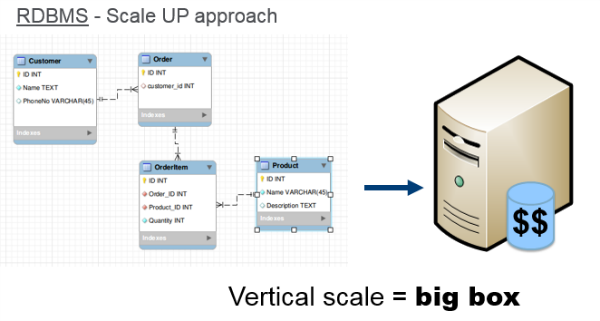
**Hbase Explained .**

**Relational Databases vs. HBase – Data Storage Model**

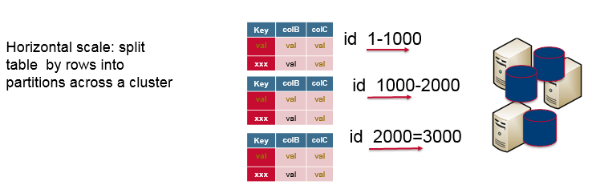
Why do we need NoSQL/HBase? First, let’s look at the pros of relational databases before we discuss its limitations:

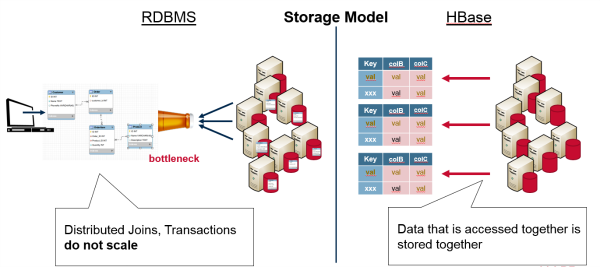
* Relational databases have provided a standard persistence model
* [SQL](https://mapr.com/why-hadoop/sql-hadoop/sql-hadoop-details) has become a de-facto standard model of data manipulation (SQL)
* Relational databases manage concurrency for transactions
* Relational database have lots of tools

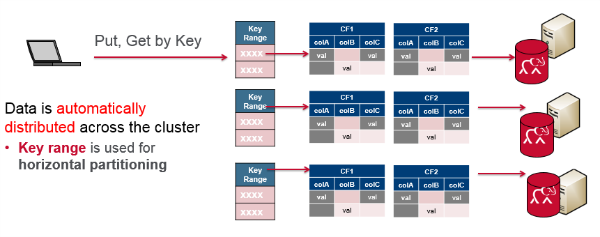


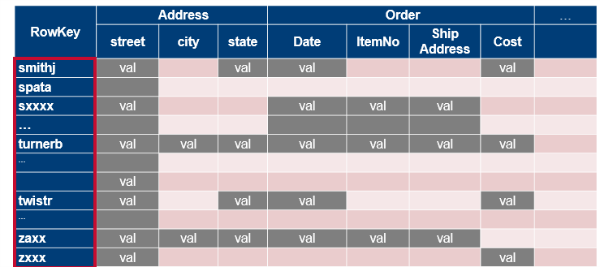
**What changed to bring on NoSQL?**

An alternative to vertical scaling is to scale horizontally with a cluster of machines, which can use commodity hardware. This can be cheaper and more reliable. To horizontally partition or shard a RDBMS, data is distributed on the basis of rows, with some rows residing on a single machine and the other rows residing on other machines, However, it’s complicated to partition or shard a relational database, and it was not designed to do this automatically. In addition, you lose the querying, transactions, and consistency controls across shards. Relational databases were designed for a single node; they were not designed to be run on clusters.









Physically data is stored per column like below Although logically its like above. But to save space for sparsely populated columns below works :

