Java Web Programming 입문 14

(Oracle PL/SQL #06)

오늘의 키워드

- 집합연산자
  - UNION
  - UNION ALL
  - MINUS
  - INTERSECT
- ORDER BY
- GROUP BY, HAVING
- Query의 해석순서
- WITH

집합 연산자

- 2개 이상의 SQL문의 결과를 연결
  - UNION
  - UNION ALL
  - MINUS
  - INTERSECT

- 연결되는 SELECT문들의 Column의 개수, Data Type이 일치해야 함

집합 연산자

- UNION
  - 합집합의 의미

```
SQL> SELECT EMPNO FROM EMP
     UNION
     SELECT DEPTNO, DNAME FROM DEPT
```

```
SQL> SELECT DNAME, DEPTNO FROM DEPT
     UNION
     SELECT DEPTNO, DNAME FROM DEPT
```

```
SQL> SELECT EMPNO, ENAME FROM EMP
     UNION
     SELECT DEPTNO, DNAME FROM DEPT
```

```
SQL> SELECT DEPTNO FROM EMP
     UNION
     SELECT DEPTNO FROM DEPT
```

집합 연산자

- UNION ALL
  - 연결될 SQL문의 결과가 모두 출력

```
SQL> SELECT EMPNO, ENAME FROM EMP
     UNION ALL
     SELECT DEPTNO, DNAME FROM DEPT
```

```
SQL> SELECT DEPTNO FROM EMP
     UNION ALL
     SELECT DEPTNO FROM DEPT
```

- UNION VS UNION ALL
  - 중복제거

집합 연산자

- MINUS
  - 차집합의 의미

```
SQL> SELECT DEPTNO FROM DEPT
     MINUS
     SELECT DEPTNO FROM EMP
```

```
SQL> SELECT DNAME, DEPTNO FROM DEPT
     MINUS
     SELECT ENAME, DEPTNO FROM EMP
```

```
SQL> SELECT DEPTNO FROM DEPT
     MINUS
     SELECT DEPTNO FROM DEPT
```

집합 연산자

- INTERSECT
  - 교집합의 의미

```
SQL> SELECT DEPTNO FROM EMP
     INTERSECT
     SELECT DEPTNO FROM DEPT
```

```
SQL> SELECT DEPTNO FROM DEPT WHERE DEPTNO = 10
     INTERSECT
     SELECT DEPTNO FROM DEPT
```

```
SQL> SELECT DEPTNO FROM DEPT
     INTERSECT
     SELECT DEPTNO FROM DEPT
```

ORDER BY

- 결과물 정렬시 사용

```
- ORDER BY [Sorting_Column_Name] ASC (Or DESC)
 -- ASC  : 오름차순 Option (Default)
 -- DESC : 내림차순 Option
```

```
SQL> SELECT ENAME, SAL
       FROM EMP
      ORDER BY SAL ASC;
```

```
SQL> SELECT ENAME, SAL
       FROM EMP
      ORDER BY SAL DESC;
```

ORDER BY

- 결과물 정렬시 사용

```
-- 별칭(Alias) 사용도 가능!
SQL> SELECT ENAME, SAL * 12 AnnSAL
       FROM EMP
      ORDER BY AnnSAL DESC;
```

```
SQL> SELECT ENAME, SAL
       FROM EMP
      ORDER BY 2 DESC
      -- 현재 SELECT문의 두번째 Column으로 정렬할것!
```

```
SQL> SELECT ENAME, SAL
       FROM EMP
      ORDER BY SAL DESC, ENAME ASC;
       --       1차 정렬, 2차 정렬
```

GROUP BY, HAVING

- 부서별(DEPTNO) 평균 급여를 알고 싶다?

```
SQL> SELECT AVG(SAL)
       FROM EMP
      WHERE DEPTNO = 10;
   -- WHERE DEPTNO = 20;
   -- WHERE DEPTNO = 30;
```

```
SQL> SELECT AVG(SAL)
       FROM EMP
      WHERE DEPTNO = 10
     UNION ALL
     SELECT AVG(SAL)
       FROM EMP
      WHERE DEPTNO = 20
     UNION ALL
     SELECT AVG(SAL)
       FROM EMP
      WHERE DEPTNO = 30
```

- 한계?

GROUP BY, HAVING

- 그룹으로 묶어버리자!

```
SQL> SELECT AVG(SAL)
       FROM EMP
      WHERE DEPTNO = 10
     UNION ALL
     SELECT AVG(SAL)
       FROM EMP
      WHERE DEPTNO = 20
     UNION ALL
     SELECT AVG(SAL)
       FROM EMP
      WHERE DEPTNO = 30
```

```
SQL> SELECT AVG(SAL)
       FROM EMP
      GROUP BY DEPTNO;
```

```
SQL> SELECT COUNT(*), DEPTNO
       FROM EMP
      GROUP BY DEPTNO;
```

```
SQL> SELECT EMPNO, AVG(SAL)
       FROM EMP
      GROUP BY DEPTNO;
```

```
-- 부서별, 직종별 평균값
SQL> SELECT DEPTNO, JOB, AVG(SAL)
       FROM EMP
      GROUP BY DEPTNO, JOB
```

GROUP BY, HAVING

- 조회될 그룹을 제한?
  - 부서별(DEPTNO) 평균 급여를 알고 싶은데 평균 급여 2,000이 안되는 부서는 제외

```
SQL> SELECT DEPTNO, AVG(SAL)
       FROM EMP
      GROUP BY DEPTNO
     HAVING AVG(SAL) >= 2000;
```

```
SQL> SELECT DEPTNO, AVG(SAL)
       FROM EMP
      GROUP BY DEPTNO
     HAVING ENAME LIKE '%S%'
```

```
- WHERE  CONDITION : 결과로 내보내줄 Table에 있는 Row를 제한
- HAVING CONDITION : 결과로 내보내줄 Group을 제한
```

Query의 해석 순서

- SELECT Query에 나오는 모든 명령이 끝났다!

```
SELECT     Column1, Column2, ... , ColumnN
  FROM     Table1
 WHERE     [WHERE Condition]
 GROUP BY Column1, Column2, ... , ColumnN
HAVING     [HAVING Condition]
 ORDER BY [Sorting Column]
```

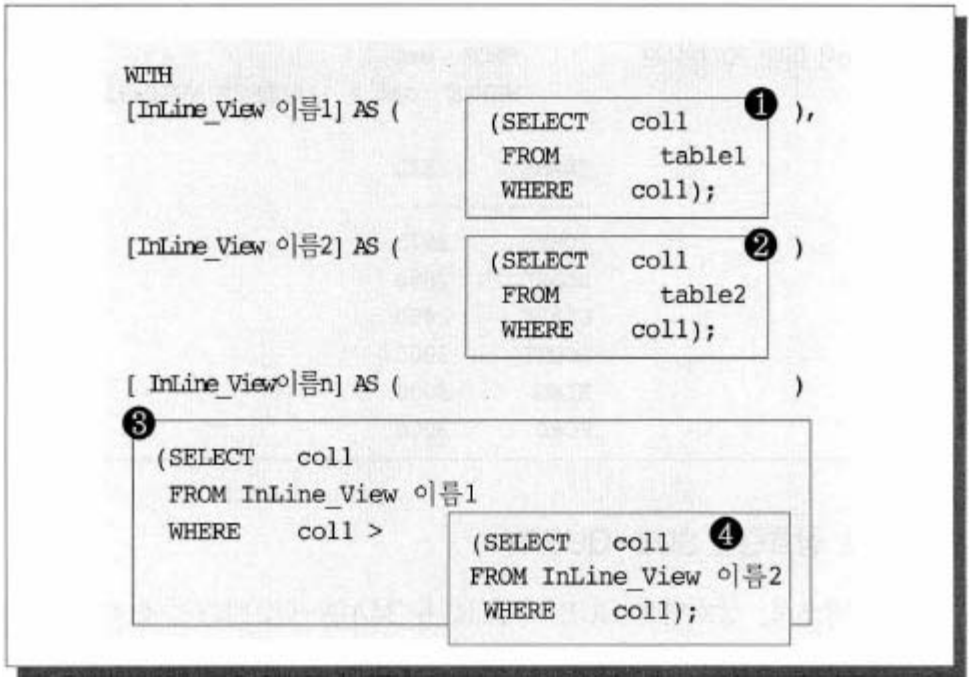
- 해석순서

```
4) SELECT
1)   FROM
2)  WHERE
3)  GROUP BY
5) HAVING
6)  ORDER BY
```

```
1) TABLE이 존재하는지 찾아본다.
2) WHERE 절에 조건을 TRUE로 만족하는 Row들을 결과로 반환
3) 결과에 반환되는 Row들을 GORUP화
   (WHERE를 만족하지 않으면 GROUP에 끼지도 못함)
4) SELECT 절에 있는 Column들과 GROUP FUNCTION 있다면 적용
5) 해당 Group 중 HAVING절 만족하는 Group 들이 결과로 반환
6) 정렬 옵션이 있다면 Sorting 후 최종 결과 출력
```

WITH

- InLine_View (Table 같은놈)



WITH

- Example

```
SQL> WITH DEPT_COSTS AS
          (SELECT   D.DEPTNO, SUM (E.SAL) AS DEPT_TOTAL
               FROM EMP E, DEPT D
              WHERE E.DEPTNO = D.DEPTNO
           GROUP BY D.DEPTNO),
          AVG_COST AS
          (SELECT   SUM (DEPT_TOTAL) / COUNT (*) AS DEPT_AVG
               FROM DEPT_COSTS)
     SELECT   *
         FROM DEPT_COSTS
        WHERE DEPT_TOTAL > (SELECT DEPT_AVG
                              FROM AVG_COST)
     ORDER BY DEPTNO;
```

WITH

- Example