

데이터베이스와 SQL

6장

집합 연산자

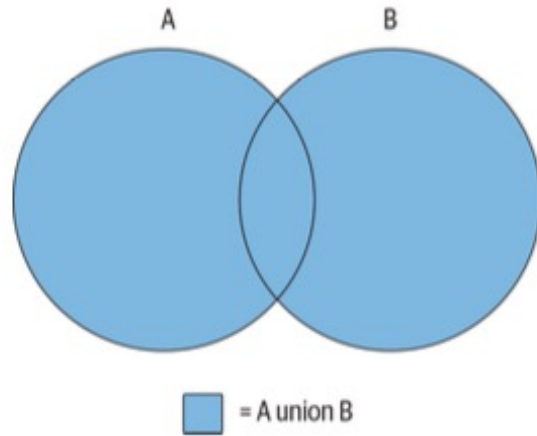
빅데이터 분석가 과정

목차

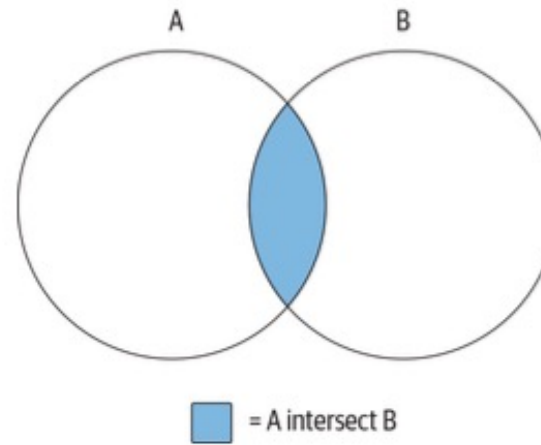
- 6.1 집합 이론
- 6.2 집합 이론 실습
- 6.3 집합 연산자
 - Union 연산자
 - Intersection 연산자
 - Except 연산자
- 6.4 집합 연산 규칙
 - 복합 쿼리의 결과 정렬
 - 집합 연산의 순서

6.1 집합 이론

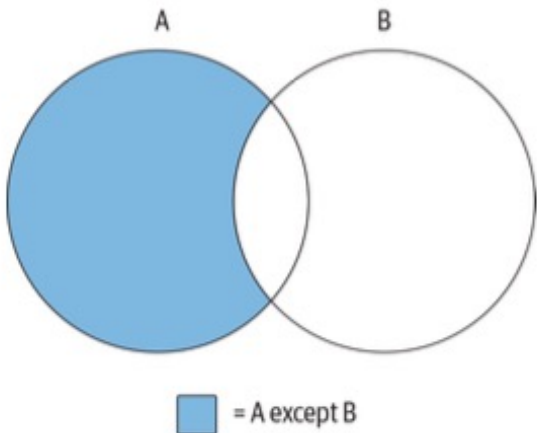
합집합



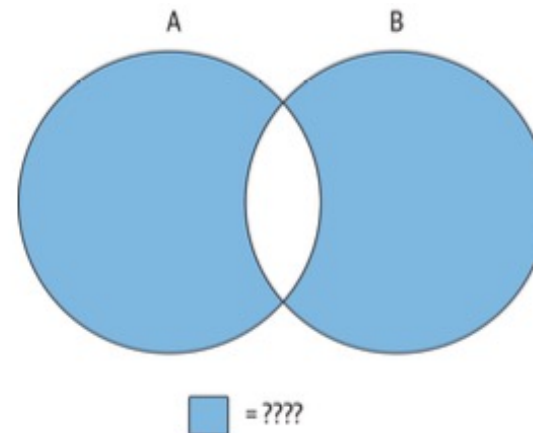
교집합



차집합



?



(A union B) except (A intersection B)

6.2 집합 이론 실습

- 집합 연산 규칙
 - 두 데이터셋 모두 같은 수의 열(column)을 가져야 됨
 - 두 데이터셋의 각 열의 자료형은 서로 동일해야 됨

```
SELECT 1 as num, 'abc' as str
UNION
SELECT 9 as num, 'xyz' as str;
```

```
num|str|
---+---+
  1|abc|
  9|xyz|
```

6.3 집합 연산자

- union 연산자
 - union 연산자
 - 결합된 집합을 정렬하고 중복을 제거
 - union all 연산자
 - 최종 데이터셋의 행의 수는 결합되는 집합의 행의 수의 총합과 같음
 - 중복되는 모든 값을 보여줌

6.3 집합 연산자

- 집합 연산을 하기 전에 customer 테이블과 actor 테이블 구성 확인
 - 두 테이블 모두, **first_name**, **last_name**이 존재하고 데이터 타입도 동일

```
desc customer;
```

Field	Type	Null	Key	Default	Extra
customer_id	smallint unsigned	NO	PRI		auto_increment
store_id	tinyint unsigned	NO	MUL		
first_name	varchar(45)	NO			
last_name	varchar(45)	NO	MUL		
email	varchar(50)	YES			
address_id	smallint unsigned	NO	MUL		
active	tinyint(1)	NO		1	
create_date	datetime	NO			
last_update	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED on update CURRENT_TIMESTAMP

customer
123 customer_id
123 store_id
ABC first_name
ABC last_name
ABC email
123 address_id
123 active
🕒 create_date
🕒 last_update

```
desc actor;
```

Field	Type	Null	Key	Default	Extra
actor_id	smallint unsigned	NO	PRI		auto_increment
first_name	varchar(45)	NO			
last_name	varchar(45)	NO	MUL		
last_update	timestamp	NO		CURRENT_TIMESTAMP	DEFAULT_GENERATED on update CURRENT_TIMESTAMP

actor
123 actor_id
ABC first_name
ABC last_name
🕒 last_update

6.3 집합 연산자

- customer 테이블과 actor 테이블 union all 연산 수행

```
SELECT 'CUST' as type1, c.first_name, c.last_name
FROM customer c
UNION ALL
SELECT 'ACTR' as type1, a.first_name, a.last_name
FROM actor a;
```

	ABC type1	ABC first_name	ABC last_name
1	CUST	MARY	SMITH
2	CUST	PATRICIA	JOHNSON
3	CUST	LINDA	WILLIAMS
4	CUST	BARBARA	JONES
5	CUST	ELIZABETH	BROWN
6	CUST	JENNIFER	DAVIS
7	CUST	MARIA	MILLER
8	CUST	SUSAN	WILSON

. . .

794	ACTR	JAYNE	SILVERSTONE
795	ACTR	BELA	WALKEN
796	ACTR	REESE	WEST
797	ACTR	MARY	KEITEL
798	ACTR	JULIA	FAWCETT
799	ACTR	THORA	TEMPLE

```
SELECT count(first_name) FROM customer;
```

```
count(first_name)|
-----+
                599|
```

```
SELECT count(first_name) FROM actor;
```

```
count(first_name)|
-----+
                200|
```

총 799개의 이름 반환,
599개는 customer 테이블에서, 200개는 actor 테이블에서 가져옴

6.3 집합 연산자

- actor 테이블에 `union_all` 연산 수행
 - 중복 항목 제거 안함 (actor 테이블의 총 개수: 200개)
 - 총 데이터수가 400개로 늘어남

```
SELECT 'ACTR1' as type, a.first_name, a.last_name
FROM actor a
UNION ALL
SELECT 'ACTR2' as type, a.first_name, a.last_name
FROM actor a;
```

type	first_name	last_name
ACTR1	PENELOPE	GUINESS
ACTR1	NICK	WAHLBERG
ACTR1	ED	CHASE
ACTR1	JENNIFER	DAVIS
ACTR1	JOHNNY	LOLLOBRIGIDA
ACTR1	BETTE	NICHOLSON
...
ACTR2	PENELOPE	GUINESS
ACTR2	NICK	WAHLBERG
ACTR2	ED	CHASE
ACTR2	JENNIFER	DAVIS
ACTR2	JOHNNY	LOLLOBRIGIDA
ACTR2	BETTE	NICHOLSON
...

- customer 테이블과 actor 테이블에서
 - 이름이 'J'로 시작하고 성은 'D'로 시작하는 사람들의 합집합: `union all` (중복)

```
SELECT c.first_name, c.last_name
FROM customer c
WHERE c.first_name LIKE 'J%' AND c.last_name LIKE 'D%'
UNION ALL
SELECT a.first_name, a.last_name
FROM actor a
WHERE a.first_name LIKE 'J%' AND a.last_name LIKE 'D%';
```

first_name	last_name
JENNIFER	DAVIS
JENNIFER	DAVIS
JUDY	DEAN
JODIE	DEGENERES
JULIANNE	DENCH

중복

6.3 집합 연산자

- union: 중복 데이터 제거

```
SELECT c.first_name, c.last_name
FROM customer c
WHERE c.first_name LIKE 'J%' AND c.last_name LIKE 'D%'
UNION
SELECT a.first_name, a.last_name
FROM actor a
WHERE a.first_name LIKE 'J%' AND a.last_name LIKE 'D%';
```

first_name	last_name
JENNIFER	DAVIS
JUDY	DEAN
JODIE	DEGENERES
JULIANNE	DENCH

6.3 집합 연산자

■ INTERSECT 연산자

■ MySQL 8.0.31 버전에서 지원

- customer 테이블과 actor 테이블에서 first_name이 'D'로 시작하고, last_name이 'T'로 시작하는 명단 검색

```
SELECT c.first_name, c.last_name
FROM customer c
WHERE c.first_name LIKE 'D%' AND c.last_name LIKE 'T%'
```



```
SELECT a.first_name, a.last_name
FROM actor a
WHERE a.first_name LIKE 'D%' AND a.last_name LIKE 'T%';
```



```
SELECT c.first_name, c.last_name
FROM customer c
WHERE c.first_name LIKE 'D%' AND c.last_name LIKE 'T%'
INTERSECT
SELECT a.first_name, a.last_name
FROM actor a
WHERE a.first_name LIKE 'D%' AND a.last_name LIKE 'T%';
```

first_name	last_name
DOROTHY	TAYLOR
DONNA	THOMPSON
DUANE	TUBBS

first_name	last_name
DUSTIN	TAUTOU
DAN	TORN

first_name	last_name
------------	-----------

교집합 없음

6.3 집합 연산자

- INTERSECT 연산자
 - inner join 연산자를 이용하여 공통 항목 검색

```
SELECT c.first_name, c.last_name
FROM customer as c
INNER JOIN actor as a
ON (c.first_name = a.first_name) and (c.last_name = a.last_name);
```

```
first_name|last_name|
-----+-----+
JENNIFER  |DAVIS    |
```

6.3 집합 연산자

■ INTERSECT(교집합)

- customer 테이블과 actor 테이블의 교집합
 - 두 테이블에서 공통으로 first_name이 'J'로 시작하고, last_name이 'D'로 시작하는 사람 검색

```
SELECT c.first_name, c.last_name
FROM customer c
WHERE c.first_name LIKE 'J%' AND c.last_name LIKE 'D%'
INTERSECT
SELECT a.first_name, a.last_name
FROM actor a
WHERE a.first_name LIKE 'J%' AND a.last_name LIKE 'D%';
```



inner join으로 동일한 결과를 얻을 수 있음

```
SELECT c.first_name, c.last_name
FROM customer as c
  INNER JOIN actor as a
    ON (c.first_name = a.first_name) and (c.last_name = a.last_name)
WHERE a.first_name LIKE 'J%' and a.last_name LIKE 'D%';
```

```
first_name|last_name|
-----+-----+
JENNIFER  |DAVIS    |
```

6.3 집합 연산자

■ EXCEPT 연산자

- MySQL 8.0.31 버전에서 지원
 - A EXCEPT B: A의 결과에 포함된 B의 내용 제거

```
SELECT a.first_name, a.last_name
FROM actor a
WHERE a.first_name LIKE 'J%' AND a.last_name LIKE 'D%';
```

```
SELECT c.first_name, c.last_name
FROM customer c
WHERE c.first_name LIKE 'J%' AND c.last_name LIKE 'D%';
```



```
SELECT a.first_name, a.last_name
FROM actor a
WHERE a.first_name LIKE 'J%' AND a.last_name LIKE 'D%'
EXCEPT
SELECT c.first_name, c.last_name
FROM customer c
WHERE c.first_name LIKE 'J%' AND c.last_name LIKE 'D%';
```



first_name	last_name
JENNIFER	DAVIS
JUDY	DEAN
JODIE	DEGENERES
JULIANNE	DENCH



first_name	last_name
JENNIFER	DAVIS

두 테이블에
모두 있음



first_name	last_name
JUDY	DEAN
JODIE	DEGENERES
JULIANNE	DENCH

6.4 집합 연산 규칙

■ 복합 쿼리의 결과 정렬

- order by 절을 쿼리 마지막에 추가
 - 열 이름 정의는 복합 쿼리의 첫 번째 쿼리에 있는 열의 이름을 사용해야 됨

```
SELECT a.first_name fname, a.last_name lname
FROM actor a
WHERE a.first_name LIKE 'J%' AND a.last_name LIKE 'D%'
UNION ALL
SELECT c.first_name, c.last_name
FROM customer c
WHERE c.first_name LIKE 'J%' AND c.last_name LIKE 'D%'
ORDER BY lname, fname;
```

fname	lname
JENNIFER	DAVIS
JENNIFER	DAVIS
JUDY	DEAN
JODIE	DEGENERES
JULIANNE	DENCH



```
SELECT a.first_name as fname, a.last_name as lname
FROM actor as a
WHERE a.first_name LIKE 'J%' AND a.last_name LIKE 'D%'
UNION ALL
SELECT c.first_name, c.last_name
FROM customer as c
WHERE c.first_name LIKE 'J%' AND c.last_name LIKE 'D%'
ORDER BY last_name, first_name;
```


order by 절에는 select문에서 사용한
열의 이름을 사용해야 됨



SQL Error [1054] [42S22]: Unknown column 'last_name' in 'order clause'

6.4 집합 연산 규칙

- 집합 연산의 순서
 - 복합 쿼리는 위에서 아래의 순서대로 실행
 - 예외: `intersect` 연산자가 다른 집합 연산자보다 우선 순위가 높음



```
SELECT a.first_name, a.last_name
FROM actor a
WHERE a.first_name LIKE 'J%' AND a.last_name LIKE 'D%'
UNION ALL
SELECT a.first_name, a.last_name
FROM actor a
WHERE a.first_name LIKE 'M%' AND a.last_name LIKE 'T%'
UNION
SELECT c.first_name, c.last_name
FROM customer c
WHERE c.first_name LIKE 'J%' AND c.last_name LIKE 'D%';
```

first_name	last_name
JENNIFER	DAVIS
JUDY	DEAN
JODIE	DEGENERES
JULIANNE	DENCH
MARY	TANDY
MENA	TEMPLE

6.5 학습 점검

■ 실습 6-2

- actor와 customer 테이블에서 성이 L로 시작하는 사람의 이름과 성을 찾는 복합 쿼리 작성

```
SELECT first_name, last_name
FROM actor
WHERE last_name like 'L%'
UNION
SELECT first_name, last_name
FROM customer
WHERE last_name like 'L%';
```

first_name	last_name
MATTHEW	LEIGH
JOHNNY	LOLLOBRIGIDA
MISTY	LAMBERT
JACOB	LANCE
RENEE	LANE
HEIDI	LARSON
DARYL	LARUE
LAURIE	LAWRENCE
JEANNE	LAWSON
LAWRENCE	LAWTON
KIMBERLY	LEE
LOUIS	LEONE
SARAH	LEWIS
GEORGE	LINTON
MAUREEN	LITTLE
DWIGHT	LOMBARDI
JACQUELINE	LONG
AMY	LOPEZ
BARRY	LOVELACE
PRISCILLA	LOWE
VELMA	LUCAS
WILLARD	LUMPKIN
LEWIS	LYMAN
JACKIE	LYNCH

6.5 학습 점검

■ 실습 6-3

- last_name 열을 기준으로 실습 6-2의 결과를 오름 차순 정렬하시오.

```
SELECT first_name, last_name
FROM actor
WHERE last_name LIKE 'L%'
UNION
SELECT first_name, last_name
FROM customer
WHERE last_name LIKE 'L%'
ORDER BY last_name;
```

first_name	last_name
MISTY	LAMBERT
JACOB	LANCE
RENEE	LANE
HEIDI	LARSON
DARYL	LARUE
LAURIE	LAWRENCE
JEANNE	LAWSON
LAWRENCE	LAWTON
KIMBERLY	LEE
MATTHEW	LEIGH
LOUIS	LEONE
SARAH	LEWIS
GEORGE	LINTON
MAUREEN	LITTLE
JOHNNY	LOLLOBRIGIDA
DWIGHT	LOMBARDI
JACQUELINE	LONG
AMY	LOPEZ
BARRY	LOVELACE
PRISCILLA	LOWE
VELMA	LUCAS
WILLARD	LUMPKIN
LEWIS	LYMAN
JACKIE	LYNCH



Questions?