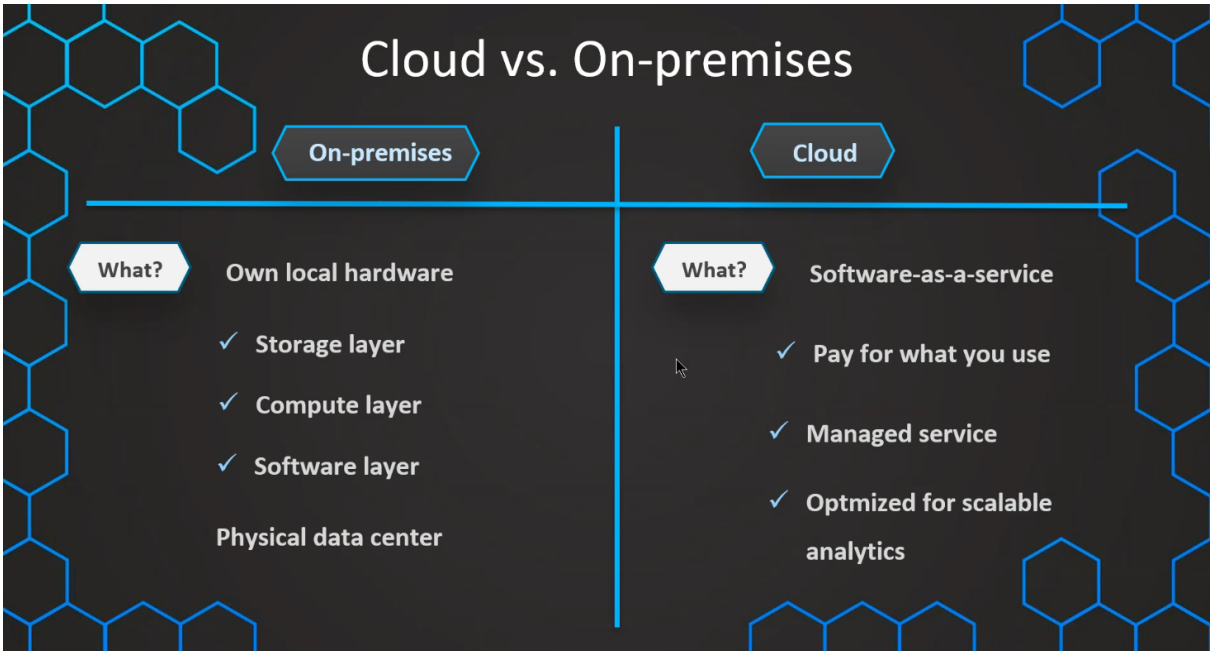
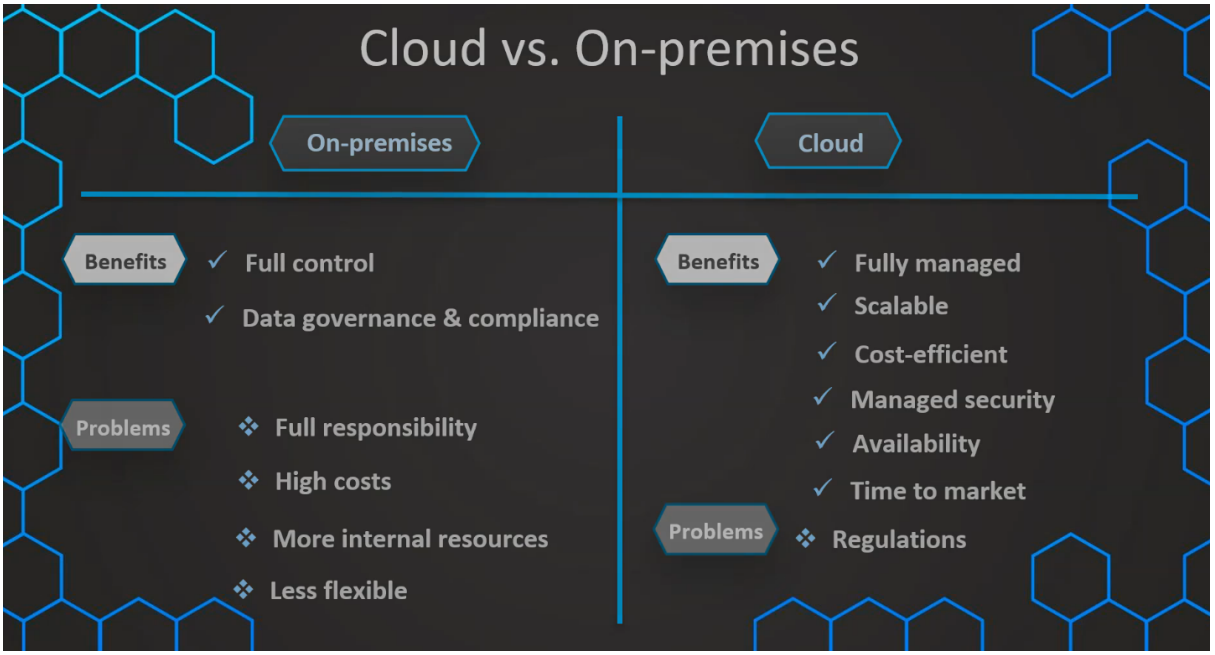
14. The Modern Data Warehouse

Cloud vs on premise

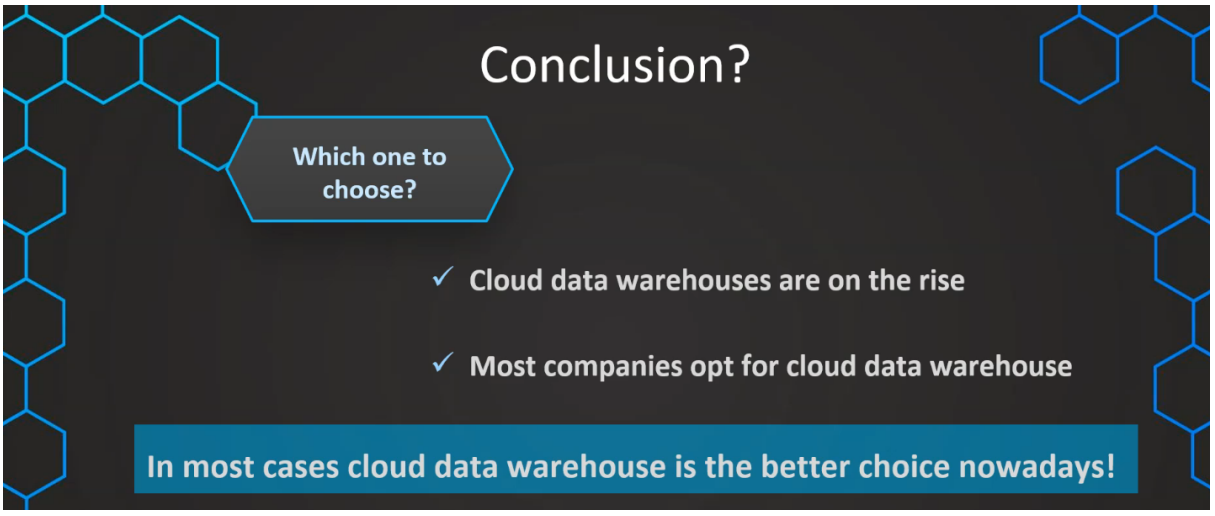


- In cloud data warehouses we don't own any of the infrastructure, but we just use software as a service.
- This means that we are not owning the infrastructure, but we are just paying for using the infrastructure.
- And one benefit is that this is fully managed so we don't need to install anything. We don't need to update anything. Don't worry about any of the administrational works, but this was just fully managed.
- on-premise data warehouse is usually also a data center that is used for other purposes and it may not be so much optimized with these new technologies that are optimized for analytical needs.

Benefits of On-premises vs Cloud



- Full control - They own all of the data infrastructure, they can decide to add additional resources, and how they manage this and this is usually also helping them in fulfilling some of the data governance regulations, or compliance regulations
- Less flexible - if we need more data storage or compute we need more time to get this up and running, on the other hand if we want to reduce the storage and compute we might not be able to do so quickly

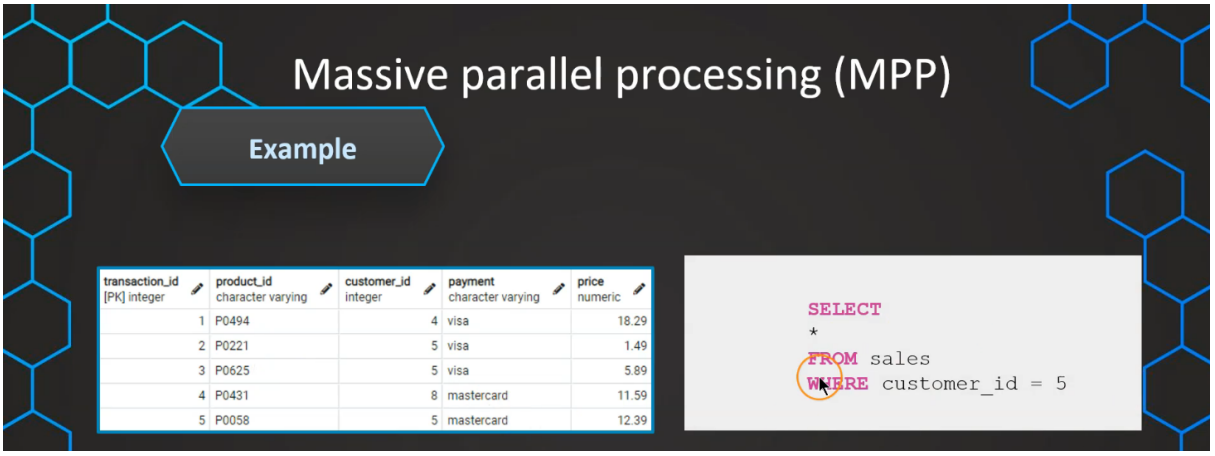


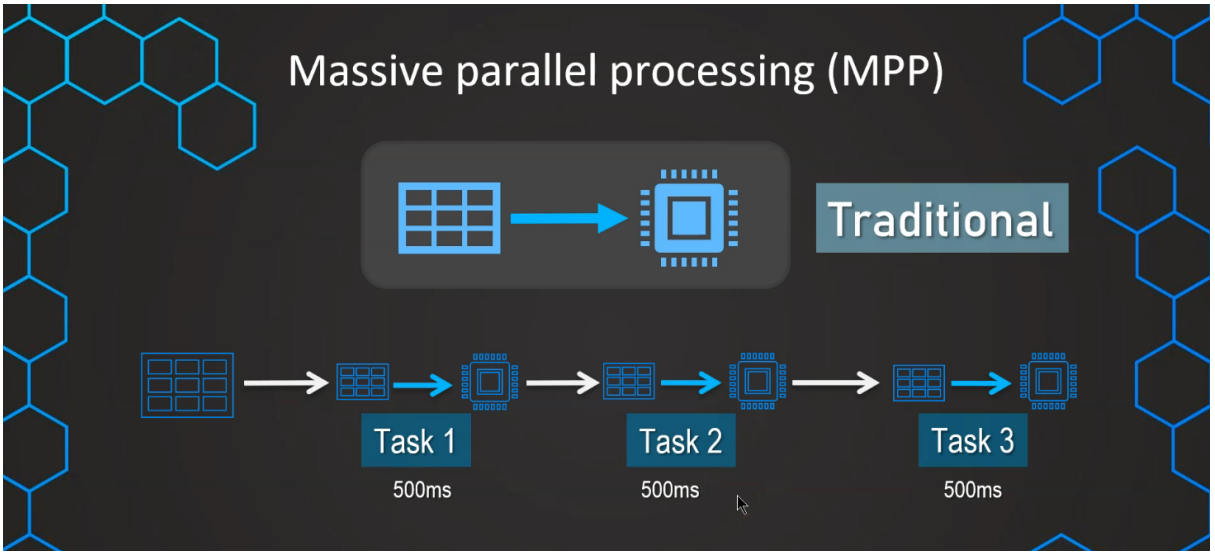
Massive parallel processing (MPP)

- We all know that the amount of data is increasing and increasing in the companies, and also there are more and more users that need to process

this data and with that, the challenge and the requirements for our database system and our query performance are increasing as well.

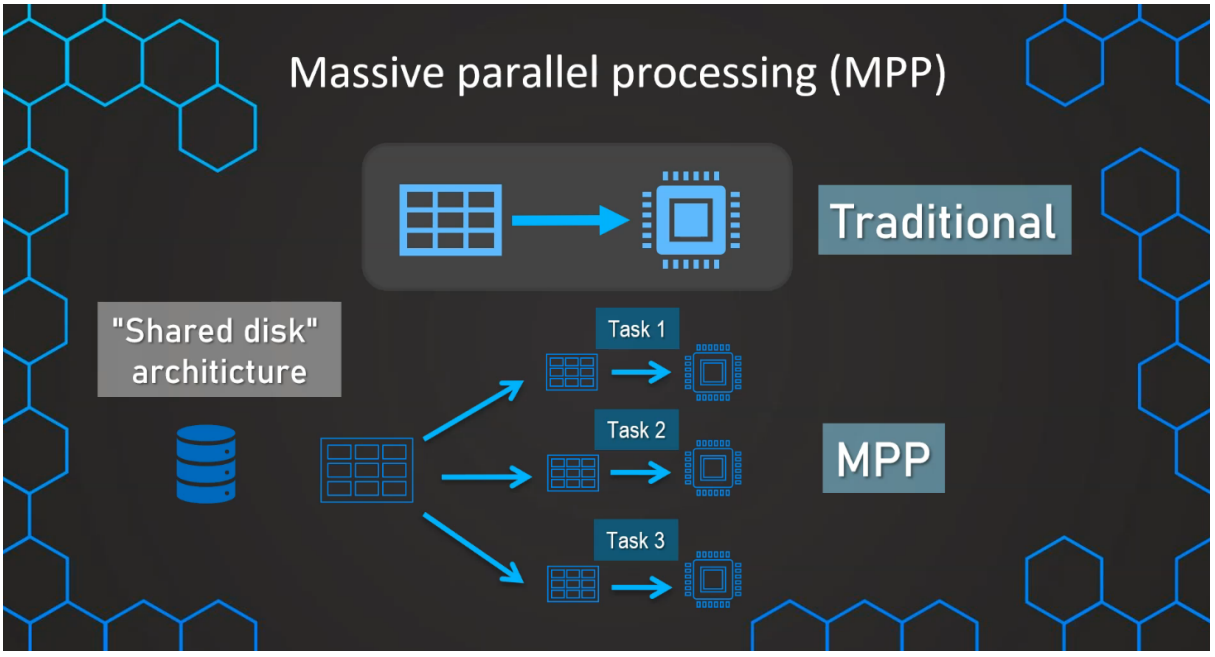
- And one of the most important modern technologies that is used to combat these challenges is so-called massive parallel processing (MPP)
- So to understand that, we want to have a look at a simple example. Let's assume we have some table and we want to get this table returned. So we just run a query against our database system.
- And then this is basically a task that needs to be processed by the database.



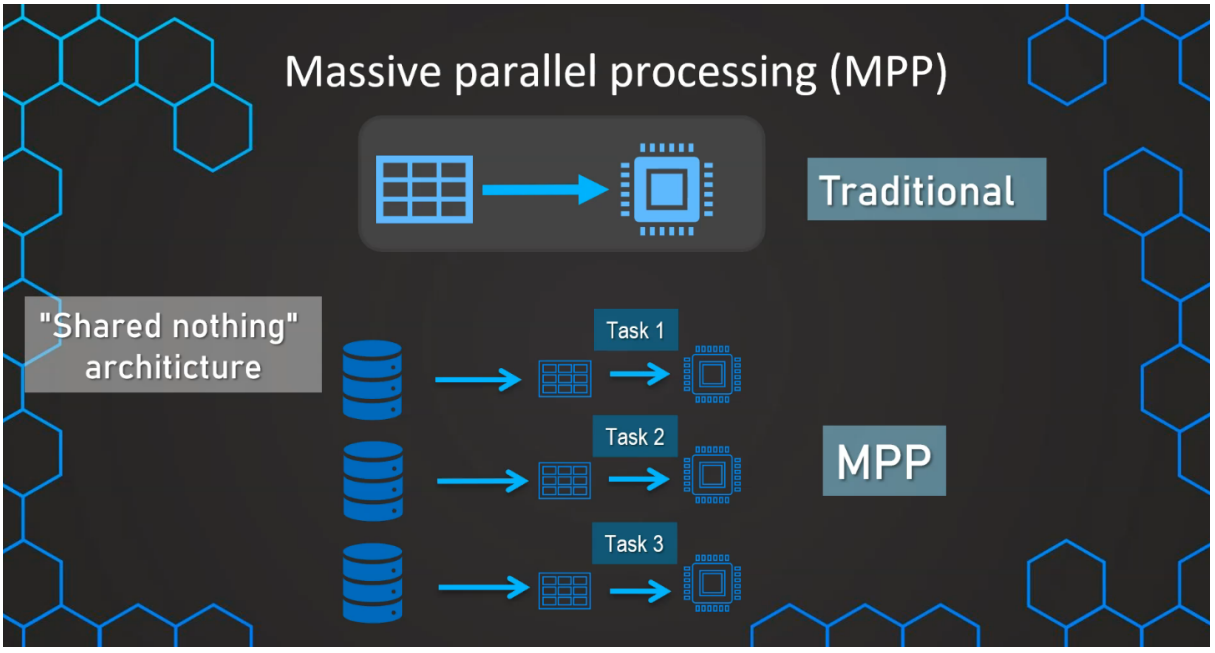


- In traditional way We have this task and this task consists usually of multiple sub-tasks. And in the traditional way, this will be processed all in a sequential order.

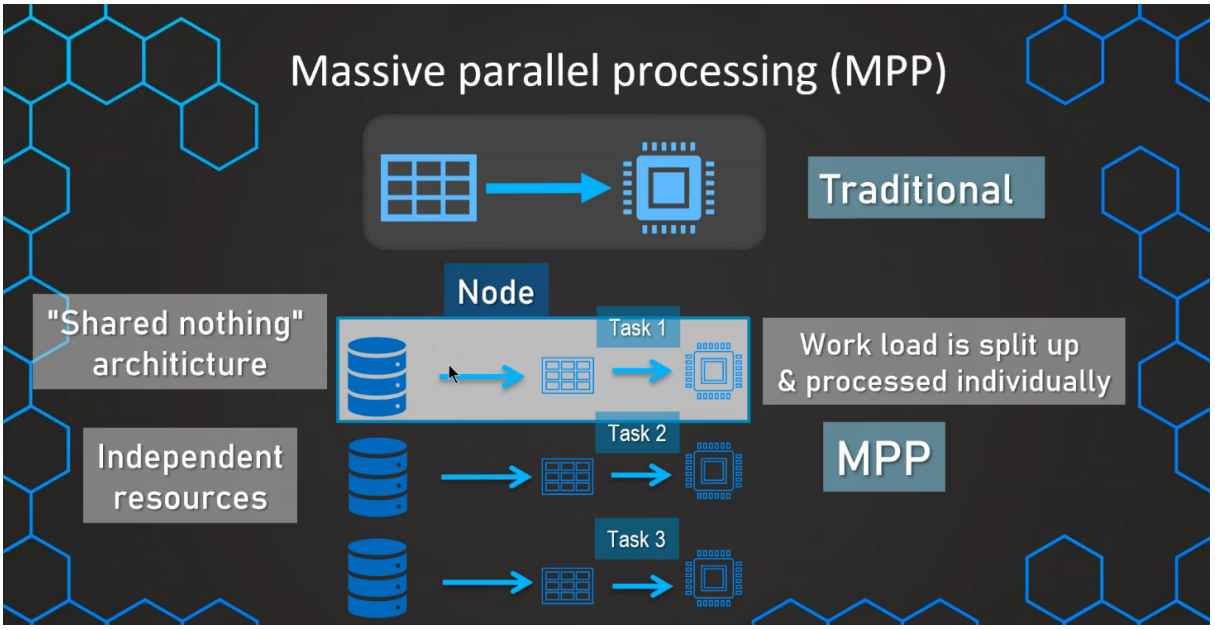
- So in this case, we need to wait until we get the result for all of these single individual tasks to be executed. And then just in the end, we get our result.
- But now with the massive parallel processing, all of these subtasks can be processed in parallel. And with that, we don't need to wait until all of the individual tasks are completed, but they can be all started at the same time.



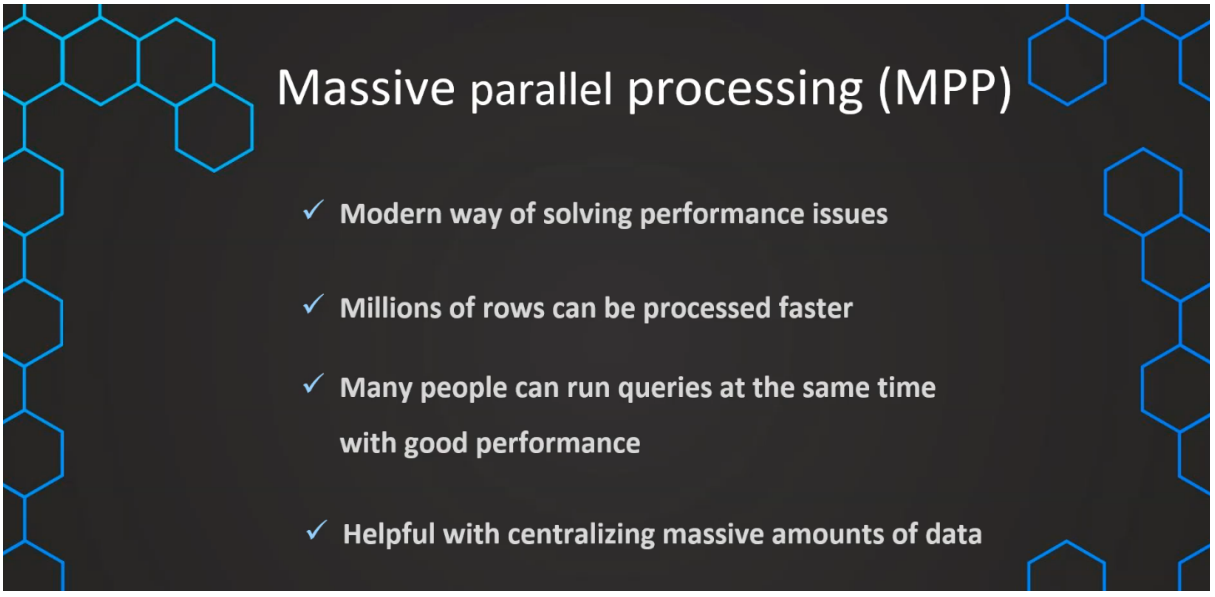
- And with that, we have of course, a much better query performance.
- So the query will be returned much faster. And those nodes, they work independently. It is possible in a so-called shared disk architecture that still we have one external disk, so where the data is stored, and this is just one central location still. And then just the task will be broken down into the subtasks.
- And just the compute resources are now not shared, but they are completely independent to process all of these subtasks in parallel.



- But aside to that, we also have the so-called shared nothing architecture. In that architecture, not only the compute resources are completely individual, but also the data storage is now also distributed to multiple data storage layers.
- So this can be, for example, as three buckets. And then in one location we have one data row. And in another location we have now another row. So they can be spread out across multiple storages. And all of those storages now also work independently.
- They all have their own compute resource, they all have their own operating system, and we call them one node. So they don't share anything. They don't share the storage and they don't share the compute resources. So they work completely independently. So you can think of them as an individual computer



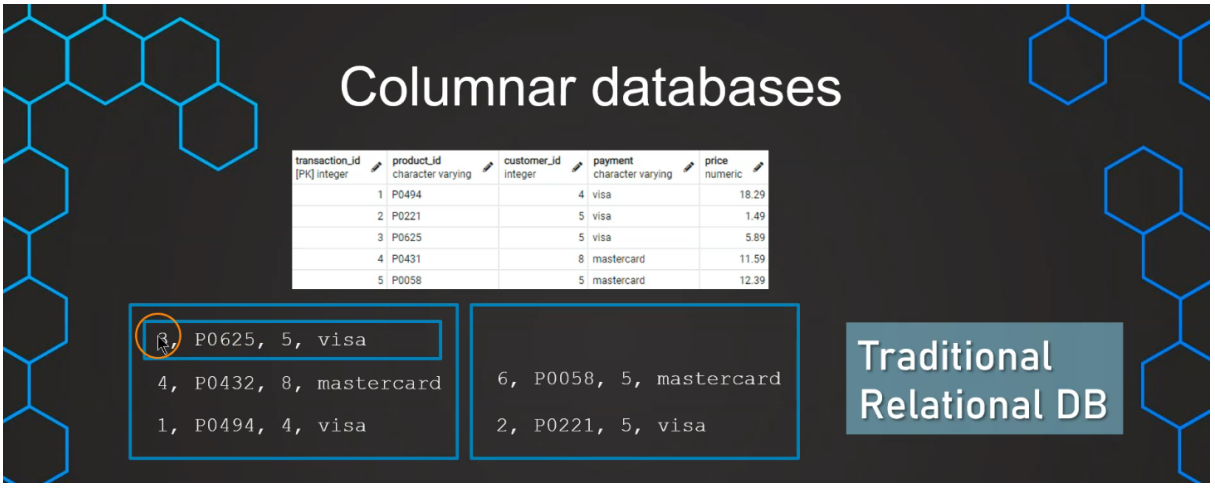
- And this is now, of course, very efficient because with that, the workload can be split up very effectively and being processed individually. And of course, between those nodes we have a high-speed connection. So that, of course, all of those nodes can work on parallel, but of course, they work on the same task. So we need to have a connection between them.
- And with these individual nodes, we can now benefit a lot because all of that work is processed in parallel



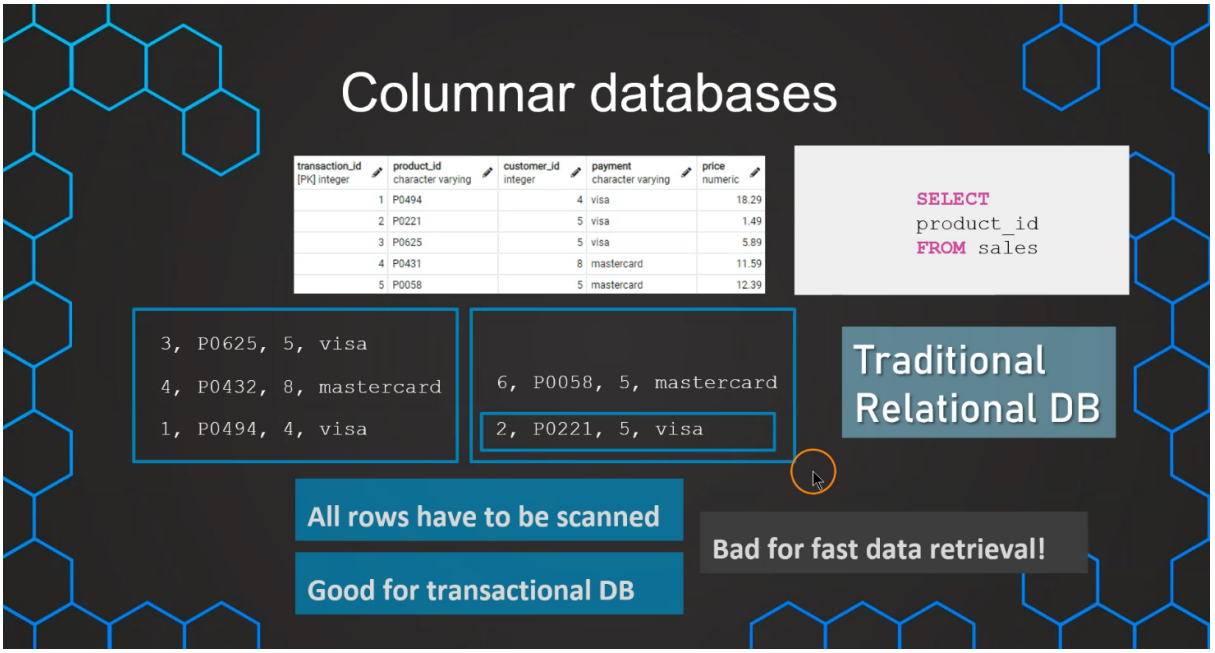
- So this massive parallel processing is now a very modern technology and an answer to solving those performance issues that come with large amounts of data.

- So with that, we can now process millions of rows much faster because we run those tasks all in parallel. And also, if we have many users that all need to wait until the queries are returned, we can now also process them in parallel which will also ensure a good query performance even if we have many people running queries against our database system.
- And with that, we have now a solution that is helping us with centralizing massive amounts of data where many users need to also query from this data.
- So this is a very important technology that is used nowadays by many cloud data warehouse providers such as, for example, Snowflake.
- So this is one of the important modern technologies that is used to ensure a high query performance. A second one is so-called columnar data storage. This is also a technology that is optimized for our data warehouse

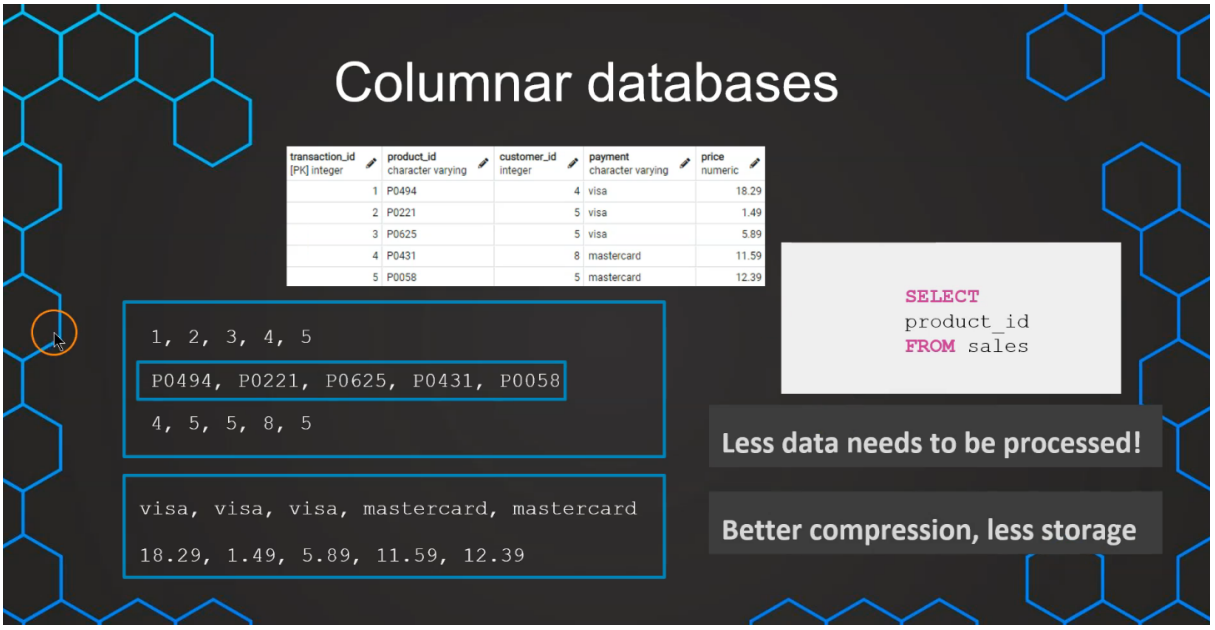
Columnar Storage



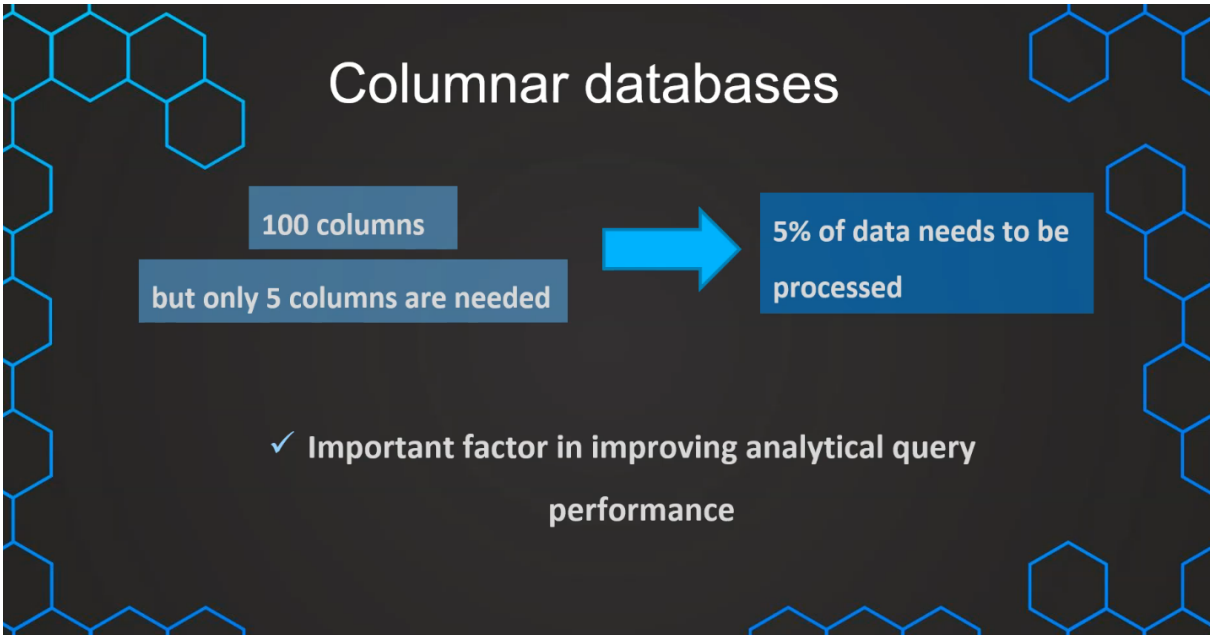
- If we have a very nice looking table with columns and rows, internally, the data is stored in blocks and it's stored row-based. So in one block there is one row.
- So we have, for example, in this case, the first column value, then we have in the second one, we have this value of the second column.
- And this is now, of course, efficient if we need to work on a transactional basis because then we need to, of course, update all of the columns in one row.



- But if we now want to retrieve the data and use it for analytical purposes, we usually have queries where we only need to process one or a few columns. And with that, now, if we use the same way of storing the data, we now need to scan again through all of the different rows to get the result.
- And now also all of them have to be sent to the memory and have to be processed. And with that, the amount of data, even if we just want to query a subset of it, is way too much, and this is really not efficient for us.
- As mentioned, of course, this is very good if we have transactional data processing where we with one data entry just need to enter values for all of the columns for one single transaction.
- So for this transactional data processing, it is very efficient. But for data retrieval, especially fast data retrieval, this is really not the best way.
- And this is why we have also so-called columnar storage that is now optimized for fast query performance when we want to retrieve the data, because now the data is again stored in these individual blocks, but now in these blocks, the data is not stored row-based, but we have now in these blocks column-stored.



- So here, for example, in the first one, we have the column transaction_id, and the second one the product_id, and so on.
- So this means if we now want to execute this select command with one column, we just need to get this single block, and we don't need to scan through all of that data and send all of that data to memory and process a lot of data.
- But we can just go with this single block in here. And now with the lower amount of data, it can be processed also much more efficiently and much faster.
- And on top of that, an additional benefit is that now that we have the data stored column-wise, the data type that is stored in one block is also just the same.
- So we have only one data type. And this makes it possible to also compress the data much more efficiently because we can choose a sufficient encoding that is specifically good for one data type. And with that, again, the storage needed is less. And again, the processing is faster with that. So this is an additional benefit.



- And this means now in practice that if we have, for example, a table with let's say 100 columns, but we only want to query five of them, so this is of course much more common if we want to analyze the data to see in one moment only a few columns, then we don't need to process all of those 100 columns, but only 5% of them.
- And with that, we have, of course, now much less data that needs to be processed. And this is now much faster and much more efficient if we want to improve our analytical query performance.
- That's why in the modern data warehouses, especially in the cloud data warehouses, they use these columnar databases.
- And nowadays this is also a very important factor in terms of our query performance.