

# 데이터베이스와 SQL

---

## 4장. 필터링

# 목차

- 4.1 조건 평가
- 4.2 조건 작성
- 4.3 조건 유형
- 4.4 Null

## 4.1 조건 평가

### ■ where 절

- and 또는 or 연산자로 하나 이상의 조건을 포함
  - and: 모든 조건이 true

```
where first_name = 'STEVE' and create_date > '2006-01-01'
```

- or: 조건 중 하나만 true이면, 해당 조건은 true

```
where first_name = 'STEVE' or create_date > '2006-01-01'
```

### ■ 괄호 사용

- 여러 개의 조건을 포함하는 경우, 괄호를 써서 의도를 명확히 표현

```
where (first_name = 'STEVE' or last_name = 'YOUNG')  
and create_date > '2006-01-01'
```

## 4.1 조건 평가

- not 연산자

```
where not (first_name = 'STEVE' or last_name = 'YOUNG')  
and create_date > '2006-01-01'
```

- 2006년 1월 1일 이후에 기록이 생성된 사람 중에 이름이 STEVE이거나 성이 YOUNG이 아닌 행만 검색

- not 연산자로 <> 사용

```
where first_name <> 'STEVE' and last_name <> 'YOUNG'  
and create_date > '2006-01-01'
```

## 4.2 조건 작성

### ■ 조건 작성

- 하나 이상의 연산자와 결합된 표현식으로 구성
- 표현식
  - 숫자
  - 테이블 또는 뷰의 열
  - 문자열
  - concat()과 같은 내장 함수
  - 서브 쿼리
  - ('Boston', 'New York', 'Chicago')와 같은 표현식 목록
- 조건 연산자
  - 비교 연산자: =, !=, <, >, <>, like, in, between
  - 산술 연산자: +, -, \*, /

## 4.3 조건 유형

- 동등 조건(equality condition): '열 = 표현/값'

```
select c.email
from customer as c
      inner join rental as r
on c.customer_id = r.customer_id
where date(r.rental_date) = '2005-06-14';
```

2개의 동등 조건

- 2005년 6월 14일에 영화를 대여한 모든 고객의 이메일 주소 출력

email
-----+
CATHERINE.CAMPBELL@sakilacustomer.org
JOYCE.EDWARDS@sakilacustomer.org
AMBER.DIXON@sakilacustomer.org
JEANETTE.GREENE@sakilacustomer.org
MINNIE.ROMERO@sakilacustomer.org
GWENDOLYN.MAY@sakilacustomer.org
SONIA.GREGORY@sakilacustomer.org
MIRIAM.MCKINNEY@sakilacustomer.org
CHARLES.KOWALSKI@sakilacustomer.org
DANIEL.CABRAL@sakilacustomer.org
MATTHEW.MAHAN@sakilacustomer.org
JEFFERY.PINSON@sakilacustomer.org
HERMAN.DEVORE@sakilacustomer.org
ELMER.NOE@sakilacustomer.org
TERRANCE.ROUSH@sakilacustomer.org
TERRENCE.GUNDERSON@sakilacustomer.org

## 4.3 조건 유형

- 부등 조건(inequality condition): 두 표현이 동일하지 않음
  - <> 또는 != 사용

```
select c.email
from customer as c
  inner join rental as r
    on c.customer_id = r.customer_id
where date(r.rental_date) <> '2005-06-14';
```

부등 조건

- 2005년 6월 14일 이외의 날짜에 영화를 대여한 사람들의 이메일 주소

email	
-----+-----	
MARY.SMITH@sakilacustomer.org	
MARY.SMITH@sakilacustomer.org	
MARY.SMITH@sakilacustomer.org	
MARY.SMITH@sakilacustomer.org	
MARY.SMITH@sakilacustomer.org	
MARY.SMITH@sakilacustomer.org	
MARY.SMITH@sakilacustomer.org	

## 4.3 조건 유형

- 동등/부등 조건 사용 예
  - 데이터를 수정할 때 사용

```
delete from rental  
where year(rental_date) = 2004;
```

```
delete from rental  
where year(rental_date) <> 2005 and year(rental_date) <> 2006;
```



## 4.3 조건 유형

### ■ 범위 조건

- 해당 식이 특정 범위 내에 있는지 확인

```
select customer_id, rental_date
from rental
where rental_date < '2005-05-25';
```

customer_id	rental_date
130	2005-05-24 22:53:30
459	2005-05-24 22:54:33
408	2005-05-24 23:03:39
333	2005-05-24 23:04:41
222	2005-05-24 23:05:21
549	2005-05-24 23:08:07
269	2005-05-24 23:11:53
239	2005-05-24 23:31:46

- 해당 날짜만 검색: `date(rental_date) = '2005-05-25'`

```
select customer_id, rental_date
from rental
where rental_date <= '2005-06-16'
      and rental_date >= '2005-06-14';
```

범위 조건  
(2005년 6월 14일 00:00:00~  
2005년 6월 16일 00:00:00)

customer_id	rental_date
416	2005-06-14 22:53:33
516	2005-06-14 22:55:13
239	2005-06-14 23:00:34
285	2005-06-14 23:07:08
...	...
95	2005-06-15 00:12:51
197	2005-06-15 00:15:15
512	2005-06-15 00:28:37
210	2005-06-15 00:33:04

- rental\_date 컬럼은 datetime 속성으로 날짜와 시간은 같이 포함
- 2005-06-16은 포함되지 않음 (시간 정보 때문)

## 4.3 조건 유형

### ■ 범위 조건

- 2005년 6월 14일부터 6월 16일까지의 데이터를 출력하기 위해
  - `date(rental_date)`를 사용: 정확한 날짜만 추출

```
select customer_id, rental_date
from rental
where date(rental_date) <= '2005-06-16'
      and date(rental_date) >= '2005-06-14';
```

```
customer_id|rental_date          |
-----+-----+
         416|2005-06-14 22:53:33|
         516|2005-06-14 22:55:13|
         239|2005-06-14 23:00:34|
         . . .
          95|2005-06-15 00:12:51|
         197|2005-06-15 00:15:15|
         512|2005-06-15 00:28:37|
         . . .
         454|2005-06-16 23:53:53|
         215|2005-06-16 23:56:11|
```

## 4.3 조건 유형

- between 연산자
  - between [범위의 하한값] and [범위의 상한값]

```
select customer_id, rental_date
from rental
where rental_date between '2005-06-14' and '2005-06-16';
```

- 숫자 범위 사용
  - 하한값과 상한값이 범위에 포함됨

```
select customer_id, payment_date, amount
from payment
where amount between 10.0 and 11.99;
```

customer_id	payment_date	amount
2	2005-07-30 13:47:43	10.99
3	2005-07-27 20:23:12	10.99
12	2005-08-01 06:50:26	10.99
13	2005-07-29 22:37:41	11.99
...		
572	2005-06-17 04:05:12	10.99
573	2005-07-31 12:14:19	10.99
591	2005-07-07 20:45:51	11.99
592	2005-07-06 22:58:31	11.99
595	2005-07-31 11:51:46	10.99

## 4.3 조건 유형

### ■ 문자열 범위

- last\_name이 'FA'와 'FRB'로 시작하는 데이터 리턴

```
select last_name, first_name
from customer
where last_name between 'FA' and 'FRB';
```

last_name	first_name
FARNSWORTH	JOHN
FENNELL	ALEXANDER
FERGUSON	BERTHA
FERNANDEZ	MELINDA
FIELDS	VICKI
. . .	
FRALEY	JUAN
FRANCISCO	JOEL
FRANKLIN	BETH
FRAZIER	GLENDA

## 4.3.3 멤버십 조건

- OR 또는 IN() 연산
  - 유한한 값의 집합으로 제한
  - IN() 연산
    - 컬럼명 IN (값1, 값2, ...)
    - 지정한 컬럼의 값이 특정 값에 해당되는 조건을 만들 때 사용 (or 대신 사용)

```
select title, rating
from film
where rating='G' or rating='PG';
```



```
select title, rating
from film
where rating in ('G', 'PG');
```

## 4.3.3 멤버십 조건

### ■ 서브 쿼리 사용

- 값의 집합을 생성할 수 있음

```
select title, rating
from film
where rating in (select rating from film where title like '%PET%');
```

#### • 서브 쿼리 내용

- 'PET'을 포함하는 영화 제목을 찾고, 그 영화 제목의 rating을 반환 ('P', 'PG')
  - » 'PET%': PET로 시작하는 단어
  - » '%PET': PET로 끝나는 단어
  - » '%PET%': PET를 포함하는 단어

```
select title, rating from film where title like '%PET%';
```



title	rating
MALKOVICH PET	G
MUPPET MILE	PG
PET HAUNTING	PG

#### • where 절 내용

- where rating in ('G', 'PG');

## 4.3.3 멤버십 조건

### ■ not in 사용

- 표현식 집합 내에 존재하지 않음

```
select title, rating
from film
where rating not in ('PG-13', 'R', 'NC-17');
```

- NOT IN('PG-13', 'R', 'NC-17')
  - 영화 등급이 'PG-13', 'R', 'NC-17'이 아닌 모든 영화를 찾음

title	rating
ACADEMY DINOSAUR	PG
ACE GOLDFINGER	G
AFFAIR PREJUDICE	G
AFRICAN EGG	G
AGENT TRUMAN	PG
ALAMO VIDEOTAPE	G
ALASKA PHANTOM	PG
ALI FOREVER	PG
AMADEUS HOLY	PG
AMISTAD MIDSUMMER	G
ANGELS LIFE	G
ANNIE IDENTITY	G
ARIZONA BANG	PG

## 4.3.4 일치 조건

### ■ 문자열 부분 가져오기

#### ■ `left`(문자열, n)

- 문자열의 가장 왼쪽부터 n개 가져옴

```
select left('abcdefg', 3)
```



```
abc
```

#### ■ `mid`(문자열, 시작 위치, n),

- `substr`(문자, 시작 위치, n)도 동일한 기능 수행

```
select mid('abcdefg', 2, 3);
```



```
bcd
```

#### ■ `right`(문자열, n)

- 문자열의 가장 오른쪽부터 n개 가져옴

```
select right('abcdefg', 2);
```



```
fg
```

### ■ 와일드 카드

- '\_': 정확히 한 문자 (underscore)
- '%': 개수에 상관없이 모든 문자 포함



## 4.3.4 일치 조건

- 일치 조건(matching condition)
  - 와일드 카드 사용시 **LIKE** 연산자를 사용

```
select last_name, first_name
from customer
where last_name like '_A_T%S';
```



last_name	first_name
MATTHEWS	ERICA
WALTERS	CASSANDRA
WATTS	SHELLY

- '\_A\_T%S':
  - 두 번째 위치에 'A', 네 번째 위치에 'T'를 포함하며 마지막은 'S'로 끝나는 문자열

```
select last_name, first_name
from customer
where last_name like 'Q%' or last_name like 'Y%';
```

- last\_name이 'Q'로 시작하거나 'Y'로 시작하는 고객 이름 검색

last_name	first_name
QUALLS	STEPHEN
QUIGLEY	TROY
QUINTANILLA	ROGER
YANEZ	LUIS
YEE	MARVIN
YOUNG	CYNTHIA

## 4.3.4 일치 조건

- 정규 표현식 사용
  - ‘`^[QY]`’: Q 또는 Y로 시작하는 단어 검색

```
select last_name, first_name
from customer
where last_name REGEXP '^[QY]';
```

last_name	first_name
YOUNG	CYNTHIA
QUALLS	STEPHEN
QUINTANILLA	ROGER
YANEZ	LUIS
YEE	MARVIN
QUIGLEY	TROY

## 4.4 Null

### ■ Null값의 다양한 경우

- 해당 사항 없음
- 아직 알려지지 않은 값
- 정의되지 않은 값

### ■ Null 확인 방법

- `is null` 사용 (= `null`)

```
select rental_id, customer_id, return_date
from rental
where return_date is null;
```

rental_id	customer_id	return_date
11496	155	
11541	335	
11563	83	
11577	219	
11593	99	
11611	192	
11646	11	
11652	597	
...		

## 4.4 Null

### ■ is not null

- 열에 값이 할당되어 있는 경우 (null이 아닌 경우)

```
select rental_id, customer_id, return_date
from rental
where return_date is not null;
```

rental_id	customer_id	return_date
1	130	2005-05-26 22:04:30
2	459	2005-05-28 19:40:33
3	408	2005-06-01 22:12:39
4	333	2005-06-03 01:43:41
5	222	2005-06-02 04:33:21
6	549	2005-05-27 01:32:07

## 4.4 Null

### ■ Null과 조건 조합

- 2005년 5월에서 8월 사이에 반납되지 않은 대여 정보 검색
  - 반납이 되지 않은 경우, 반납 날짜의 값이 NULL
  - 또는 반납 날짜가 2005년 5월~ 2005년 8월 사이가 아닌 경우

```
select rental_id, customer_id, return_date
from rental
where return_date is null
or return_date not between '2005-05-01' and '2005-09-01';
```

rental_id	customer_id	return_date
11496	155	
11541	335	
11563	83	
11577	219	
. . .		
15717	208	
15722	26	2005-09-01 13:24:29
15729	193	
15735	295	
15745	43	
15752	17	2005-09-01 14:06:38
15753	144	2005-09-01 09:07:30
15759	361	2005-09-01 11:28:37
15775	173	2005-09-01 16:56:44

## 4.5 실습

### ■ 4.5 서브셋 조건 설정

payment_id	customer_id	amount	payment_date
101	4	8.99	2005-08-18
102	4	1.99	2005-08-19
103	4	2.99	2005-08-20
104	4	6.99	2005-08-20
105	4	4.99	2005-08-21
106	4	2.99	2005-08-22
107	4	1.99	2005-08-23
108	5	0.99	2005-05-29
109	5	6.99	2005-05-31
110	5	1.99	2005-05-31
111	5	3.99	2005-06-15
112	5	2.99	2005-06-16
113	5	4.99	2005-06-17
114	5	2.99	2005-06-19
115	5	4.99	2005-06-20
116	5	4.99	2005-07-06
117	5	2.99	2005-07-08
118	5	4.99	2005-07-09
119	5	5.99	2005-07-09
120	5	1.99	2005-07-09

# 실습 4-1 서브셋

```
select payment_id, customer_id, amount, date(payment_date) payment_date
from payment
where (payment_id between 101 and 120);
```

## 4.5 실습

### ■ 실습 4-1

- 아래의 필터조건에 따라 반환되는 payment\_id는 무엇입니까?
  - customer\_id가 5가 아니고
  - amount가 8보다 크거나 payment\_date가 2005년 8월 23일인 경우

```
select payment_id, customer_id, amount, date(payment_date) payment_date
from payment
where (payment_id between 101 and 120)
and customer_id != 5 and (amount > 8 or date(payment_date) = '2005-08-23');
```

payment_id	customer_id	amount	payment_date
101	4	8.99	2005-08-18
107	4	1.99	2005-08-23

- payment\_id는 101, 107

## 4.5 실습

### ■ 실습 4-2

- 다음 필터조건에 따라 반환되는 payment\_id는 무엇입니까?
  - customer\_id는 5이고
  - amount가 6보다 크거나 payment\_date가 2005년 6월 19일이 아닌 payment\_id

```
select payment_id, customer_id, amount, date(payment_date) payment_date
from payment
where (payment_id between 101 and 120)
and customer_id = 5 and not (amount > 6 or date(payment_date) = '2005-06-19');
```

payment_id	customer_id	amount	payment_date
108	5	0.99	2005-05-29
110	5	1.99	2005-05-31
111	5	3.99	2005-06-15
112	5	2.99	2005-06-16
113	5	4.99	2005-06-17
115	5	4.99	2005-06-20
116	5	4.99	2005-07-06
117	5	2.99	2005-07-08
118	5	4.99	2005-07-09
119	5	5.99	2005-07-09
120	5	1.99	2005-07-09



## 4.5 실습

### ■ 실습 4-3

- payments 테이블에서 금액이 1.98, 7.98 또는 9.98인 모든 행을 검색하는 쿼리

```
select amount from payment  
where amount in (1.98, 7.98, 9.98);
```

```
amount|  
-----+  
  7.98|  
  9.98|  
  1.98|  
  7.98|  
  7.98|  
  7.98|  
  7.98|
```



# Questions?