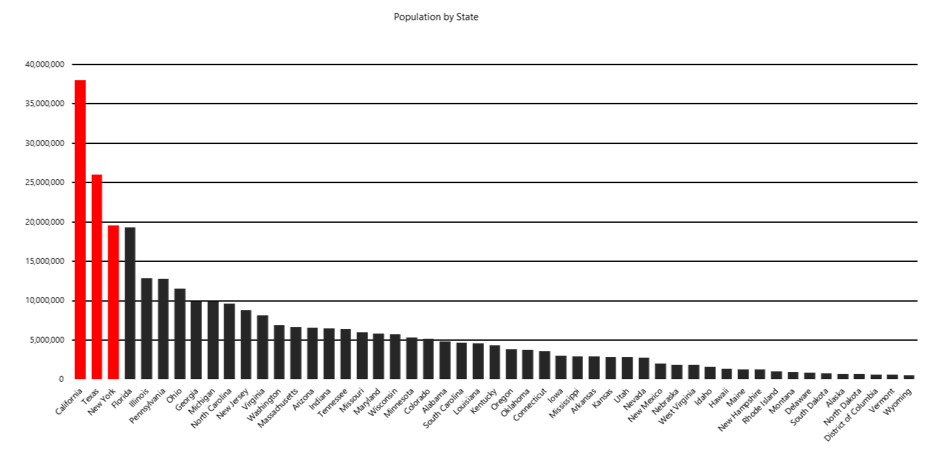
We are going to discuess about below concept:

* Check for skewed data and space usage
* Understand column store storage details
* Study the impact of materialized views
* Explore rules for minimally logged operations

**Understand skewed data and space usage**

In simple terms, data skew is an over-represented value. Imagine that you have assigned 50 tax examiners to audit tax returns, one examiner for each US state. The Wyoming examiner, because the population there is small, has little to do. In California, however, the examiner is kept very busy because of the state's large population.



In our scenario, the data is unevenly distributed across all tax examiners, which means that some examiners must work more than others. In your job, you frequently experience situations like the tax-examiner example here. In more technical terms, one vertex gets much more data than its peers, a condition that makes the vertex work more than the others and that eventually slows down an entire job. What's worse, the job might fail because vertices might have, for example, a 5-hour runtime limitation and a 6-GB memory limitation.

A quick way to check for data skew is to use DBCC PDW\_SHOWSPACEUSED. The following SQL code returns the number of table rows that are stored in each of the 60 distributions. For balanced performance, the rows in your distributed table should be spread evenly across all the distributions.

-- Find data skew for a distributed table

DBCC PDW\_SHOWSPACEUSED('dbo.FactInternetSales');

Another aspect of data storage in Azure Synapse dedicated SQL pools is to monitor the table data space usage and observe its relationship with different table distribution types. Additionally, it is helpful to know the number of rows and the storage space used for indexing. Below is a list of System Dynamic Management Views (DMVs) that you can use to dig for the information. During the next exercise you will create a view using these DMVs to get a better view of the data.

|  |  |
| --- | --- |
| **Table Name** | **Description** |
| sys.schemas | All schemas in the database. |
| sys.tables | All tables in the database. |
| sys.indexes | All indexes in the database. |
| sys.columns | All columns in the database. |
| sys.pdw\_table\_mappings | Maps each table to local tables on physical nodes and distributions. |
| sys.pdw\_nodes\_tables | Contains information on each local table in each distribution. |
| sys.pdw\_table\_distribution\_properties | Holds distribution information for tables (the type of distribution tables have). |
| sys.pdw\_column\_distribution\_properties | Holds distribution information for columns. Filtered to include only columns used to distribute their parent tables (distribution\_ordinal = 1). |
| sys.pdw\_distributions | Holds information about the distributions from the SQL pool. |
| sys.dm\_pdw\_nodes | Holds information about the nodes from the SQL pool. Filtered to include only compute nodes (type = COMPUTE). |
| sys.dm\_pdw\_nodes\_db\_partition\_stats | Returns page and row-count information for every partition in the current database. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |