

29 - Operadores lógicos (and - or - not)

Hasta el momento, hemos aprendido a establecer una condición con "where" utilizando operadores relacionales. Podemos establecer más de una condición con la cláusula "where", para ello aprenderemos los operadores lógicos.

Son los siguientes:

- and, significa "y",
- or, significa "y/o",
- not, significa "no", invierte el resultado
- (), paréntesis

Los operadores lógicos se usan para combinar condiciones.

Si queremos recuperar todos los libros cuyo autor sea igual a "Borges" y cuyo precio no supere los 20 pesos, necesitamos 2 condiciones:

```
select * from libros
  where (autor='Borges') and
        (precio<=20);
```

Los registros recuperados en una sentencia que une 2 condiciones con el operador "and", cumplen con las 2 condiciones.

Queremos ver los libros cuyo autor sea "Borges" y/o cuya editorial sea "Planeta":

```
select * from libros
  where autor='Borges' or
        editorial='Planeta';
```

En la sentencia anterior usamos el operador "or"; indicamos que recupere los libros en los cuales el valor del campo "autor" sea "Borges" y/o el valor del campo "editorial" sea "Planeta", es decir, seleccionará los registros que cumplan con la primera condición, con la segunda condición o con ambas condiciones.

Los registros recuperados con una sentencia que une 2 condiciones con el operador "or", cumplen 1 de las condiciones o ambas.

Queremos recuperar los libros que NO cumplan la condición dada, por ejemplo, aquellos cuya editorial NO sea "Planeta":

```
select * from libros
  where not editorial='Planeta';
```

El operador "not" invierte el resultado de la condición a la cual antecede.

Los registros recuperados en una sentencia en la cual aparece el operador "not", no cumplen con la condición a la cual afecta el "NOT".

Los paréntesis se usan para encerrar condiciones, para que se evalúen como una sola expresión. Cuando explicitamos varias condiciones con diferentes operadores lógicos (combinamos "and", "or")

permite establecer el orden de prioridad de la evaluación; además permite diferenciar las expresiones más claramente.

Por ejemplo, las siguientes expresiones devuelven un resultado diferente:

```
select* from libros
  where (autor='Borges') or
  (editorial='Paidos' and precio<20);
```

```
select * from libros
  where (autor='Borges' or editorial='Paidos') and
  (precio<20);
```

Si bien los paréntesis no son obligatorios en todos los casos, se recomienda utilizarlos para evitar confusiones.

El orden de prioridad de los operadores lógicos es el siguiente: "not" se aplica antes que "and" y "and" antes que "or", si no se especifica un orden de evaluación mediante el uso de paréntesis.

El orden en el que se evalúan los operadores con igual nivel de precedencia es indefinido, por ello se recomienda usar los paréntesis.

Entonces, para establecer más de una condición en un "where" es necesario emplear operadores lógicos. "and" significa "y", indica que se cumplan ambas condiciones; "or" significa "y/o", indica que se cumpla una u otra condición (o ambas); "not" significa "no", indica que no se cumpla la condición especificada.

Servidor de SQL Server instalado en forma local.

Ingresemos el siguiente lote de comandos en el SQL Server Management Studio:

```
if object_id ('libros') is not null
  drop table libros;
```

```
create table libros(
  codigo int identity,
  titulo varchar(40) not null,
  autor varchar(20) default 'Desconocido',
  editorial varchar(20),
  precio decimal(6,2),
);
```

```
go
```

```
insert into libros
  values('El aleph','Borges','Emece',15.90);
insert into libros
  values('Antología poética','Borges','Planeta',39.50);
insert into libros
  values('Java en 10 minutos','Mario Molina','Planeta',50.50);
insert into libros
  values('Alicia en el pais de las maravillas','Lewis Carroll','Emece',19.90);
insert into libros
  values('Martin Fierro','Jose Hernandez','Emece',25.90);
insert into libros
  values('Martin Fierro','Jose Hernandez','Paidos',16.80);
```

```

insert into libros
  values('Aprenda PHP','Mario Molina','Emece',19.50);
insert into libros
  values('Cervantes y el quijote','Borges','Paidos',18.40);

-- Recuperamos los libros cuyo autor sea igual a "Borges" y
-- cuyo precio no supere los 20 pesos:
select * from libros
  where (autor='Borges') and
    (precio<=20);

-- Seleccionamos los libros cuyo autor es "Borges" y/o cuya editorial es "Planeta":
select * from libros
  where autor='Borges' or
    editorial='Planeta';

-- Recuperamos los libros cuya editorial NO es "Planeta":
select * from libros
  where not editorial='Planeta';

-- Veamos cómo el uso de paréntesis hace que SQL Server evalúe en forma diferente
-- ciertas consultas aparentemente iguales:
select * from libros
  where (autor='Borges') or
    (editorial='Paidos' and precio<20);

select * from libros
  where (autor='Borges' or editorial='Paidos') and
    (precio<20);

```