Selection Operator: `WHERE` Clause

DSC 301: Lecture 5

February 5, 2021

Lecture Objectives

- `WHERE` clause
- SQL operators
- Rename operator (alias)

Last time we covered *projection* operators, $\Pi_{(a,b,c)}(R)$, that filtered attributes by selecting a subset of the columns. In this lecture, we cover the *selection* operator from *relational algebra*. Recall these operators are written as $\sigma_p(R)$ where p is a predicate. The form of this predicate is our focus.

The `WHERE` Clause

The `WHERE` clause is used to filter records based on a conditional predicate. A predicate is a statement containing a conditional expression that is either TRUE or FALSE.

The general syntax is:

```
SELECT <columns> FROM <TableX> WHERE [condition];
```

Comparison Operators

Comparison operators include equal (=), non-equal (<>), less than (<), less than or equal (<=), greater than (>), and greater than or equal (>=). The equal sign is used to test equality.

Example 1. *Select all carriers and flight numbers whose destination is San Francisco.*

```
SELECT flight FROM Flights WHERE dest = `SFO';
```

Example 2. *Select all flights that whose departure was delayed at least 30 minutes.*

```
SELECT flight FROM Flights WHERE dep_delay >= 30;
```

Conjunctive Operators

Conjunctions are used to build more complex queries. For **example**, if we want to know the flights to San Francisco that arrive on time, we would write:

```
SELECT flight FROM Flights WHERE dest='SFO' AND arr_delay <=0;
```

Here are a few more examples.

Example 3. *Find all flights going to Chicago, either O'hara or Midway.*

```
SELECT flight FROM Flights WHERE dest='ORD' OR dest='MDW';
```

Example 4. *Find all flight IDs going to Chicago Midway on Christmas day.*

```
SELECT fid FROM Flights WHERE dest='MDW' AND month=12 AND day=25;
```

Logical Operators

Logical operators used with the `WHERE` clause include[1]:

IS NULL	BETWEEN	IN
LIKE	EXISTS	UNIQUE
ALL	SOME	ANY

Table 1: Logical Operators in SQL

Example 5. *Select fights with delays ranging between 10 and 20 minutes.*

[1] ALL, ANY, EXIST, SOME, UNIQUE will be used with subqueries.

LIKE Wildcards	Description
%	Match any string of any length (including zero length)
_	Match on a single character

Table 2: Wildcards for LIKE

```
SELECT
    flight, dep_delay
FROM
    Flights
WHERE
    dep_delay BETWEEN 10 AND 30;
```

IN

Records where field is in a list of criteria. For example, flights whose destination matches one of many. For **example**, flights with destination San Francisco, Los Angles, or Oakland California.

```
SELECT fid FROM Flights WHERE dest IN ('SFO','LAX','OAK');
```

LIKE

The LIKE condition is used in the WHERE clause and allows the use of wildcards to perform pattern matching in a query. Wildcards include:

```
SELECT dest FROM Flights where dest like 'H%';
```

```
SELECT dest FROM Flights where dest like '_N_';
```

Negation

How about not? Many times you want the compliment of a set or results that do **not** include some parameter. Equality is negated using <> (or !=). See the next example.

Example 6. *Find all flight IDs not going to Chicago Midway on Christmas day.*

```sql
SELECT fid FROM Flights WHERE dest<>'MDW' AND month=12 AND day=25;
```

Negate logical operators using the keyword NOT. It can be used with any logical operator, e.g., NOT LIKE, NOT IN, NOT BETWEEN, NOT EXISTS, NOT ANY, etc. Here are two examples using NOT[2].

Example 7. *Find all carriers departing SEA excluding AS, DL, WN, and UA.*

```sql
SELECT
    carrier
FROM
    Flights
WHERE
    carrier NOT IN ('AS' , 'DL', 'WN', 'UA');
```

Example 8. *Find flgiths departing SEA whose distances are less than 500 or greater than 1000 using NOT BETWEEN. Can you do this another way?*

```sql
SELECT
    flight
FROM
    Flights
WHERE
    distance NOT BETWEEN 500 AND 1000;
```

```sql
SELECT flight FROM Flights WHERE distance<500 OR distance > 1000;
```

Rename Operator

We studied *rename* operators, $\rho_a(R)$, in the section on relational algebra. The idea is simple, sometimes we want to rename an attribute or relation[3]. Here is an example illustrating the rename operator. This operation is also called **aliasing**.

Example 9 (Column alias)**.** *List the departure delays from flights to San Francisco in June. Rename departure delay field to just "Delay".*

[2] NOT is often used together with the IS NULL operator.

[3] We will not rename tables until working with multiple tables.

```
SELECT
    dep_delay AS Delay
FROM
    Flights
WHERE
    dest = 'SFO' AND month = 6;
```

Note 1: This displays all delays (even if early). How can you change it to display only flights that were actually late departing?

Note 2: If you want to rename column with multiple words (e.g., Departure Delay), you must use single quotations. The example above would be:

```
SELECT
    dep_delay AS 'Departure Delay'
FROM
    Flights
WHERE
    dest = 'SFO' AND month = 6;
```
