

[참고1] 원격 오라클 DBMS 접속 환경

클라이언트^{Client}의 TNSNAMES.ORA 파일에 기록된 접속 대상 DBMS 정보를 이용하여 접속함



```
VCC0 =  
  (DESCRIPTION =  
    (ADDRESS_LIST =  
      (ADDRESS = (PROTOCOL = TCP)(HOST = VCC0)(PORT = 1521))  
    )  
    (CONNECT_DATA =  
      (SERVICE_NAME = VCC0)  
    )  
  )
```

오라클 설치 폴더\NETWORK\ADMIN\TNSNAMES.ORA

```
SID_LIST_VCC0 =  
  (SID_LIST =  
    (SID_DESC =  
      (SID_NAME = PLSExtProc)  
      (ORACLE_HOME = C:\oracle\app\oracle\product\10.2.0\server)  
      (PROGRAM = extproc)  
    )  
  )  
VCC0 =  
  (DESCRIPTION_LIST =  
    (DESCRIPTION =  
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))  
      (ADDRESS = (PROTOCOL = TCP)(HOST = VCC0)(PORT = 1521))  
    )  
  )
```

오라클 설치 폴더\NETWORK\ADMIN\LISTENER.ORA

[참고2] SQL 작성 표준 예시

가로 길이는 최대 80자
이내

세로 줄
에 맞춤

괄호
앞/뒤로
공백

```
SELECT TO_DATE(A.DELY_DATE, 'YYYYMMDD') DELYED_DATE_D,  
       B.ORDER_NO, B.SALES_SEQ, COUNT(*), MAX(D.CNT),  
       :  
FROM   TM_SAL_MST A,  
       TM_SAL_DTL B,  
       (  
       SELECT CUST_CD, COM_CODE, FULL_NAME,  
              CREDIT_TO  
       FROM   TB_CCUST  
       WHERE  CUST_CD = BILL_TO  
       ) C,  
       TM_MODEL D  
WHERE  B.ORDER_NO = A.ORDER_NO  
AND    B.COM_CODE = A.COM_CODE  
...  
AND    EXISTS ( SELECT 'Y'  
                FROM DEPT E  
                WHERE E.DEPT_NO = A.DIV_CD )  
AND    B.ORDER_NO IN  
       ( SELECT F.ORDER_NO  
       FROM   TM_SAL_MST F  
       WHERE  F.TXN_TYPE = 'Y' )  
GROUP BY DELYED_DATE_D, B.ORDER_NO, B.SALES_SEQ  
HAVING COUNT(*) > 0  
ORDER BY B.ORDER_NO
```

테이블은 다른 줄에 기술하고
테이블 별칭 Alias은 A, B, C, D
순서로 기술

함수 괄호 사이는
공백 없음

',' 전엔 공백 없고
다음엔 공백을 넣는다

Sub query가 길면
다음 줄에 기술

[참고 3-1] 샘플 테이블 구조(교재용)

[COUNTRY - 국가 정보 테이블]

	COUNTRY_ID	COUNTRY_NAME
설명	국가 코드	국가 이름
데이터 타입	CHAR(2)	VARCHAR2(30)
제약조건	PK	NOT NULL

[LOCATION - 지역 정보 테이블]

	LOCATION_ID	COUNTRY_ID	LOC_DESCRIBE
설명	지역 코드	국가 코드	지역 설명
데이터 타입	CHAR(2)	CHAR(2)	VARCHAR2(40)
제약조건	PK	FK,NN	
참조테이블		COUNTRY	
참조 컬럼		COUNTRY_ID	

[DEPARTMENT - 부서 정보 테이블]

	DEPT_ID	DEPT_NAME	LOC_ID
설명	부서코드	부서 이름	지역 코드
데이터 타입	CHAR(2)	VARCHAR2(30)	CHAR(2)
제약조건	PK		FK, NOT NULL
참조테이블			LOCATION
참조 컬럼			LOCATION_ID

[SAL_GRADE - 급여 정보 테이블]

	SLEVEL	LOWEST	HIGHEST
설명	급여 레벨	하한 값	상한 값
데이터 타입	CHAR(1)	NUMBER	NUMBER
제약조건	PK		

[JOB - 직급 정보 테이블]

	JOB_ID	JOB_TITLE	MIN_SAL	MAX_SAL
설명	직급코드	직급 이름	최소 급여	최대 급여
데이터 타입	CHAR(2)	VARCHAR2(35)	NUMBER	NUMBER
제약조건	PK			

[EMPLOYEE - 사원 정보 테이블]

	EMP_ID	EMP_NAME	EMP_NO	EMAIL	PHONE	HIRE_DATE
설명	사번	이름	주민번호	e-mail	전화번호	입사일
데이터 타입	CHAR(3)	VARCHAR2(20)	CHAR(14)	VARCHAR2(25)	VARCHAR2(12)	DATE
제약조건	PK	NOT NULL	NOT NULL, UNIQUE			
	JOB_ID	SALARY	BONUS_PCT	MARRIAGE	MGR_ID	DEPT_ID
설명	직급 코드	급여	보너스 지급률	결혼여부	관리자 사번	부서 코드
데이터 타입	CHAR(2)	NUMBER	NUMBER	CHAR(1)	CHAR(3)	CHAR(2)
제약조건	FK			DEFAULT 'N'	FK	FK
참조테이블	JOB				EMPLOYEE	DEPARTMENT
참조 컬럼	JOB_ID				EMP_ID	DEPT_ID

[참고 3-2] 샘플 테이블 구조(SET Operator)

[ROLE_HISTORY - 직원 ROLE 이력 테이블]

	EMP_ID	START_DATE	END_DATE	ROLE_NAME	DEPT_ID
설명	사번	ROLL 시작일	ROLL 종료일	ROLL 이름	부서코드
데이터 타입	CHAR(3)	DATE	DATE	VARCHAR2(50)	CHAR(2)
제약조건					

[EMPLOYEE_ROLE - 직원 현재 ROLE 정보 테이블]

	EMP_ID	EMP_NAME	EMP_NO	EMAIL	PHONE	HIRE_DATE
설명	사번	이름	주민번호	e-mail	전화번호	입사일
데이터 타입	CHAR(3)	VARCHAR2(20)	CHAR(14)	VARCHAR2(25)	VARCHAR2(12)	DATE
제약조건	PK	NOT NULL	NOT NULL, UNIQUE			
	JOB_ID	SALARY	BONUS_PCT	MARRIAGE	MGR_ID	DEPT_ID
설명	직급 코드	급여	보너스 지급률	결혼여부	관리자 사번	부서 코드
데이터 타입	CHAR(2)	NUMBER	NUMBER	CHAR(1)	CHAR(3)	CHAR(2)
제약조건	FK			DEFAULT 'N'	FK	FK
참조테이블	JOB				EMPLOYEE	DEPARTMENT
참조 컬럼	JOB_ID				EMP_ID	DEPT_ID

	ROLE_NAME
설명	ROLE 이름
데이터 타입	VARCHAR2(50)
제약조건	
참조테이블	
참조 컬럼	

[참고 4] ORDER BY 사용 시 NULL 위치 조정 구문

NULLS FIRST/LAST 키워드를 사용하여 NULL 표시 위치 변경 가능

```
SELECT  EMP_NAME,  
        HIRE_DATE,  
        DEPT_ID  
FROM    EMPLOYEE  
WHERE   HIRE_DATE > TO_DATE('20030101','YYYYMMDD')  
ORDER BY DEPT_ID DESC NULLS LAST,  
        HIRE_DATE ASC,  
        EMP_NAME;
```

EMP_NAME	HIRE_DATE	DEPT_ID
강중훈	04/04/30	90
한선기	90/04/01	90
엄정하	04/07/21	80
성해교	03/08/16	50
정지현	04/07/15	50
박하일	04/11/10	50
이중기	04/10/01	20
감우섭	05/07/31	20
고승우	03/04/11	10
염정하	03/09/17	
심하균	04/09/30	

[참고 5-1] Analytic Functions - RANK

지정한 값의 순위를 반환하거나, 지정한 값을 기준으로 순위를 지정

```
SELECT RANK(2300000) WITHIN GROUP (ORDER BY SALARY DESC) AS RANK  
FROM   EMPLOYEE;
```

RANK
15

"SALARY" 값을 큰 순서대로 정렬했을 때, "2300000"은 몇 번째 순위인지 반환

[결과(일부)]

```
SELECT EMP_NAME, SALARY,  
       RANK() OVER (ORDER BY SALARY DESC) AS RANK  
FROM   EMPLOYEE;
```

"SALARY" 값을 큰 순서대로 정렬한 순위를 반환

EMP_NAME	SALARY	RANK
한선기	90000000	1
강중훈	55000000	2
조재형	38000000	3
최만식	36000000	4
안석규	35000000	5
권상후	34100000	6
김순이	34000000	7
임영애	26400000	8
정도연	26000000	9
박하일	26000000	9
이준기	25000000	11

[참고 5-2] Analytic Functions - RANK

RANK를 이용한 Top-N 분석 예

```
SELECT *  
FROM (SELECT EMP_NAME, SALARY,  
            RANK() OVER (ORDER BY SALARY DESC) AS RANK  
      FROM EMPLOYEE)  
WHERE RANK <= 5;
```

EMP_NAME	SALARY	RANK
한선기	9000000	1
강중훈	5500000	2
조재형	3800000	3
최만식	3600000	4
안석규	3500000	5

인라인 뷰를 사용하여 순위를 활용할 수 있음

```
SELECT *  
FROM (SELECT EMP_NAME, SALARY,  
            RANK() OVER (ORDER BY SALARY DESC) AS RANK  
      FROM EMPLOYEE)  
WHERE RANK = 11;
```

EMP_NAME	SALARY	RANK
김술오	2500000	11
이중기	2500000	11
감우섭	2500000	11

ROWNUM을 사용하는 경우와 달리 특정 순위를 지정할 수 있음

[참고 6] ROWID

데이터 파일에 저장된 각 행의 물리적인 위치를 나타내는 논리적인 가상 컬럼

```
SELECT DEPT_ID, ROWID
FROM   DEPARTMENT;
```

DEPT_ID	ROWID
10	AAACikAAEAAAAlAAB
20	AAACikAAEAAAAlAAA
30	AAACikAAEAAAAlAAG
50	AAACikAAEAAAAlAAC
60	AAACikAAEAAAAlAAD
80	AAACikAAEAAAAlAAE
90	AAACikAAEAAAAlAAF

[ROWID Format]

000000FFFBBBBBBRRR	
000000	객체 고유번호
FFF	데이터 파일 번호
BBBBBB	데이터 블록 번호
RRR	행 번호