# Locking and Blocking

In SQL Server, locking and blocking are mechanisms that are used to protect data integrity and consistency by controlling access to data by concurrent transactions. Locking is the process of identifying and reserving data resources for use by a specific task, while blocking is the process of preventing other tasks from accessing data that is currently being used by another task.

## Locks

Locks are used to ensure that data is not modified by multiple tasks at the same time, which can lead to data inconsistencies. SQL Server uses different types of locks, such as shared locks, exclusive locks, and update locks, to control access to data in different situations.

## Blocking

Blocking occurs when one task is holding a lock on a resource that another task is trying to access. This can cause the second task to wait until the first task releases the lock, which can lead to performance issues if the wait time is long. SQL Server uses lock escalation and timeout settings to manage blocking and prevent it from becoming a problem.

Overall, locking and blocking are important mechanisms for maintaining the integrity and consistency of data in SQL Server, but they can also impact performance if not managed properly.

SQL Server uses locking to manage concurrent access to data and prevent conflicts that could result in data corruption. When one task acquires a lock on a resource, other tasks that want to access the same resource are blocked until the lock is released. This is known as blocking, and it can occur at different levels of granularity depending on the type of lock that is acquired.

## There are two main types of locks in SQL Server:

- Shared locks allow multiple tasks to read data at the same time, but they prevent any task from writing to the data until all shared locks are released.

- Exclusive locks allow a single task to write to data, but they prevent any other task from reading or writing to the data until the exclusive lock is released.

Blocking can be a natural and necessary part of the way that SQL Server manages concurrency. However, excessive blocking can cause performance problems and decrease the overall throughput of the database. To avoid these problems, it is important to carefully design the database and the applications that access it, and to monitor for and troubleshoot blocking issues when they arise.

As part of Database Health Monitor, I have created the SQL Performance monitor to catch excessive locking that turns into a block for an extensive amount of time. This is incredibly useful to track the problem queries that are having performance issues on your SQL Server.

### Download Database Health Monitor here:

?**http://databasehealth.com/download2/**

I hope this helps you better understand Locking and Blocking on SQL Server.