5-2. 테이블 간 관계 맺기 – 조인2
(ANSI조인, Cartesian Product, 셀프조인)

홍형경
chariehong@gmail.com
2020.01

1. ANSI 조인 (ANSI Join)

· ANSI 표준 문법으로 작성한 조인 방법

· 내부조인, 외부조인을 ANSI 문법에 맞게 작성한 쿼리

· 내부조인 : INNER JOIN

· 외부조인 : LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN

· FULL OUTER JOIN은 ANSI 문법으로만 구현 가능

· ANSI 문법은 다른 DBMS에서도 사용 가능 → (+) 기호는 오라클에서만 사용

1. ANSI 조인 (ANSI Join)

· ANSI 조인 문법의 특징

- 조인 조건 절을 WHERE 절이 아닌 FROM 절에 기술

- 조인 조건은 ON 다음에 기술

- 조인 조건이 여러 개이면 AND 연산자 사용해 조건 기술

- 조인 조건 외에 다른 조건은 WHERE 절에서 기술

2. ANSI 내부조인

· 기존문법

```
SELECT  a.employee_id    emp_id,
        a.department_id  a_dept_id,
        b.department_id b_dept_id,
        b.department_name dept_name
   FROM employees a, departments b
  WHERE a.department_id = b.department_id
  ORDER BY a.department_id;
```

· ANSI 문법

```
SELECT  a.employee_id    emp_id,
        a.department_id  a_dept_id,
        b.department_id b_dept_id,
        b.department_name dept_name
   FROM employees a
   INNER JOIN departments b
      ON a.department_id = b.department_id
  ORDER BY a.department_id;
```

* INNER 생략 가능

2. ANSI 내부조인

· ANSI 문법

```
SELECT  a.employee_id    emp_id,
            a.department_id  a_dept_id,
             b.department_id b_dept_id,
             b.department_name dept_name
    FROM employees a
    INNER JOIN departments b
        ON a.department_id = b.department_id
  ORDER BY a.employee_id;
```

	EMP_ID	A_DEPT_ID	B_DEPT_ID	DEPT_NAME
1	100	90	90	Executive
2	101	90	90	Executive
3	102	90	90	Executive
4	103	60	60	IT
5	104	60	60	IT
6	105	60	60	IT
7	106	60	60	IT
8	107	60	60	IT
9	108	100	100	Finance
10	109	100	100	Finance
11	110	100	100	Finance
12	111	100	100	Finance
13	112	100	100	Finance
14	113	100	100	Finance

3. ANSI 외부조인 – Left outer join

· 기존문법

```
SELECT  a.employee_id    emp_id,
        a.department_id  a_dept_id,
        b.department_id b_dept_id,
        b.department_name dept_name
   FROM employees a, departments b
  WHERE a.department_id = b.department_id(+)
  ORDER BY a.employee_id;
```

· ANSI 문법

```
SELECT  a.employee_id    emp_id,
        a.department_id  a_dept_id,
        b.department_id b_dept_id,
        b.department_name dept_name
   FROM employees a
   LEFT OUTER JOIN departments b
     ON a.department_id = b.department_id
  ORDER BY a.employee_id;
```

* OUTER 생략 가능

3. ANSI 외부조인 – Left outer join

· ANSI 문법

```
SELECT  a.employee_id    emp_id,
        a.department_id  a_dept_id,
        b.department_id b_dept_id,
        b.department_name dept_name
   FROM employees a
   LEFT OUTER JOIN departments b
      ON a.department_id = b.department_id
ORDER BY a.employee_id;
```

	EMP_ID	A_DEPT_ID	B_DEPT_ID	DEPT_NAME
1	100	90	90	Executive
2	101	90	90	Executive
3	102	90	90	Executive
4	103	60	60	IT
5	104	60	60	IT
6	105	60	60	IT
7	106	60	60	IT
8	107	60	60	IT
9	108	100	100	Finance
10	109	100	100	Finance

	EMP_ID	A_DEPT_ID	B_DE...	DEPT_NAME
76	175	80	80	Sales
77	176	80	80	Sales
78	177	80	80	Sales
79	178	(null)	(null)	(null)
80	179	80	80	Sales
81	180	50	50	Shipping
82	181	50	50	Shipping

3. ANSI 외부조인 – Right outer join

· 기존문법

```
SELECT  a.employee_id    emp_id,
        a.department_id  a_dept_id,
        b.department_id b_dept_id,
        b.department_name dept_name
   FROM employees a, departments b
  WHERE a.department_id(+) = b.department_id
  ORDER BY a.employee_id;
```

· ANSI 문법

```
SELECT  a.employee_id    emp_id,
        a.department_id  a_dept_id,
        b.department_id b_dept_id,
        b.department_name dept_name
   FROM employees a
   RIGHT OUTER JOIN departments b
     ON a.department_id = b.department_id
  ORDER BY a.employee_id;
```

* OUTER 생략 가능

3. ANSI 외부조인 – Right outer join

· ANSI 문법

```
SELECT  a.employee_id    emp_id,
        a.department_id  a_dept_id,
        b.department_id b_dept_id,
        b.department_name dept_name
  FROM employees a
  RIGHT OUTER JOIN departments b
       ON a.department_id = b.department_id
 ORDER BY a.employee_id, b.department_id;
```

	EMP_ID	A_DEPT_ID	B_DEPT_ID	DEPT_NAME
1	100	90	90	Executive
2	101	90	90	Executive
3	102	90	90	Executive
4	103	60	60	IT
5	104	60	60	IT
6	105	60	60	IT
7	106	60	60	IT
8	107	60	60	IT

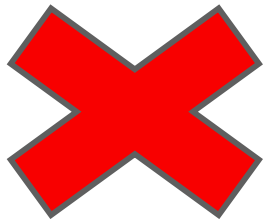
	EMP_ID	A_DEPT_ID	B_DEPT_ID	DEPT_NAME
76	175	80	80	Sales
77	176	80	80	Sales
78	177	80	80	Sales
79	179	80	80	Sales
80	180	50	50	Shipping

	EMP_ID	A_DEPT_ID	B_DEPT_ID	DEPT_NAME
105	205	110	110	Accounting
106	206	110	110	Accounting
107	(null)	(null)	120	Treasury
108	(null)	(null)	130	Corporate Tax
109	(null)	(null)	140	Control And Credit
110	(null)	(null)	150	Shareholder Services
111	(null)	(null)	160	Benefits
112	(null)	(null)	170	Manufacturing
113	(null)	(null)	180	Construction
114	(null)	(null)	190	Contracting
115	(null)	(null)	200	Operations
116	(null)	(null)	210	IT Support
117	(null)	(null)	220	NOC
118	(null)	(null)	230	IT Helpdesk
119	(null)	(null)	240	Government Sales
120	(null)	(null)	250	Retail Sales
121	(null)	(null)	260	Recruiting

3. ANSI 외부조인 – Full outer join

· 기존문법

```
SELECT  a.employee_id    emp_id,
        a.department_id  a_dept_id,
        b.department_id b_dept_id,
        b.department_name dept_name
   FROM employees a, departments b
  WHERE a.department_id(+) = b.department_id(+)
  ORDER BY b.department_id;
```



· ANSI 문법

```
SELECT  a.employee_id    emp_id,
        a.department_id  a_dept_id,
        b.department_id b_dept_id,
        b.department_name dept_name
   FROM employees a
   FULL OUTER JOIN departments b
     ON a.department_id = b.department_id
  ORDER BY a.employee_id,
           b.department_id;
```

* OUTER 생략 가능

3. ANSI 외부조인 – Full outer join

· 기존문법

```
SELECT  a.employee_id    emp_id,
        a.department_id  a_dept_id,
        b.department_id b_dept_id,
        b.department_name dept_name
   FROM employees a, departments b
  WHERE a.department_id(+) = b.department_id(+)
  ORDER BY b.department_id;
```

```
ORA-01468: outer-join된 테이블은 1개만 지정할 수 있습니다
01468, 00000 - "a predicate may reference only one outer-joined table"
*Cause:
*Action:
6행, 27열에서 오류 발생
```


3. ANSI 외부조인 – Full outer join

· ANSI 문법

```
SELECT  a.employee_id    emp_id,
        a.department_id  a_dept_id,
        b.department_id b_dept_id,
        b.department_name dept_name
   FROM employees a
   FULL OUTER JOIN departments b
     ON a.department_id = b.department_id
 ORDER BY a.employee_id,
          b.department_id;
```


3. ANSI 외부조인 – Full outer join

FULL OUTER JOIN 결과

emp_id	A_DEPT_ID	B_DEPT_ID	DEPT_NAME
100	90	90	Executive
101	90	90	Executive
102	90	90	Executive
103	60	60	IT
…	…	…	..
178	(null)	(null)	(null)
179	80	80	Sales
…	…	…	…
(null)	(null)	120	Treasury
(null)	(null)	130	Corporate Tax
…	…	…	…

3. ANSI 외부조인 – Full outer join

FULL OUTER JOIN 결과

	EMP_ID	A_DEPT_ID	B_DEPT_ID	DEPT_NAME
1	100	90	90	Executive
2	101	90	90	Executive
3	102	90	90	Executive
4	103	60	60	IT
5	104	60	60	IT
6	105	60	60	IT
7	106	60	60	IT

	EMP_ID	A_DEPT_ID	B_DEPT_ID	DEPT_NAME
76	175	80	80	Sales
77	176	80	80	Sales
78	177	80	80	Sales
79	178	(null)	(null)	(null)
80	179	80	80	Sales

	EMP_ID	A_DEPT_ID	B_DEPT_ID	DEPT_NAME
105	204	70	70	Public Relations
106	205	110	110	Accounting
107	206	110	110	Accounting
108	(null)	(null)	120	Treasury
109	(null)	(null)	130	Corporate Tax
110	(null)	(null)	140	Control And Credit
111	(null)	(null)	150	Shareholder Services
112	(null)	(null)	160	Benefits
113	(null)	(null)	170	Manufacturing
114	(null)	(null)	180	Construction
115	(null)	(null)	190	Contracting
116	(null)	(null)	200	Operations
117	(null)	(null)	210	IT Support
118	(null)	(null)	220	NOC
119	(null)	(null)	230	IT Helpdesk
120	(null)	(null)	240	Government Sales
121	(null)	(null)	250	Retail Sales
122	(null)	(null)	260	Recruiting

4. ANSI 조인 실습

```
SELECT a.employee_id,
       a.first_name || ' ' || a.last_name emp_names,
       a.job_id, b.job_id, b.job_title
  FROM employees a
  INNER JOIN jobs b
    ON a.job_id = b.job_id
 ORDER BY 1;
```

	EMPLOYEE_ID	EMP_NAMES	JOB_ID	JOB_ID_1	JOB_TITLE
1	100	Steven King	AD_PRES	AD_PRES	President
2	101	Neena Kochhar	AD_VP	AD_VP	Administration Vice President
3	102	Lex De Haan	AD_VP	AD_VP	Administration Vice President
4	103	Alexander Hunold	IT_PROG	IT_PROG	Programmer
5	104	Bruce Ernst	IT_PROG	IT_PROG	Programmer
6	105	David Austin	IT_PROG	IT_PROG	Programmer
7	106	Valli Pataballa	IT_PROG	IT_PROG	Programmer
8	107	Diana Lorentz	IT_PROG	IT_PROG	Programmer
9	108	Nancy Greenberg	FI_MGR	FI_MGR	Finance Manager
10	109	Daniel Faviet	FI_ACCOUNT	FI_ACCOUNT	Accountant
11	110	John Chen	FI_ACCOUNT	FI_ACCOUNT	Accountant

4. ANSI 조인 실습

```
SELECT a.employee_id,
       a.first_name || ' ' || a.last_name emp_names,
      b.job_title
     ,c.department_id ,c.department_name
 FROM employees a
 INNER JOIN jobs b
     ON a.job_id        = b.job_id
 INNER JOIN departments c
   ON a.department_id = c.department_id
ORDER BY 1;
```

	EMPLOYEE_ID	EMP_NAMES	JOB_TITLE	DEPARTMENT_ID	DEPARTMENT_NAME
1	100	Steven King	President	90	Executive
2	101	Neena Kochhar	Administration Vice President	90	Executive
3	102	Lex De Haan	Administration Vice President	90	Executive
4	103	Alexander Hunold	Programmer	60	IT
5	104	Bruce Ernst	Programmer	60	IT
6	105	David Austin	Programmer	60	IT
7	106	Valli Pataballa	Programmer	60	IT
8	107	Diana Lorentz	Programmer	60	IT
9	108	Nancy Greenberg	Finance Manager	100	Finance
10	109	Daniel Faviet	Accountant	100	Finance
11	110	John Chen	Accountant	100	Finance

4. ANSI 조인 실습

```
SELECT a.employee_id,
       a.first_name || ' ' || a.last_name emp_names,
       a.job_id, b.job_id, b.job_title
      ,c.department_id ,c.department_name
  FROM employees a
  INNER JOIN jobs b
    ON a.job_id       = b.job_id
  INNER JOIN departments c
    ON a.department_id = c.department_id
 WHERE b.job_id = 'SH_CLERK'
 ORDER BY 1;
```

	EMPLOYEE_ID	EMP_NAMES	JOB_ID	JOB_ID_1	JOB_TITLE	DEPARTMENT_ID	DEPARTMENT_NAME
1	180	Winston Taylor	SH_CLERK	SH_CLERK	Shipping Clerk	50	Shipping
2	181	Jean Fleaur	SH_CLERK	SH_CLERK	Shipping Clerk	50	Shipping
3	182	Martha Sullivan	SH_CLERK	SH_CLERK	Shipping Clerk	50	Shipping
4	183	Girard Geoni	SH_CLERK	SH_CLERK	Shipping Clerk	50	Shipping
5	184	Nandita Sarchand	SH_CLERK	SH_CLERK	Shipping Clerk	50	Shipping
6	185	Alexis Bull	SH_CLERK	SH_CLERK	Shipping Clerk	50	Shipping
7	186	Julia Dellinger	SH_CLERK	SH_CLERK	Shipping Clerk	50	Shipping
8	187	Anthony Cabrio	SH_CLERK	SH_CLERK	Shipping Clerk	50	Shipping
9	188	Kelly Chung	SH_CLERK	SH_CLERK	Shipping Clerk	50	Shipping

4. ANSI 조인 실습

```
SELECT a.employee_id,
       a.first_name || ' ' || a.last_name emp_names,
       c.department_id, c.department_name,
       d.location_id, d.street_address, d.city
  FROM employees a
  LEFT JOIN departments c
    ON a.department_id = c.department_id
 INNER JOIN locations d
    ON c.location_id   = d.location_id
 ORDER BY 1;
```

	EMPLOYEE_ID	EMP_NAMES	DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	STREET_ADDRESS	CITY
73	172	Elizabeth Bates	80	Sales	2500	Magdalen Centre, The Oxford Science Park	Oxford
74	173	Sundita Kumar	80	Sales	2500	Magdalen Centre, The Oxford Science Park	Oxford
75	174	Ellen Abel	80	Sales	2500	Magdalen Centre, The Oxford Science Park	Oxford
76	175	Alyssa Hutton	80	Sales	2500	Magdalen Centre, The Oxford Science Park	Oxford
77	176	Jonathon Taylor	80	Sales	2500	Magdalen Centre, The Oxford Science Park	Oxford
78	177	Jack Livingston	80	Sales	2500	Magdalen Centre, The Oxford Science Park	Oxford
79	179	Charles Johnson	80	Sales	2500	Magdalen Centre, The Oxford Science Park	Oxford
80	180	Winston Taylor	50	Shipping	1500	2011 Interiors Blvd	South San Francisco
81	181	Jean Fleaur	50	Shipping	1500	2011 Interiors Blvd	South San Francisco
82	182	Martha Sullivan	50	Shipping	1500	2011 Interiors Blvd	South San Francisco
83	183	Girard Geoni	50	Shipping	1500	2011 Interiors Blvd	South San Francisco
84	184	Nandita Sarchand	50	Shipping	1500	2011 Interiors Blvd	South San Francisco

4. ANSI 조인 실습

```
SELECT a.employee_id,
       a.first_name || ' ' || a.last_name emp_names,
       c.department_id, c.department_name,
       d.location_id, d.street_address, d.city
  FROM employees a
  LEFT JOIN departments c
    ON a.department_id = c.department_id
  LEFT JOIN locations d
    ON c.location_id   = d.location_id
 ORDER BY 1;
```

	EMPLOYEE_ID	EMP_NAMES	DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	STREET_ADDRESS	CITY
76	175	Alyssa Hutton	80	Sales	2500	Magdalen Centre, The Oxford Science Park	Oxford
77	176	Jonathon Taylor	80	Sales	2500	Magdalen Centre, The Oxford Science Park	Oxford
78	177	Jack Livingston	80	Sales	2500	Magdalen Centre, The Oxford Science Park	Oxford
79	178	Kimberely Grant	(null)	(null)	(null)	(null)	(null)
80	179	Charles Johnson	80	Sales	2500	Magdalen Centre, The Oxford Science Park	Oxford
81	180	Winston Taylor	50	Shipping	1500	2011 Interiors Blvd	South San Francisco
82	181	Jean Fleaur	50	Shipping	1500	2011 Interiors Blvd	South San Francisco
83	182	Martha Sullivan	50	Shipping	1500	2011 Interiors Blvd	South San Francisco

5. 일반 조인과 ANSI 조인 문법

· 일반 조인을 사용해야 할까? 아니면 ANSI 조인을 써야 할까?

· 내부 조인

- 일반 조인 문법 (WHERE 절에 조인 조건 기술)
- 가독성 측면에서 좋음

· 외부 조인

- ANSI 문법 사용
- (+)는 오라클 고유의 문법 ➔ 다른 DBMS에서 사용 불가
- ANSI 외부조인 문법이 가독성이 더 좋음
- FULL OUTER JOIN은 ANSI 문법만 가능

6. Cartesian Product

· 조인 조건이 없는 조인

· 조인 참여 테이블을 FROM 절에 기술하고 WHERE 절에 조인 조건 기술하지 않음

· 조인 조건이 없으므로, 두 테이블 기준 모든 조합(경우의 수)의 로우가 조회됨
  - A 테이블(3건), B 테이블(5건) ➔ 3 * 5 = 15건이 조회됨

· 거의 사용되지 않음

· ANSI 문법 ➔ CROSS JOIN

6. Cartesian Product

```
SELECT a.region_name, b.department_id,
       b.department_name
  FROM regions a
      ,departments b
  WHERE 1=1;
```

-- 4 * 24 = 104

	REGION_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
87	Middle East and Africa	90	Executive
88	Middle East and Africa	100	Finance
89	Middle East and Africa	110	Accounting
90	Middle East and Africa	120	Treasury
91	Middle East and Africa	130	Corporate Tax
92	Middle East and Africa	140	Control And Credit
93	Middle East and Africa	150	Shareholder Services
94	Middle East and Africa	160	Benefits
95	Middle East and Africa	170	Manufacturing
96	Middle East and Africa	180	Construction
97	Middle East and Africa	190	Contracting
98	Middle East and Africa	200	Operations
99	Middle East and Africa	210	IT Support
100	Middle East and Africa	220	NOC
101	Middle East and Africa	230	IT Helpdesk
102	Middle East and Africa	240	Government Sales
103	Middle East and Africa	250	Retail Sales
104	Middle East and Africa	260	Recruiting

6. Cartesian Product

· 조인이라고는 하지만 엄밀히 말하면 조인은 아님

➔ 조인조건이 없으므로…

· 실제 사용되는 경우는 거의 없음

· 만약 이런 조인 결과를 보게 되면 …

➔ 아, 조인 조건이 누락됐구나!

7. 셀프 조인 (Self Join)

· 자기 자신과 조인

· 동일한 테이블 끼리 조인

```
SELECT a.employee_id
      ,a.first_name || ' ' || a.last_name emp_name
      ,a.manager_id
      ,b.first_name || ' ' || b.last_name manager_name

 FROM employees a
         ,employees b
WHERE a.manager_id = b.employee_id

ORDER BY 1;
```

employee_id	emp_name	manager_id	manager_name
101	Neena Kochhar	100	Steven King
102	Lex De Hann	100	Steven King
…	…	…	…
105	David Austin	103	Alexander Hunold
…	…	..	…

7. 셀프 조인 (Self Join)

```
SELECT a.employee_id
      ,a.first_name || ' ' || a.last_name emp_name
      ,a.manager_id
      ,b.first_name || ' ' || b.last_name manager_name

 FROM employees a
         ,employees b
WHERE a.manager_id = b.employee_id

ORDER BY 1;
```

	EMPLOYEE_ID	EMP_NAME	MANAGER_ID	MANAGER_NAME
1	101	Neena Kochhar	100	Steven King
2	102	Lex De Haan	100	Steven King
3	103	Alexander Hunold	102	Lex De Haan
4	104	Bruce Ernst	103	Alexander Hunold
5	105	David Austin	103	Alexander Hunold
6	106	Valli Pataballa	103	Alexander Hunold
7	107	Diana Lorentz	103	Alexander Hunold
8	108	Nancy Greenberg	101	Neena Kochhar
9	109	Daniel Faviet	108	Nancy Greenberg
10	110	John Chen	108	Nancy Greenberg
11	111	Ismael Sciarra	108	Nancy Greenberg
12	112	Jose Manuel Urman	108	Nancy Greenberg
13	113	Luis Popp	108	Nancy Greenberg
14	114	Den Raphaely	100	Steven King
15	115	Alexander Khoo	114	Den Raphaely
16	116	Shelli Baida	114	Den Raphaely
17	117	Sigal Tobias	114	Den Raphaely
18	118	Guy Himuro	114	Den Raphaely
19	119	Karen Colmenares	114	Den Raphaely
20	120	Matthew Weiss	100	Steven King
21	121	Adam Fripp	100	Steven King

7. 셀프 조인 (Self Join)

```
SELECT a.employee_id
      ,a.first_name || ' ' || a.last_name emp_name
      ,a.manager_id
      ,b.first_name || ' ' || b.last_name manager_name
 FROM employees a
 INNER JOIN employees b
    ON a.manager_id = b.employee_id
ORDER BY 1;
```

	EMPLOYEE_ID	EMP_NAME	MANAGER_ID	MANAGER_NAME
1	101	Neena Kochhar	100	Steven King
2	102	Lex De Haan	100	Steven King
3	103	Alexander Hunold	102	Lex De Haan
4	104	Bruce Ernst	103	Alexander Hunold
5	105	David Austin	103	Alexander Hunold
6	106	Valli Pataballa	103	Alexander Hunold
7	107	Diana Lorentz	103	Alexander Hunold
8	108	Nancy Greenberg	101	Neena Kochhar
9	109	Daniel Faviet	108	Nancy Greenberg
10	110	John Chen	108	Nancy Greenberg
11	111	Ismael Sciarra	108	Nancy Greenberg
12	112	Jose Manuel Urman	108	Nancy Greenberg
13	113	Luis Popp	108	Nancy Greenberg
14	114	Den Raphaely	100	Steven King
15	115	Alexander Khoo	114	Den Raphaely
16	116	Shelli Baida	114	Den Raphaely
17	117	Sigal Tobias	114	Den Raphaely
18	118	Guy Himuro	114	Den Raphaely
19	119	Karen Colmenares	114	Den Raphaely
20	120	Matthew Weiss	100	Steven King
21	121	Adam Fripp	100	Steven King

학습정리

· ANSI 조인은 ANSI 문법에 맞게 내부조인과 외부조인을 구현한 쿼리를 말한다.

· ANSI 외부조인 중 FULL OUTER JOIN은 ANSI 문법으로만 처리할 수 있다.

· 셀프조인은 동일한 테이블에 대해 자기 자신과의 조인을 말한다.

Quiz

1. Jobs 테이블에는 min_salary와 max_salary란 컬럼이 있는데, 이는 해당 job_id에 대한 최소와 최대급여 금액을 담고 있습니다. Jobs 테이블과 employees 테이블을 조인하고 사원의 급여가 최소와 최대급여 금액을 벗어난 사원이 있는지 조회하는 쿼리를 작성해 보세요.

Quiz

2. 아래 외부조인 문장을 실행하면 내부조인을 한 것과 결과가 같습니다. 왜 이런 결과가 나왔는지 설명해 보세요.

```
SELECT a.employee_id, a.first_name || ' ' || a.last_name emp_names, b.*
  FROM employees a,
       job_history b
 WHERE a.employee_id(+) = b.employee_id
 ORDER BY 1;
```


Quiz

3. 실습시간 마지막에 배웠던 셀프조인의 경우 사번이 100번인 Steven King은 조회되지 않습니다. 그 이유는 뭘까요?

Quiz

4. 실습시간 마지막에 배웠던 셀프조인에서 누락된 사번이 100번인 Steven King 까지 조회되도록 쿼리를 작성해 보세요.

