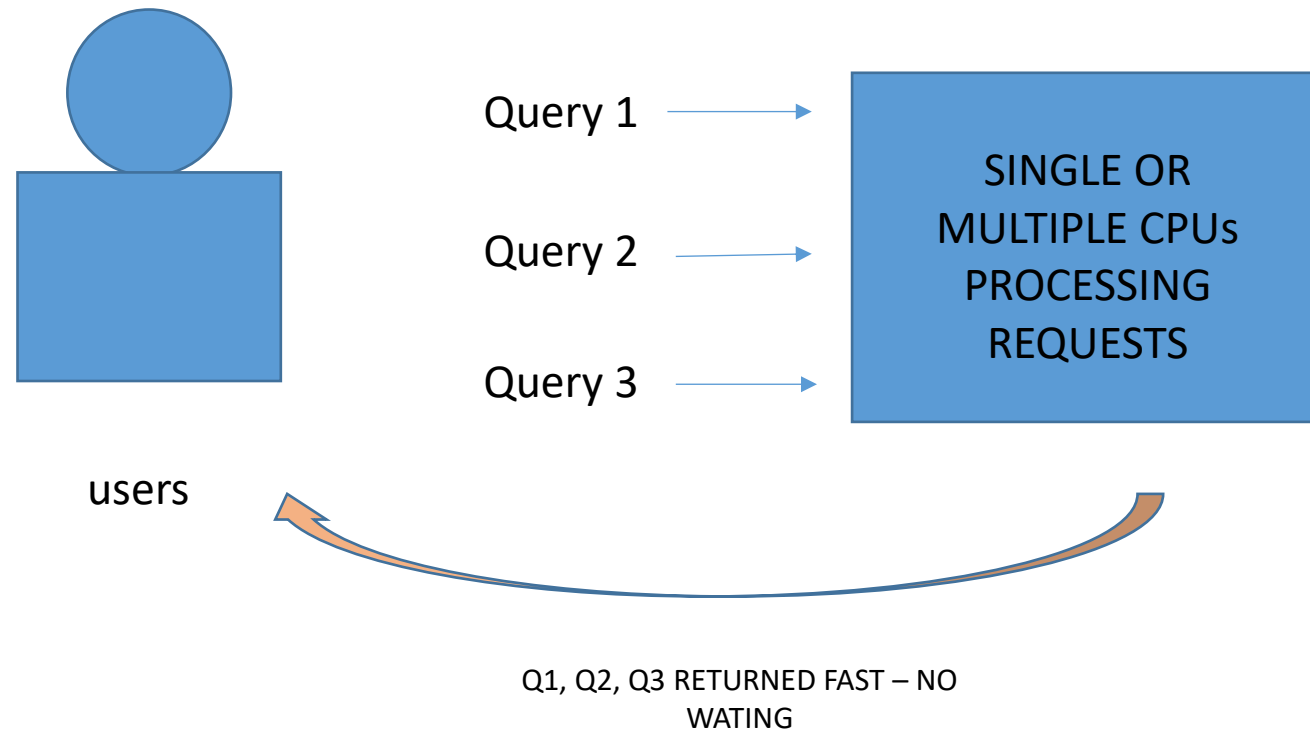


Wait statistics

- What are wait statistics?
- Cycle of a wait stat (query)
- DMV used to discover issues
- Demo

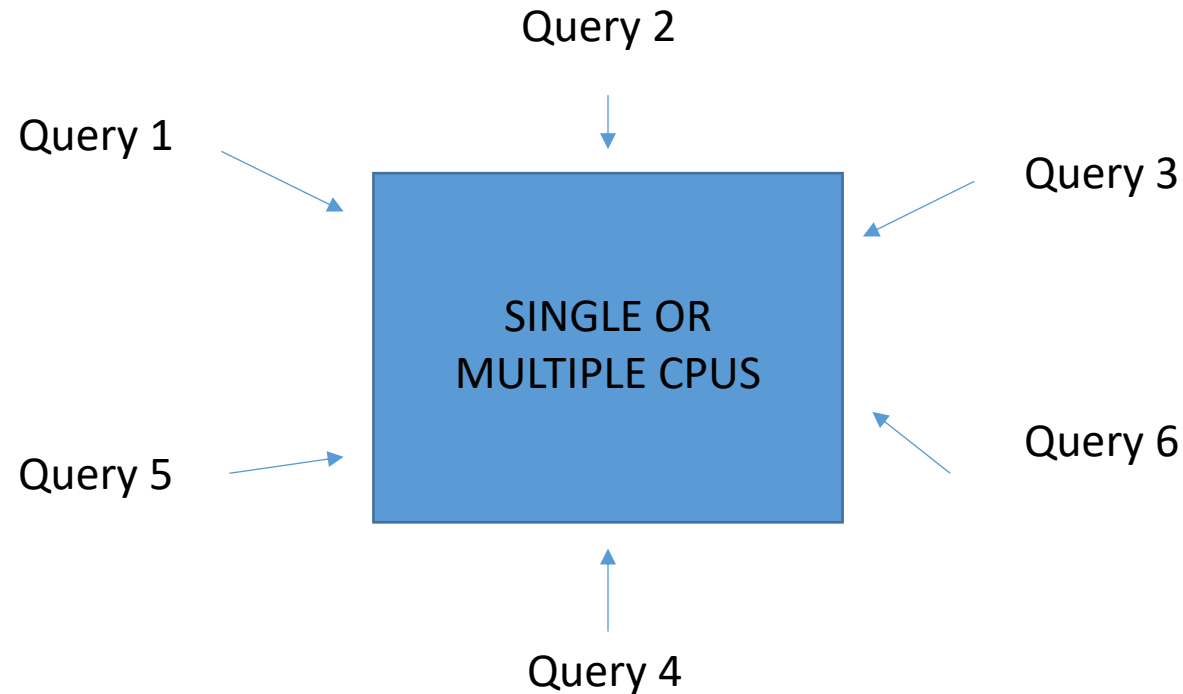
What are wait statistics?

- When a query is sent from an application they must 'compete ' for the limited CPU processor.
- On a single core CPU with little transactional activity, accessing the CPU for processing may not be a problem



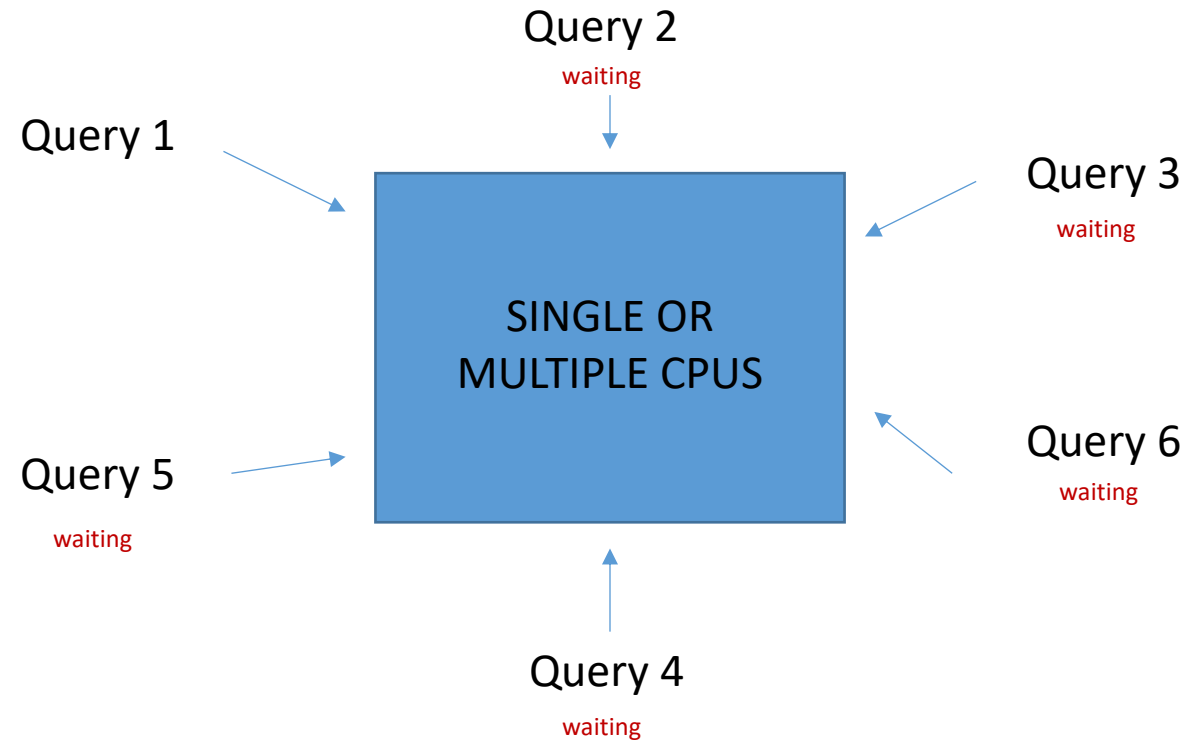
What are wait statistics?

- However, on a very busy transactional database, where you have thousands of users requesting their query to be processed this can become an issue. As these requests 'compete' for the limited CPU resource, **WAITING** for the CPU to process the request can become an issue. And the CPU can be taxed.



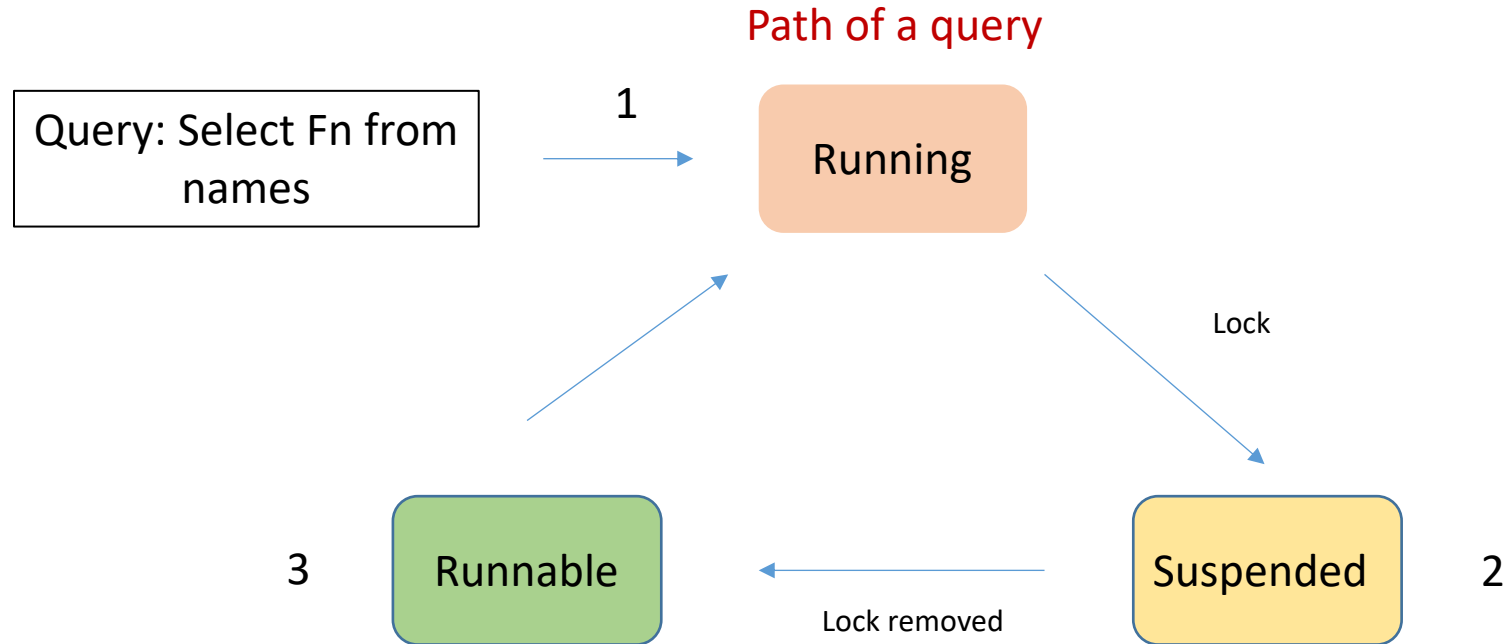
What are wait statistics?

- Here we see queries waiting for the processor to execute the request. There can be many factors causing this waiting, such as locking, poor index, badly designed query and more.



What are wait statistics?

As seen on previous slide, every time an applications sends a request to the CPU for processing, the query always has to wait. This waiting can be tracked by wait statistics. (Generally this wait time is 4 ms). The best way to demonstrate this wait process is via the diagram below. As you can see, there are three states that a query can be in: Running, Suspended, and Runnable



1. When a query is being processed by the CPU it is said to be in a **running** state. (Ideal position)
2. If there is a lock on the query or a missing resource the query needs, then the query moves to a state **suspended** and moves to the queue for waiting for the resource to be obtained
3. When the missing resource the query needs is provided (lock is removed) it goes into **runnable** state, and the entire cycle repeats for query and all other queries

DMVs to view the wait statistics

If the query is forced to wait for a resource, SQL Server records that wait stats and wait types. There are many wait types and the following link will help you understand the types:

<https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-os-wait-stats-transact-sql>

We can view the wait statistics by examining the following DMVs:

- `Select * from sys.dm_os_wait_stats`
- `Select * from sys.dm_os_waiting_tasks`
- `Select * from sys.dm_exec_requests`

Wait types

Commonly encountered wait types include:

ASYNC_NETWORK_IO

SQL Server holds data in the output buffer until it receives an acknowledgement from the client that it has finished consuming that data. ASYNC_NETWORK_IO is an indication that your client application is not able to efficiently retrieve the data it needs from the system. You may also see this wait if there are fundamental issues with your network, which can lead to long waits while the data is processed by the client and then signals returned back to the server.

CXPACKET

This wait type is involved in parallel query execution, and indicates that the SPID is waiting on a parallel process to complete or start.

OLEDB

Object Linking and Embedding Database (OLEDB) is a mechanism used within SQL Server to connect to other database resources. OLEDB waits indicate that SQL Server has been making calls to remote OLEDB resources from within the system. Waits may be affected by any issues on the remote resource. The wait itself defines the length of time that the OLEDB call is taking to return.

PAGEIOLATCH_EX

A query is waiting for exclusive write access to a page in order to add data to the page, but the page is not currently in memory and has to be loaded from disk.