

데이터베이스와 SQL

5장

다중 테이블 쿼리

빅데이터 분석가 과정

목차

- 5.1 조인(join): 내부 조인
- 5.2 세 개 이상 테이블 조인
- 5.3 셀프 조인

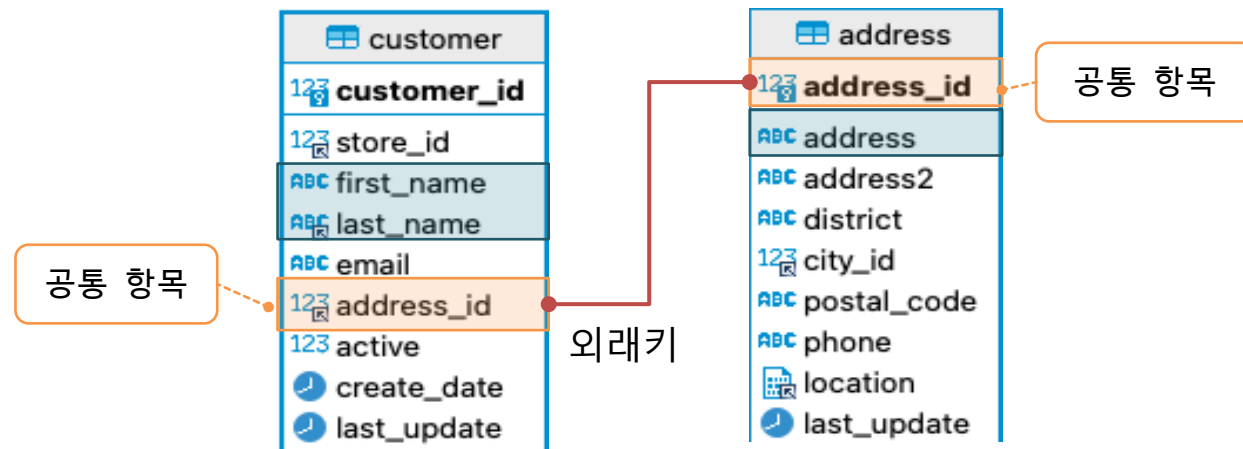
5.1 Join

■ 다중 테이블 쿼리

- 대부분의 쿼리는 여러 테이블을 필요로 함
- 외래 키(foreign key): 여러 테이블을 연결할 **연결 수단**
- **조인(join)**
 - 두 개 이상의 테이블을 서로 묶어서(연결) 하나의 결과 집합으로 만들어 내는 것

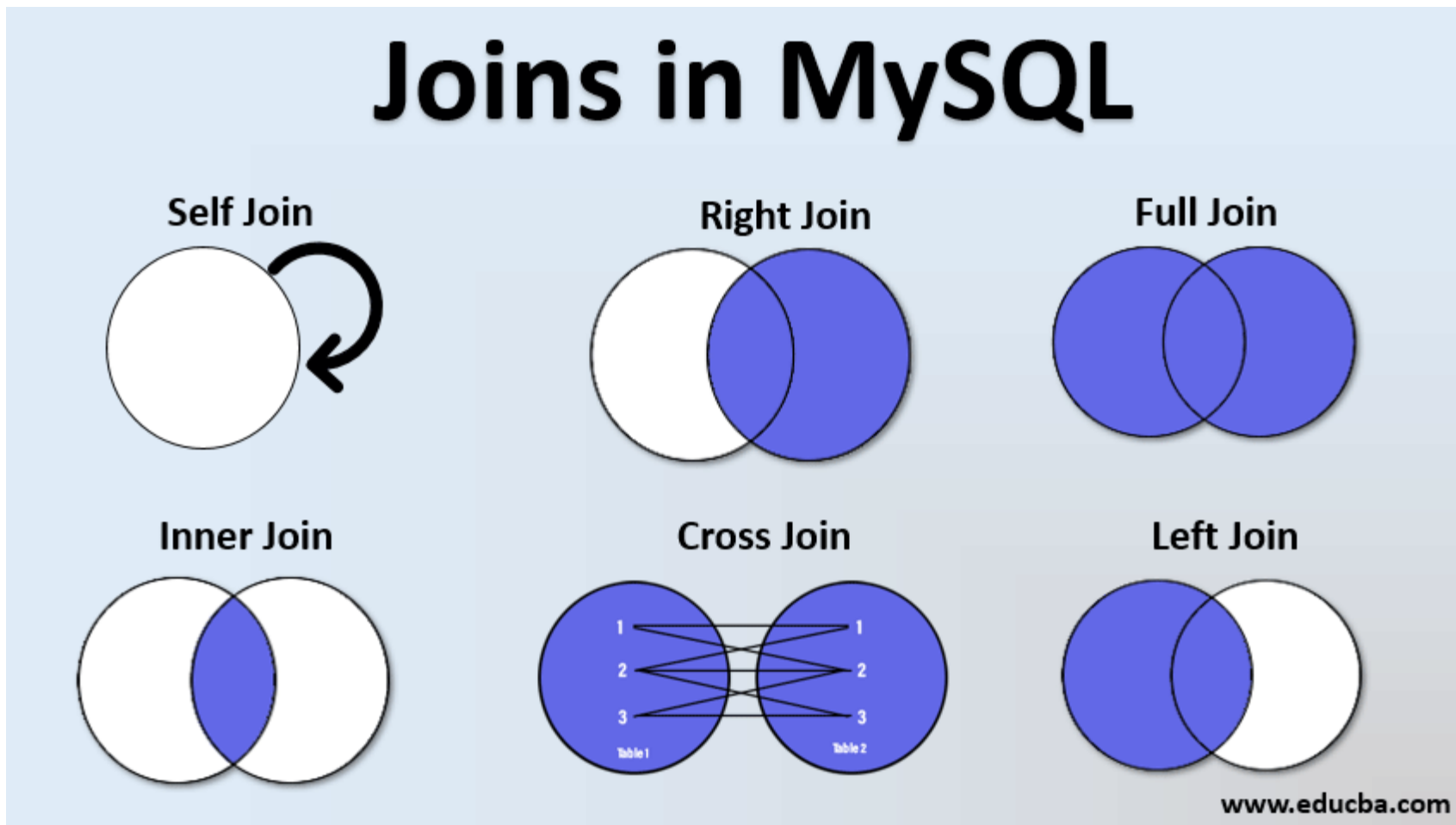
■ 고객의 성과 이름, 주소를 검색할 경우

- customer 테이블의 first_name, last_name 열을 검색
- address 테이블의 address 열을 검색
- 연결 수단: **address_id**는 두 테이블에 공통으로 포함되어 있음



5.1 Join: Join의 종류별 그림

■ Join의 종류



5.1 Join: 교차 조인

■ 데카르트 곱 (Cartesian Product): 교차 조인

- 두 개의 테이블에서 한 테이블의 모든 행들과 다른 테이블의 모든 행을 결합
- 교차 조인(cross join): join의 조건이 없이 모든 행을 결합

```
SELECT (조회 컬럼)  
FROM 테이블명1 CROSS JOIN 테이블명2;
```

- 상품 테이블 (3개의 행) x 재고 테이블 (3개의 행) = 총 9 개의 행

상품 테이블			재고 테이블		
product_no	name	price	product_no	count	Receiving date
1	A	1,000	1	10	2018-01-04
2	B	5,000	2	50	2018-01-07
3	C	3,000	3	20	2018-01-10

↓

product_no	name	price	product_no	count	Receiving data
1	A	1,000	1	10	2018-01-04
2	B	5,000	1	10	2018-01-04
3	C	3,000	1	10	2018-01-04
1	A	1,000	2	50	2018-01-07
2	B	5,000	2	50	2018-01-07
3	C	3,000	2	50	2018-01-07
1	A	1,000	3	20	2018-01-10
2	B	5,000	3	20	2018-01-10
3	C	3,000	3	20	2018-01-10

5.1 Join: 교차 조인

■ 교차 조인

- customer 및 address 테이블을 교차 조인
 - 결합(join)의 조건 없이 모든 행을 결합
 - 조인의 조건(on)이 없음

```
select c.first_name, c.last_name, a.address
from customer as c cross join address as a;
```

on 조건 없음

first_name	last_name	address
AUSTIN	CINTRON	47 MySakila Drive
WADE	DELVALLE	47 MySakila Drive
FREDDIE	DUGGAN	47 MySakila Drive
ENRIQUE	FORSYTHE	47 MySakila Drive
...		

```
select count(*) from customer;
```

count(*)

599

X

```
select count(*) from address;
```

count(*)

603

(599 x 603 = 361,197)

교차 조인

```
select count(*) from customer as c join address as a;
```

on 조건 없음
- 교차 조인으로 처리

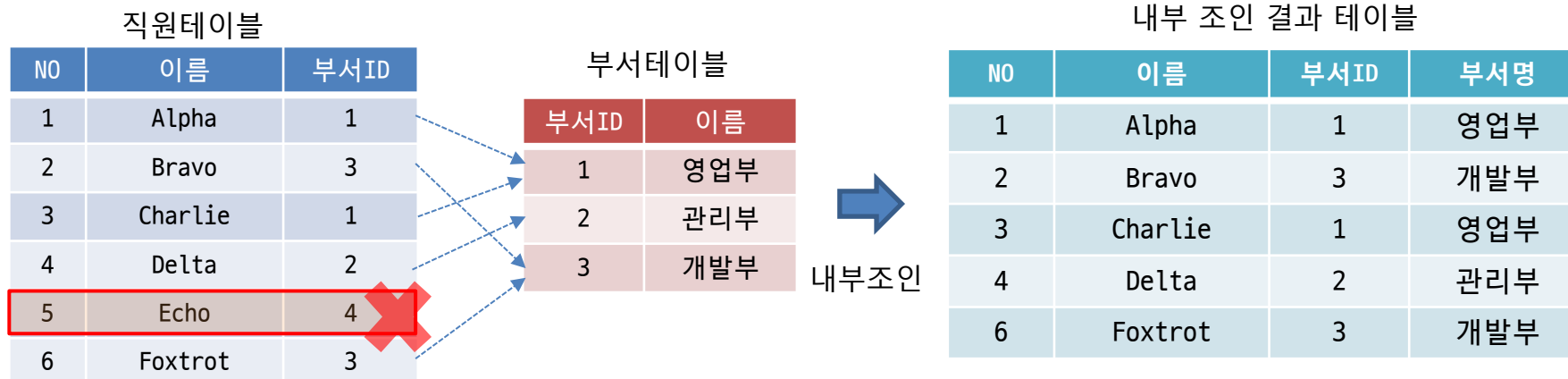
count(*)

361197

5.1 Join: 내부 조인

- 내부 조인 (inner join): 가장 일반적인 join 유형
 - 두 개 이상의 테이블을 묶어서 하나의 결과 집합을 만들어 내는 것
 - 일반적으로 join은 inner join을 지칭
 - 일치하지 않는 데이터는 검색하지 않음

```
SELECT <열 목록>  
FROM <기준 테이블> [INNER] JOIN <참조할 테이블>  
    ON <조인 조건>  
[WHERE 검색 조건]
```



부서ID값 4는 부서테이블에 존재
하지 않기 때문에 조인하지 않음

5.1 Join: 내부 조인

■ 내부 조인 예제

- customer 테이블의 `address_id`와 address 테이블의 `address_id`가 같은 경우에만 join

```
select c.first_name, c.last_name, a.address, a.district
from customer as c inner join address as a
on c.address_id = a.address_id;
```

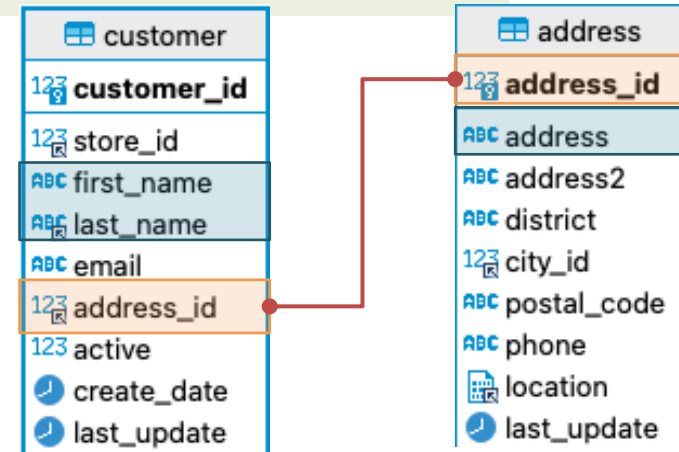
join 조건

inner join에서 inner
생략 가능

first_name	last_name	address	district
MARY	SMITH	1913 Hanoi Way	Nagasaki
PATRICIA	JOHNSON	1121 Loja Avenue	California
LINDA	WILLIAMS	692 Joliet Street	Attika
BARBARA	JONES	1566 Inegl Manor	Mandalay
ELIZABETH	BROWN	53 Idfu Parkway	Nantou
JENNIFER	DAVIS	1795 Santiago de Compostela Way	Texas
MARIA	MILLER	900 Santiago de Compostela Parkway	Central Serbia
. . .			

- 총 599개의 행 출력

```
select count(*)
from customer as c inner join address as a
on c.address_id = a.address_id;
```



5.1 Join: 외부 조인

■ 외부 조인(Outer join): 10장

■ 외부 조인

- 한쪽 테이블에만 존재하는 데이터들을 다른 테이블에 결합하는 방식

```
SELECT <열 목록>  
FROM <첫 번째 테이블 (LEFT)>  
    <LEFT | RIGHT | FULL> OUTER JOIN <두 번째 테이블 (RIGHT)>  
ON <조인 조건>  
[WHERE 검색조건];
```

NO	이름	부서ID
1	Alpha	1
2	Bravo	3
3	Charlie	1
4	Delta	2
5	Echo	4
6	Foxtrot	3

부서ID	이름
1	영업부
2	관리부
3	개발부

외부조인



NO	이름	부서ID	부서명
1	Alpha	1	영업부
2	Bravo	3	개발부
3	Charlie	1	영업부
4	Delta	2	관리부
5	Echo	4	
6	Foxtrot	3	개발부

5.1.3 ANSI join 문법

- 교재의 테이블 조인 표기법
 - ANSI SQL 표준의 SQL92 버전 사용
- 이전 문법 표기
 - 조인 조건 및 필터 조건: **WHERE**절에 표기

```
select c.first_name, c.last_name, a.address
from customer as c join address as a
where c.address_id = a.address_id and a.postal_code = 52137;
```

- 조인 조건과 필터 조건을 구분하기 어려움

- SQL92 문법 표기
 - Join 조건: **ON** 절
 - 필터 조건(검색 조건): **WHERE** 절

```
select c.first_name, c.last_name, a.address, a.postal_code
from customer as c join address as a
    on c.address_id = a.address_id
where a.postal_code = 52137;
```

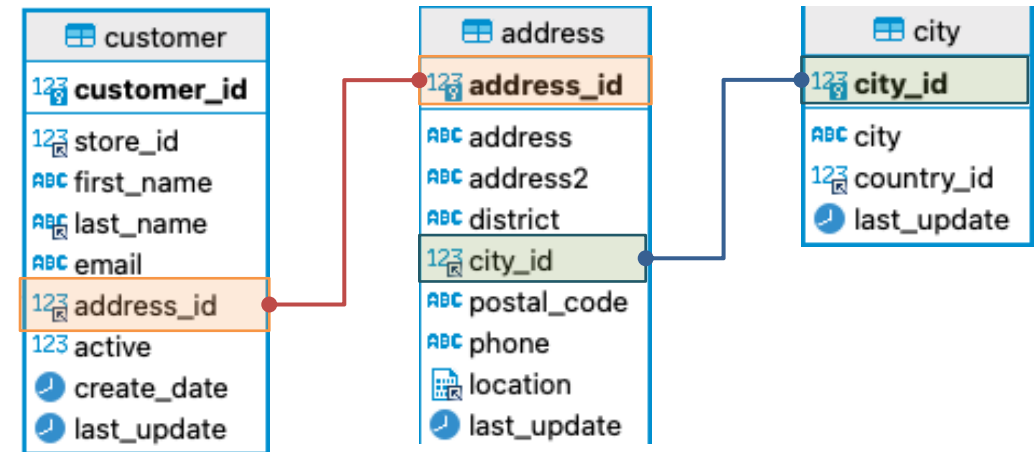
on절: join 조건

first_name	last_name	address	postal_code
JAMES	GANNON	1635 Kuwana Boulevard	52137
FREDDIE	DUGGAN	1103 Quilmes Boulevard	52137

5.2 세 개 이상 테이블 조인

■ 세 개 이상 테이블 조인

- customer 테이블, address 테이블, city 테이블 사용
 - 고객이 사는 도시를 반환하는 쿼리 작성



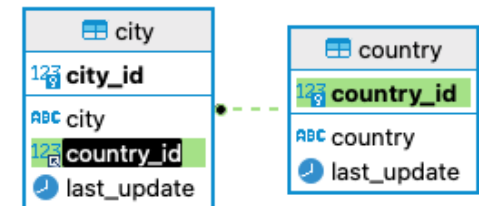
■ city 테이블 구조 확인

- Key 컬럼의 **MUL**: 다른 테이블의 기본 키를 참조하는 외래 키인 경우
- MUL(Multiple Occurrences Column): 여러 행이 동일한 값을 가질 수 있음을 의미

```
desc city;
```

Field	Type	Null	Key	Default	Extra
city_id	smallint unsigned	NO	PRI		auto_increment
city	varchar(50)	NO			
country_id	smallint unsigned	NO	MUL		
last_update	timestamp	NO		CURRENT_TIMESTAMP	DEFAULT_GENERATED on update CURRENT_TIMESTAMP

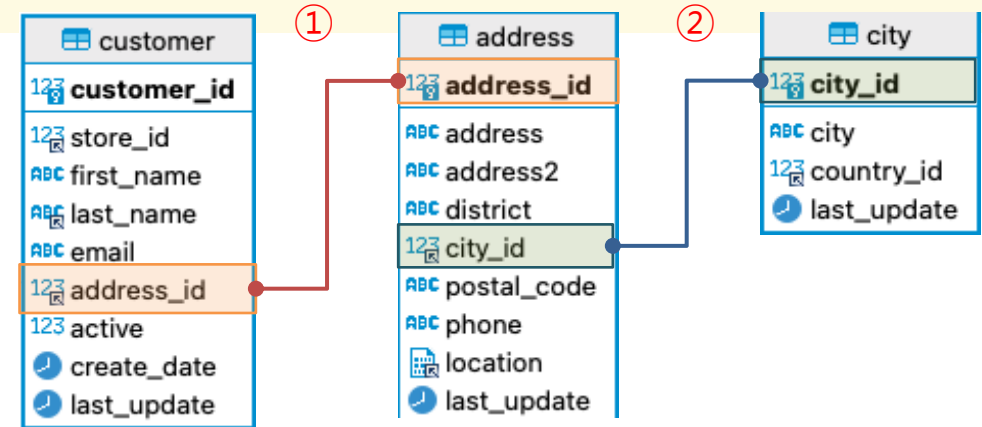
중복 가능



5.2 세 개 이상 테이블 조인

■ 세 개의 테이블 조인

- customer 테이블과 address 테이블 inner join
 - address_id를 사용하여 연결
- address 테이블과 city 테이블 inner join
 - city_id를 사용하여 연결



- Join 과정에서 from절의 테이블 순서는 중요하지 않음: 데이터베이스 내부에서 결정

```
select c.first_name, c.last_name, ct.city, a.address, a.district, a.postal_code
from customer as c
      inner join address as a on c.address_id = a.address_id
      inner join city as ct on a.city_id = ct.city_id;
```

customer과 address 조인

address와 city 조인

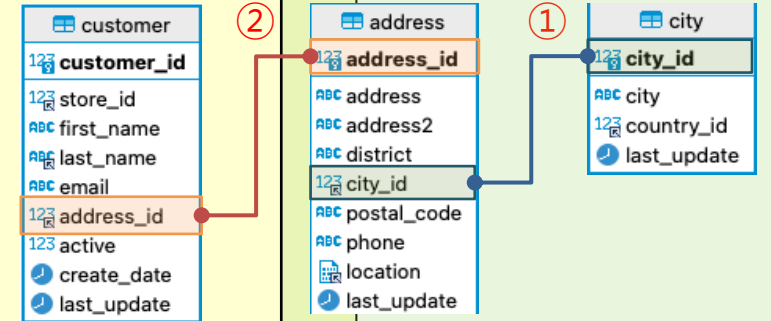
first_name	last_name	city	address	district	postal_code
JULIE	SANCHEZ	A Coruña (La Coruña)	939 Probolinggo Loop	Galicia	4166
PEGGY	MYERS	Abha	733 Mandaluyong Place	Asir	77459
TOM	MILNER	Abu Dhabi	535 Ahmadnagar Manor	Abu Dhabi	41136
GLEN	TALBERT	Acuña	1789 Saint-Denis Parkway	Coahuila de Zaragoza	8268
LARRY	THRASHER	Adana	663 Baha Blanca Parkway	Adana	33463
. . .					

5.2 세 개 이상 테이블 조인

■ 서브 쿼리 사용

```
select c.first_name, c.last_name, addr.address, addr.city, addr.district
from customer as c
  inner join
    (select a.address_id, a.address, ct.city, a.district
     from address as a
       inner join city as ct
         on a.city_id = ct.city_id
      where a.district = 'California'
     ) as addr
on c.address_id = addr.address_id;
```

서브 쿼리 별칭 addr



- 서브 쿼리(addr) 결과를 customer 테이블과 inner join
- 쿼리 결과
 - California에 거주하는 모든 고객들의 이름, 성, 주소, 도시, 지역을 검색

first_name	last_name	address	city	district
PATRICIA	JOHNSON	1121 Loja Avenue	San Bernardino	California
BETTY	WHITE	770 Bydgoszcz Avenue	Citrus Heights	California
ALICE	STEWART	1135 Izumisano Parkway	Fontana	California
ROSA	REYNOLDS	793 Cam Ranh Avenue	Lancaster	California
RENEE	LANE	533 al-Ayn Boulevard	Compton	California
KRISTIN	JOHNSTON	226 Brest Manor	Sunnyvale	California
CASSANDRA	WALTERS	920 Kumbakonam Loop	Salinas	California
JACOB	LANCE	1866 al-Qatif Avenue	El Monte	California
RENE	MCALISTER	1895 Zhezqazghan Drive	Garden Grove	California

5.2 세 개 이상 테이블 조인

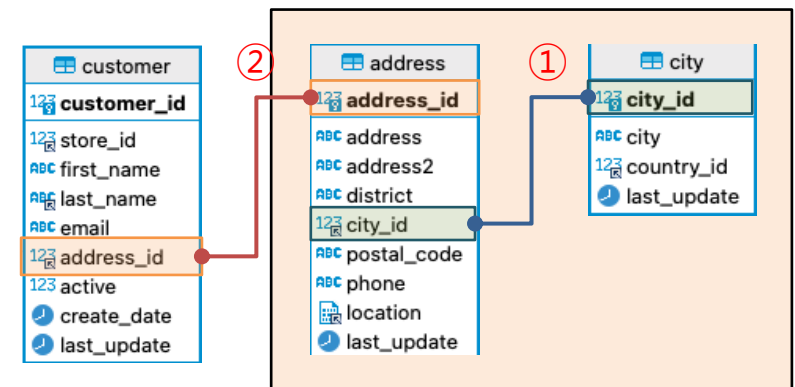
■ 서브 쿼리 사용

■ 서브 쿼리만 단독으로 실행

- address 테이블과 city 테이블을 내부 조인
- 필터링 조건: district의 이름이 'California'

```
select a.address_id, a.address, ct.city, a.district
  from address as a
    inner join city ct
    on a.city_id = ct.city_id
 where a.district = 'California';
```

address_id	address	city	district
6	1121 Loja Avenue	San Bernardino	California
18	770 Bydgoszcz Avenue	Citrus Heights	California
55	1135 Izumisano Parkway	Fontana	California
116	793 Cam Ranh Avenue	Lancaster	California
186	533 al-Ayn Boulevard	Compton	California
218	226 Brest Manor	Sunnyvale	California
274	920 Kumbakonam Loop	Salinas	California
425	1866 al-Qatif Avenue	El Monte	California
599	1895 Zhezqazghan Drive	Garden Grove	California



5.2 세 개 이상 테이블 조인

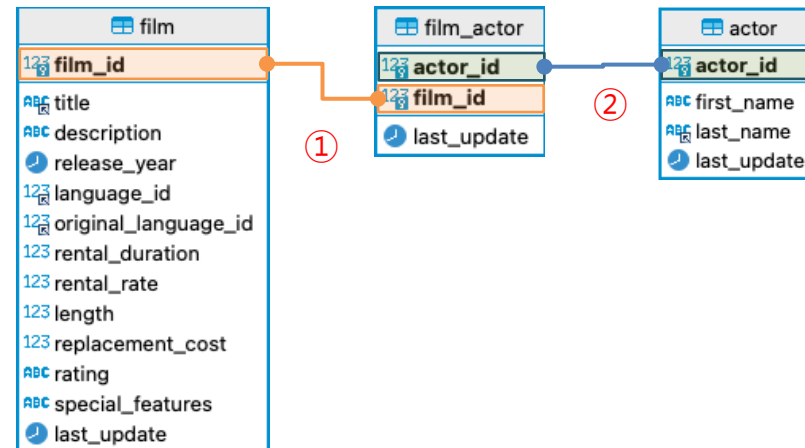
■ 테이블 재사용

- 여러 테이블을 join할 경우, 같은 테이블을 두 번 이상 join 할 수 있음
- 두 명의 특정 배우가 출연한 영화 제목 검색
 - 조인 테이블: `film`, `film_actor`, `actor` 테이블

```
select f.title
from film as f
  inner join film_actor as fa
    on f.film_id = fa.film_id
  inner join actor a
    on fa.actor_id = a.actor_id
where ((a.first_name = 'CATE' and a.last_name = 'MCQUEEN')
       or (a.first_name = 'CUBA' and a.last_name = 'BIRCH'));
```

Cate McQueen과 Cuba Birch가
출연한 모든 영화 제목 검색

title
ATLANTIS CAUSE
BLOOD ARGONAUTS
COMMANDMENTS EXPRESS
DYNAMITE TARZAN
EDGE KISSING
FINDING ANACONDA
GREATEST NORTH
JUNGLE CLOSER
...



5.2 세 개 이상 테이블 조인

- 두 배우가 같이 출연한 영화만 검색 #1
 - film 테이블에서 film_actor 테이블에 두 행(두 배우)가 있는 모든 행을 검색
 - 같은 테이블을 여러 번 사용하기 때문에 테이블 별칭 사용

```
select f.title
from film as f
  inner join film_actor as fa1
    on f.film_id = fa1.film_id
  inner join actor a1      # film, film_actor, actor 내부 조인 #1
    on fa1.actor_id = a1.actor_id
  inner join film_actor as fa2
    on f.film_id = fa2.film_id
  inner join actor a2      # film, film_actor, actor 내부 조인 #2
    on fa2.actor_id = a2.actor_id
where (a1.first_name = 'CATE' and a1.last_name = 'MCQUEEN')
and (a2.first_name = 'CUBA' and a2.last_name = 'BIRCH');
```

film	
123	film_id
ABC	title
ABC	description
🕒	release_year
123	language_id
123	original_language_id
123	rental_duration
123	rental_rate
123	length
123	replacement_cost
ABC	rating
ABC	special_features
🕒	last_update

film_actor	
123	actor_id
123	film_id
🕒	last_update

actor	
ABC	first_name
ABC	last_name
🕒	last_update

```
title      |
-----+
BLOOD ARGONAUTS |
TOWERS HURRICANE|
```


5.2 세 개 이상 테이블 조인

- 두 배우가 같이 출연한 영화만 출력 #2
 - 앞 예제를 2개의 쿼리로 분리한 다음 AND 연산 수행

```
# 두 명의 배우가 같이 출연한 영화 검색
# CATE MCQUEEN만 검색
select f.title, f.film_id, a1.first_name, a1.last_name
from film as f
  inner join film_actor as fa1
    on f.film_id = fa1.film_id
  inner join actor a1
    on fa1.actor_id = a1.actor_id
where (a1.first_name = 'CATE' and a1.last_name = 'MCQUEEN');
```

AND

```
# CUBA BIRCH만 검색
select f.title, f.film_id, a2.first_name, a2.last_name
from film as f
  inner join film_actor as fa2
    on f.film_id = fa2.film_id
  inner join actor a2
    on fa2.actor_id = a2.actor_id
where (a2.first_name = 'CUBA' and a2.last_name = 'BIRCH');
```



title	film_id	first_name	last_name
	...	CATE	MCQUEEN
	...	CATE	MCQUEEN
	...	CATE	MCQUEEN

AND



title	film_id	first_name	last_name
	...	CUBA	BIRCH
	...	CUBA	BIRCH
	...	CUBA	BIRCH

5.2 세 개 이상 테이블 조인

```
where (a1.first_name = 'CATE' and a1.last_name = 'MCQUEEN');
```

CATE MCQUEEN
만 검색

title	film_id	first_name	last_name
ANNIE IDENTITY	26	CATE	MCQUEEN
BLOOD ARGONAUTS	82	CATE	MCQUEEN
CAPER MOTIONS	119	CATE	MCQUEEN
COMANCHEROS ENEMY	168	CATE	MCQUEEN
DARN FORRESTER	212	CATE	MCQUEEN
DOCTOR GRAIL	238	CATE	MCQUEEN
FACTORY DRAGON	299	CATE	MCQUEEN
FIDDLER LOST	312	CATE	MCQUEEN
FLYING HOOK	326	CATE	MCQUEEN
FRENCH HOLIDAY	336	CATE	MCQUEEN
GABLES METROPOLIS	345	CATE	MCQUEEN
HAWK CHILL	407	CATE	MCQUEEN
INSIDER ARIZONA	462	CATE	MCQUEEN
JERSEY SASSY	485	CATE	MCQUEEN
LEGEND JEDI	516	CATE	MCQUEEN
MESSAGE IMAGE	564	CATE	MCQUEEN
NAME DETECTIVE	614	CATE	MCQUEEN
PACIFIC AMISTAD	650	CATE	MCQUEEN
PATTON INTERVIEW	665	CATE	MCQUEEN
PERDITION FARGO	671	CATE	MCQUEEN
POTTER CONNECTICUT	693	CATE	MCQUEEN
PRIDE ALAMO	696	CATE	MCQUEEN
SALUTE APOLLO	759	CATE	MCQUEEN
SEARCHERS WAIT	774	CATE	MCQUEEN
SNATCH SLIPPER	814	CATE	MCQUEEN
TOWERS HURRICANE	899	CATE	MCQUEEN
TROJAN TOMORROW	912	CATE	MCQUEEN
VIRGIN DAISY	944	CATE	MCQUEEN
VOLCANO TEXAS	949	CATE	MCQUEEN
WATERSHIP FRONTIER	965	CATE	MCQUEEN

```
where (a2.first_name = 'CUBA' and a2.last_name = 'BIRCH');
```

CUBA BIRCH
만 검색

title	film_id	first_name	last_name
ATLANTIS CAUSE	43	CUBA	BIRCH
BLOOD ARGONAUTS	82	CUBA	BIRCH
COMMANDMENTS EXPRESS	171	CUBA	BIRCH
DYNAMITE TARZAN	266	CUBA	BIRCH
EDGE KISSING	272	CUBA	BIRCH
FINDING ANACONDA	315	CUBA	BIRCH
GREATEST NORTH	378	CUBA	BIRCH
JUNGLE CLOSER	492	CUBA	BIRCH
LANGUAGE COWBOY	509	CUBA	BIRCH
LEAGUE HELLFIGHTERS	512	CUBA	BIRCH
LIBERTY MAGNIFICENT	519	CUBA	BIRCH
LOST BIRD	533	CUBA	BIRCH
MAGNIFICENT CHITTY	548	CUBA	BIRCH
MARS ROMAN	560	CUBA	BIRCH
NORTHWEST POLISH	628	CUBA	BIRCH
ROAD ROXANNE	734	CUBA	BIRCH
RUGRATS SHAKESPEARE	748	CUBA	BIRCH
SHIP WONDERLAND	788	CUBA	BIRCH
SONS INTERVIEW	820	CUBA	BIRCH
STRANGER STRANGERS	853	CUBA	BIRCH
TENENBAUMS COMMAND	882	CUBA	BIRCH
TOOTSIE PILOT	896	CUBA	BIRCH
TOWERS HURRICANE	899	CUBA	BIRCH
VICTORY ACADEMY	940	CUBA	BIRCH

5.3 셀프 조인

■ 셀프 조인(self-join)

- 자기 자신에게 join

customer_id	first_name	last_name	birth_date	spouse_id
1	John	Mayer	1983-05-12	2
2	Mary	Mayer	1990-07-30	1
3	Lisa	Ross	1989-04-15	5
4	Anna	Timothy	1988-12-26	6
5	Tim	Ross	1957-08-15	3
6	Steve	Donell	1967-07-09	4
7	Donna	Trapp	1978-06-23	.

- customer_id 1(John)과 2(Mary)는 서로 배우자
- customer_id 3(Lisa)과 5(Tim)는 서로 배우자
- 고객의 이름과 배우자의 ID는 알 수 있지만, 배우자의 성과 이름은 바로 알 수 없음
- self-join을 이용하여 customer 테이블에 배우자의 성과 이름을 붙여줄 수 있음

5.3 셀프 조인

- customer 테이블 생성 및 데이터 추가

- sqlclass_db 사용: `use sqlclass_db;`

```
use sqlclass_db;
create table customer
(customer_id smallint unsigned,
first_name varchar(20),
last_name varchar(20),
birth_date date,
spouse_id smallint unsigned,
constraint primary key (customer_id));
```

- sqlclass_db 데이터베이스에 생성된 customer 테이블 확인

```
desc customer;
```

Field	Type	Null	Key	Default	Extra
customer_id	smallint unsigned	NO	PRI		
first_name	varchar(20)	YES			
last_name	varchar(20)	YES			
birth_date	date	YES			
spouse_id	smallint unsigned	YES			

5.3 셀프 조인

■ customer 테이블에 데이터 추가

```
insert into customer (customer_id, first_name, last_name, birth_date, spouse_id)
values
```

```
(1, 'John', 'Mayer', '1983-05-12', 2),
(2, 'Mary', 'Mayer', '1990-07-30', 1),
(3, 'Lisa', 'Ross', '1989-04-15', 5),
(4, 'Anna', 'Timothy', '1988-12-26', 6),
(5, 'Tim', 'Ross', '1957-08-15', 3),
(6, 'Steve', 'Donell', '1967-07-09', 4);
```

```
insert into customer (customer_id, first_name, last_name, birth_date)
values (7, 'Donna', 'Trapp', '1978-06-23');
```

```
select * from customer;
```

customer_id	first_name	last_name	birth_date	spouse_id
1	John	Mayer	1983-05-12	2
2	Mary	Mayer	1990-07-30	1
3	Lisa	Ross	1989-04-15	5
4	Anna	Timothy	1988-12-26	6
5	Tim	Ross	1957-08-15	3
6	Steve	Donell	1967-07-09	4
7	Donna	Trapp	1978-06-23	

5.3 셀프 조인

■ self-join 예제

select

```
cust.customer_id,  
cust.first_name,  
cust.last_name,  
cust.birth_date,  
cust.spouse_id,
```

```
spouse.first_name as spouse_firstname,  
spouse.last_name as spouse_lastname
```

from customer **as** cust

join customer **as** spouse

on cust.spouse_id = spouse.customer_id;

새롭게 보여줄 열을 추가

self-join:
customer 테이블을 spouse로 가칭 사용

customer_id	first_name	last_name	birth_date	spouse_id	spouse_firstname	spouse_lastname
1	John	Mayer	1983-05-12	2	Mary	Mayer
2	Mary	Mayer	1990-07-30	1	John	Mayer
3	Lisa	Ross	1989-04-15	5	Tim	Ross
4	Anna	Timothy	1988-12-26	6	Steve	Donell
5	Tim	Ross	1957-08-15	3	Lisa	Ross
6	Steve	Donell	1967-07-09	4	Anna	Timothy

self-join 조건

```
customer_id|first_name|last_name|birth_date|spouse_id|  
-----+-----+-----+-----+-----+  
7|Donna      |Trapp    |1978-06-23|
```

Donna Trapp는 제외됨

실습 5-2

- JOHN이라는 이름의 배우가 출연한 모든 영화의 제목을 출력

```
select f.title
from film as f
    inner join film_actor as fa
    on f.film_id = fa.film_id
    inner join actor as a
    on fa.actor_id = a.actor_id
where a.first_name = 'JOHN';
```

title
ALLEY EVOLUTION
BEVERLY OUTLAW
CANDLES GRAPES
CLEOPATRA DEVIL
COLOR PHILADELPHIA
CONQUERER NUTS
DAUGHTER MADIGAN
GLEAMING JAWBREAKER
GOLDMINE TYCOON
HOME PITY
INTERVIEW LIAISONS
ISHTAR ROCKETEER
JAPANESE RUN
JERSEY SASSY
LUKE MUMMY
MILLION ACE
MONSTER SPARTACUS
NAME DETECTIVE
NECKLACE OUTBREAK
NEWSIES STORY
PET HAUNTING
PIANIST OUTFIELD
PINOCCHIO SIMON
PITTSBURGH HUNCHBACK
QUILLS BULL
RAGING AIRPLANE
ROXANNE REBEL
SATISFACTION CONFIDENTIAL
SONG HEDWIG

5.4 학습 점검

- 같은 도시에 있는 모든 주소를 반환하는 쿼리 작성
 - address 테이블을 self-join, 각 행에는 서로 다른 주소가 포함

```
select a1.address as addr1, a2.address as addr2, a1.city_id, a1.district
from address as a1
      inner join address as a2
where (a1.city_id = a2.city_id)
      and (a1.address_id != a2.address_id);
```

self-join:
address 테이블을 a1, a2의 가칭으로 사용

addr1	addr2	city_id district
-----+	-----+	-----+
47 MySakila Drive	23 Workhaven Lane	300 Alberta
28 MySQL Boulevard	1411 Lillydale Drive	576 QLD
23 Workhaven Lane	47 MySakila Drive	300 Alberta
1411 Lillydale Drive	28 MySQL Boulevard	576 QLD
1497 Yuzhou Drive	548 Uruapan Street	312 England
587 Benguela Manor	43 Vilnius Manor	42 Illinois
548 Uruapan Street	1497 Yuzhou Drive	312 Ontario
43 Vilnius Manor	587 Benguela Manor	42 Colorado



Questions?