

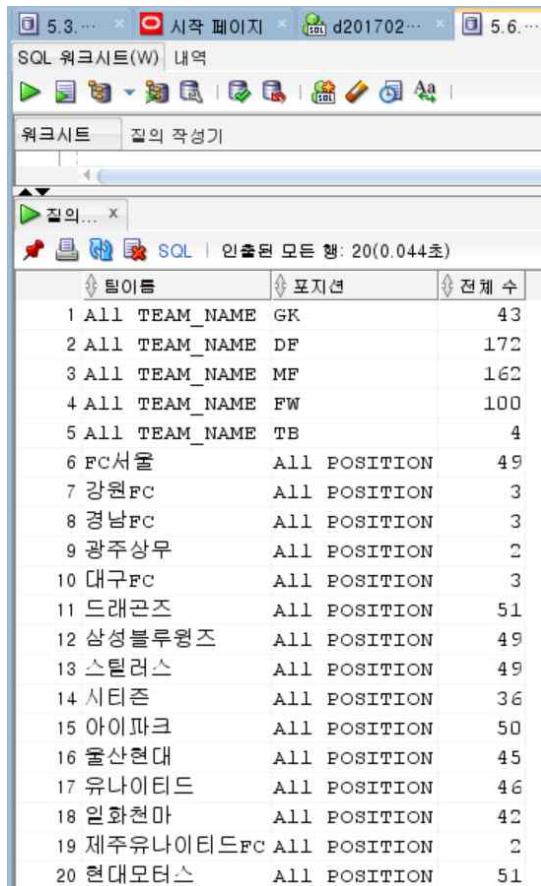
## [실습과제-9] SQL 그룹 함수, 윈도우 함수 실습

※ 다음 [실습과제9-1], [실습과제9-2] 문제는 SQL 실습노트9-1 파일을 참고하여, 본인이 작성한 SQL문과 함께 실행 결과를 포함하여 제출하여야 합니다.

[실습과제9-1] [예제12]를 참고하여 PLAYER와 TEAM 테이블에서 팀 별 선수들의 수와 포지션 별 선수들의 수를 모두 조회하는 SQL문을 작성하고 그 결과를 출력하시오. 단, 포지션이 NULL인 데이터는 제외하고 팀이름 기준 오름차순으로 정렬하여 출력한다.

[실습과제 9-1] 답안

```
SELECT
    DECODE(GROUPING(Team_NAME),1,
    'All TEAM_NAME', TEAM_NAME) AS 팀이름,
    DECODE(GROUPING(POSITION), 1,
    'All POSITION', POSITION) AS 포지션,
    COUNT(*) "전체 수"
FROM PLAYER , TEAM
WHERE PLAYER.TEAM_ID = TEAM.TEAM_ID
    AND POSITION IS NOT NULL
GROUP BY GROUPING SETS (TEAM_NAME, POSITION)
ORDER BY 팀이름;
```

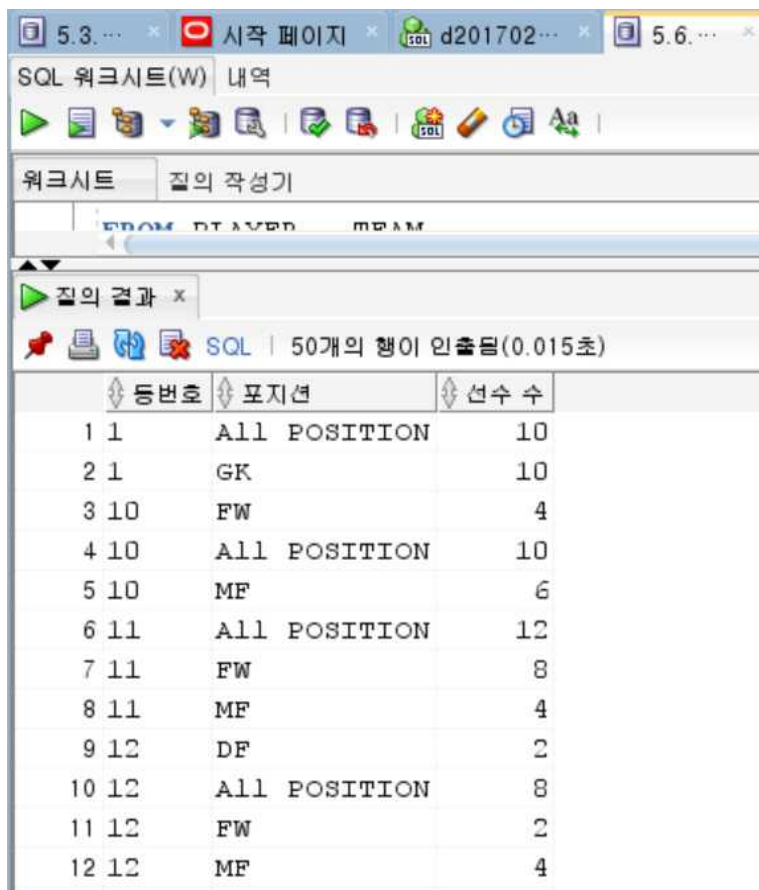


팀이름	포지션	전체 수
1 All TEAM_NAME	GK	43
2 All TEAM_NAME	DF	172
3 All TEAM_NAME	MF	162
4 All TEAM_NAME	FW	100
5 All TEAM_NAME	TB	4
6 FC서울	All POSITION	49
7 강원FC	All POSITION	3
8 경남FC	All POSITION	3
9 광주상무	All POSITION	2
10 대구FC	All POSITION	3
11 드래곤즈	All POSITION	51
12 삼성블루윙즈	All POSITION	49
13 스틸러스	All POSITION	49
14 시티즌	All POSITION	36
15 아이파크	All POSITION	50
16 울산현대	All POSITION	45
17 유나이티드	All POSITION	46
18 일화천마	All POSITION	42
19 제주유나이티드FC	All POSITION	2
20 현대모터스	All POSITION	51

[실습과제9-2] PLAYER 테이블에서 등번호 별, 포지션 별 선수들의 수와 총 선수 수를 조회하는 SQL문을 작성하고 그 결과를 출력하시오. 단, 등번호가 없거나 포지션이 없는 선수들은 제외하고 등번호 기준 오름차순으로 정렬하여 출력한다.  
(Hint: CUBE 함수를 사용한다. ROLLUP 함수와 GROUPING SETS 함수를 사용하면 안되는 이유는 무엇인가?)

[실습과제 9-2] 답안

```
SELECT
    DECODE(GROUPING(BACK_NO),1,
    'All TEAM_NAME', BACK_NO) AS 등번호,
    DECODE(GROUPING(POSITION), 1,
    'All POSITION', POSITION) AS 포지션,
    COUNT(*) "선수 수"
FROM PLAYER
WHERE BACK_NO IS NOT NULL
    AND POSITION IS NOT NULL
GROUP BY CUBE (BACK_NO, POSITION)
ORDER BY 등번호;
```



등번호	포지션	선수 수
1 1	All POSITION	10
2 1	GK	10
3 10	FW	4
4 10	All POSITION	10
5 10	MF	6
6 11	All POSITION	12
7 11	FW	8
8 11	MF	4
9 12	DF	2
10 12	All POSITION	8
11 12	FW	2
12 12	MF	4

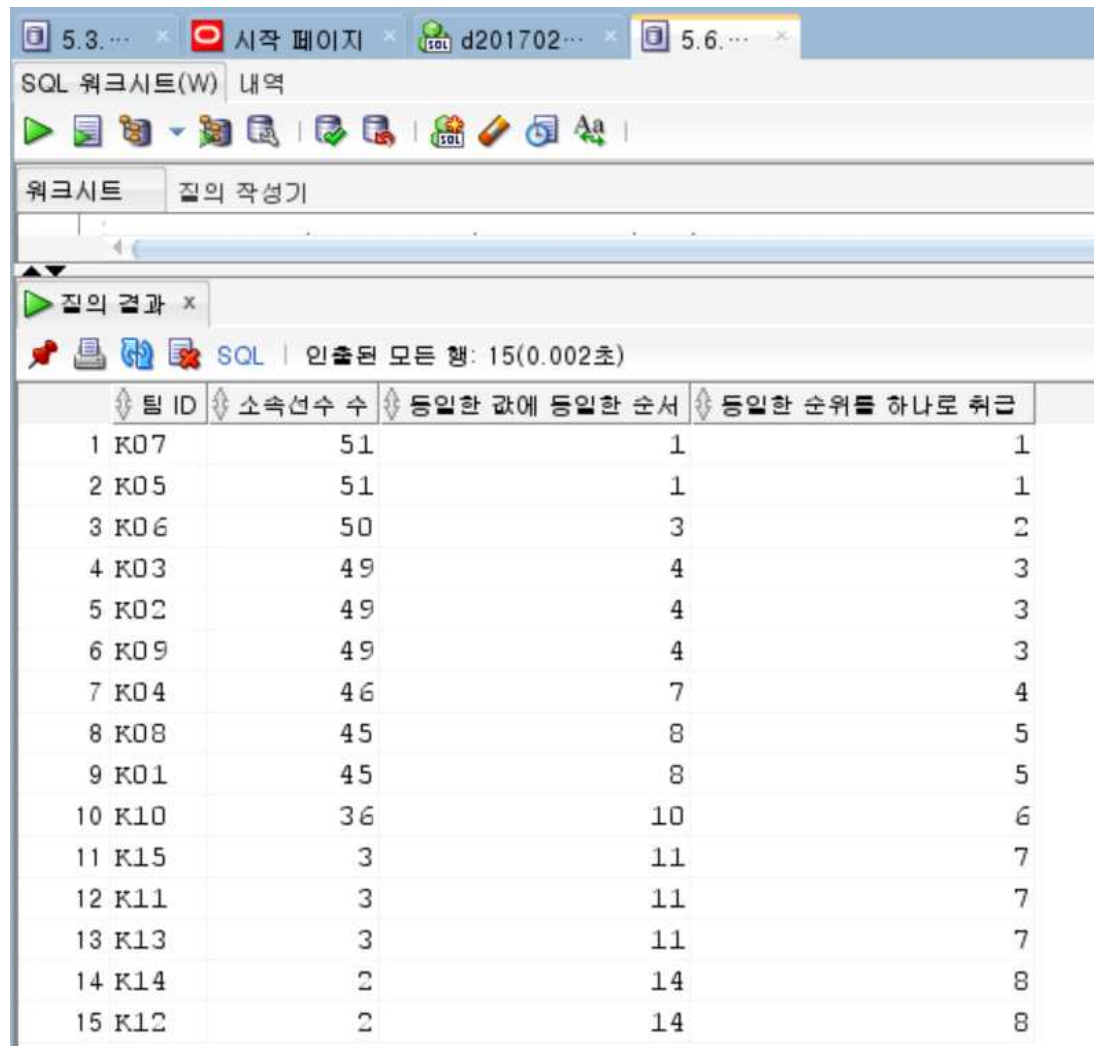
CUBE를 사용해야만 표시된 인수들에 대한 계층별 집계를 구할 수 있기 때문이다

※ 다음 [실습과제9-3] ~ [실습과제9-7] 문제는 SQL 실습노트9-2 파일을 참고하여, 본인이 작성한 SQL문과 함께 실행 결과를 포함하여 제출하여야 합니다.

[실습과제9-3] [예제2]를 참고하여 PLAYER 테이블에서 팀ID, 소속선수 수, 소속선수 수가 많은 순으로 팀 순위를 조회하는 SQL문을 작성하고 그 결과를 출력하시오. 단, 하나의 SQL문으로 작성되어야 하며, 팀 순위는 동일한 값에 대해서 동일한 순서를 부여하는 방식과 동일한 순위를 하나의 건수로 취급하는 방식 두 가지를 모두 사용하여 출력한다.

[실습과제 9-3] 답안

```
SELECT TEAM_ID "팀 ID", COUNT(*) "소속선수 수",
RANK() OVER (ORDER BY COUNT(*)DESC) "동일한 값에 동일한 순서",
DENSE_RANK() OVER (ORDER BY COUNT(*)DESC) "동일한 순위를 하나로 취급 "
FROM PLAYER
GROUP BY TEAM_ID
ORDER BY "소속선수 수" DESC;
```

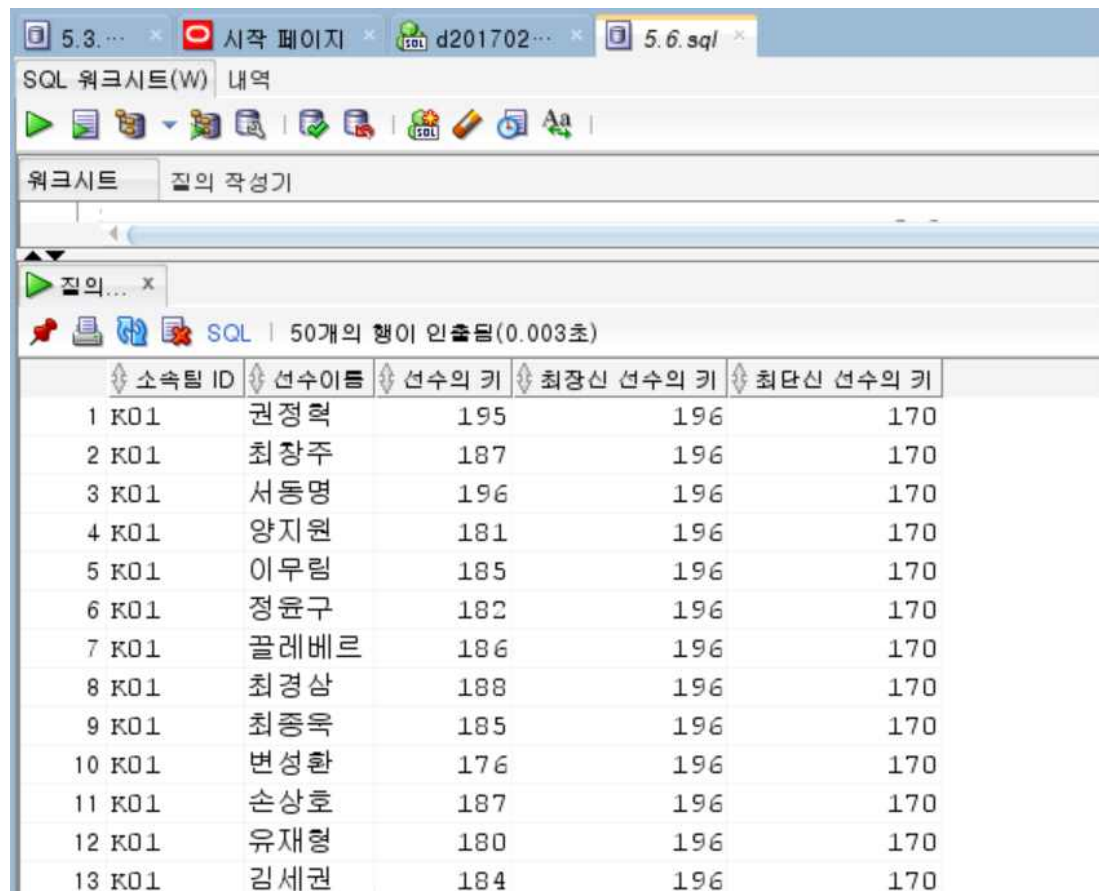


팀 ID	소속선수 수	동일한 값에 동일한 순서	동일한 순위를 하나로 취급
1 K07	51	1	1
2 K05	51	1	1
3 K06	50	3	2
4 K03	49	4	3
5 K02	49	4	3
6 K09	49	4	3
7 K04	46	7	4
8 K08	45	8	5
9 K01	45	8	5
10 K10	36	10	6
11 K15	3	11	7
12 K11	3	11	7
13 K13	3	11	7
14 K14	2	14	8
15 K12	2	14	8

[실습과제9-4] [예제4-1], [예제5]를 참고하여 PLAYER 테이블에서 선수들의 소속팀 ID, 선수이름, 선수의 키, 소속팀 선수들 중 최장신 선수의 키, 최단신 선수의 키를 조회하는 SQL문을 작성하고 그 결과를 출력하시오.

[실습과제 9-4] 답안

```
SELECT TEAM_ID "소속팀 ID", PLAYER_NAME 선수이름, HEIGHT "선수의 키",
       MAX(HEIGHT) OVER (PARTITION BY TEAM_ID) AS "최장신 선수의 키",
       MIN(HEIGHT) OVER (PARTITION BY TEAM_ID) AS "최단신 선수의 키"
FROM PLAYER;
```

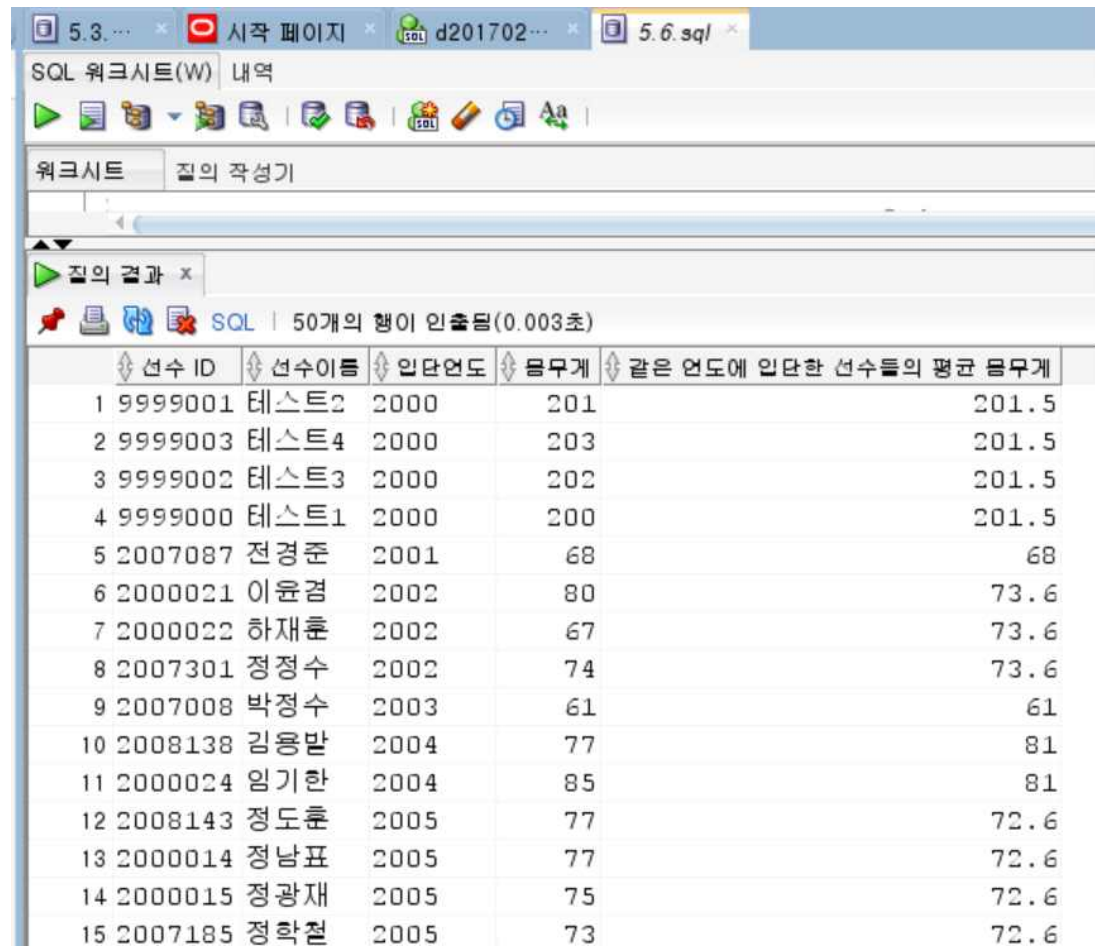


	소속팀 ID	선수이름	선수의 키	최장신 선수의 키	최단신 선수의 키
1	K01	권정혁	195	196	170
2	K01	최창주	187	196	170
3	K01	서동명	196	196	170
4	K01	양지원	181	196	170
5	K01	이무림	185	196	170
6	K01	정윤구	182	196	170
7	K01	글레베르	186	196	170
8	K01	최경삼	188	196	170
9	K01	최종욱	185	196	170
10	K01	변성환	176	196	170
11	K01	손상호	187	196	170
12	K01	유재형	180	196	170
13	K01	김세권	184	196	170

[실습과제9-5] [예제7]을 참고하여 PLAYER 테이블에서 각 선수들에 대하여 선수ID, 선수이름, 입단연도, 몸무게와 해당 선수와 같은 연도에 입단한 선수들의 평균 몸무게를 조회하는 SQL문을 작성하고 그 결과를 출력하시오. 단, 입단연도가 없거나 몸무게 값이 없는 선수는 제외하고, 평균 몸무게는 소수점 첫째자리까지 출력한다.

[실습과제 9-5] 답안

```
SELECT PLAYER_ID "선수 ID", PLAYER_NAME 선수이름, JOIN_YYYY 입단연도,
WEIGHT 몸무게,
      TRUNC (AVG(WEIGHT) OVER (PARTITION BY JOIN_YYYY,1) AS "같은 연도
에 입단한 선수들의 평균 몸무게"
FROM PLAYER
WHERE JOIN_YYYY IS NOT NULL
      AND WEIGHT IS NOT NULL;
```



선수 ID	선수이름	입단연도	몸무게	같은 연도에 입단한 선수들의 평균 몸무게
1 9999001	테스트2	2000	201	201.5
2 9999003	테스트4	2000	203	201.5
3 9999002	테스트3	2000	202	201.5
4 9999000	테스트1	2000	200	201.5
5 2007087	전경준	2001	68	68
6 2000021	이윤경	2002	80	73.6
7 2000022	하재훈	2002	67	73.6
8 2007301	정정수	2002	74	73.6
9 2007008	박정수	2003	61	61
10 2008138	김용발	2004	77	81
11 2000024	임기한	2004	85	81
12 2008143	정도훈	2005	77	72.6
13 2000014	정남표	2005	77	72.6
14 2000015	정광재	2005	75	72.6
15 2007185	정학철	2005	73	72.6

[실습과제9-6] [예제9], [예제10]을 참고하여 PLAYER 테이블에서 포지션 별 선수들의 입단연도가 빠른 순으로 포지션명, 선수이름, 입단연도, 포지션 내에서 가장 입단연도가 빠른 선수와 가장 늦은 선수 이름을 조회하는 SQL문을 작성하고 그 결과를 출력하시오. 단, 입단연도가 같으면 이름순으로 정렬하여 출력한다.

[실습과제 9-6] 답안

```
SELECT POSITION 포지션명, PLAYER_NAME 선수이름, JOIN_YYYY 입단연도,
      FIRST_VALUE(PLAYER_NAME) OVER
        (PARTITION BY POSITION ORDER BY JOIN_YYYY, PLAYER_NAME ASC
          ROWS UNBOUNDED PRECEDING) AS "입단년도 가장 빠른 선수",
      LAST_VALUE(PLAYER_NAME) OVER
        (PARTITION BY POSITION ORDER BY JOIN_YYYY, PLAYER_NAME ASC
          ROWS BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING) AS "입단
년도 가장 늦은 선수"
FROM PLAYER
WHERE JOIN_YYYY IS NOT NULL;
```

	포지션명	선수이름	입단연도	입단년도 가장 빠른 선수	입단년도 가장 늦은 선수
1	DF	이윤경	2002	이윤경	홍명보
2	DF	정정수	2002	이윤경	홍명보
3	DF	하재훈	2002	이윤경	홍명보
4	DF	임기한	2004	이윤경	홍명보
5	DF	류병훈	2005	이윤경	홍명보
6	DF	윤희준	2005	이윤경	홍명보
7	DF	정광재	2005	이윤경	홍명보
8	DF	정남표	2005	이윤경	홍명보
9	DF	정학철	2005	이윤경	홍명보
10	DF	최상인	2005	이윤경	홍명보
11	DF	고병운	2006	이윤경	홍명보
12	DF	김현준	2006	이윤경	홍명보



[실습과제9-7] [예제17]을 참고하여 EMP 테이블에서 전체 사원을 입사일 순서로 정렬하고, 입사일을 기준으로 5개 그룹으로 분류하여 각 사원의 이름, 봉급, 입사일, 해당 그룹을 출력하는 SQL문을 작성하고 그 결과를 출력하시오.

(Hint: NTILE 함수 사용)

[실습과제 9-7] 답안

```
SELECT ENAME 이름, SALARY*12 봉급, HIREDATE 입사일, EMPNO "해당 그룹",
       NTILE(5) OVER (ORDER BY HIREDATE DESC) AS QUAR_TILE
FROM EMP
ORDER BY HIREDATE DESC;
```

	이름	봉급	입사일	해당 그룹	QUAR_TILE
1	ADAMS	13200	87/07/13	7876	1
2	SCOTT	36000	87/07/13	7788	1
3	MILLER	15600	82/01/23	7934	1
4	FORD	36000	81/12/03	7902	2
5	JAMES	11400	81/12/03	7900	2
6	KING	60000	81/11/17	7839	2
7	MARTIN	15000	81/09/28	7654	3
8	TURNER	18000	81/09/08	7844	3
9	CLARK	29400	81/06/09	7782	3
10	BLAKE	34200	81/05/01	7698	4
11	JONES	35700	81/04/02	7566	4
12	WARD	15000	81/02/22	7521	4
13	ALLEN	19200	81/02/20	7499	5
14	SMITH	9600	80/12/17	7369	5

\* 실습과제 제출물

- 본 문서에 답안을 작성하여 파일명을 “[LA-9]학번\_이름”으로 하여 제출한다.  
(예: [LA-9]202201234\_홍길동.hwp)