In this video, we continue to explore **SQL queries** and **filters**, but now we’ll apply them to **new data types**: string, numeric, and date/time.

**Understanding the Data Types**

1. **String Data**: This type of data consists of an ordered sequence of characters, which could include letters, numbers, or symbols. For example, **user names** like analyst10 are considered string data.
2. **Numeric Data**: This refers to data that consists of numbers, such as a count of **log-in attempts**. Numeric data allows you to perform **mathematical operations**, like addition or multiplication.
3. **Date and Time Data**: This represents a **date** and/or **time**, which is useful for queries involving events that happened at specific moments, like log-in attempts within a specific time window.

**Applying Filters to Numeric and Date/Time Data**

While we’ve previously used filters with string data, now we’ll focus on **numeric** and **date/time** data. As a **security analyst**, you may need to filter for data based on numbers (like **log-in attempts**) or time ranges (like **patch dates**). SQL has operators that help with this.

**Common Operators for Numeric and Date/Time Data**

Some of the operators you’ll use for numeric or date/time data include:

* **= (equals)**
* **> (greater than)**
* **< (less than)**
* **<> (not equal to)**
* **>= (greater than or equal to)**
* **<= (less than or equal to)**

**Example: Filtering Log-In Attempts After 6 PM**

Let’s say you want to find log-in attempts made **after 6 PM**, perhaps to detect suspicious patterns. You’d use the **greater than** (>) operator in your query. Here’s how the query would look:

SELECT \* FROM log\_in\_attempts WHERE time > '18:00';

This SQL statement does the following:

* Selects all columns from the **log\_in\_attempts** table.
* Filters rows where the **time** is greater than '18:00', which represents **6 PM**.

**Using the BETWEEN Operator**

Another way to filter for numbers or dates within a specific range is by using the **BETWEEN** operator. For instance, if you want to find all patches installed **between March 1st, 2021 and September 1st, 2021**, you can filter using BETWEEN. Here’s how you’d write that query:

SELECT \* FROM machines WHERE OS\_patch\_date BETWEEN '2021-03-01' AND '2021-09-01';

Explanation:

* We select all records from the **machines** table.
* The **WHERE** clause filters the **OS\_patch\_date** column to return all dates within the range of **March 1st, 2021** to **September 1st, 2021**.

**Key Notes on Data Filtering:**

* For **string, date, and time** data types, we use **quotation marks** (') to specify values.
* For **numeric** values, we do not use quotation marks.

**Conclusion**

With these filtering techniques, you're now able to refine your SQL queries to work with different types of data. You can filter log-in attempts by time or date ranges for patches, and use various operators to tailor your searches. In the next video, we’ll explore even more advanced filtering by using **multiple conditions** in a single query.

# Operators for filtering dates and numbers

Previously, you examined operators like less than (<) or greater than (>) and explored how they can be used in filtering numeric and date and time data types. This reading summarizes what you learned and provides new examples of using operators in filters.

## Numbers, dates, and times in cybersecurity

Security analysts work with more than just **string data**, or data consisting of an ordered sequence of characters.

They also frequently work with **numeric data**, or data consisting of numbers. A few examples of numeric data that you might encounter in your work as a security analyst include:

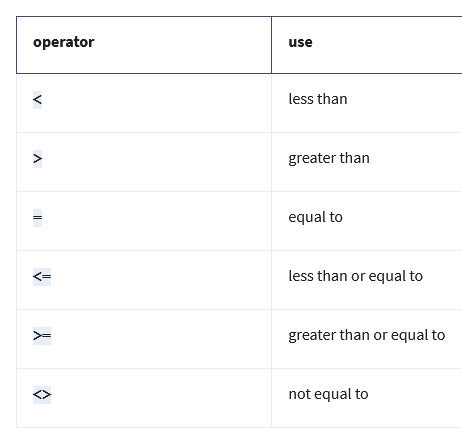
* the number of login attempts
* the count of a specific type of log entry
* the volume of data being sent from a source
* the volume of data being sent to a destination

You'll also encounter **date and time data**, or data representing a date and/or time. As a first example, logs will generally timestamp every record. Other time and date data might include:

* login dates
* login times
* dates for patches
* the duration of a connection

## Comparison operators

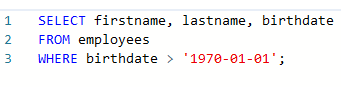
In SQL, filtering numeric and date and time data often involves operators. You can use the following operators in your filters to make sure you return only the rows you need:



**Note:** You can also use != as an alternative operator for not equal to.

### Incorporating operators into filters

These comparison operators are used in the WHERE clause at the end of a query. The following query uses the > operator to filter the birthdate column. You can run this query to explore its output:

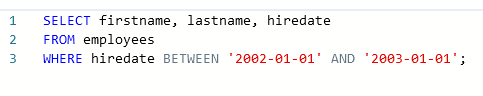


This query returns the first and last names of employees born after, but not on, '1970-01-01' (or January 1, 1970). If you were to use the >= operator instead, the results would also include results on exactly '1970-01-01'.

In other words, the > operator is exclusive and the >= operator is inclusive.  An **exclusive operator** is an operator that does not include the value of comparison.An **inclusive operator** is an operator that includes the value of comparison.

### ****BETWEEN****

Another operator used for numeric data as well as date and time data is the BETWEEN operator. BETWEEN filters for numbers or dates within a range. For example, if you want to find the first and last names of all employees hired between January 1, 2002 and January 1, 2003, you can use the BETWEEN operator as follows:



**Note:** The BETWEEN operator is inclusive. This means records with a hiredate of January 1, 2002 or January 1, 2003 are included in the results of the previous query.

## Key takeaways

Operators are important when filtering numeric and date and time data. These include exclusive operators such as < and inclusive operators such as  <=. The BETWEEN operator, another inclusive operator, helps you return the data you need within a range.