2. 데이터 조회를 위한 고급 기법들

1. 집합 연산자들
2. 단일행 함수
3. **그룹함수**

목차

- 기본적인 GROUP 함수
- GROUP BY절
- HAVING 절
- 자동 소계/합계 함수
- GROUPPING 함수 살펴보기
- 그룹함수 연습문제
- 기타 그룹 함수

그룹함수의 종류

함수이름	의 미	사 용 예
COUNT	입력되는 데이터들의 건수를 출력	COUNT(sal)
SUM	입력되는 데이터들의 합계값을 출력	SUM(sal)
AVG	입력되는 데이터들의 평균값을 출력	AVG(sal)
MAX	입력되는 데이터들 중 최고 값을 출력	MAX(sal)
MIN	입력되는 데이터들 중 최저 값을 출력	MIN(sal)
STDDEV	입력되는 데이터 값들의 표준 편차값 출력	STDDEV(sal)
VARIANCE	입력되는 데이터 값들의 분산값 출력	VARIANCE(sal)
ROLLUP	입력되는 데이터들의 소계값을 자동으로 계산해서 출력	아래 예 참조
CUBE	입력되는 데이터들의 소계 및 전체 총계를 자동 계산 후 출력	아래 예 참조
GROUPING	해당 칼럼이 그룹에 사용되었는지 여부를 1 또는 0으로 반환	아래 예 참조
GROUPINGSET	한번의 질의로 여러개의 그룹화 가능	아래 예 참조
LISTAGG		아래 예 참조
PIVOT		아래 예 참조
LAG		아래 예 참조
LEAD		아래 예 참조
RANK		아래 예 참조
DENSE_RANK		아래 예 참조
누계집계하기		아래 예 참조

3.그룹함수- 기본 그룹 함수

- **COUNT 함수**
  - 데이터의 총 건수를 반환

```
SELECT COUNT(*), COUNT(hpage)  FROM professor;
```

- **SUM 함수**
  - 입력된 데이터들의 합계 값을 구하는 함수

```
SELECT name, bonus FROM professor;
```

```
SELECT count(bonus), sum(bonus) FROM professor;
```

- **AVG**
  - 입력된 값들의 평균을 구하는 함수

```
SELECT count(bonus), sum(bonus), avg(bonus), avg(NVL(bonus,0))
FROM professor;
```

3.그룹함수- 기본 그룹 함수

- **MAX/MIN**

```
SELECT MAX(sal), MIN(sal) FROM emp;
SELECT MAX(hiredate), MIN(hiredate) FROM emp;
```

-속도와 성능 부분에서 문제가 될 수 있으므로 인덱스를 활용하는 방법을 사용 할 것을 적극 권장함

- **STDDEV 함수/ VARIANCE 함수**

```
SELECT STDDEV(pay), VARIANCE(pay) FROM professor;
```

3. 그룹함수- GROUP BY 절

GROUP BY

- 특정조건으로 세부적인 그룹화하기

- Professor 테이블에서 학과별 교수들의 평균 급여를 출력하세요.

```
SELECT deptno, AVG(NVL(pay,0)) "평균급여"
FROM professor
GROUP BY deptno;
```

- Professor 테이블에서 학과별, 직급별로 교수들의 평균 급여를 출력하세요.

```
SELECT deptno, position, AVG(NVL(pay,0)) "평균급여"
FROM professor
GROUP BY deptno, position;
```

GROUP BY 절 사용 시 주의 사항

1. **SELECT 절에 사용된 그룹함수 이외의 칼럼이나 표현식은 반드시 GROUP BY 절에 사용되어야 합니다. 그렇지 않을 경우 아래와 같은 에러가 발생합니다.**

```
SELECT deptno, position, AVG(NVL(pay,0)) "평균급여"
FROM professor GROUP BY deptno;
```

2. GROUP BY 절에 사용된 칼럼은 SELECT 절에 사용되지 않아도 됩니다.

```
SELECT deptno, AVG(NVL(pay,0)) "평균급여"
FROM professor GROUP BY deptno, position;
```

3. GROUP BY 절에는 반드시 칼럼 명이 사용되어야 하며 칼럼 Alias 는 사용하면 안됩니다.

```
SELECT deptno "dno", AVG(NVL(pay,0)) "평균급여"
FROM professor GROUP BY dno;
```

HAVING 절

HAVING

- GROUP BY 절에 조건을 지정할 때 사용

- 평균 급여가 450 이상인 부서의 부서번호와 평균급여를 구하세요.

```
SELECT deptno, AVG(NVL(pay,0))
FROM professor
WHERE AVG(pay)>450
GROUP BY deptno;
```

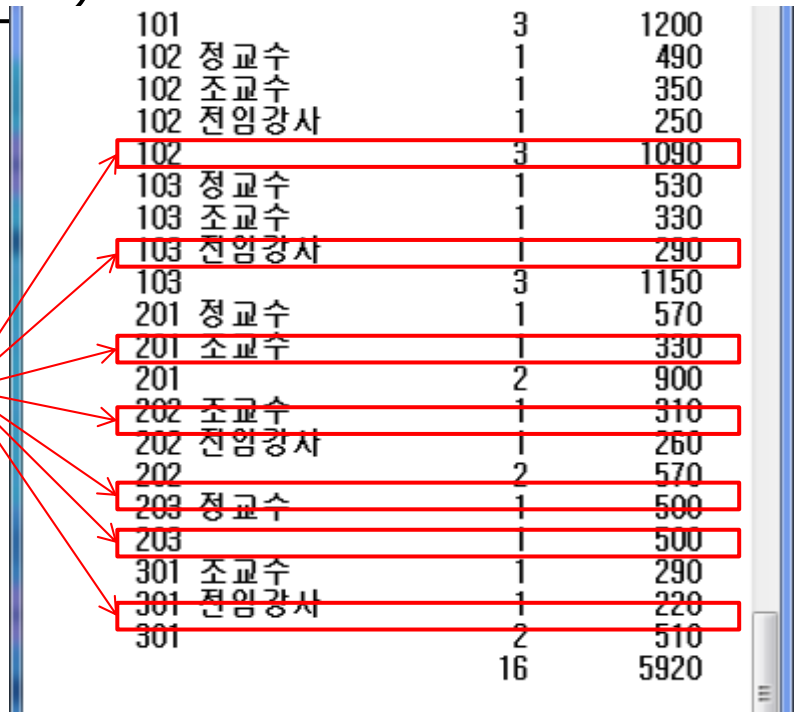
```
SELECT deptno, AVG(NVL(pay,0))
FROM professor
GROUP BY deptno
HAVING AVG(pay)>450;
```

자동 소계/합계 함수

- **ROLLUP 함수**
  - 자동으로 소계 값을 구해주는 함수

```
SELECT deptno, position,
COUNT(*), SUM(pay)
FROM professor
GROUP BY ROLLUP(deptno, position);
```

이 부분들이 ROLLUP에 의해 자동으로 구해진 소계 부분 입니다.



실습

```
SELECT deptno, position, SUM(pay)
FROM professor
GROUP BY position, ROLLUP(deptno);
```

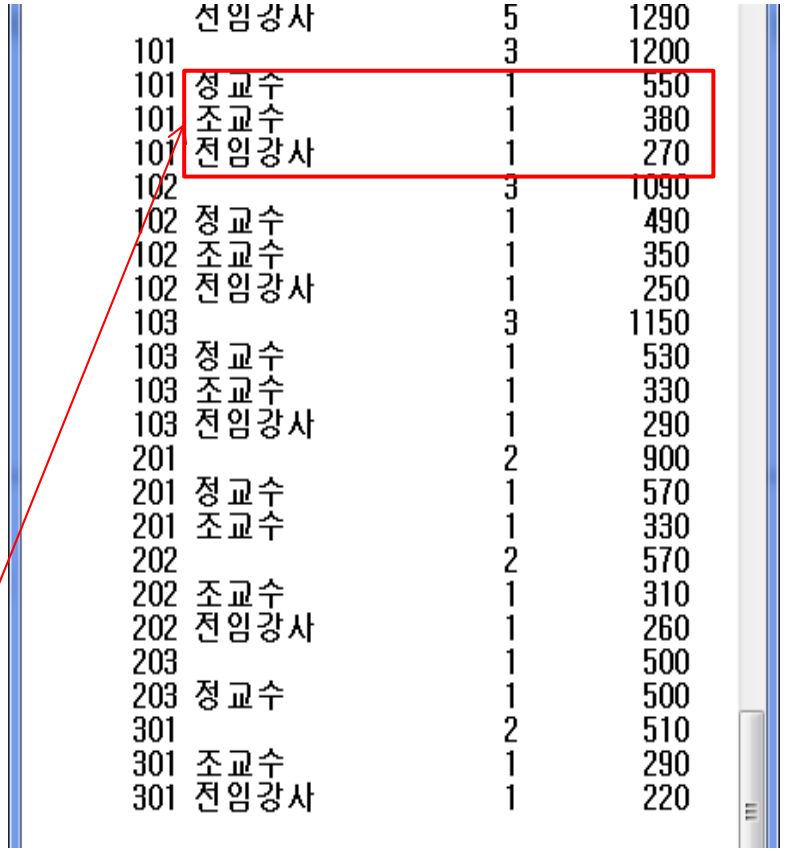
```
SELECT deptno, position, SUM(pay)
FROM professor
GROUP BY deptno, ROLLUP(position);
```

CUBE 함수

- ROLLUP 함수와 같이 각 소계도 출력하고 전체 총계까지 출력.

```
SELECT deptno, position,
count(*), SUM(pay)
FROM professor
GROUP BY CUBE(deptno, position);
```

이 부분이 ROLLUP 함수와 다른 전체 총계 출력부분입니다



	전임강사	5	1290
101		3	1200
101	정교수	1	550
101	조교수	1	380
101	전임강사	1	270
102		3	1090
102	정교수	1	490
102	조교수	1	350
102	전임강사	1	250
103		3	1150
103	정교수	1	530
103	조교수	1	330
103	전임강사	1	290
201		2	900
201	정교수	1	570
201	조교수	1	330
202		2	570
202	조교수	1	310
202	전임강사	1	260
203		1	500
203	정교수	1	500
301		2	510
301	조교수	1	290
301	전임강사	1	220

실습 예:

```
SELECT deptno, position,SUM(pay)
FROM professor
GROUP BY deptno, CUBE(position);
```

```
SELECT deptno, position,SUM(pay)
FROM professor
GROUP BY position, CUBE(deptno);
```

다른 그룹핑 관련 함수들 살펴보기

- **GROUPING 함수**
  - 그루핑 작업에 사용 유무를 확인하는 함수

```
SELECT deptno, SUM(pay),
GROUPING(deptno) g_deptno
FROM professor
GROUP BY ROLLUP(deptno);
```

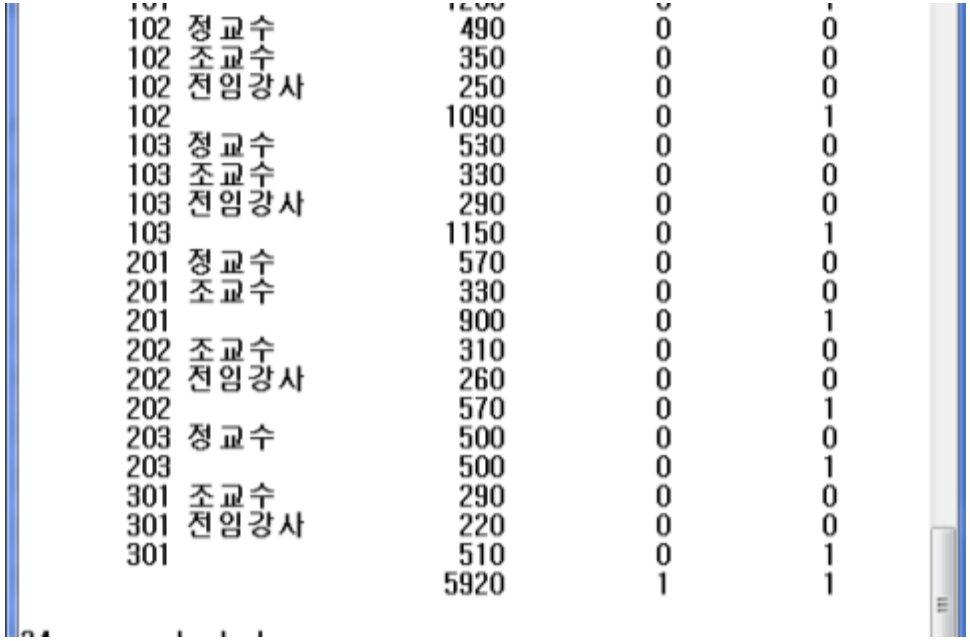
DEPTNO	SUM(PAY)	G_DEPTNO
101	1200	0
102	1090	0
103	1150	0
201	900	0
202	570	0
203	500	0
301	510	0
	5920	1

8 rows selected.

부서별로 급여 합계를 구하는 쿼리에서 deptno 컬럼이 그룹핑 하는데 사용되었는지 살펴보기 위해 grouping 함수를 사용, 가장 마지막 합계부분만 1로 사용되지 않았고 나머지는 모두 0으로 사용되었음을 확인할 수 있음.

실습

```
SELECT deptno, position, SUM(pay),
GROUPING(deptno) g_deptno,
GROUPING(position) g_position
FROM professor
GROUP BY ROLLUP(deptno, position);
```



102	정교수	490	0	0
102	조교수	350	0	0
102	전임강사	250	0	0
102		1090	0	1
103	정교수	530	0	0
103	조교수	330	0	0
103	전임강사	290	0	0
103		1150	0	1
201	정교수	570	0	0
201	조교수	330	0	0
201		900	0	1
202	조교수	310	0	0
202	전임강사	260	0	0
202		570	0	1
203	정교수	500	0	0
203		500	0	1
301	조교수	290	0	0
301	전임강사	220	0	0
301		510	0	1
		5920	1	1

두 개의 컬럼을 그룹핑 하면서 각 컬럼의 사용 유무를 확인. G_DEPTNO 컬럼은 모두 그룹핑하는데 사용되었고 G_POSITION 컬럼은 각 부서별 소계 값을 구할 때는 그룹핑에 사용되지 않았음을 보여줌. 당연히 부서별 소계 값을 구하는 것이니 직급 컬럼은 사용되지 않을 것임

GROUPING_ID 함수

GROUPING 컬럼	BIT	GROUPING 결과	의 미
A , B	0 0	0	두 컬럼 다 GROUPING 에 사용됨
A	0 1	1	A 컬럼만 GROUPING 에 사용됨
B	1 0	2	B 컬럼만 GROUPING 에 사용됨
-	1 1	3	두 컬럼 모두 사용 안됨

실습 예

```
SELECT deptno, position, SUM(pay),
GROUPING_ID(deptno, position) g_dp,
GROUPING_ID(position, deptno) g_pd
FROM professor
GROUP BY ROLLUP(deptno, position);
```

101		1200	1	2
102	정교수	490	0	0
102	조교수	350	0	0
102	전임강사	250	0	0
102		1090	1	2
103	정교수	530	0	0
103	조교수	330	0	0
103	전임강사	290	0	0
103		1150	1	2
201	정교수	570	0	0
201	조교수	330	0	0
201		900	1	2
202	조교수	310	0	0
202	전임강사	260	0	0
202		570	1	2
203	정교수	500	0	0
203		500	1	2
301	조교수	290	0	0
301	전임강사	220	0	0

결과를 보면 GDP 부분에 0 으로 되어 있는 건 두 컬럼 모두 그룹핑에 사용되었다는 뜻이고 1 인 컬럼은 01 비트란 의미이므로 deptno 는 그룹핑에 사용되었으나 position은 사용되지 않았다 라는 의미, 이렇게 여러 개의 컬럼 이 있 을 경 우 GROUPING_ID 를 활용하여 보다 간편하게 조회 할 수 있음.

GROUPING SETS

기존 방법

```
SELECT grade, COUNT(*)
FROM student
GROUP BY(grade)
UNION
SELECT deptno1, COUNT(*)
FROM student
GROUP BY(deptno1);
```

```
     GRADE   COUNT(*)
---------- ----------
         1          5
         2          5
         3          5
         4          5
       101          4
       102          4
       103          2
       201          6
       202          2
       301          2

10 rows selected.

SCOTT>
```

GROUPING SETS 이용

```
SELECT grade, COUNT(*)
FROM student
GROUP BY GROUPING
SETS(grade,deptno1);
```

```
     GRADE    DEPTNO1   COUNT(*)
---------- ---------- ----------
                   102          4
                   201          6
                   301          2
                   101          4
                   202          2
                   103          2
         1                      5
         2                      5
         4                      5
         3                      5

10 rows selected.

SCOTT>
```

LISTAGG 함수(11g에서 추가됨)

문법 : LISTAGG(list_column, ['구분자']) WITHIN GROUP(ORDER BY column)

```
SELECT deptno,
  LISTAGG(name, '**') WITHIN GROUP(ORDER BY hiredate) "LISTAGG"
FROM professor GROUP BY deptno;
```

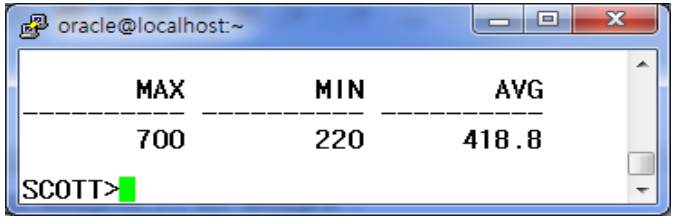
```
   DEPTNO LISTAGG
--------- ----------------------------------------
      101 조인형**박승곤**송도권
      102 주승재**김영조**양선희
      103 김도형**나한열**김현정
      201 심슨**최슬기
      202 박원범**차범철
      203 바비
      301 허은**전민

rows selected.
```

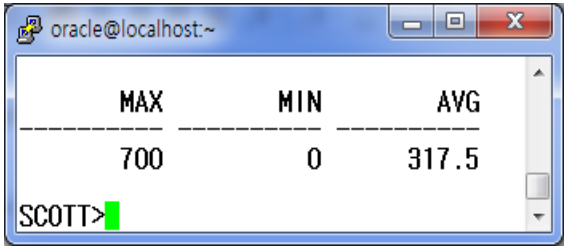
```
SELECT deptno,
  LISTAGG(name) WITHIN GROUP(ORDER BY hiredate) "LISTAGG"
FROM professor GROUP BY deptno;
```

SQL 그룹 함수 연습 문제

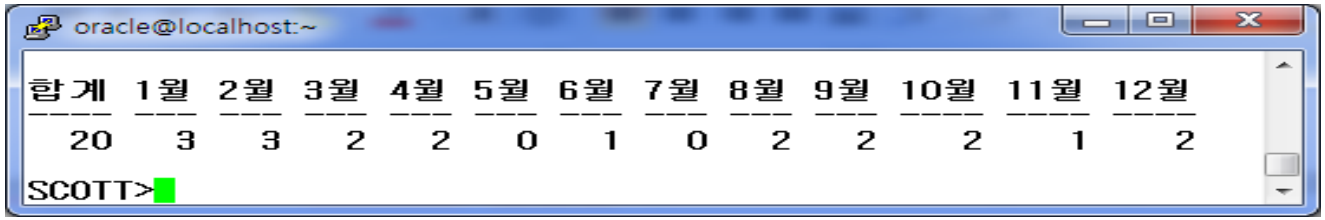
1) Professor 테이블을 사용하여 교수 중에서 급여(Pay)와 보너스(bonus)를 합친 금액이 가장 많은 경우와 가장 적은 경우 , 평균 금액을 구하세요. 단 보너스가 없을 경우는 보너스를 0 으로 계산하고 출력 금액은 모두 소수점 첫째 자리까지만 나오게 하세요.



2) Professor 테이블을 사용하여 교수 중에서 급여(Pay)와 보너스(bonus)를 합친 금액이 가장 많은 경우와 가장 적은 경우 , 평균 금액을 구하세요. 단 보너스가 없을 경우는 급여를 0 으로 계산하고 출력 금액은 모두 소수점 첫째 자리까지만 나오게 하세요.



3) Student 테이블의 birthday 칼럼을 사용하여 아래 화면처럼 월별로 태어난 인원수를 출력하세요.



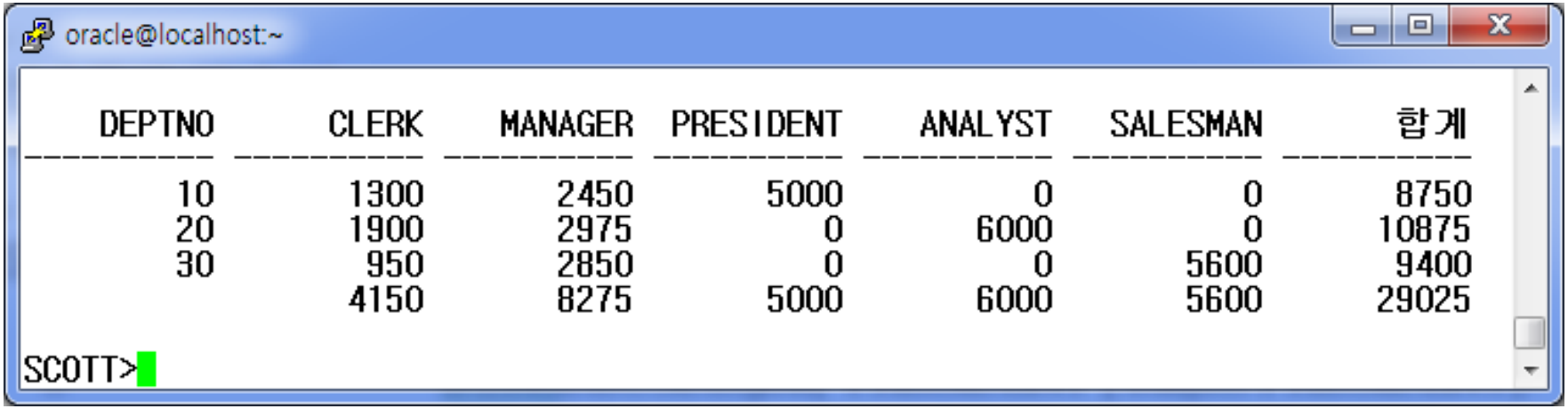
4) Student 테이블의 tel 칼럼을 참고하여 아래와 같이 지역별 인원수를 출력하세요. 단 02 –서울 , 031 – 경기 , 051 – 부산 , 052 – 울산 , 053 – 대구 , 055 – 경남으로 출력하세요

5) Emp 테이블을 사용하여 아래의 화면과 같이 부서별로 직급별로 급여 합계 결과를 출력하세요. 먼저 아래의 두 건의 데이터를 입력 하신 후 작업하세요.

```
SQL>insert into emp (empno , deptno , ename , sal)
2 values (1000,10,'홍길동',3600) ;

SQL> insert into emp (empno , deptno , ename , sal)
2  values (2000,30,'일지매',3000);

SQL> commit;
```

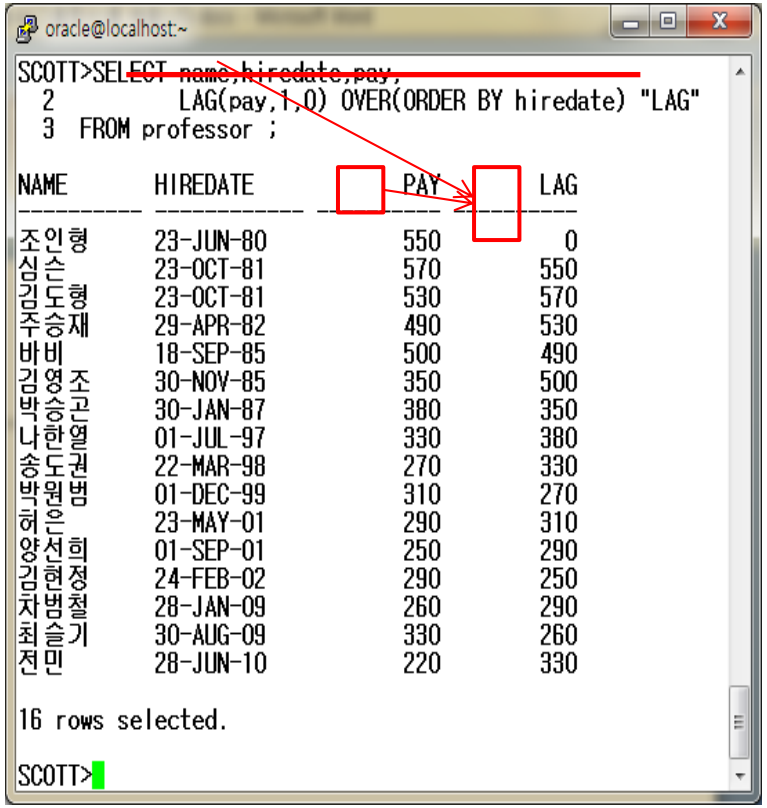


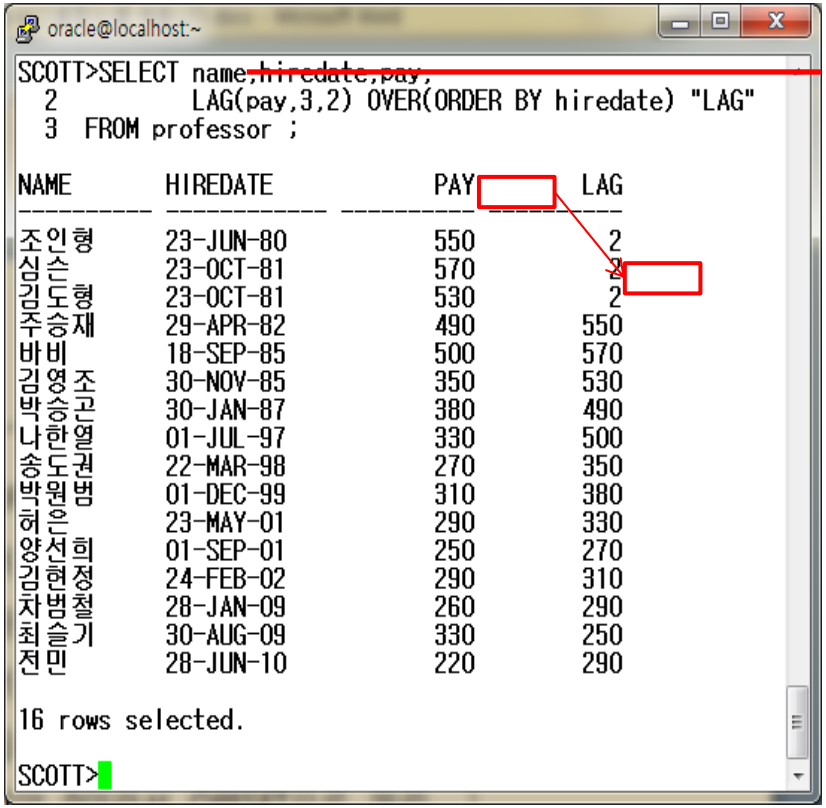
기타 함수

LAG 함수 :

- 이전 행 값을 가져 올 때 사용하는 함수.

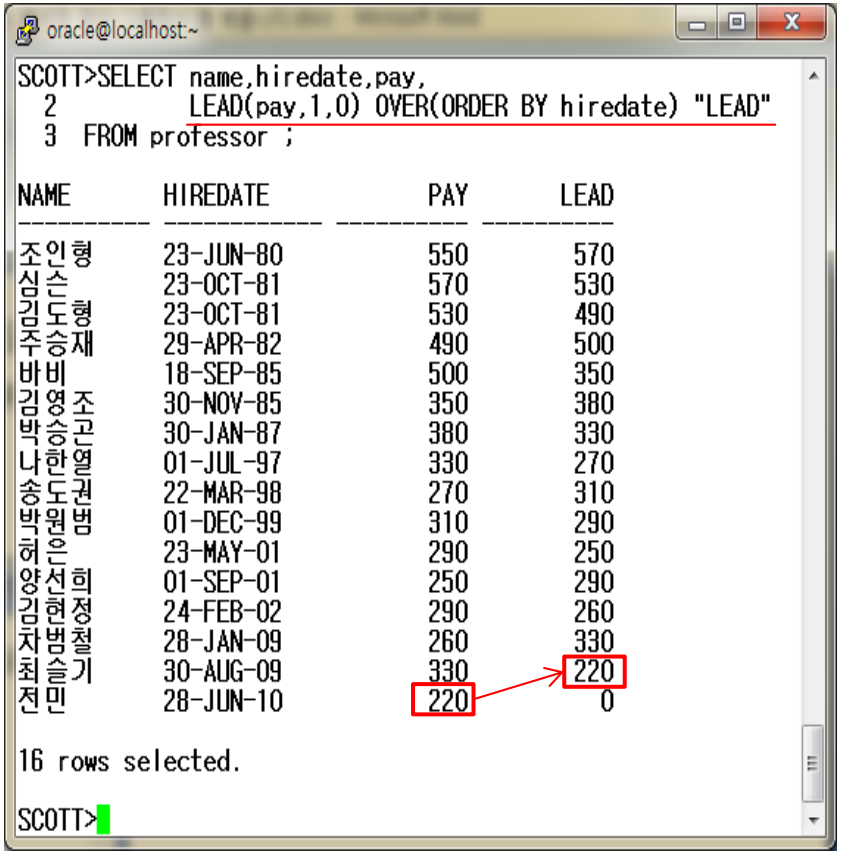
문 법 : LAG(출력할 컬럼명 , OFFSET , 기본 출력값)
OVER (Query_partition구문 , ORDER BY 정렬할 컬럼)

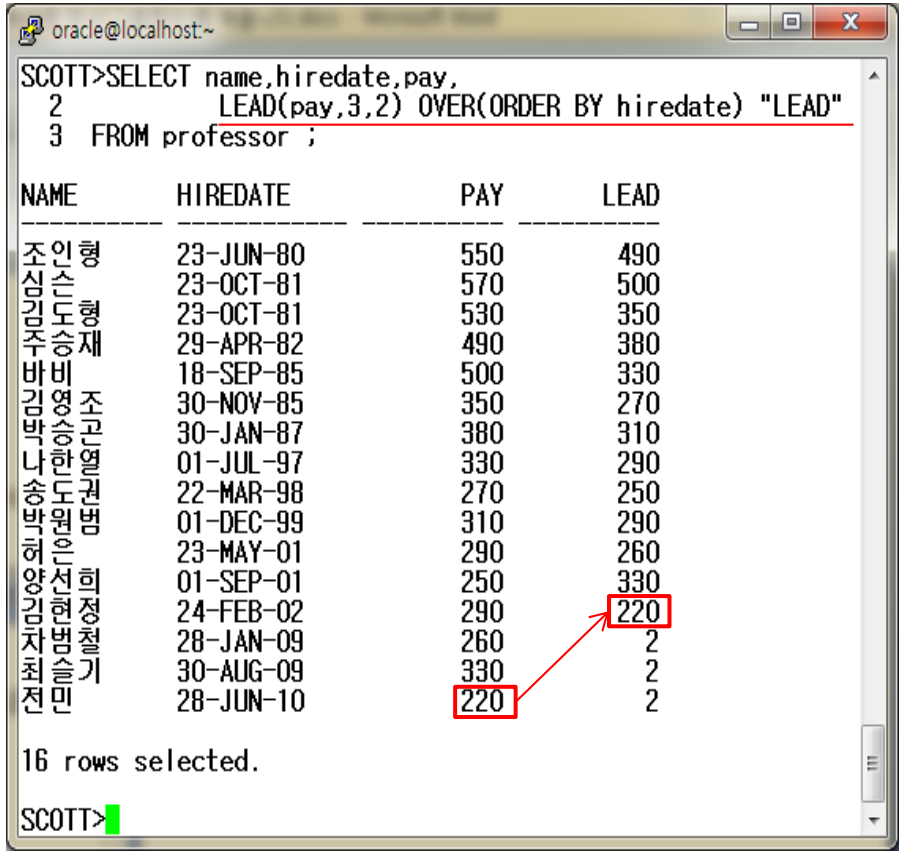




) LEAD 함수

- LEAD 함수는 LAG 함수와 반대로 이후의 값을 가져오는 함수.



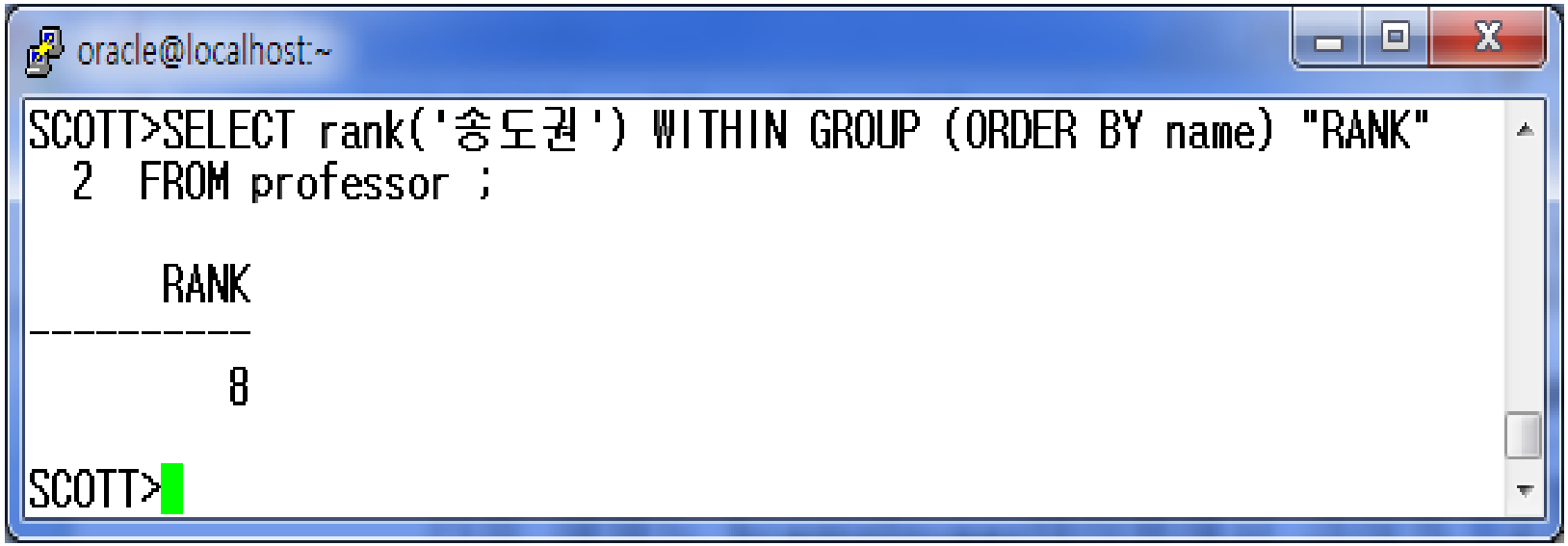


RANK 함수

- 순위 출력 함수

RANK(조건값) WITHIN GROUP (ORDER BY 조건값 컬럼명 [ASC | DESC])

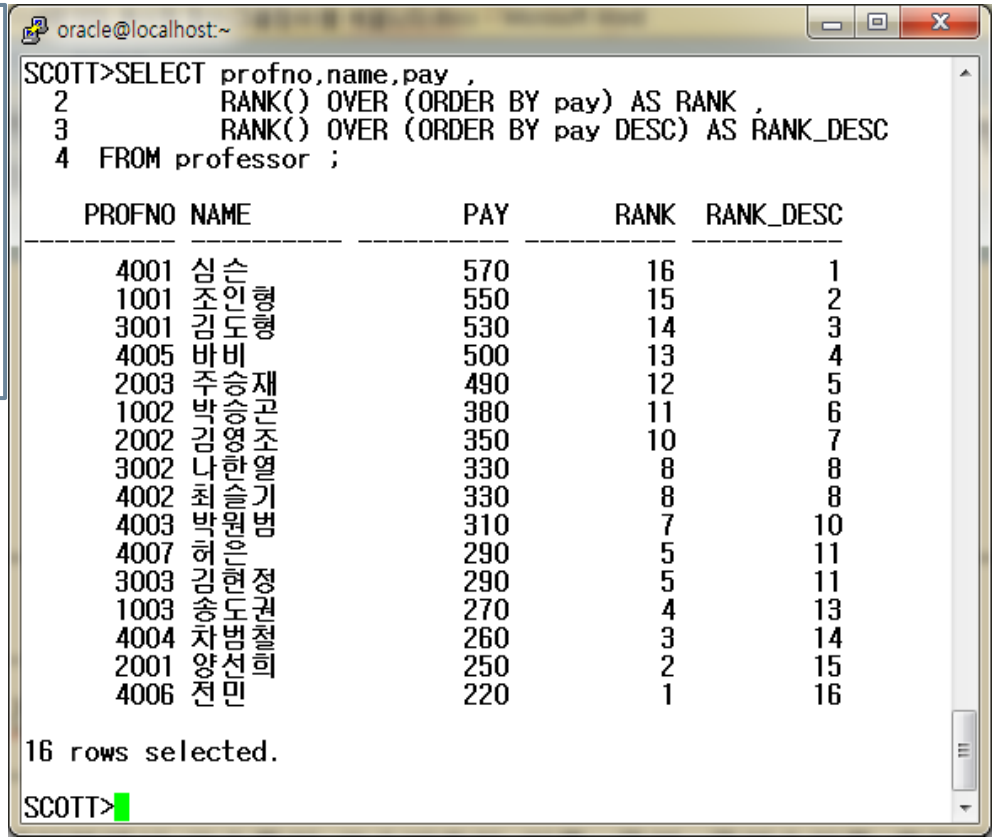
- **사용 예 :** 이름이 '송도권' 인 교수의 순위를 조회하세요.



RANK 함수 – 집계용

RANK ( ) (ORDER BY 조건컬럼명 [ASC | DESC])

①교수 테이블 (professor) 테이블에서 교수들의 교수번호와 이름, 급여, 급여순위를 출력하세요.



누적 합계 구하기

① panmae 테이블을 사용하여 1000 번 대리점의 판매 내역을 출력하되 판매 일자, 제품코드, 판매량, 누적 판매금액을 아래와 같이 출력하세요.