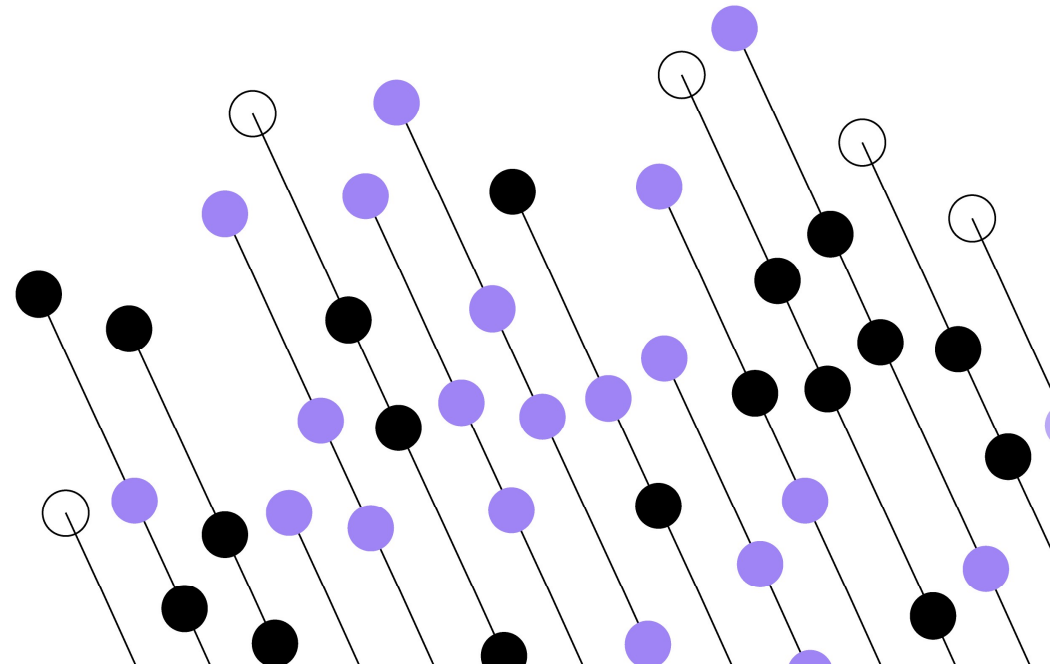


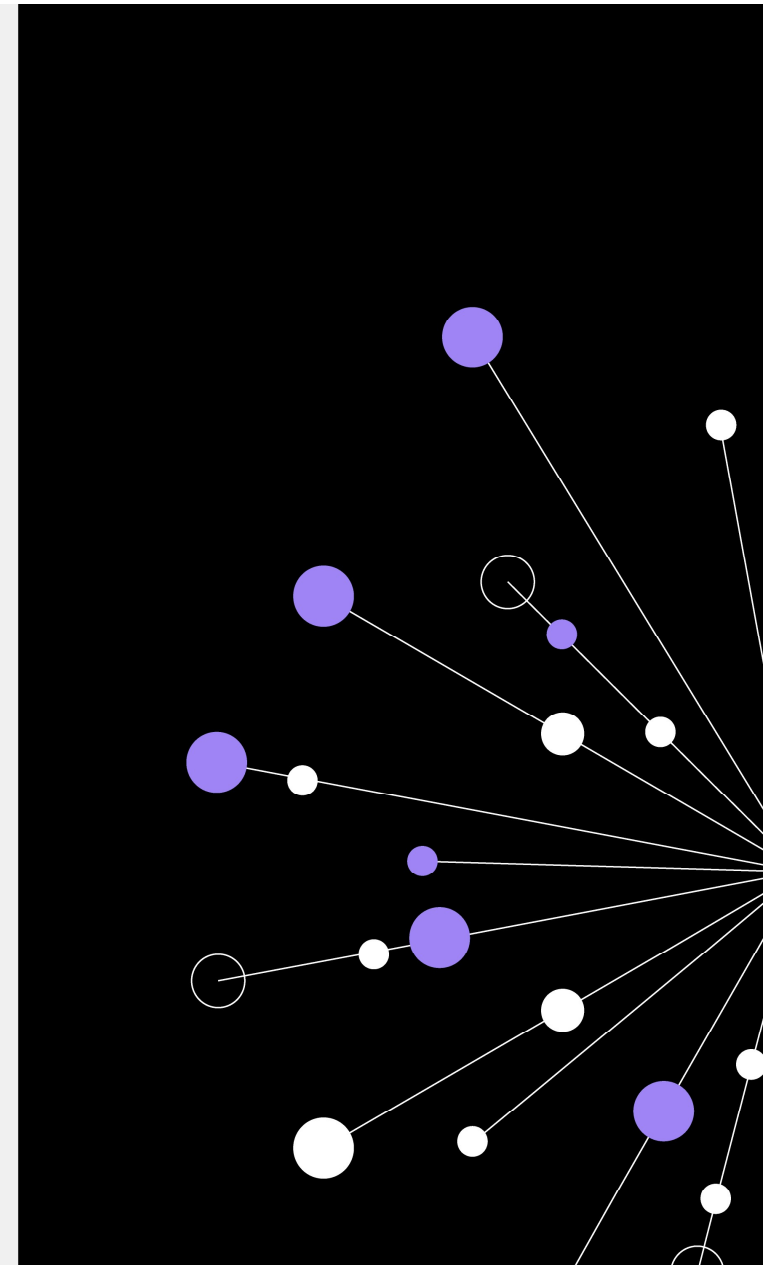
SQL: Filtering Rows



SQL

Contents

- Filtering rows
- Comparison, logical and special operators



SELECT statement: where

SELECT <<field(s)>>

FROM <<table(s)>>

WHERE <<condition(s)>>

GROUP BY <<field(s)>>

HAVING <<condition(s)>>

ORDER BY <<field(s)>>



Operators

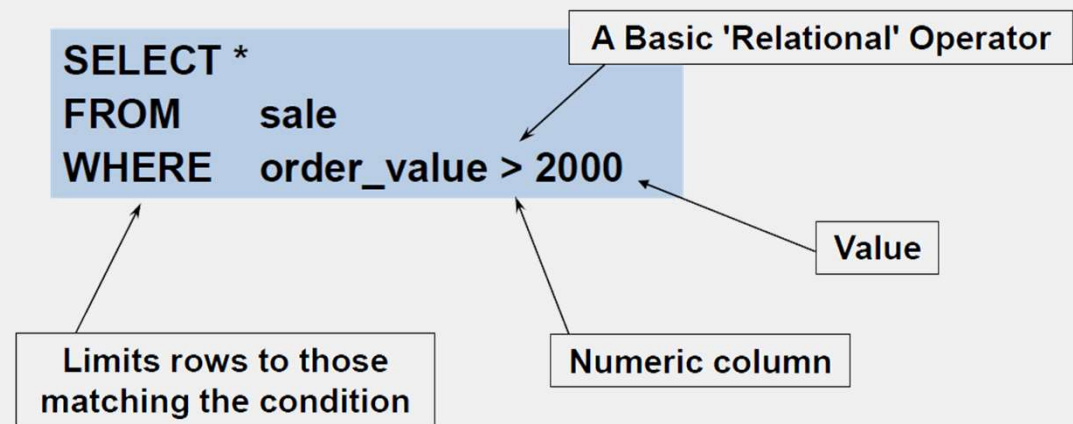
Type	Operators
• Comparison operators	• = < > <> != >= <=
• Logical operators	UnitsInStock > 0 • AND OR NOT UnitsInStock = 0 AND UnitsOnOrder = 0
• "Special" operators	• BETWEEN IN LIKE UnitsInStock BETWEEN 10 AND 100

A row is included in the result set if the condition is satisfied (the test returns true).

Comparison operators

Using database Northwind:

- Display the first name, last name and city from table Employees for the employees based in London.
- Display the product ID, product name and unit price from table Products for the products with price higher than 100.



Filtering on calculated values

Alias cannot be used in WHERE clause.

```
SELECT
    ProductID,
    ProductName,
    UnitsInStock + UnitsOnOrder AS
FutureStock
FROM
    dbo.Products
WHERE
    --FutureStock < 100 --won't work!
    --use:
    (UnitsInStock + UnitsOnOrder) < 100
```

Logical operators

Using logical operators AND / OR

```
SELECT *  
FROM employee  
WHERE county = 'Surrey' AND name = 'Smith'
```

Both conditions must be true

Every time you say AND you are likely to get less rows

```
SELECT *  
FROM employee  
WHERE county = 'Surrey' OR name = 'Smith'
```

Either (or both) conditions can be true

Every time you say OR you are likely to get more rows

Notice with AND or OR in WHERE clause both Column and Value must be specified

Multiple conditions

Multiple conditions

In the operators' precedence, AND precedes OR

- ❖ In absence of brackets AND will be executed first then OR

Say you want all employees in either Surrey or Harlow but must have Name of Smith

This will be the order SQL will run the query if no brackets applied

Anything different must be explicitly stated with brackets around what to execute first

```
SELECT *  
FROM employee  
WHERE county = 'Surrey'  
OR town = 'Harlow'  
AND name = 'Smith'
```

```
SELECT *  
FROM employee  
WHERE county = 'Surrey'  
OR (town = 'Harlow'  
AND name = 'Smith')
```

```
SELECT *  
FROM employee  
WHERE (county = 'Surrey'  
OR town = 'Harlow')  
AND name = 'Smith'
```


Have a go at multiple conditions

Using database Northwind, display the product ID, product name, category ID and unit price from table Products for the products from categories 7 or 8 with unit price higher than 30.



Special operator between

Using database Northwind, display the product ID, product name and unit price from table Products for the products with unit price between 35 and 40.

How would the query change if the unit price is to be greater than 35 and less or equal to 40?

```
SELECT *  
FROM    salesperson  
WHERE    sales_target BETWEEN 100 AND 500
```

Starting Value

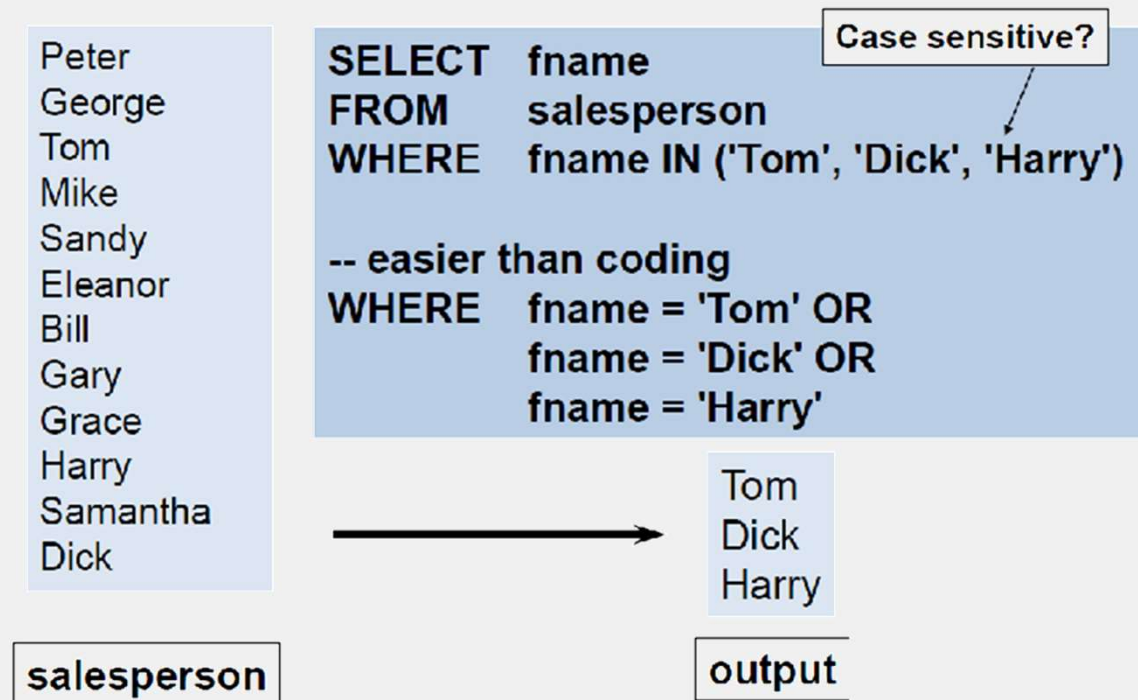
Stopping Value

- A lot easier than typing

```
WHERE    sales_target >= 100 AND  
         sales_target <= 500
```

- Values are inclusive

Special operator in



Special operator in

Revisit the query that you did earlier:

- Using database Northwind, display the product ID, product name, category ID and unit price from table Products for the products from categories 7 or 8 with unit price higher than 30.
- Rework the query using the IN operator.



Special operator like

Using database Northwind, display the first name, last name and title from table Employees for those whose title:

- starts with "sales".
- Includes "sales".

- **When used with 'Like'**

- '_' Matches any single character
- '%' Matches any number of characters (incl 0!)

WHERE Iname LIKE 'a%'

← starts with 'A'

WHERE Iname LIKE '%a'

← ends with 'A'

WHERE Iname LIKE '%a%'

← contains an 'A'

WHERE Iname LIKE '_t%n%r_'

← has a 'T' in pos 2
has an 'R' 1 char from end
and an 'N' between them

- Note 'Like' is only used with character columns

| e.g 'stationary' |

Finding unknown values (nulls)



```
SELECT
    CompanyName, Phone,
    Fax
FROM
    dbo.Suppliers
WHERE
    Fax = NULL
```

The screenshot shows a SQL Server Enterprise Manager window with a query result grid. The grid has columns for CompanyName, Phone, and Fax. The status bar at the bottom indicates 0 rows.

	CompanyName	Phone	Fax
--	-------------	-------	-----

8R.2 (10.50 RTM) QATSQL\Student (53) Northwind 00:00:00 0 rows



```
SELECT
    CompanyName, Phone,
    Fax
FROM
    dbo.Suppliers
WHERE
    Fax IS NULL
```

The screenshot shows a SQL Server Enterprise Manager window with a query result grid. The grid has columns for CompanyName, Phone, and Fax. The status bar at the bottom indicates 16 rows.

	CompanyName	Phone	Fax
1	Exotic Liquids	(171) 555-2222	NULL
2	New Orleans Cajun Delights	(100) 555-4822	NULL
3	Tokyo Traders	(03) 3555-5011	NULL
4	Cooperativa de Quesos 'Las Cabras'	(98) 598 76 54	NULL
5	Mayumi's	(06) 431-7877	NULL
6	Specialty Biscuits, Ltd.	(161) 555-4448	NULL
7	Refrescos Americanas LTDA	(11) 555 4640	NULL
8	Heli Süßwaren GmbH & Co. KG	(010) 9984510	NULL
9	Plutzer Lebensmittelgefäße AG	(030) 892755	NULL
10	... (truncated)

R.2 (10.50 RTM) QATSQL\Student (53) Northwind 00:00:00 16 rows

The case expression

```
USE Northwind
SELECT ProductID, ProductName, UnitPrice, CASE
WHEN UnitPrice > 100 THEN 'Price Level 1'
WHEN UnitPrice <= 100 AND UnitPrice > 70 THEN 'Price Level 2'
WHEN UnitPrice <= 70 AND UnitPrice > 40 THEN 'Price Level 3'
WHEN UnitPrice <= 40 AND UnitPrice > 20 THEN 'Price Level 4'
ELSE 'Price Level 5'
END AS 'Price Level'
FROM Products
ORDER BY ProductID
```

NOTE: The conditions must not overlap, otherwise the result is reliant on their order.

Fix these

The following queries from the Northwind database don't return results. Why?

```
SELECT ProductID, ProductName FROM Products  
WHERE UnitPrice <= 30 AND UnitPrice >= 40
```

```
SELECT ProductID, ProductName FROM Products  
WHERE UnitPrice BETWEEN 40 AND 30
```

