

## Motivation

### Data warehouse

Decision support tool for business intelligence applications

#### Traditional pattern

- Stable
- Fixed
- Inflexible

#### Modern pattern

- Dynamic
- Unexpected
- Schema-comes-second



### How schema design affect different quality measures ?

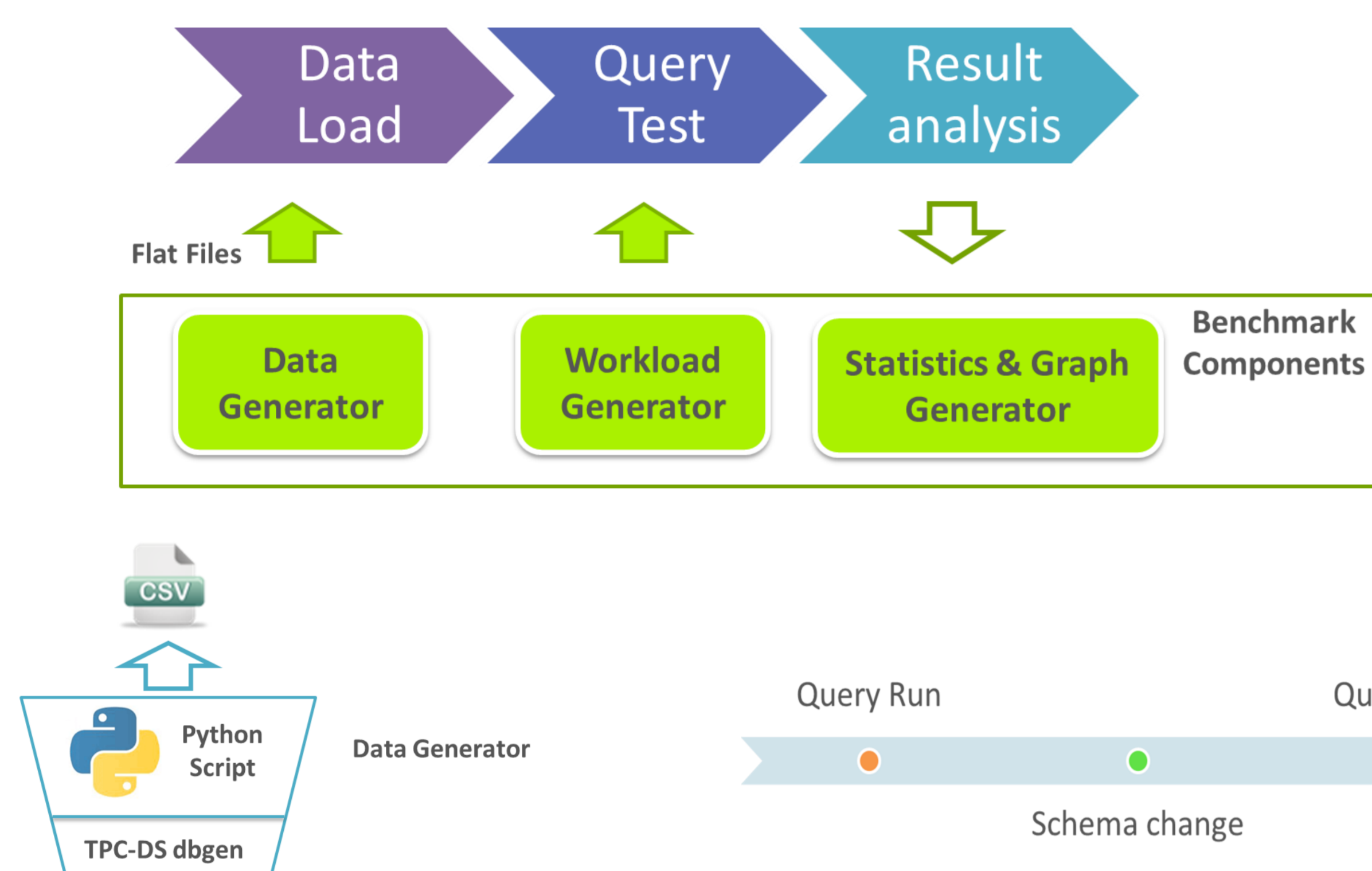
- Query execution time
- Storage efficiency
- Understandability

### Requirements for schema-flexible Benchmark

- Compare different design alternatives
- Cover Typical schema change scenarios
- Compatibility with DBMS from different vendors
- Automatically rewrite queries

## A Schema-aware Benchmark

### Benchmark Execution Order

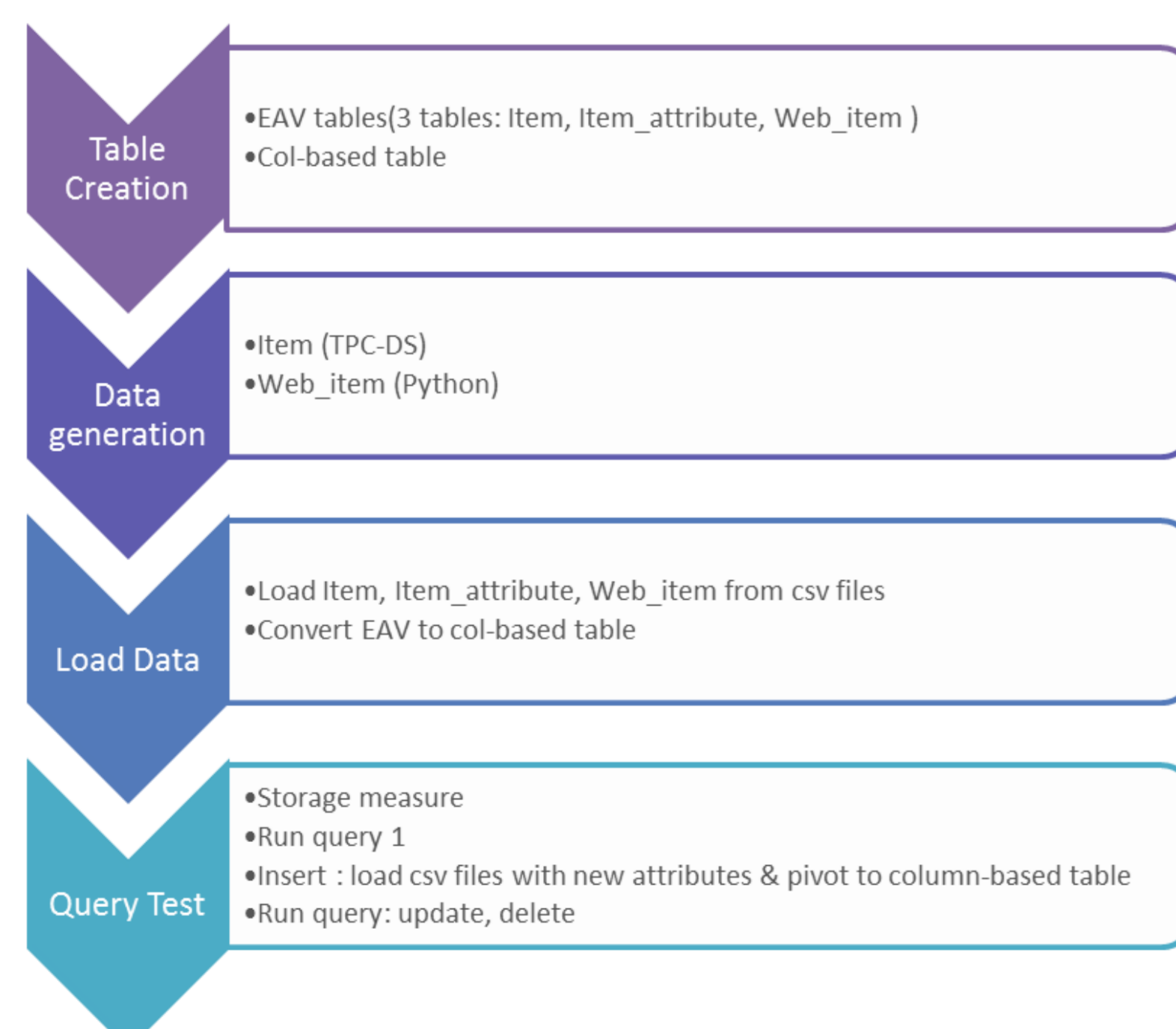


### Schema change scenarios

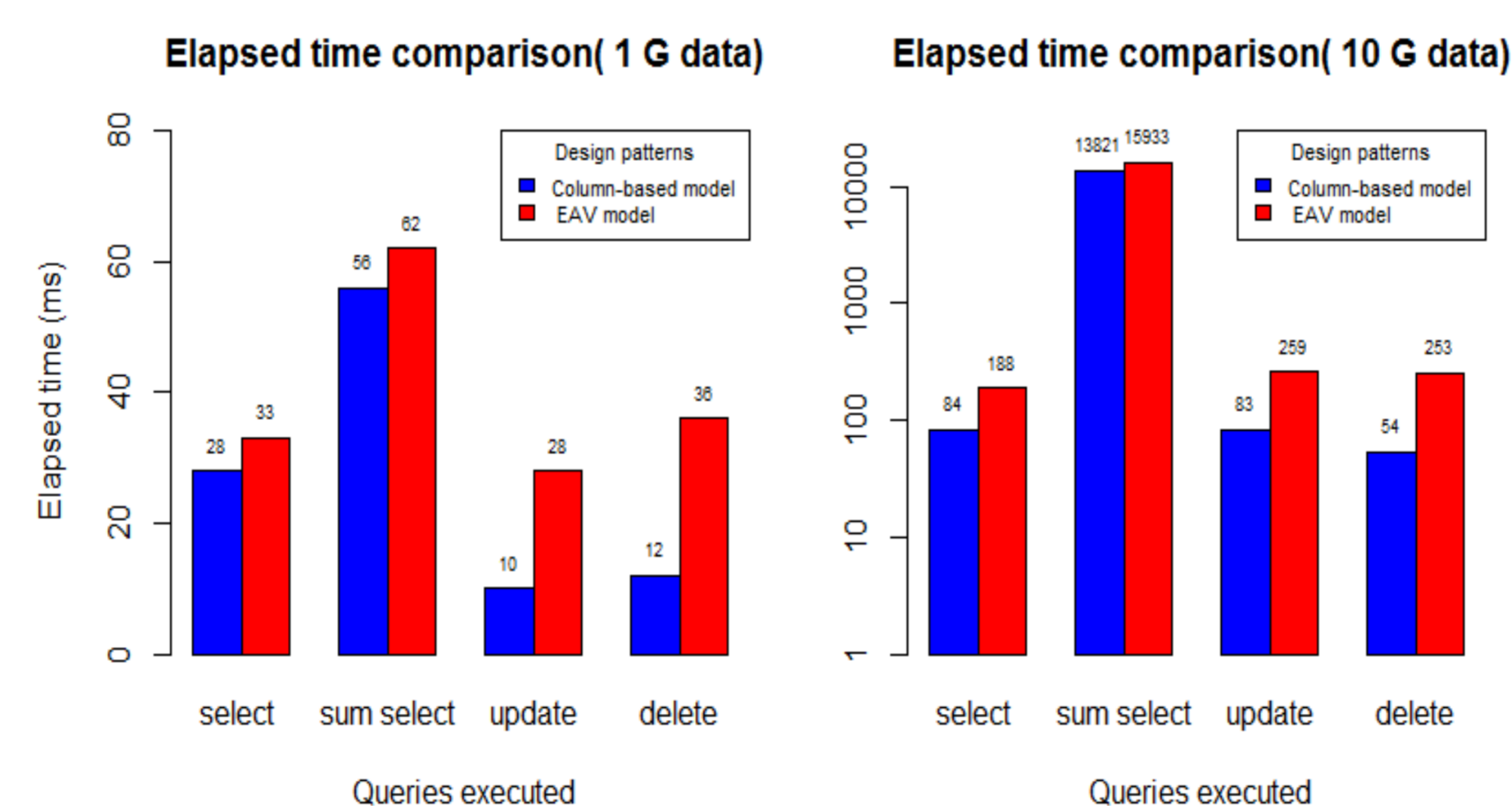
Dimension changes	Scenarios
Fast changing dimensions	Dimension attributes involve fast and in multiple rows
EAV tables	Attributes schemas are highly variable
Bridge tables	Traverse an unbalanced hierarchy (navigate up/down the hierarchy)
Ragged dimensions	Branch levels have inconsistent depth
Slowly changing dimensions	Dimension attributes involve slowly over time
Snowflaking	Low-cardinality or text attributes need to be moved separated normalized tables
Garbage dimension	Add new low-cardinality attributes to the schema
Hot swappable dimensions	The dimension needs to be swapped between multiple alternate versions during query time

## Evaluation on Microsoft SQL Server

### Implementation Process



### Elapsed time comparison



### Storage efficiency comparison

	1G		10G	
	Column-based	Row-based (EAV)	Column-based	Row-based (EAV)
Data space(MB)	0.695	3.758	5.352	21.125
Row count	18000	99539	102000	560615
Index space(MB)	0.016	0.031	0.031	0.109
<b>Before</b>				
<hr/>				
<b>After</b>				
	Column-based	Row-based (EAV)	Column-based	Row-based (EAV)
Data space(MB)	4.117	9.094	21.367	49.516
Row count	18000	135767	102000	764937
Index space(MB)	0.016	0.047	0.070	0.211

\*Partial data from TPC-DS (1G,10G)

Decisions and Speedup changes for different scale factors

