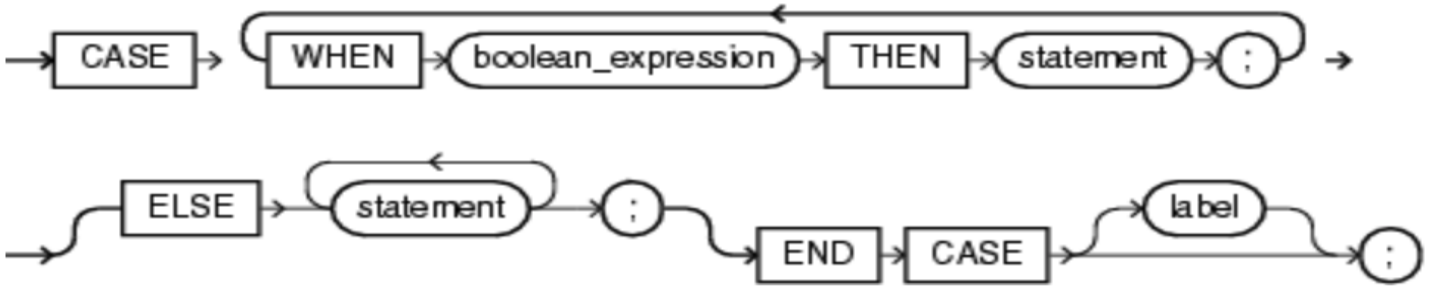
* **조건절(IF ~ ELSE IF ~ELSE) 연산자**
* **DECODE 연산자 기능 확장 & 성능 향상**
  - CASE: ANSI-SQL , DECODE : Oracle SQL
  - 기능: 비교 연산자, 논리 연산자 사용, SQL 연산자(ex LIKE , IN)
  - 성능: DECODE ~ 함수 , CASE ~ Statement (SQL, PL/SQL)

* CASE Syntax Diagram

simple_case_statement **::=**



searched_case_statement **::=**



8. DECODE, CASE

❑ CASE

```
① SELECT DEPTNO,ENAME,                                          -- Simple case
          CASE DEPTNO WHEN  '10'  THEN  'ACCOUNTING'            -- 암시적(x), 에러 원인? 수정후 실행
                      WHEN   20   THEN  'RESEARCH'
                      WHEN   30   THEN  'SALES'
                       ELSE             'ETC'
           END  AS DEPARTMENT
FROM  EMP
ORDER BY DEPTNO;
```

```
SELECT DEPTNO, ENAME,
       DECODE(DEPTNO,10,'ACCOUNTING',20,'RESEARCH',30,'SALES','ETC')
FROM  EMP
ORDER BY DEPTNO ;
```

```
② SELECT DEPTNO, ENAME, SAL,                                    -- Searched case
           CASE     WHEN SAL >= 4800                    THEN  'HIGH'     // 비교연산자
                    WHEN SAL BETWEEN 3000 AND 4799      THEN  'MID'      // SQL 연산자
                    WHEN SAL >= 1000 AND SAL <=2999     THEN  'LOW'      //  비교 & 논리 연산자
           ELSE                                                 'Passion pay'
           END      SAL_GRADE
 FROM EMP
 ORDER BY DEPTNO;
```

8. DECODE, CASE

❑ CASE

```
③ SELECT DEPTNO, ENAME, COMM,
          CASE    WHEN COMM >= 1000 THEN  'Great'
                  WHEN COMM >=  500 THEN  'Good'
                  WHEN COMM >=    0 THEN  'Bad'
          ELSE                            'Dreadful'          // ELSE에서 NULL  Catch
          END     COMM_GRADE
  FROM EMP
  ORDER BY DEPTNO;
```

```
④ 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_GRADE
FROM  EMP
ORDER BY SAL;
```

```
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  "Sal Grade"
FROM EMP
ORDER BY SAL;
```

- 여러 개의 WHEN 절에서 조건이 중첩되는 경우 올바른 처리방식 ?? When 기술순서

8. DECODE, CASE

❑ 과제

1) 부서별 차등 보너스를 계산하는 SQL 작성
- 10번 부서 급여의 0.3% , 20번부서 급여의 20%, 30번 부서 급여의 10%, 나머지 모든 부서 1%
- 부서 번호, 이름,직무,급여,보너스 출력
- 부서별, 최고 보너스 순서로 정렬
- 소수점 반올림
- 컬럼헤딩 변경시 컬럼 Alias 사용

2) DECODE 와 CASE이 차이점 정리후 발표

3) 아래의 SQL이 ③ SQL과 같은 결과를 출력하도록 NULL 처리

```
SELECT DEPTNO, ENAME, SAL,COMM,
            DECODE(GREATEST(COMM,1000),COMM,'Great',
                                    DECODE(GREATEST(COMM,500),COMM,'Good','Bad'))
FROM EMP
ORDER BY DEPTNO;
```

4) CASE 구문에서 LIKE 연산자 와 IN 연산자를 사용하는 임의의 SQL 작성

5) 실행후 결과 설명

```
SELECT DEPTNO, ENAME, DECODE(DEPTNO,10,'ACCOUNTING',20,'RESEARCH','ETC') FROM EMP ORDER BY DEPTNO;
SELECT DEPTNO, ENAME, DECODE(DEPTNO,10,'ACCOUNTING',20,'RESEARCH') FROM EMP ORDER BY DEPTNO;
```

8. DECODE, CASE

❑ 과제

6) 중첩 decode를 사용한 아래의 결과와 동일한 결과를 decode + || (합성연산자)를 사용하여 작성 하십시요

```
select deptno,job,decode(deptno, 10 ,decode(job,'CLERK','OK','NO'),'NO')
from emp
order by deptno,job;
```

9. ROWNUM

❑ ROWNUM

* For each row returned by a query, the ROWNUM **pseudocolumn returns a number indicating the order in which Oracle selects the row from a table** or set of joined rows.
The first row selected has a ROWNUM of 1, the second has 2, and so on.

// SQL-Developer 메뉴 창 > Configure Window > 세로로분할(V) 후 왼쪽창(① 실행) 오른쪽창(② 실행)하여 ROWNUM이 각 행에 부여되는 고유번호인지 비교

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 : **WHERE > ROWNUM > ORDER BY**

④ SELECT ENAME,DEPTNO,SAL FROM EMP WHERE ROWNUM = 1; // O , 1 Row = 1

⑤ SELECT ENAME,DEPTNO,SAL FROM EMP WHERE ROWNUM = 3; // X

* 1 Row != 3 ➔ 필터링 ➔ 2 Row가 1 Row가 되어 1 Row != 3 비교를 반복

⑥ SELECT ENAME,DEPTNO,SAL FROM EMP WHERE ROWNUM > 3; // X

⑦ SELECT ENAME,DEPTNO,SAL FROM EMP WHERE ROWNUM <= 3; // O

⑧ SELECT ENAME,DEPTNO,SAL FROM EMP WHERE ROWNUM < 3; // O

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;

// 연산자 우선순위 , 좋은 방식의 SQL? , OPTIMIZER(SQL 최적화기)가 AND 와 OR중 선호하는 연산자는?

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 ENAME,JOB,SAL FROM EMP WHERE JOB != 'CLERK';
⑧ SELECT ENAME,JOB,SAL FROM EMP WHERE JOB NOT IN('CLERK','MANAGER'); ⑨

❑ 과제

1) OR 연산시 중복되는 ROW는 어떻게 처리되는지 아래의 SQL을 실행한후 결과를 설명
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
```

11. 함수

❑ ORACLE DEFINED FUNCTION

① SINGLE ROW FUNCTION (단일행 함수)

문자함수
숫자함수
날짜함수
변환함수(DATA TYPE CONVERSION)
기타함수

② GROUP ROW FUNCTION (그룹행 함수)

❑ USER DEFINED FUNCTION (by PL/SQL)

<table>
<tr><td rowspan="6">ORACLE 제공 함수</td><td rowspan="5">단일행 함수</td><td>문자 함수</td></tr>
<tr><td>숫자 함수</td></tr>
<tr><td>날짜 함수</td></tr>
<tr><td>형변환 함수</td></tr>
<tr><td>기타 함수</td></tr>
<tr><td colspan="2">그룹행 함수</td></tr>
<tr><td>사용자 정의 함수</td><td colspan="2">PL/SQL로 사용자가 작성</td></tr>
</table>

11. 함수

❑ SINGLE ROW FUNCTION 개요 테스트

* Single Row function 정의 : 1개의 row에 적용되고 1 row당 1개의 결과 return.(출력)

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

① SELECT ENAME,EMPNO,SAL,COMM FROM EMP; // 14건 입력, 14건 출력

② SELECT ENAME,LOWER(ENAME),UPPER(LOWER(ENAME)),LENGTH(ENAME), // 14건 입력, 14건 출력
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 length(ENAME);

❑ GROUP ROW FUNCTION 개요 테스트

* Group Row function 정의: N개의 row에 적용(입력)되고 그룹당 1개의 결과 return(출력).

④ SELECT AVG(SAL),SUM(SAL),SUM(COMM),COUNT(*) FROM EMP; // 14건 입력, 1건 출력

⑤ SELECT DEPTNO,COUNT(*),SUM(SAL),AVG(SAL) FROM EMP // 부서번호 기준으로 그룹핑
GROUP BY DEPTNO; // 14건 입력, 3건 출력

⑥ SELECT DEPTNO,JOB,COUNT(*),SUM(SAL),AVG(SAL) FROM EMP // 부서번호,직무 기준으로 그룹핑
GROUP BY DEPTNO,JOB // 2개 컬럼의 조합의 그룹핑
ORDER BY DEPTNO,JOB; // 14건 입력, 9건 출력

11. 함수 – 단일행 함수

❑ 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

11. 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; // 문자열 내위치

④ SELSELECT ENAME,rpad(ENAME,10,' '),rpad(ENAME,10), rpad(ENAME,10,'*'),lpad(ENAME,10,'*+') FROM EMP;
SELECT length(rpad('X',1000,'X')), rpad('X',1000,'X') FROM DUAL; // dummy data

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(' ' from ' 대한민국 '), trim('*' from '**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;
- **DBMS 내부 저장시 : 영문자 1 Byte , 한글: 2~3 Bytes 할당 , length: 문자열 함수**

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

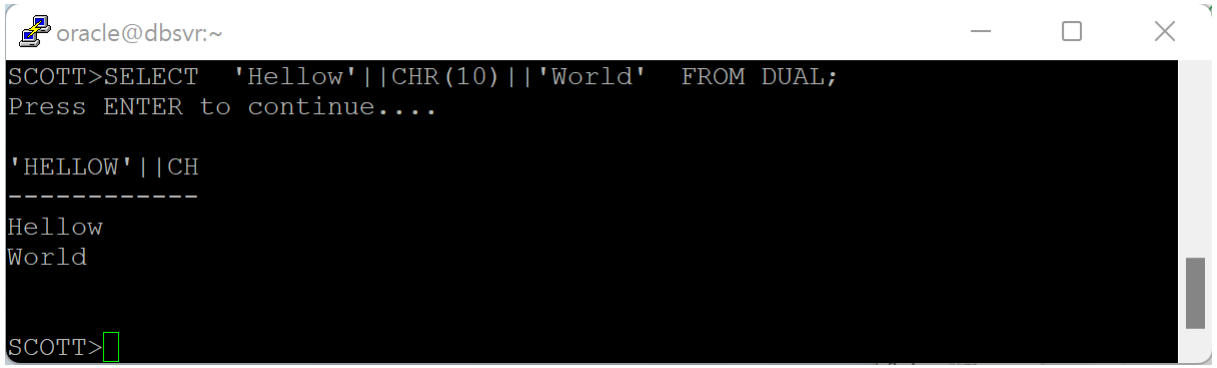
⑩ SELECT length('abcd'), vsize('abcd'), length('대한민국'), vsize('대한민국') FROM dual;
- VSIZE : 할당된 Bytes수 리턴

11. 함수 – 단일행 함수

❑ 문자 함수

⑪ SELECT ASCII('A'), ASCII('a') FROM dual; -- 65, 97

⑫ SELECT CHR(65), CHR(97) FROM DUAL;
SELECT 'Hellow'||CHR(10)||'World' FROM DUAL; -- Line feed ?? ⓐ run as script ⓑ double-click cell



⑬ SELECT job, TRANSLATE(job,'CL','ab'),TRANSLATE(job,'CL','a'),replace(job,'CL','ab') FROM EMP;

⑭ SELECT job, TRANSLATE(job,'!ABC','!'),TRANSLATE(job,'ABC',' ') FROM EMP;

11. 함수 – 단일행 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

❑ 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;

④ SELECT ENAME,SAL,SAL*0.053 as tax, round(SAL*0.053,0) as rtax FROM EMP; // 급여의5.3%세금,원단위 반올림

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;

11. 함수 – 단일행 함수

❑ 숫자 함수

① SELECT sign(-999),sign(999) FROM dual;

② SELECT 2**3,power(2,3), power(-2,3), power(-2,4) FROM dual; -- ??

③ variable x number
begin
:x := 2**3;
end;
/
print x

11. 함수 – 단일행 함수

❑ DATE type

* **DATE TYPE은 연산 가능**

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;

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

- DATE는 날짜와 시간 정보를 가지고 있다. ?? **왜 날짜만 보이고 시간은 보이지 않는가?**

④ ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY-MM-DD:HH24:MI:SS'; // 날짜&시간 표시 포맷 변경
SELECT ename,sysdate,hiredate FROM emp; // default date format ??
SELECT SYSDATE FROM DUAL;

❑ 과제

1) 세션(Session)의 의미 및 역할 2~3줄 정리 및 설명
연결(Connection) 과 세션(Session) 관계 및 차이 그림으로 그리고 설명

2) ALTER SESSION 명령어의 의미 및 적용 범위 설명
SQL-Developer에서 2개의 세션을 생성후 위의 예제를 사용하여 적용 범위 설명

11. 함수 – 단일행 함수

함수	설명
SYSDATE	DBMS Server의 현재 날짜와 시간을 DATE 타입으로 반환하는 함수 * DBMS 내부에서 DATE 타입은 숫자 데이타 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

❑ 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...

11. 함수 – 단일행 함수

❑ 날짜 함수

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;

❑ 과제

1) EXTRACT 함수의 결과 데이터 타입이 무엇인지 오라클 공식 메뉴얼을 검색하여 설명

2) 아래 SQL을 참고하여 해당월의 마지막 법정 영업일자를 구하는 SQL을 작성(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;
```

3) PROGRAMMING방식을 해결 하십시요

```
실습:  ALTER SESSION SET NLS_DATE_FORMAT = 'RR/MM/DD';
       SELECT * FROM EMP WHERE HIREDATE LIKE '82%';      // 검색(O)

       ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MON-RR';
       SELECT * FROM EMP WHERE HIREDATE LIKE '82%';      // 날짜포맷이 변경되면 기존 검색(X)
       SELECT * FROM EMP WHERE HIREDATE LIKE '%82';      // 검색조건을 바꾸어야 검색(O)
```

4) 다음의 날짜 포맷을 조사하여 [암기후] 설명

5) DBMS Server의 현재 시간 + 1/100초를 조회하는 SQL 작성

구분	날짜 포맷
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

11. 함수 – 단일행 함수

❑ 과제

```
select systimestamp, to_char(systimestamp,'YYYY-MM-DD HH24:MI:SS.FF2'),
                     to_char(systimestamp,'YYYY-MM-DD HH24:MI:SS.FF3'),
                     to_char(systimestamp,'YYYY-MM-DD HH24:MI:SS.FF')        -- TIMESTAMP WITH TIME ZONE
from dual;

select sysdate,systimestamp from dual;
ALTER SESSION SET NLS_TIMESTAMP_TZ_FORMAT ='YYYY-MM-DD HH24:MI:SS.FF3';

select sysdate,systimestamp from dual;
ALTER SESSION SET NLS_TIMESTAMP_TZ_FORMAT ='YYYY-MM-DD HH24:MI:SS.FF';

select sysdate,systimestamp from dual;
```

❑ 과제

```
drop table d_test;
create table d_test
(
    orderdt     varchar2(6),                    -- 가변길이  6 Bytes
    paydt       date,                           -- 고정길이  7 Bytes
    deliverdt   timestamp                       -- 고정길이 11 Bytes  , 밀리세컨드 (default) , 나노세컨드
);

desc d_test

insert into d_test values('220701',sysdate,systimestamp);

select * from d_test;
```

11. 함수 – 단일행 함수

❑ Data Type 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(format modifier): 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;
- 변환시 포맷문자(999,999)가 데이터 보다 크거나 같아야 정상 출력

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;

11. 함수 – 그룹행 함수

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;

- **그룹행 함수에서는 NULL을 무시하고 연산 수행**

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

- DISTINCT 연산 수행후 COUNT 수행

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

- SUM(COMM)/COUNT(*) 와 AVG 결과가 다른 이유?

⑤ 그룹행 함수 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
```

❑ 과제

1) ② SQL에서 COUNT(*) 와 COUNT(EMPNO)의 결과가 항상 일치하는 이유를 DESC EMP 수행하여 관찰후 설명

2) [선택] 아래의 SQL 실행

```
select * from all_arguments where object_name = 'LENGTH';
select * from v$sqlfn_metadata where name = 'SUM';
```

11. 함수 – 기타함수

Session 환경 함수

```
select USER,UID from dual;
select   USERENV('CLIENT_INFO'),
         USERENV('TERMINAL'),
         USERENV('LANG'),
         USERENV('LANGUAGE')
from dual;

select   sys_context('USERENV','SERVER_HOST'),
          sys_context('USERENV','INSTANCE'),
          sys_context('USERENV','INSTANCE_NAME')
from dual;

select sys_context('USERENV','SID'),
        sys_context('USERENV','IP_ADDRESS'),
        sys_context('USERENV','HOST'),
        sys_context('USERENV','OS_USER'),
        sys_context('USERENV','TERMINAL'),
        sys_context('USERENV','CLIENT_PROGRAM_NAME')
from dual;

select   sys_context('USERENV','NLS_DATE_FORMAT'),
        sys_context('USERENV','NLS_DATE_LANGUAGE'),
        sys_context('USERENV','NLS_TERRITORY')
from dual;
```

[과제] 위의 데이터를 V$SESSION에서 확인하는 SQL 작성

12. GROUP BY , HAVING

❑ GROUP BY

* 기준 컬럼으로 데이터 그룹핑

① SELECT DEPTNO,COUNT(*) FROM SCOTT.EMP GROUP BY DEPTNO;

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

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

③ SELECT DEPTNO,AVG(SAL) ,SUM(SAL) FROM EMP
GROUP BY DEPTNO
ORDER BY DEPTNO;

④ SELECT DEPTNO,ROUND(AVG(SAL),1) ,SUM(SAL) FROM EMP
GROUP BY DEPTNO
ORDER BY DEPTNO;

⑤ SELECT COMM,COUNT(*) FROM EMP GROUP BY COMM; // NULL 그룹핑 대상인가 ?

12. GROUP BY , HAVING

❑ HAVING

* GROUP BY 결과 집합의 조건절

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

DEPTNO	COUNT(*)	SUM(SAL)	ROUND(AVG(SAL),1)
30	6	9400	1566.7
20	5	10875	2175
10	3	8750	2916.7

② SELECT DEPTNO,COUNT(*),SUM(SAL),ROUND(AVG(SAL),1) FROM EMP
GROUP BY DEPTNO
HAVING SUM(SAL) >= 9000 ;

DEPTNO	COUNT(*)	SUM(SAL)	ROUND(AVG(SAL),1)
30	6	9400	1566.7
20	5	10875	2175

③ SELECT DEPTNO,COUNT(*),SUM(SAL),ROUND(AVG(SAL),1) FROM EMP
GROUP BY DEPTNO
HAVING DEPTNO in (10,20);

DEPTNO	COUNT(*)	SUM(SAL)	ROUND(AVG(SAL),1)
20	5	10875	2175
10	3	8750	2916.7

④ SELECT DEPTNO,ROUND(AVG(SAL),1) ,SUM(SAL) FROM EMP
WHERE DEPTNO IN (10,20)
GROUP BY DEPTNO
HAVING SUM(SAL) >= 9000
ORDER BY DEPTNO DESC;

- GROUP 실행순서 : FROM > WHERE > GROUP BY > HAVING > SELECT > ORDER BY

12. GROUP BY , HAVING

❑ 과제

1) ② SQL 참조, GROUP BY 실행 방식이 10g 부터 정렬된 결과 집합이 되지 않아 ③ SQL 처럼 정렬된 결과를 원하면 ORDER BY 연산을 해야 한다.
Oracle 9i 에서 GROUP BY 연산 방식 과 Oracle 10g 에서 GROUP BY 연산 방식의 차이점 조사 발표

2) 부서별 최대 ,최소 급여를 조회 하십시요, 부서별 정렬

3) 입사년도별 최대 ,최소 급여를 조회 하십시요, 년도별 정렬
hint> group by to_char(hiredate,'YYYY')

3) 다음과 같은 결과를 생성하는 SQL문장을 작성

10번부서	20번부서	30번부서
3	5	6

❑ 과제

1) Customer 테이블에서 sample(10)을 사용하여 10% 이내의 임의의 Row를 조회하는 SQL 작성 , sample(5)을 사용하여 5% 이내의 항상 동일한 Row를 조회하는 SQL 작성 하십시요

2) 부서별 최대 ,최소 급여를 조회 하십시요

3) 다음과 같은 결과를 생성하는 SQL문장을 작성

10번부서	20번부서	30번부서
3	5	6