

5-1. 테이블 간 관계 맺기 - 조인1 (내부조인, 외부조인)

홍형경

chariehong@gmail.com

2020.06

1. 조인 (Join)

- RDBMS 특징 : **중복 데이터 저장 회피** 목적으로 데이터 성격에 따라 **테이블 분리**
(예, 사원 테이블, 부서 테이블, ...)
- 한 테이블만 읽어서는 필요한 정보가 부족
- employees 테이블에는 부서번호(department_id) 컬럼만 있어 부서명을 알 수 없음
- 부서명을 가져오려면 departments 테이블과 연결 필요
- 이런 테이블 간 연결 작업을 조인(Join)이라 함

1. 조인 (Join)

- 테이블 간 연결(조인)을 위해서는 연결고리 역할을 하는 컬럼이 필요
- 조인에 참여하는 테이블 간 같은 값을 가진 컬럼 → 조인 컬럼
(예, employees와 departments 테이블 조인 컬럼 : department_id)
- 각 테이블의 조인 컬럼 명이 같을 필요는 없으나 동일하게 만드는 것이 좋음
- 조인 컬럼은 한 개 이상으로 구성될 수 있고, 뷰(View)도 조인 가능
- 조인 방식에 따라 크게 내부조인, 외부조인 으로 구분

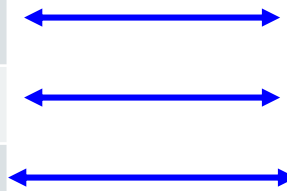
1. 조인 (Join)

사원 테이블

emp_id	emp_name	...	dept_id
301	홍길동		10
302	김유신		20
303	강감찬		30
304	이성계		

부서 테이블

dept_id	dept_name	...
10	Administration	
20	Marketing	
30	Purchasing	
40	Human Resources	



- 301번 홍길동이 속한 부서의 이름을 알려면 사원 테이블만 조회해서는 알 수 없음
- 사원 테이블의 dept_id와 부서 테이블의 dept_id를 연결해 부서 테이블의 부서명을 가져올 수 있음

2. 내부 조인 (Inner Join)

- 가장 기본적인 조인 방식
- 조인 참여 테이블 간 **조인 컬럼 값이 같은** 건을 가져옴
- WHERE 절에서 각 테이블의 조인 컬럼과 연산자를 사용해 조건 명시
→ 조인 조건
- 일반적으로 조인 조건에 **동등 연산자(=)** 사용 → 조인 컬럼 값이 같은 건이 조회됨
(예, where a.seq_id = b.seq_id)
- 조인 조건을 만족한 데이터만 조회됨

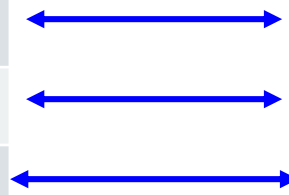
2. 내부 조인 (Inner Join)

사원 테이블(a)

emp_id	emp_name	...	dept_id
301	홍길동		10
302	김유신		20
303	강감찬		30
304	이성계		

부서 테이블(b)

dept_id	dept_name	...
10	Administration	
20	Marketing	
30	Purchasing	
40	Human Resources	



조인 조건

WHERE **a.dept_id = b.dept_id**

2. 내부 조인 (Inner Join)

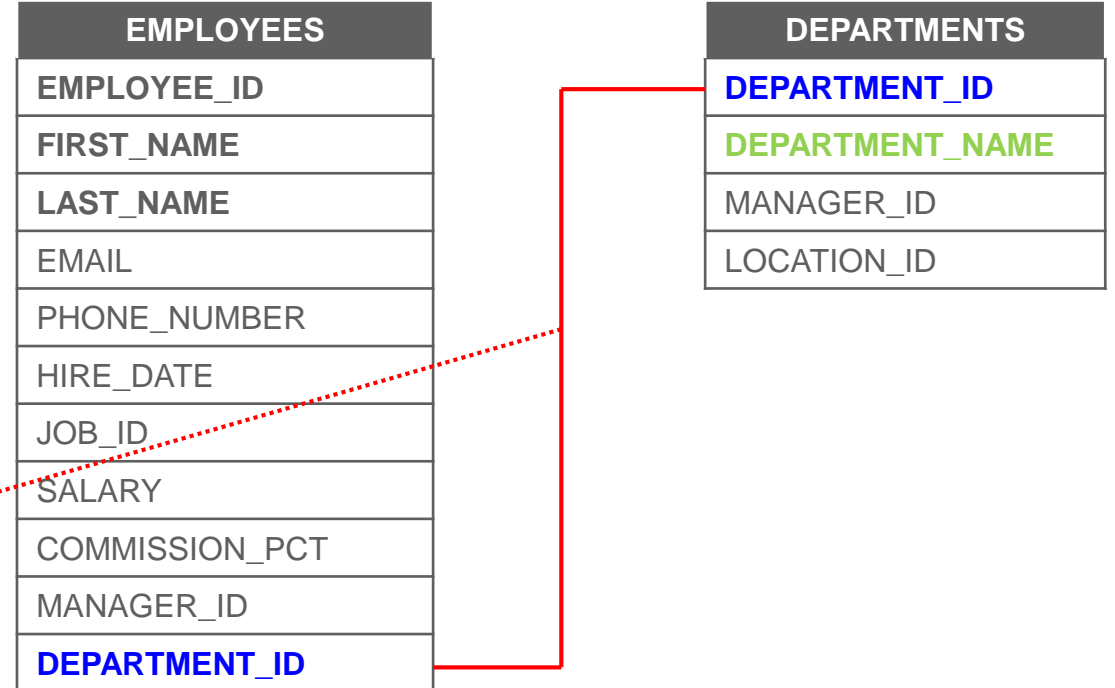
. 사용 예

```
SELECT a.employee_id,  
       a.first_name,  
       a.department_id,  
       b.department_name  
FROM employees a,  
     departments b  
WHERE a.department_id = b.department_id  
ORDER BY a.department_id;
```

- FROM 절에 조인에 참여할 테이블 명시 (콤마로 구분)
- 각 테이블에 Alias를 주는 것이 좋음
- 모든 컬럼은 테이블명.컬럼명 혹은 테이블 alias명.컬럼명 형태로 사용
- WHERE 절에서는 조인 조건과 일반 조건 함께 사용
- 조인 조건을 만족하는 데이터만 조회됨

3. 내부 조인 실습

```
SELECT a.employee_id,  
       a.first_name,  
       a.last_name,  
       a.department_id,  
       b.department_name  
FROM employees a, departments b  
WHERE a.department_id = b.department_id  
ORDER BY a.employee_id;
```



3. 내부 조인 실습

```
SELECT a.employee_id,  
       a.first_name,  
       a.last_name,  
       a.department_id,  
       b.department_name  
FROM employees a, departments b  
WHERE a.department_id = b.department_id  
ORDER BY a.employee_id;
```

※ 178번 사원이 빠져 있음

EMPLOYEE_ID	FIRST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
100	Steven	90	Executive
101	Neena	90	Executive
102	Lex	90	Executive
103	Alexander	60	IT
104	Bruce	60	IT
105	David	60	IT
106	Valli	60	IT
107	Diana	60	IT
108	Nancy	100	Finance

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
169	Harrison	Bloom	80	Sales
170	Tayler	Fox	80	Sales
171	William	Smith	80	Sales
172	Elizabeth	Bates	80	Sales
173	Sundita	Kumar	80	Sales
174	Ellen	Abel	80	Sales
175	Alyssa	Hutton	80	Sales
176	Jonathon	Taylor	80	Sales
177	Jack	Livingston	80	Sales
179	Charles	Johnson	80	Sales
180	Winston	Taylor	50	Shipping

3. 내부 조인 실습

```
SELECT a.employee_id,  
       a.first_name, a.last_name,  
       a.department_id  
FROM employees a  
WHERE a.department_id IS NULL  
ORDER BY a.employee_id;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
178	Kimberely	Grant	(null)

➔ 사번이 178번인 Kimberely Grant는 **department_id 값이 NULL** 이어서 조회가 되지 않음

➔ WHERE **a.department_id = b.department_id** 조건 불만족

3. 내부 조인 실습

```
SELECT a.employee_id, a.first_name || ' ' || a.last_name emp_names,
```

```
       job_id, b.job_title
```

```
FROM employees a,
```

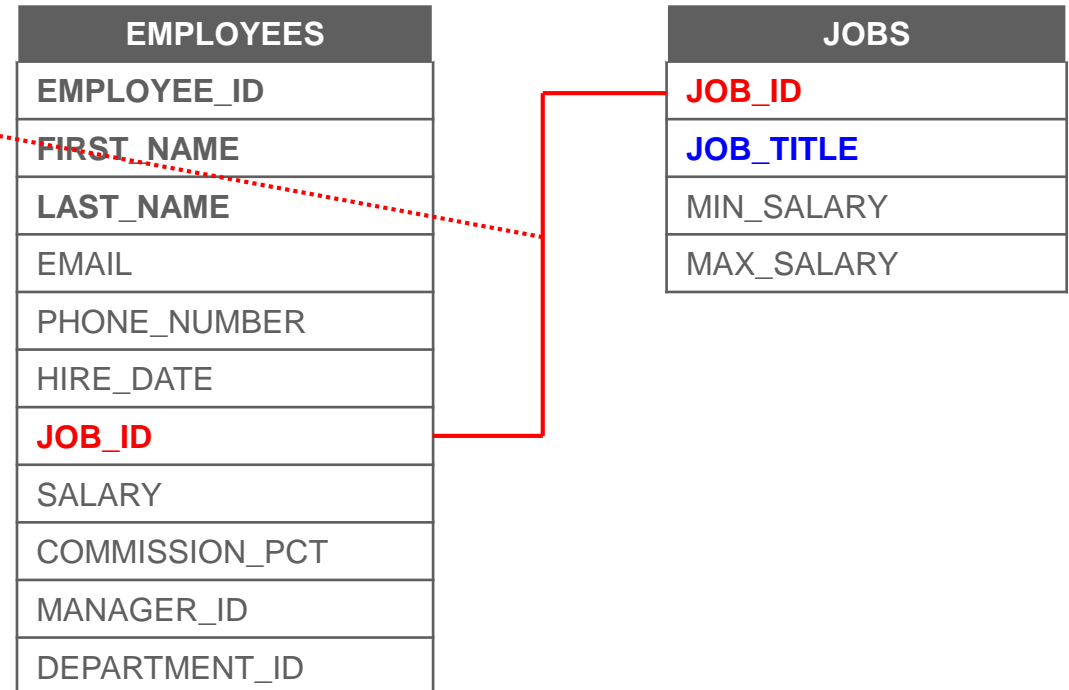
```
     jobs b
```

```
WHERE a.job_id = b.job_id
```

```
ORDER BY 1;
```

→ job_id 컬럼은 두 테이블 모두 존재,
어느 테이블의 job_id를 가져올 것인지 명시해야 함

```
ORA-00918: 열의 정의가 애매합니다  
00918, 00000 - "column ambiguously defined"  
*Cause:  
*Action:  
2행, 8열에서 오류 발생
```



3. 내부 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,  
     jobs b  
WHERE a.job_id = b.job_id  
ORDER BY 1;
```

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

EMPLOYEES
EMPLOYEE_ID
FIRST_NAME
LAST_NAME
EMAIL
PHONE_NUMBER
HIRE_DATE
JOB_ID
SALARY
COMMISSION_PCT
MANAGER_ID
DEPARTMENT_ID

JOB
JOB_ID
JOB_TITLE
MIN_SALARY
MAX_SALARY

3. 내부 조인 실습

```
SELECT a.employee_id, a.first_name || ' ' || a.last_name emp_names,
```

```
       a.job_id, b.job_id, job_title
```

```
FROM employees a,
```

```
     jobs b
```

```
WHERE a.job_id = b.job_id
```

```
ORDER BY 1;
```

EMPLOYEE_ID	EMP_NAMES	JOB_ID	JOB_ID_1	JOB_TITLE
100	Steven King	AD_PRES	AD_PRES	President
101	Neena Kochhar	AD_VP	AD_VP	Administration Vice President
102	Lex De Haan	AD_VP	AD_VP	Administration Vice President
103	Alexander Hunold	IT_PROG	IT_PROG	Programmer
104	Bruce Ernst	IT_PROG	IT_PROG	Programmer
105	David Austin	IT_PROG	IT_PROG	Programmer
106	Valli Pataballa	IT_PROG	IT_PROG	Programmer
107	Diana Lorentz	IT_PROG	IT_PROG	Programmer
108	Nancy Greenberg	FI_MGR	FI_MGR	Finance Manager

➔ job_title은 jobs 테이블에만 존재, 별칭이 없어도 오류 없으나 붙여 주는 것이 좋다

3. 내부 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,  
     jobs b,  
     departments c  
WHERE a.job_id = b.job_id  
      AND a.department_id = c.department_id  
ORDER BY 1;
```

JOB
JOB_ID
JOB_TITLE
MIN_SALARY
MAX_SALARY

EMPLOYEES
EMPLOYEE_ID
FIRST_NAME
LAST_NAME
EMAIL
PHONE_NUMBER
HIRE_DATE
JOB_ID
SALARY
COMMISSION_PCT
MANAGER_ID
DEPARTMENT_ID

DEPARTMENTS
DEPARTMENT_ID
DEPARTMENT_NAME
MANAGER_ID
LOCATION_ID

3. 내부 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,  
     jobs b,  
     departments c  
WHERE a.job_id      = b.job_id  
      AND a.department_id = c.department_id  
ORDER BY 1;
```

EMPLOY...	EMP_NAMES	JOB.TITLE	DEPARTMENT_ID	DEPARTMENT_NAME
100	Steven King	President	90	Executive
101	Neena Kochhar	Administration Vice President	90	Executive
102	Lex De Haan	Administration Vice President	90	Executive
103	Alexander Hunold	Programmer	60	IT
104	Bruce Ernst	Programmer	60	IT
105	David Austin	Programmer	60	IT
106	Valli Pataballa	Programmer	60	IT
107	Diana Lorentz	Programmer	60	IT
108	Nancy Greenberg	Finance Manager	100	Finance
...

3. 내부 조인 실습

```
SELECT a.employee_id,  
       a.first_name || ' ' || a.last_name emp_names,  
       b.job_title, c.department_name,  
       d.location_id, d.street_address, d.city, d.state_province  
FROM employees a,  
     jobs b,  
     departments c,  
     locations d  
WHERE a.job_id = b.job_id  
      AND a.department_id = c.department_id  
      AND c.location_id = d.location_id  
ORDER BY 1;
```

JOB_ID
JOB_ID
JOB_TITLE
MIN_SALARY
MAX_SALARY

EMPLOYEES
EMPLOYEE_ID
FIRST_NAME
LAST_NAME
EMAIL
PHONE_NUMBER
HIRE_DATE
JOB_ID
SALARY
COMMISSION_PCT
MANAGER_ID
DEPARTMENT_ID

DEPARTMENTS
DEPARTMENT_ID
DEPARTMENT_NAME
MANAGER_ID
LOCATION_ID

LOCATIONS
LOCATION_ID
STREET_ADDRESS
POSTAL_CODE
CITY
STATE_PROVINCE
COUNTRY_ID

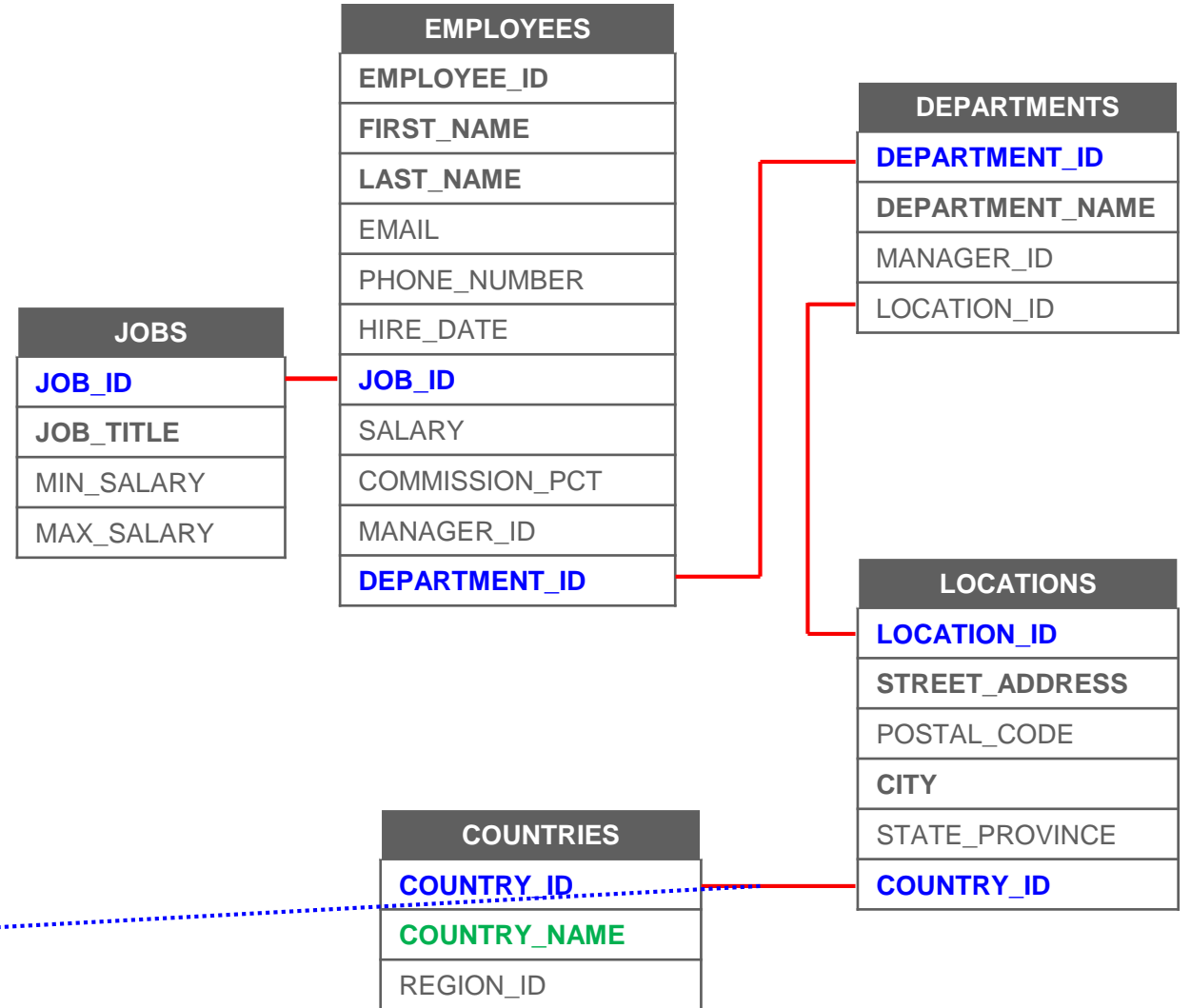
3. 내부 조인 실습

```
SELECT a.employee_id,  
       a.first_name || ' ' || a.last_name emp_names,  
       b.job_title, c.department_name,  
       d.location_id, d.street_address, d.city, d.state_province  
  
FROM employees a,  
     jobs b,  
     departments c,  
     locations d  
  
WHERE a.job_id      = b.job_id  
      AND a.department_id = c.department_id  
      AND c.location_id  = d.location_id  
  
ORDER BY 1;
```

EMPLOYEE_ID	EMP_NAMES	JOB_TITLE	DEPARTMENT_NAME	LOCATION_ID	STREET_ADDRESS	CITY	STATE_PROVINCE
100	Steven King	President	Executive	1700	2004 Charade Rd	Seattle	Washington
101	Neena Kochhar	Administration Vice President	Executive	1700	2004 Charade Rd	Seattle	Washington
102	Lex De Haan	Administration Vice President	Executive	1700	2004 Charade Rd	Seattle	Washington
103	Alexander Hunold	Programmer	IT	1400	2014 Jabberwocky Rd	Southlake	Texas
104	Bruce Ernst	Programmer	IT	1400	2014 Jabberwocky Rd	Southlake	Texas
105	David Austin	Programmer	IT	1400	2014 Jabberwocky Rd	Southlake	Texas
106	Valli Pataballa	Programmer	IT	1400	2014 Jabberwocky Rd	Southlake	Texas
107	Diana Lorentz	Programmer	IT	1400	2014 Jabberwocky Rd	Southlake	Texas
108	Nancy Greenberg	Finance Manager	Finance	1700	2004 Charade Rd	Seattle	Washington
109	Daniel Faviet	Accountant	Finance	1700	2004 Charade Rd	Seattle	Washington
110	John Chen	Accountant	Finance	1700	2004 Charade Rd	Seattle	Washington
111	Ismael Sciarra	Accountant	Finance	1700	2004 Charade Rd	Seattle	Washington

3. 내부 조인 실습

```
SELECT a.employee_id
      ,a.first_name || ' ' || a.last_name emp_names
      ,b.job_title ,c.department_name
      ,d.street_address, d.city
      ,e.country_name
FROM employees a,
      jobs b,
      departments c,
      locations d,
      countries e
WHERE a.job_id      = b.job_id
      AND a.department_id = c.department_id
      AND c.location_id  = d.location_id
      AND d.country_id   = e.country_id
ORDER BY 1;
```



3. 내부 조인 실습

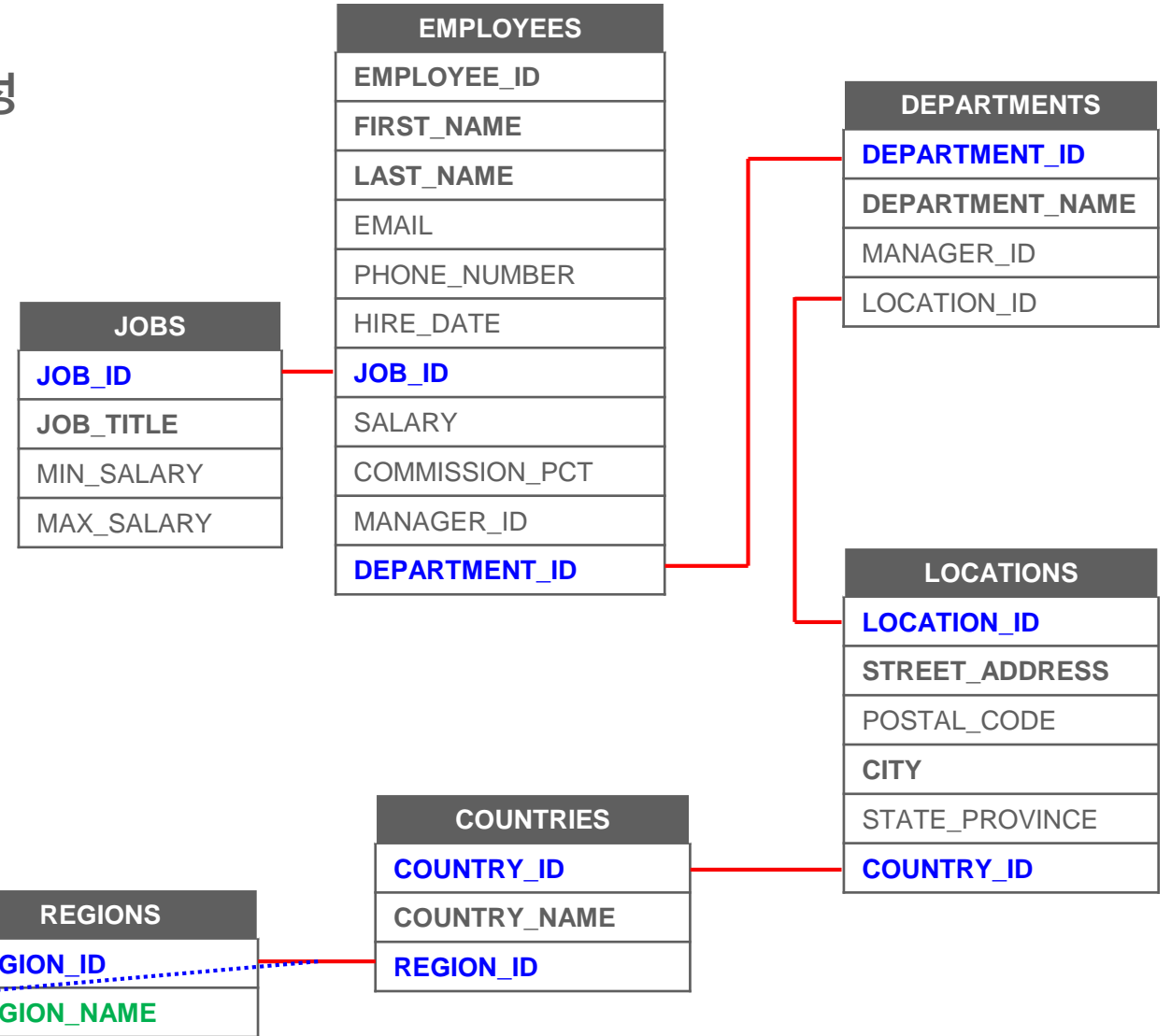
```
SELECT a.employee_id
       ,a.first_name || ' ' || a.last_name emp_names
       ,b.job_title ,c.department_name
       ,d.street_address, d.city
       ,e.country_name
FROM employees a,
     jobs b,
     departments c,
     locations d,
     countries e
WHERE a.job_id      = b.job_id
     AND a.department_id = c.department_id
     AND c.location_id  = d.location_id
     AND d.country_id   = e.country_id
ORDER BY 1;
```

EMPLOYEE_ID	EMP_NAMES	JOB_TITLE	DEPARTMENT_NAME	STREET_ADDRESS	CITY	COUNTRY_NAME
100	Steven King	President	Executive	2004 Charade Rd	Seattle	United States of America
101	Neena Kochhar	Administration Vice President	Executive	2004 Charade Rd	Seattle	United States of America
102	Lex De Haan	Administration Vice President	Executive	2004 Charade Rd	Seattle	United States of America
103	Alexander Hunold	Programmer	IT	2014 Jabberwocky Rd	Southlake	United States of America
104	Bruce Ernst	Programmer	IT	2014 Jabberwocky Rd	Southlake	United States of America
105	David Austin	Programmer	IT	2014 Jabberwocky Rd	Southlake	United States of America
106	Valli Pataballa	Programmer	IT	2014 Jabberwocky Rd	Southlake	United States of America
107	Diana Lorentz	Programmer	IT	2014 Jabberwocky Rd	Southlake	United States of America
108	Nancy Greenberg	Finance Manager	Finance	2004 Charade Rd	Seattle	United States of America
109	Daniel Faviet	Accountant	Finance	2004 Charade Rd	Seattle	United States of America
110	John Chen	Accountant	Finance	2004 Charade Rd	Seattle	United States of America
111	Ismael Sciarra	Accountant	Finance	2004 Charade Rd	Seattle	United States of America

3. 내부 조인 실습

regions 테이블과 조인해 대륙명을 가져오는 쿼리 작성

```
SELECT a.employee_id
      ,a.first_name || ' ' || a.last_name emp_names
      ,b.job_title ,c.department_name
      ,d.street_address, d.city
      ,e.country_name ,f.region_name
FROM employees a,
     jobs b,
     departments c,
     locations d,
     countries e,
     regions f
WHERE a.job_id      = b.job_id
     AND a.department_id = c.department_id
     AND c.location_id  = d.location_id
     AND d.country_id   = e.country_id
     AND e.region_id    = f.region_id
ORDER BY 1;
```



3. 내부 조인 실습

방금 전 쿼리를 기준으로 regions 테이블과 조인해 대륙명을 가져오는 쿼리 작성해 보세요

```
SELECT a.employee_id
```

```
,a.first_name || ' ' || a.last_name emp_names
```

```
,b.job_title ,c.department_name
```

```
,d.street_address, d.city
```

```
,e.country_name ,f.region_name
```

```
FROM employees a,
```

```
jobs b,
```

```
departments c,
```

```
locations d,
```

```
countries e,
```

```
regions f
```

```
WHERE a.job_id      = b.job_id
```

```
AND a.department_id = c.department_id
```

```
AND c.location_id   = d.location_id
```

```
AND d.country_id    = e.country_id
```

```
AND e.region_id     = f.region_id
```

```
ORDER BY 1;
```

EMPLOYEE_ID	EMP_NAMES	JOB_TITLE	DEPARTMENT_NAME	STREET_ADDRESS	CITY	COUNTRY_NAME	REGION_NAME
100	Steven King	President	Executive	2004 Charade Rd	Seattle	United States of America	Americas
101	Neena Kochhar	Administration Vic...	Executive	2004 Charade Rd	Seattle	United States of America	Americas
102	Lex De Haan	Administration Vic...	Executive	2004 Charade Rd	Seattle	United States of America	Americas
103	Alexander Hunold	Programmer	IT	2014 Jabberwocky Rd	Southlake	United States of America	Americas
104	Bruce Ernst	Programmer	IT	2014 Jabberwocky Rd	Southlake	United States of America	Americas
105	David Austin	Programmer	IT	2014 Jabberwocky Rd	Southlake	United States of America	Americas
106	Valli Pataballa	Programmer	IT	2014 Jabberwocky Rd	Southlake	United States of America	Americas
107	Diana Lorentz	Programmer	IT	2014 Jabberwocky Rd	Southlake	United States of America	Americas
108	Nancy Greenberg	Finance Manager	Finance	2004 Charade Rd	Seattle	United States of America	Americas
109	Daniel Faviet	Accountant	Finance	2004 Charade Rd	Seattle	United States of America	Americas
110	John Chen	Accountant	Finance	2004 Charade Rd	Seattle	United States of America	Americas
111	Ismael Sciarra	Accountant	Finance	2004 Charade Rd	Seattle	United States of America	Americas
112	Jose Manuel Urman	Accountant	Finance	2004 Charade Rd	Seattle	United States of America	Americas
113	Luis Popp	Accountant	Finance	2004 Charade Rd	Seattle	United States of America	Americas
114	Den Raphaely	Purchasing Manager	Purchasing	2004 Charade Rd	Seattle	United States of America	Americas

3. 내부 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,  
     jobs b,  
     departments c  
WHERE a.job_id      = b.job_id  
      AND a.department_id = c.department_id  
      AND c.department_id = 30  
ORDER BY 1;
```

EMPLOYEE_ID	EMP_NAMES	JOB_TITLE	DEPARTMENT_ID	DEPARTMENT_NAME
114	Den Raphaely	Purchasing Manager	30	Purchasing
115	Alexander Khoo	Purchasing Clerk	30	Purchasing
116	Shelli Baida	Purchasing Clerk	30	Purchasing
117	Sigal Tobias	Purchasing Clerk	30	Purchasing
118	Guy Himuro	Purchasing Clerk	30	Purchasing
119	Karen Colmenares	Purchasing Clerk	30	Purchasing

4. 외부 조인 (Outer Join)

- 조인 조건을 만족하는 것은 물론 만족하지 않는 데이터(로우) 까지 포함해 조회
- A, B 두 테이블 기준, 조인조건에 부합하지 않는 상대방 테이블 데이터도 조회됨
→ 조인 조건에 (+)를 붙여야 함
- 조인조건을 만족하지 않는 a 테이블의 데이터까지 조회 시
→ WHERE a.department_id = b.department_id (+)
- 조인조건을 만족하지 않는 b 테이블의 데이터까지 조회 시
→ WHERE a.department_id(+) = b.department_id

4. 외부 조인 (Outer Join)

- 외부조인 시 조인조건에 (+)를 붙이는 것은 오라클 전용 문법임
- 다른 DBMS에서는 (+) 기호 붙이면 오류 발생

4. 외부 조인 (Outer Join)

사원 테이블(a)

emp_id	last_name	...	dept_id
301	홍길동		10
302	김유신		20
303	강감찬		30
304	이성계		

부서 테이블(b)

dept_id	dept_name	...
10	Administration	
20	Marketing	
30	Purchasing	
40	Human Resources	

조인 조건

WHERE a.dept_id = b.dept_id (+)

4. 외부 조인 (Outer Join)

사원 테이블(a)

emp_id	emp_name	...	dept_id
301	홍길동		10
302	김유신		20
303	강감찬		30
304	이성계		

부서 테이블(b)

dept_id	dept_name	...
10	Administration	
20	Marketing	
30	Purchasing	
40	Human Resources	

조인 조건

WHERE a.dept_id (+) = b.dept_id

4. 외부 조인 (Outer Join)

. 사용 예

```
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;
```



emp_id	a_dept_id	b_dept_id	dept_name
200	10	10	Administration
201	20	20	Marketing
202	20	20	Marketing
...
178			

Employees 테이블에서 사번이 178번인 사원의 부서번호는 Null

4. 외부 조인 (Outer Join)

. 사용 예

```
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;
```



emp_id	a_dept_id	b_dept_id	dept_name
200	10	10	Administration
201	20	20	Marketing
202	20	20	Marketing
...
		120	Treasury
		130	Corporate Tax
...

부서번호가 120번 이상인 부서는 employees 테이블의 department_id에 할당된 건이 없음

4. 외부 조인 (Outer Join)

- (+) 기호를 사용하는 오라클 외부 조인 제약사항

- 조인 컬럼이 여러 개일 경우, 조인조건에서 (+) 기호를 모두 붙여야 제대로 조회됨

예)

where a.col1 = b.col1(+)

and a.col2 = b.col2(+)

...

- 조인 조건 양쪽에 (+) 기호 붙일 수 없음

5. 외부조인 실습

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

	EMPLOYEE_ID	EMP_NAMES	DEPARTMENT_ID	DEPARTMENT_NAME
1	100	Steven King	90	Executive
2	101	Neena Kochhar	90	Executive
3	102	Lex De Haan	90	Executive
4	103	Alexander Hunold	60	IT
5	104	Bruce Ernst	60	IT
6	105	David Austin	60	IT
7	106	Valli Pataballa	60	IT
8	107	Diana Lorentz	60	IT
9	108	Nancy Greenberg	100	Finance
10	109	Daniel Faviet	100	Finance
11	110	John Chen	100	Finance
12	111	Ismael Sciarra	100	Finance
13	112	Jose Manuel Urman	100	Finance
14	113	Luis Popp	100	Finance

	EMPLOYEE_ID	EMP_NAMES	DEPARTMENT_ID	DEPARTMENT_NAME
73	172	Elizabeth Bates	80	Sales
74	173	Sundita Kumar	80	Sales
75	174	Ellen Abel	80	Sales
76	175	Alyssa Hutton	80	Sales
77	176	Jonathon Taylor	80	Sales
78	177	Jack Livingston	80	Sales
79	178	Kimberely Grant	(null)	(null)
80	179	Charles Johnson	80	Sales

5. 외부조인 실습

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

	EMPLOYEE_ID	EMP_NAMES	DEPARTMENT_ID	DEPARTMENT_NAME
1	100	Steven King	90	Executive
2	101	Neena Kochhar	90	Executive
3	102	Lex De Haan	90	Executive
4	103	Alexander Hunold	60	IT
5	104	Bruce Ernst	60	IT
6	105	David Austin	60	IT
7	106	Valli Pataballa	60	IT
8	107	Diana Lorentz	60	IT
9	108	Nancy Greenberg	100	Finance
0	109	Daniel Faviet	100	Finance
1	110	John Chen	100	Finance
2	111	Ismael Sciarra	100	Finance
3	112	Jose Manuel Urman	100	Finance
4	113	Luis Popp	100	Finance
78	177	Jack Livingston	80	Sales
79	179	Charles Johnson	80	Sales
106	206	William Gietz	110	Accounting
107	(null)		180	Construction
108	(null)		190	Contracting
109	(null)		200	Operations
110	(null)		210	IT Support
111	(null)		220	NOC
112	(null)		230	IT Helpdesk
113	(null)		240	Government Sales
114	(null)		250	Retail Sales
115	(null)		260	Recruiting
116	(null)		270	Payroll (Merge)
117	(null)		300	경리부 (Merge)
118	(null)		160	Benefits
119	(null)		150	Shareholder Services
120	(null)		140	Control And Credit
121	(null)		130	Corporate Tax
122	(null)		120	Treasury
123	(null)		170	Manufacturing

5. 외부조인 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,  
     departments c,  
     locations d  
WHERE a.department_id = c.department_id(+)  
      AND c.location_id = d.location_id  
ORDER BY 1;
```

※ 외부조인을 했으므로 178번인 Kimberly Grant가
조회되어야 하지만,
외부조인한 결과 178번의 department_id는 null.
결국 locations 테이블과 내부조인을 했기 때문에
조회되지 않음

EMPLOYEE_ID	EMP_NAMES	DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	STREET_ADDRESS
175	Alyssa Hutton	80	Sales	2500	Magdalen Centre, The Oxford Sci
176	Jonathon Taylor	80	Sales	2500	Magdalen Centre, The Oxford Sci
177	Jack Livingston	80	Sales	2500	Magdalen Centre, The Oxford Sci
179	Charles Johnson	80	Sales	2500	Magdalen Centre, The Oxford Sci
180	Winston Taylor	50	Shipping	1500	2011 Interiors Blvd
181	Jean Fleaur	50	Shipping	1500	2011 Interiors Blvd

5. 외부조인 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,  
     departments c,  
     locations d  
WHERE a.department_id = c.department_id(+)  
      AND c.location_id = d.location_id(+)  
ORDER BY 1;
```

※ 178번의 department_id는 null, departments 테이블도 null 이지만 locations와 외부조인을 했기 때문에 Kimberly Grant가 조회됨

	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

※ 외부 조인은 왜 사용할까?

- 테이블 설계가 제대로 되어 있고, 데이터가 정확히 입력되어 있다면 굳이 외부 조인을 사용할 필요가 없음
 - 하지만 현실은 그렇지 않음
 - 테이블 설계를 완벽히 할 수 없음
 - 애초에 제대로 설계했더라도 업무가 변경되면 로직 수정이 필요
 - 설계가 제대로 되어 있더라도, 데이터 입력 시 오류로 인해 잘못된 데이터 입력, 누락 데이터 발생
- 예) 178번 Kimberly Grant는 부서번호가 없음

부서가 없는 사원이 존재할까?

설사 부서 발령이 안되더라도 미발령부서 정보를 부서테이블에 등록하는 것이 정상적

학습정리

- 조인은 여러 테이블들을 조인조건으로 연결해 데이터를 조회하는 기법이다.
- 조인은 크게 내부조인과 외부조인으로 나뉜다.
- 내부조인은 두 테이블간 조인 컬럼을 사용한 조인조건에 맞는 데이터가 조회된다.
- 외부조인은 내부조인과 동작방식이 같은데, 이에 더해 조인 조건을 만족하지 않는 데이터까지도 조회할 수 있다.