5-3. 분석함수, 그리고 MSSQL

홍형경
chariehong@gmail.com
2021.06

1. 분석함수 (Analytic Function)

· 로우별 그룹을 지정해서 값을 집계하는 함수

· GROUP BY 절과는 다름

· GROUP BY 절 사용 시, 집계 대상에 따라 로우 수가 줄어들지만, 분석함수는 그렇지 않음

➔ **로우 수는 그대로, 집계 값 산출이 가능**

· 분석함수에서 말하는 로우별 그룹 ➔ 윈도우(Window) 절

· **분석 함수**와 **윈도우 절**이 같이 사용됨

1. 분석함수 (Analytic Function)

· 일반 집계 함수(SUM, MAX, MIN, AVG 등)를 분석 함수로 사용 가능

· 그 외에 ROW_NUMBER, RANK, DENSE_RANK, LAG, LEAD 함수가 있음

1. 분석함수 (Analytic Function)

· 분석 함수 구문

분석 함수 **OVER** (**PARTITION BY** col1, col2, …
ORDER BY col1, col2…)

· PARTITION BY : 분석 함수 집계 대상이 되는 로우 값의 범위, 그룹

· PARTITION BY 절 생략 시, 전체 로우가 분석 함수 집계 대상이 됨

· ORDER BY : 분석 함수 계산 시, 고려되는 로우 순서

2. 분석함수 실습 – row_number() : 일련번호

· 부서별로 사원의 급여 순으로 순번을 구하라

```
SELECT b.department_id, b.department_name,
       a.first_name || ' ' || a.last_name as emp_name,
       ROW_NUMBER() OVER
              (PARTITION BY b.department_id
               ORDER BY a.salary ) dept_sal_seq,
       a.salary
  FROM employees a,
       departments b
 WHERE a.department_id = b.department_id
 ORDER BY 2, 4 ;
```

	DEPARTMENT_ID	DEPARTMENT_NAME	EMP_NAME	DEPT_SAL_SEQ	SALARY
1	110	Accounting	William Gietz	1	8300
2	110	Accounting	Shelley Higgins	2	12008
3	10	Administration	Jennifer Whalen	1	4400
4	90	Executive	Lex De Haan	1	17000
5	90	Executive	Neena Kochhar	2	17000
6	90	Executive	Steven King	3	24000
7	100	Finance	Luis Popp	1	6900
8	100	Finance	Ismael Sciarra	2	7700
9	100	Finance	Jose Manuel Urman	3	7800
10	100	Finance	John Chen	4	8200
11	100	Finance	Daniel Faviet	5	9000
12	100	Finance	Nancy Greenberg	6	12008
13	40	Human Resources	Susan Mavris	1	6500
14	60	IT	Diana Lorentz	1	4200
15	60	IT	David Austin	2	4800
16	60	IT	Valli Pataballa	3	4800
17	60	IT	Bruce Ernst	4	6000
18	60	IT	Alexander Hunold	5	9000

2. 분석함수 실습 – row_number() : 일련번호

· 부서별로 사원의 급여가 높은 순으로 순번을 구하라

```
SELECT b.department_id, b.department_name,
       a.first_name || ' ' || a.last_name as emp_name,
       ROW_NUMBER() OVER
              (PARTITION BY b.department_id
               ORDER BY a.salary desc) dept_sal_seq,
       a.salary
  FROM employees a,
       departments b
 WHERE a.department_id = b.department_id
 ORDER BY 2, 4 ;
```

	DEPARTMENT_ID	DEPARTMENT_NAME	EMP_NAME	DEPT_SAL_SEQ	SALARY
1	110	Accounting	Shelley Higgins	1	12008
2	110	Accounting	William Gietz	2	8300
3	10	Administration	Jennifer Whalen	1	4400
4	90	Executive	Steven King	1	24000
5	90	Executive	Neena Kochhar	2	17000
6	90	Executive	Lex De Haan	3	17000
7	100	Finance	Nancy Greenberg	1	12008
8	100	Finance	Daniel Faviet	2	9000
9	100	Finance	John Chen	3	8200
10	100	Finance	Jose Manuel Urman	4	7800
11	100	Finance	Ismael Sciarra	5	7700
12	100	Finance	Luis Popp	6	6900
13	40	Human Resources	Susan Mavris	1	6500
14	60	IT	Alexander Hunold	1	9000
15	60	IT	Bruce Ernst	2	6000
16	60	IT	David Austin	3	4800
17	60	IT	Valli Pataballa	4	4800
18	60	IT	Diana Lorentz	5	4200

2. 분석함수 실습 – row_number() : 일련번호

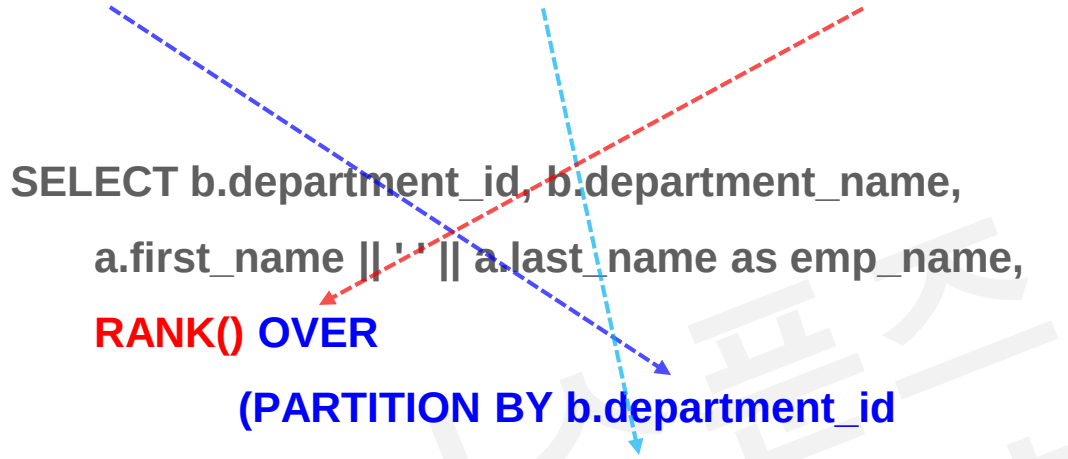
· 전 사원의 급여가 높은 순으로 순번을 구하라

```
SELECT b.department_id, b.department_name,
       a.first_name || ' ' || a.last_name as emp_name,
       ROW_NUMBER() OVER
              ( ORDER BY a.salary desc) dept_sal_seq,
       a.salary
  FROM employees a,
       departments b
 WHERE a.department_id = b.department_id
 ORDER BY 4 ;
```

	DEPARTMENT_ID	DEPARTMENT_NAME	EMP_NAME	DEPT_SAL_SEQ	SALARY
1	90	Executive	Steven King	1	24000
2	90	Executive	Neena Kochhar	2	17000
3	90	Executive	Lex De Haan	3	17000
4	80	Sales	John Russell	4	14000
5	80	Sales	Karen Partners	5	13500
6	20	Marketing	Michael Hartstein	6	13000
7	110	Accounting	Shelley Higgins	7	12008
8	100	Finance	Nancy Greenberg	8	12008
9	80	Sales	Alberto Errazuriz	9	12000
10	80	Sales	Lisa Ozer	10	11500
11	80	Sales	Gerald Cambrault	11	11000
12	30	Purchasing	Den Raphaely	12	11000
13	80	Sales	Ellen Abel	13	11000
14	80	Sales	Clara Vishney	14	10500
15	80	Sales	Eleni Zlotkey	15	10500
16	70	Public Relations	Hermann Baer	16	10000
17	80	Sales	Harrison Bloom	17	10000
18	80	Sales	Janette King	18	10000
19	80	Sales	Peter Tucker	19	10000
20	80	Sales	Tayler Fox	20	9600
21	80	Sales	David Bernstein	21	9500

2. 분석함수 실습 – RANK() : 순위

· 부서별로 사원의 급여가 높은 순으로 순위를 구하라



```
SELECT b.department_id, b.department_name,
       a.first_name || ' ' || a.last_name as emp_name,
       RANK() OVER
              (PARTITION BY b.department_id
               ORDER BY a.salary desc) dept_sal_seq,
       a.salary
  FROM employees a,
       departments b
 WHERE a.department_id = b.department_id
 ORDER BY 2, 4 ;
```

	DEPARTMENT_ID	DEPARTMENT_NAME	EMP_NAME	DEPT_SAL_SEQ	SALARY
1	110	Accounting	Shelley Higgins	1	12008
2	110	Accounting	William Gietz	2	8300
3	10	Administration	Jennifer Whalen	1	4400
4	90	Executive	Steven King	1	24000
5	90	Executive	Lex De Haan	2	17000
6	90	Executive	Neena Kochhar	2	17000
7	100	Finance	Nancy Greenberg	1	12008
8	100	Finance	Daniel Faviet	2	9000
9	100	Finance	John Chen	3	8200
10	100	Finance	Jose Manuel Urman	4	7800
11	100	Finance	Ismael Sciarra	5	7700
12	100	Finance	Luis Popp	6	6900
13	40	Human Resources	Susan Mavris	1	6500
14	60	IT	Alexander Hunold	1	9000
15	60	IT	Bruce Ernst	2	6000
16	60	IT	David Austin	3	4800
17	60	IT	Valli Pataballa	3	4800
18	60	IT	Diana Lorentz	5	4200
19	20	Marketing	Michael Hartstein	1	13000
20	20	Marketing	Pat Fay	2	6000

2. 분석함수 실습 – DENSE_RANK() : 누적순위

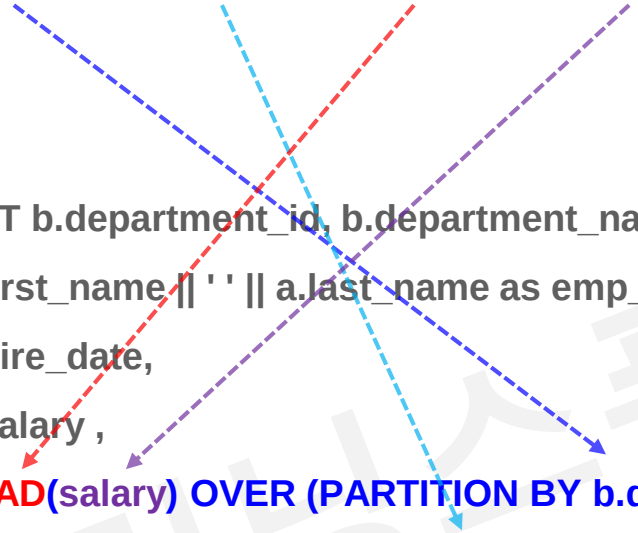
· 부서별로 사원의 급여가 높은 순 누적순위

```
SELECT b.department_id, b.department_name,
       a.first_name || ' ' || a.last_name as emp_name,
       DENSE_RANK() OVER
              (PARTITION BY b.department_id
               ORDER BY a.salary desc) dept_sal_seq,
       a.salary
  FROM employees a,
       departments b
 WHERE a.department_id = b.department_id
 ORDER BY 2, 4 ;
```

	DEPARTMENT_ID	DEPARTMENT_NAME	EMP_NAME	DEPT_SAL_SEQ	SALARY
1	110	Accounting	Shelley Higgins	1	12008
2	110	Accounting	William Gietz	2	8300
3	10	Administration	Jennifer Whalen	1	4400
4	90	Executive	Steven King	1	24000
5	90	Executive	Lex De Haan	2	17000
6	90	Executive	Neena Kochhar	2	17000
7	100	Finance	Nancy Greenberg	1	12008
8	100	Finance	Daniel Faviet	2	9000
9	100	Finance	John Chen	3	8200
10	100	Finance	Jose Manuel Urman	4	7800
11	100	Finance	Ismael Sciarra	5	7700
12	100	Finance	Luis Popp	6	6900
13	40	Human Resources	Susan Mavris	1	6500
14	60	IT	Alexander Hunold	1	9000
15	60	IT	Bruce Ernst	2	6000
16	60	IT	David Austin	3	4800
17	60	IT	Valli Pataballa	3	4800
18	60	IT	Diana Lorentz	4	4200
19	20	Marketing	Michael Hartstein	1	13000
20	20	Marketing	Pat Fay	2	6000

2. 분석함수 실습 – LEAD(expr) : 후행 로우값

· 부서별, 입사일자 순, 직후 사원의 급여를 구하라

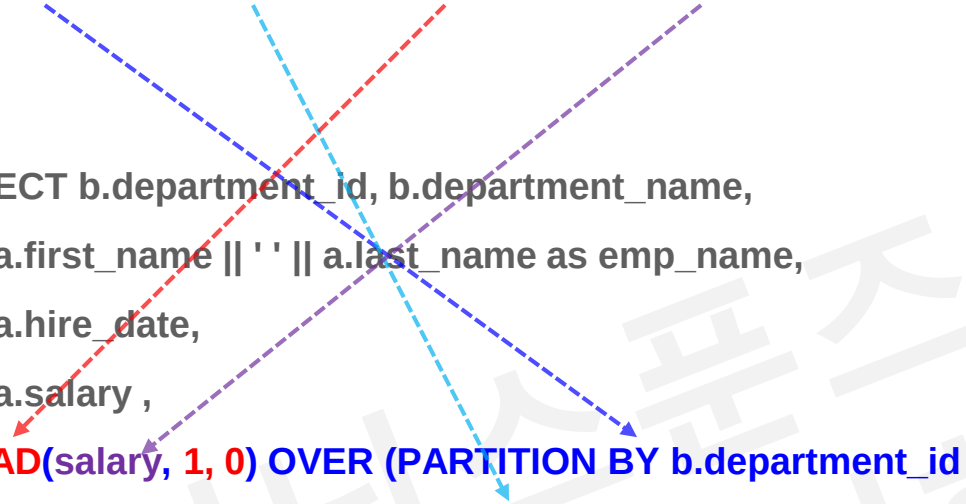


```
SELECT b.department_id, b.department_name,
       a.first_name || ' ' || a.last_name as emp_name,
       a.hire_date,
       a.salary ,
       LEAD(salary) OVER (PARTITION BY b.department_id
                          ORDER BY a.hire_date ) lead_salary
  FROM employees a,
       departments b
 WHERE a.department_id = b.department_id
 ORDER BY 2, 4 ;
```

	DEPARTME...	DEPARTMENT_NAME	EMP_NAME	HIRE_DATE	SALARY	LEAD_SALARY
1	110	Accounting	William Gietz	2002-06-07 00:00:00	8300	(null)
2	110	Accounting	Shelley Higgins	2002-06-07 00:00:00	12008	8300
3	10	Administration	Jennifer Whalen	2003-09-17 00:00:00	4400	(null)
4	90	Executive	Lex De Haan	2001-01-13 00:00:00	17000	24000
5	90	Executive	Steven King	2003-06-17 00:00:00	24000	17000
6	90	Executive	Neena Kochhar	2005-09-21 00:00:00	17000	(null)
7	100	Finance	Daniel Faviet	2002-08-16 00:00:00	9000	12008
8	100	Finance	Nancy Greenberg	2002-08-17 00:00:00	12008	8200
9	100	Finance	John Chen	2005-09-28 00:00:00	8200	7700
10	100	Finance	Ismael Sciarra	2005-09-30 00:00:00	7700	7800
11	100	Finance	Jose Manuel Urman	2006-03-07 00:00:00	7800	6900
12	100	Finance	Luis Popp	2007-12-07 00:00:00	6900	(null)
13	40	Human Resources	Susan Mavris	2002-06-07 00:00:00	6500	(null)
14	60	IT	David Austin	2005-06-25 00:00:00	4800	9000
15	60	IT	Alexander Hunold	2006-01-03 00:00:00	9000	4800
16	60	IT	Valli Pataballa	2006-02-05 00:00:00	4800	4200
17	60	IT	Diana Lorentz	2007-02-07 00:00:00	4200	6000
18	60	IT	Bruce Ernst	2007-05-21 00:00:00	6000	(null)
19	20	Marketing	Michael Hartstein	2004-02-17 00:00:00	13000	6000
20	20	Marketing	Pat Fay	2005-08-17 00:00:00	6000	(null)

2. 분석함수 실습 – LEAD(expr, offset, default) : 후행 로우값

· 부서별, 입사일자 순, 직후 사원의 급여를 구하라



```
SELECT b.department_id, b.department_name,
       a.first_name || ' ' || a.last_name as emp_name,
       a.hire_date,
       a.salary ,
  LEAD(salary, 1, 0) OVER (PARTITION BY b.department_id
                           ORDER BY a.hire_date ) lead_salary
  FROM employees a,
       departments b
 WHERE a.department_id = b.department_id
 ORDER BY 2, 4 ;
```

	DEPARTMENT_ID	DEPARTMENT_NAME	EMP_NAME	HIRE_DATE	SALARY	LEAD_SALARY
1	110	Accounting	William Gietz	2002-06-07 00:00:00	8300	0
2	110	Accounting	Shelley Higgins	2002-06-07 00:00:00	12008	8300
3	10	Administration	Jennifer Whalen	2003-09-17 00:00:00	4400	0
4	90	Executive	Lex De Haan	2001-01-13 00:00:00	17000	24000
5	90	Executive	Steven King	2003-06-17 00:00:00	24000	17000
6	90	Executive	Neena Kochhar	2005-09-21 00:00:00	17000	0
7	100	Finance	Daniel Faviet	2002-08-16 00:00:00	9000	12008
8	100	Finance	Nancy Greenberg	2002-08-17 00:00:00	12008	8200
9	100	Finance	John Chen	2005-09-28 00:00:00	8200	7700
10	100	Finance	Ismael Sciarra	2005-09-30 00:00:00	7700	7800
11	100	Finance	Jose Manuel Urman	2006-03-07 00:00:00	7800	6900
12	100	Finance	Luis Popp	2007-12-07 00:00:00	6900	0
13	40	Human Resources	Susan Mavris	2002-06-07 00:00:00	6500	0
14	60	IT	David Austin	2005-06-25 00:00:00	4800	9000
15	60	IT	Alexander Hunold	2006-01-03 00:00:00	9000	4800
16	60	IT	Valli Pataballa	2006-02-05 00:00:00	4800	4200
17	60	IT	Diana Lorentz	2007-02-07 00:00:00	4200	6000
18	60	IT	Bruce Ernst	2007-05-21 00:00:00	6000	0

2. 분석함수 실습 – LEAD(expr, offset, default) : 후행 로우값

· 부서별, 입사일자 순, 2 로우 후 사원의 급여를 구하라

```
SELECT b.department_id, b.department_name,
       a.first_name || ' ' || a.last_name as emp_name,
       a.hire_date,
       a.salary ,
  LEAD(salary, 2, 0) OVER (PARTITION BY b.department_id
                           ORDER BY a.hire_date ) lead_salary
  FROM employees a,
       departments b
 WHERE a.department_id = b.department_id
 ORDER BY 2, 4 ;
```

	DEPARTMENT_ID	DEPARTMENT_NAME	EMP_NAME	HIRE_DATE	SALARY	LEAD_SALARY
1	110	Accounting	William Gietz	2002-06-07 00:00:00	8300	0
2	110	Accounting	Shelley Higgins	2002-06-07 00:00:00	12008	0
3	10	Administration	Jennifer Whalen	2003-09-17 00:00:00	4400	0
4	90	Executive	Lex De Haan	2001-01-13 00:00:00	17000	17000
5	90	Executive	Steven King	2003-06-17 00:00:00	24000	0
6	90	Executive	Neena Kochhar	2005-09-21 00:00:00	17000	0
7	100	Finance	Daniel Faviet	2002-08-16 00:00:00	9000	8200
8	100	Finance	Nancy Greenberg	2002-08-17 00:00:00	12008	7700
9	100	Finance	John Chen	2005-09-28 00:00:00	8200	7800
10	100	Finance	Ismael Sciarra	2005-09-30 00:00:00	7700	6900
11	100	Finance	Jose Manuel Urman	2006-03-07 00:00:00	7800	0
12	100	Finance	Luis Popp	2007-12-07 00:00:00	6900	0
13	40	Human Resources	Susan Mavris	2002-06-07 00:00:00	6500	0

2. 분석함수 실습 – LAG(expr, offset, default) : 선행 로우값

· 부서별, 입사일자 순, 직전 사원의 급여를 구하라

```
SELECT b.department_id, b.department_name,
       a.first_name || ' ' || a.last_name as emp_name,
       a.hire_date,
       a.salary ,
LAG(salary, 1, 0) OVER (PARTITION BY b.department_id
                        ORDER BY a.hire_date ) lag_salary
 FROM employees a,
      departments b
WHERE a.department_id = b.department_id
ORDER BY 2, 4 ;
```

	DEPARTMENT_ID	DEPARTMENT_NAME	EMP_NAME	HIRE_DATE	SALARY	LAG_SALARY
1	110	Accounting	William Gietz	2002-06-07 00:00:00	8300	12008
2	110	Accounting	Shelley Higgins	2002-06-07 00:00:00	12008	0
3	10	Administration	Jennifer Whalen	2003-09-17 00:00:00	4400	0
4	90	Executive	Lex De Haan	2001-01-13 00:00:00	17000	0
5	90	Executive	Steven King	2003-06-17 00:00:00	24000	17000
6	90	Executive	Neena Kochhar	2005-09-21 00:00:00	17000	24000
7	100	Finance	Daniel Faviet	2002-08-16 00:00:00	9000	0
8	100	Finance	Nancy Greenberg	2002-08-17 00:00:00	12008	9000
9	100	Finance	John Chen	2005-09-28 00:00:00	8200	12008
10	100	Finance	Ismael Sciarra	2005-09-30 00:00:00	7700	8200
11	100	Finance	Jose Manuel Urman	2006-03-07 00:00:00	7800	7700
12	100	Finance	Luis Popp	2007-12-07 00:00:00	6900	7800
13	40	Human Resources	Susan Mavris	2002-06-07 00:00:00	6500	0
14	60	IT	David Austin	2005-06-25 00:00:00	4800	0
15	60	IT	Alexander Hunold	2006-01-03 00:00:00	9000	4800
16	60	IT	Valli Pataballa	2006-02-05 00:00:00	4800	9000
17	60	IT	Diana Lorentz	2007-02-07 00:00:00	4200	4800
18	60	IT	Bruce Ernst	2007-05-21 00:00:00	6000	4200
19	20	Marketing	Michael Hartstein	2004-02-17 00:00:00	13000	0
20	20	Marketing	Pat Fay	2005-08-17 00:00:00	6000	13000

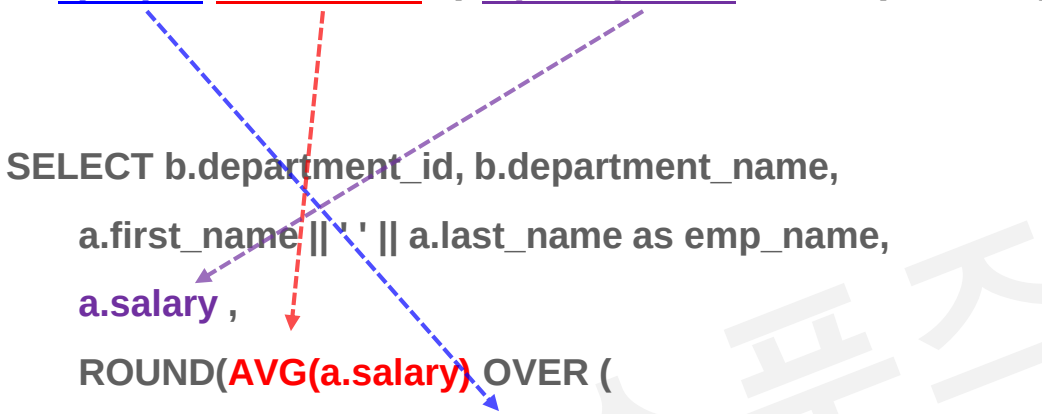
2. 분석함수 실습 – LAG와 LEAD

```
SELECT b.department_id, b.department_name,
       a.first_name || ' ' || a.last_name as emp_name,
       a.hire_date,
       LAG(salary, 1, 0) OVER (PARTITION BY b.department_id
                               ORDER BY a.hire_date ) PrevSal,
       a.salary ,
       LEAD(salary, 1, 0) OVER (PARTITION BY b.department_id
                               ORDER BY a.hire_date ) NextSal
  FROM employees a,
       departments b
 WHERE a.department_id = b.department_id
 ORDER BY 2, 4 ;
```

	DEPARTMENT_ID	DEPARTMENT_NAME	EMP_NAME	HIRE_DATE	PREVSAL	SALARY	NEXTSAL
1	110	Accounting	William Gietz	2002-06-07 00:00:00	12008	8300	0
2	110	Accounting	Shelley Higgins	2002-06-07 00:00:00	0	12008	8300
3	10	Administration	Jennifer Whalen	2003-09-17 00:00:00	0	4400	0
4	90	Executive	Lex De Haan	2001-01-13 00:00:00	0	17000	24000
5	90	Executive	Steven King	2003-06-17 00:00:00	17000	24000	17000
6	90	Executive	Neena Kochhar	2005-09-21 00:00:00	24000	17000	0
7	100	Finance	Daniel Faviet	2002-08-16 00:00:00	0	9000	12008
8	100	Finance	Nancy Greenberg	2002-08-17 00:00:00	9000	12008	8200
9	100	Finance	John Chen	2005-09-28 00:00:00	12008	8200	7700
10	100	Finance	Ismael Sciarra	2005-09-30 00:00:00	8200	7700	7800
11	100	Finance	Jose Manuel Urman	2006-03-07 00:00:00	7700	7800	6900
12	100	Finance	Luis Popp	2007-12-07 00:00:00	7800	6900	0
13	40	Human Resources	Susan Mavris	2002-06-07 00:00:00	0	6500	0
14	60	IT	David Austin	2005-06-25 00:00:00	0	4800	9000
15	60	IT	Alexander Hunold	2006-01-03 00:00:00	4800	9000	4800
16	60	IT	Valli Pataballa	2006-02-05 00:00:00	9000	4800	4200
17	60	IT	Diana Lorentz	2007-02-07 00:00:00	4800	4200	6000
18	60	IT	Bruce Ernst	2007-05-21 00:00:00	4200	6000	0

2. 분석함수 실습 – 집계 함수 사용

· 부서별 평균 급여와 사원의 급여를 동시에 조회



```
SELECT b.department_id, b.department_name,
    a.first_name || ' ' || a.last_name as emp_name,
    a.salary ,
    ROUND(AVG(a.salary) OVER (
            PARTITION BY b.department_id
            ORDER BY b.department_id),0) dept_avg_sal
 FROM employees a,
    departments b
 WHERE a.department_id = b.department_id
 ORDER BY 2, 3;
```

	DEPARTMENT_ID	DEPARTMENT_NAME	EMP_NAME	SALARY	DEPT_AVG_SAL
1	110	Accounting	Shelley Higgins	12008	10154
2	110	Accounting	William Gietz	8300	10154
3	10	Administration	Jennifer Whalen	4400	4400
4	90	Executive	Lex De Haan	17000	19333
5	90	Executive	Neena Kochhar	17000	19333
6	90	Executive	Steven King	24000	19333
7	100	Finance	Daniel Faviet	9000	8601
8	100	Finance	Ismael Sciarra	7700	8601
9	100	Finance	John Chen	8200	8601
10	100	Finance	Jose Manuel Urman	7800	8601
11	100	Finance	Luis Popp	6900	8601
12	100	Finance	Nancy Greenberg	12008	8601
13	40	Human Resources	Susan Mavris	6500	6500
14	60	IT	Alexander Hunold	9000	5760
15	60	IT	Bruce Ernst	6000	5760
16	60	IT	David Austin	4800	5760
17	60	IT	Diana Lorentz	4200	5760
18	60	IT	Valli Pataballa	4800	5760
19	20	Marketing	Michael Hartstein	13000	9500
20	20	Marketing	Pat Fay	6000	9500

2. 분석함수 예시 – 집계 함수 사용

· 사원의 급여와 부서별 누적 급여 조회

```
SELECT b.department_id, b.department_name,
       a.first_name || ' ' || a.last_name as emp_name,
       a.salary ,
       ROUND(SUM(a.salary) OVER (
                  PARTITION BY b.department_id
                  ORDER BY a.salary  ),0) dept_cum_sum
  FROM employees a,
       departments b
 WHERE a.department_id = b.department_id
 ORDER BY 2, 4;
```

	DEPARTMENT_ID	DEPARTMENT_NAME	EMP_NAME	SALARY	DEPT_CUM_SUM
1	110	Accounting	William Gietz	8300	8300
2	110	Accounting	Shelley Higgins	12008	20308
3	10	Administration	Jennifer Whalen	4400	4400
4	90	Executive	Lex De Haan	17000	34000
5	90	Executive	Neena Kochhar	17000	34000
6	90	Executive	Steven King	24000	58000
7	100	Finance	Luis Popp	6900	6900
8	100	Finance	Ismael Sciarra	7700	14600
9	100	Finance	Jose Manuel Urman	7800	22400
10	100	Finance	John Chen	8200	30600
11	100	Finance	Daniel Faviet	9000	39600
12	100	Finance	Nancy Greenberg	12008	51608
13	40	Human Resources	Susan Mavris	6500	6500
14	60	IT	Diana Lorentz	4200	4200
15	60	IT	David Austin	4800	13800
16	60	IT	Valli Pataballa	4800	13800
17	60	IT	Bruce Ernst	6000	19800
18	60	IT	Alexander Hunold	9000	28800
19	20	Marketing	Pat Fay	6000	6000
20	20	Marketing	Michael Hartstein	13000	19000
21	70	Public Relations	Hermann Baer	10000	10000

2. 분석함수 예시 – RATIO_TO_REPORT()

· 부서별 사원 급여의 비율

```
SELECT b.department_id, b.department_name,
       a.first_name || ' ' || a.last_name as emp_name,
       a.salary ,
       ROUND(RATIO_TO_REPORT(a.salary)
            OVER (PARTITION BY b.department_id ),2) rates
  FROM employees a,
       departments b
 WHERE a.department_id = b.department_id
 ORDER BY 2, 4 DESC;
```

	DEPARTMENT_ID	DEPARTMENT_NAME	EMP_NAME	SALARY	RATES
1	110	Accounting	Shelley Higgins	12008	0.59
2	110	Accounting	William Gietz	8300	0.41
3	10	Administration	Jennifer Whalen	4400	1
4	90	Executive	Steven King	24000	0.41
5	90	Executive	Lex De Haan	17000	0.29
6	90	Executive	Neena Kochhar	17000	0.29
7	100	Finance	Nancy Greenberg	12008	0.23
8	100	Finance	Daniel Faviet	9000	0.17
9	100	Finance	John Chen	8200	0.16
10	100	Finance	Jose Manuel Urman	7800	0.15
11	100	Finance	Ismael Sciarra	7700	0.15
12	100	Finance	Luis Popp	6900	0.13
13	40	Human Resources	Susan Mavris	6500	1
14	60	IT	Alexander Hunold	9000	0.31
15	60	IT	Bruce Ernst	6000	0.21
16	60	IT	David Austin	4800	0.17
17	60	IT	Valli Pataballa	4800	0.17
18	60	IT	Diana Lorentz	4200	0.15
19	20	Marketing	Michael Hartstein	13000	0.68
20	20	Marketing	Pat Fay	6000	0.32
21	70	Public Relations	Hermann Baer	10000	1
22	30	Purchasing	Den Raphaely	11000	0.44
23	30	Purchasing	Alexander Khoo	3100	0.12
24	30	Purchasing	Shelli Baida	2900	0.12
25	30	Purchasing	Sigal Tobias	2800	0.11
26	30	Purchasing	Guy Himuro	2600	0.1
27	30	Purchasing	Karen Colmenares	2500	0.1

3. MSSQL

· MSSQL 혹은 SQL Server 라고 부름

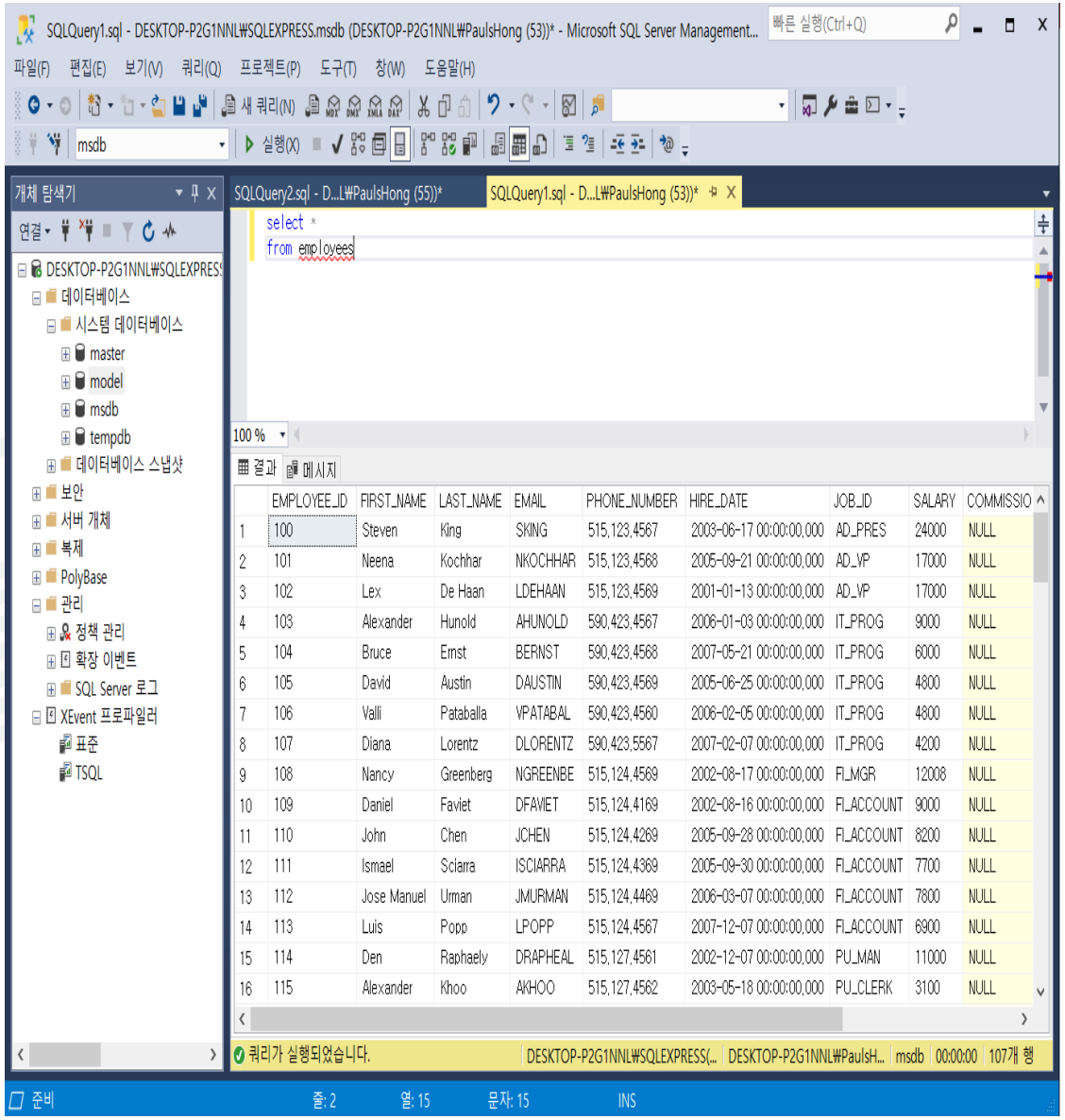
· Oracle 18c Express 버전처럼 무료 express 버전 제공

- SQL Server 2017 Express Edition

Express	데스크톱 및 소형 서버 데이터 기반의 애플리케이션(최대 10GB)을 교육하고 빌드하는 데 적합한 초급자용 무료 데이터베이스입니다.	해당 없음	체험판 다운로드하기	체험판

3. MSSQL

· SSMS (Sql Server Management Studio)
  - 오라클의 SQL Developer 같은 툴
  - DB 백업과 복구까지 할 수 있는 관리용 GUI 도구
  - SQL을 작성하고 결과도 확인 가능
  - 별로도 download & 설치



3. MSSQL

· 기본적인 SQL은 오라클과 동일

· 외부조인은 ANSI 문법 사용할 것

· 빌트인 함수, 컬럼의 데이터 형은 오라클과 차이 있음
  - 문자형 : VARCHAR
  - 날짜형 : DATETIME
  - 숫자형 : INT, FLOAT, DOUBLE, DECIMAL

3. MSSQL – 오라클과 차이점

· **sp_help** 테이블명 : 테이블 상세 정보, 테이블 외에 다른 객체도 사용 가능

· select **getdate()** : 현재 일자 반환

· **대소문자 구분 안함**

```
select *
from employees
 where first_name = 'steven'
```

결과 메시지

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE
1	100	Steven	King	SKING	515.123.4567	2003-06-17 00:0
2	128	Steven	Markle	SMARKLE	650.124.1434	2008-03-08 00:0

3. MSSQL – 오라클과 차이점

· 문자열 결합 : select 'a' + 'b' ➔ 'ab'

· SUBSTRING 함수 : select substring('abc', 2, 1) ➔ 'b'

· INSTR 함수 : select CHARINDEX('A', 'AB') ➔ 1

- 오라클 INSTR 함수와 매개변수 순서가 반대

· 문자열 길이 : select len('홍'), DATALENGTH('홍') ➔ 1, 2

- LENGTH → LEN, LENGTHB → DATALENGTH

3. MSSQL – 오라클과 차이점

· select **left**('abc', 1), **right**('abc', 1) ➔ a, c

- LEFT, RIGHT 함수는 MSSQL에만 있음

· select **IsNull**(null, ' b ') ➔ b

- NVL 함수 → IsNull

· select 17 **%** 3 ➔ 2

- 나머지 Mod 함수 → % 연산자

3. MSSQL – 오라클과 차이점

· SELECT ISNUMERIC('A'), ISNUMERIC('123')

	(열 이름 없음)	(열 이름 없음)
1	0	1

· SELECT ISDATE('20200110'), ISDATE('20200132')

(열 이름 없음)	(열 이름 없음)
1	0

· 현재일자 기준 1개월 후

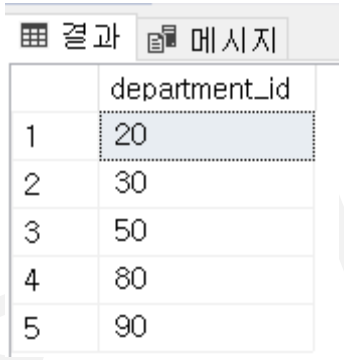
- SELECT ADD_MONTHS(SYSDATE, 1)
  FROM DUAL;

- SELECT DATEADD(MM, 1, GETDATE())

(열 이름 없음)
2020-02-21 15:34:42.700

3. MSSQL – 오라클과 차이점 (MINUS)

```
· SELECT department_id
   FROM employees
  WHERE manager_id = 100
  EXCEPT
 SELECT department_id
   FROM employees
  WHERE manager_id = 101;
```


결과 메시지

	department_id
1	20
2	30
3	50
4	80
5	90

3. MSSQL – 오라클과 차이점 (NULL)

· 오라클에서는 empty string('')은 Null임

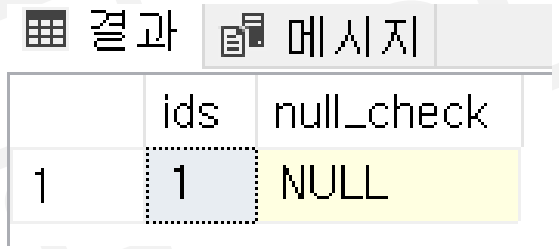
```
CREATE TABLE NULL_TEST (
        ids                 NUMBER,
        null_check      VARCHAR2(10) );

INSERT INTO NULL_TEST VALUES (1, NULL);
INSERT INTO NULL_TEST VALUES (2, '');
INSERT INTO NULL_TEST VALUES (3, 'A');

SELECT *
FROM NULL_TEST
WHERE null_check IS NULL;
```

	IDS	NULL_CHECK
1	1	(null)
2	2	(null)

3. MSSQL – 오라클과 차이점 (NULL)

· MSSQL에서는 empty string('')은 Null이 아님

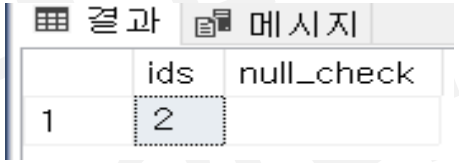
```
CREATE TABLE NULL_TEST (
        ids                int,
        null_check      VARCHAR(10) );

INSERT INTO NULL_TEST VALUES (1, NULL);
INSERT INTO NULL_TEST VALUES (2, '');
INSERT INTO NULL_TEST VALUES (3, 'A');

SELECT *
FROM NULL_TEST
WHERE null_check IS NULL;
```



3. MSSQL – 오라클과 차이점 (NULL)

· MSSQL에서는 empty string('')은 Null이 아님

```
SELECT *
FROM NULL_TEST
WHERE null_check = '';
```



3. MSSQL – 오라클과 차이점 (NULL)

· MSSQL에서는 empty string('')은 Null이 아님

```
CREATE TABLE NULL_TEST2 (
        ids             int,
        null_check      VARCHAR(10)  NOT NULL );

INSERT INTO NULL_TEST2 VALUES (1, NULL);
```

3. MSSQL – 오라클과 차이점 (NULL)

· MSSQL에서는 empty string('')은 Null이 아님

```
INSERT INTO NULL_TEST2 VALUES (2, '');
INSERT INTO NULL_TEST2 VALUES (3, 'A');

SELECT *
FROM null_test2;
```

메시지

(1개 행 적용됨)

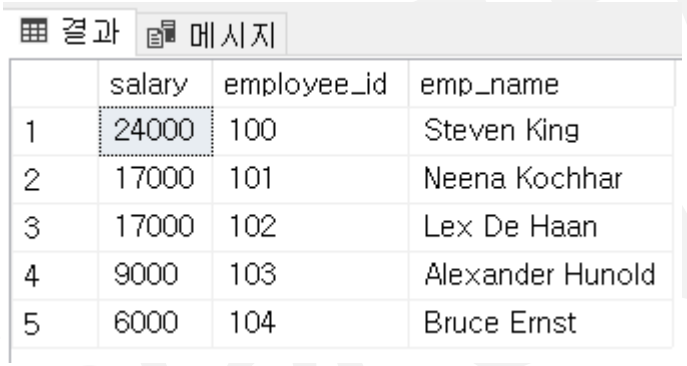
(1개 행 적용됨)

결과 | 메시지

	ids	null_check
1	2	
2	3	A

3. MSSQL – 오라클과 차이점

```
select top 5 salary, employee_id
      ,first_name + ' ' + last_name emp_name
from employees;
```



결과 메시지

	salary	employee_id	emp_name
1	24000	100	Steven King
2	17000	101	Neena Kochhar
3	17000	102	Lex De Haan
4	9000	103	Alexander Hunold
5	6000	104	Bruce Ernst

3. MSSQL – 오라클과 차이점 (프로시저)

```
CREATE PROCEDURE sp_emp
As
SELECT a.employee_id,
       a.first_name + ' ' + a.last_name emp_name,
   a.job_id,
   a.salary,
   a.department_id,
   b.department_name
  into #emp
  FROM employees a,
       departments b
WHERE a.department_id = b.department_id;

SELECT * FROM #emp;
go
```


3. MSSQL – 주 차이점 (프로시저 실행)

exec sp_emp

	employee_id	emp_name	job_id	salary	department_id	department_name
1	100	Steven King	AD_PRES	24000	90	Executive
2	101	Neena Kochhar	AD_VP	17000	90	Executive
3	102	Lex De Haan	AD_VP	17000	90	Executive
4	103	Alexander Hunold	IT_PROG	9000	60	IT
5	104	Bruce Ernst	IT_PROG	6000	60	IT
6	105	David Austin	IT_PROG	4800	60	IT
7	106	Valli Pataballa	IT_PROG	4800	60	IT
8	107	Diana Lorentz	IT_PROG	4200	60	IT
9	108	Nancy Greenberg	FI_MGR	12008	100	Finance
10	109	Daniel Faviet	FI_ACCOUNT	9000	100	Finance
11	110	John Chen	FI_ACCOUNT	8200	100	Finance
12	111	Ismael Sciarra	FI_ACCOUNT	7700	100	Finance
13	112	Jose Manuel Urman	FI_ACCOUNT	7800	100	Finance
14	113	Luis Popp	FI_ACCOUNT	6900	100	Finance

학습정리

· 분석함수는 로우를 특정 값 별로 그룹으로 묶어, 이 그룹별 집계 값을 산출한다.

· 분석함수는 GROUP BY 절과는 달리 그룹별로 집계 값을 산출하지만, 로우 수를 줄이지는 않는다.

· 분석함수에는 일반 집계 함수와 ROW_NUMBER, RANK, DENSE_RANK, LAG, LEAD 등이 있다.

