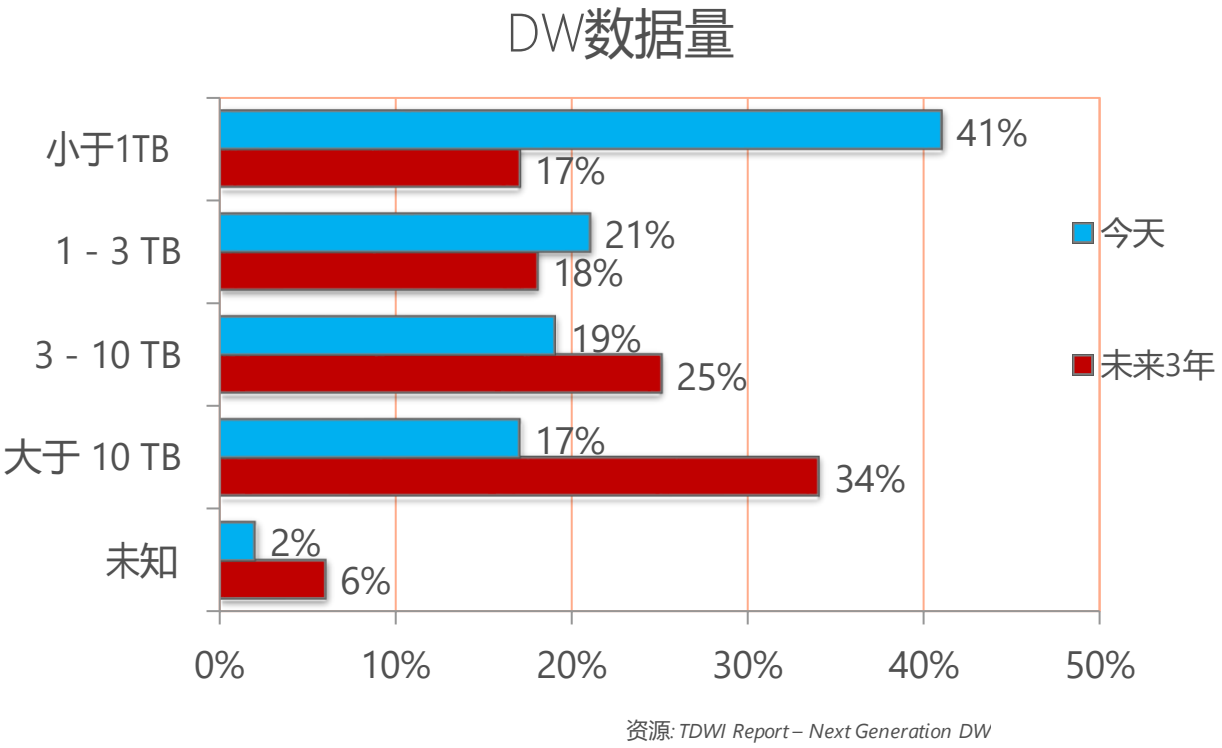
SQL Server 列存储索引

01 | 商业推动理由

Microsoft

数据仓库使用空间



资源: *TDWI Report – Next Generation DW*

- **规模更大**: 数据持续增长，弹性扩充是考量的主要关键10s of TBs, to 100s of TB, to PBs。
- **性能**: 分析大量数据的能力。
- **大众化**: 降低每TB的价格。

列存储索引是为解决以上需求所设计.

02 | 列存储体系结构

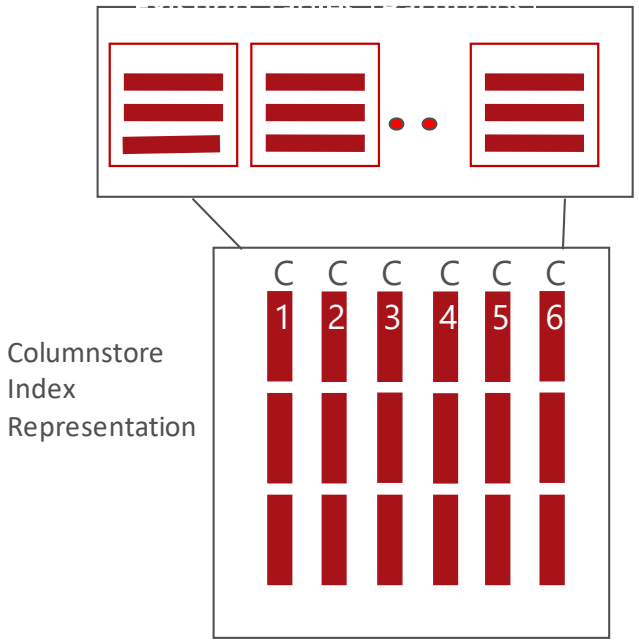
Microsoft

In-Memory 数据仓库
数据以行存储: 堆, b-树

- In-Memory列存储
- 内存和硬盘都有
- 与RDBMS 引擎集成
- 客户收益:
  - 10-100倍更快
  - 减少了设计花费
  - 在原有的硬件上即可使用
  - 方便升级，方便部署

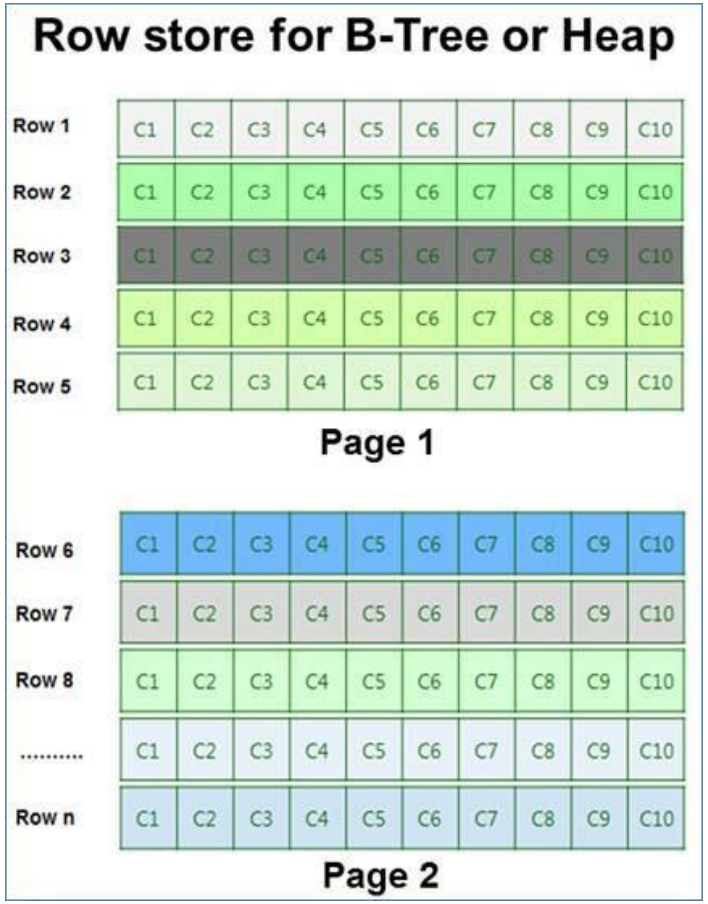
“通过使用SQL Server 2012 列存储索引, 我们可以在**2 到 3** 秒内获取100百万行记录，原来要**30分钟** “

- Atsuo Nakajima Asst Director, Bank of Nagoya



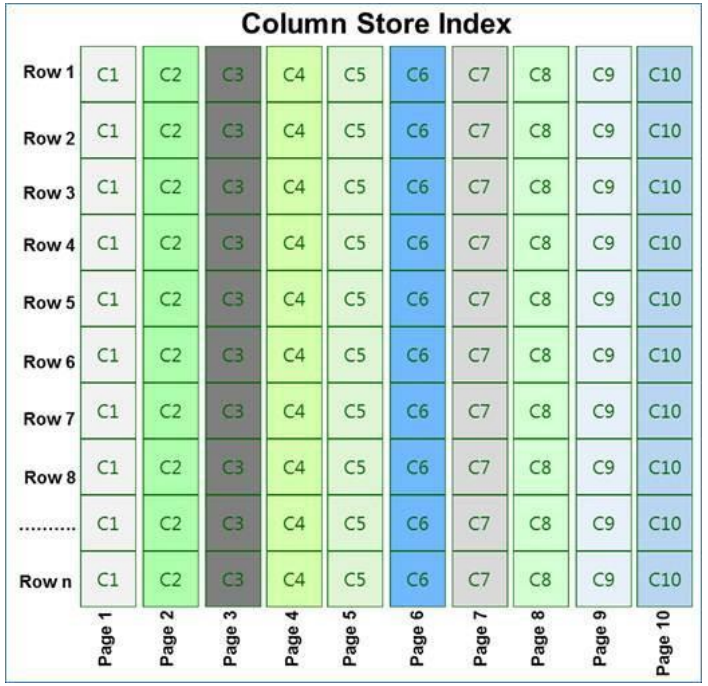
列存储索引好处

数据按行存储

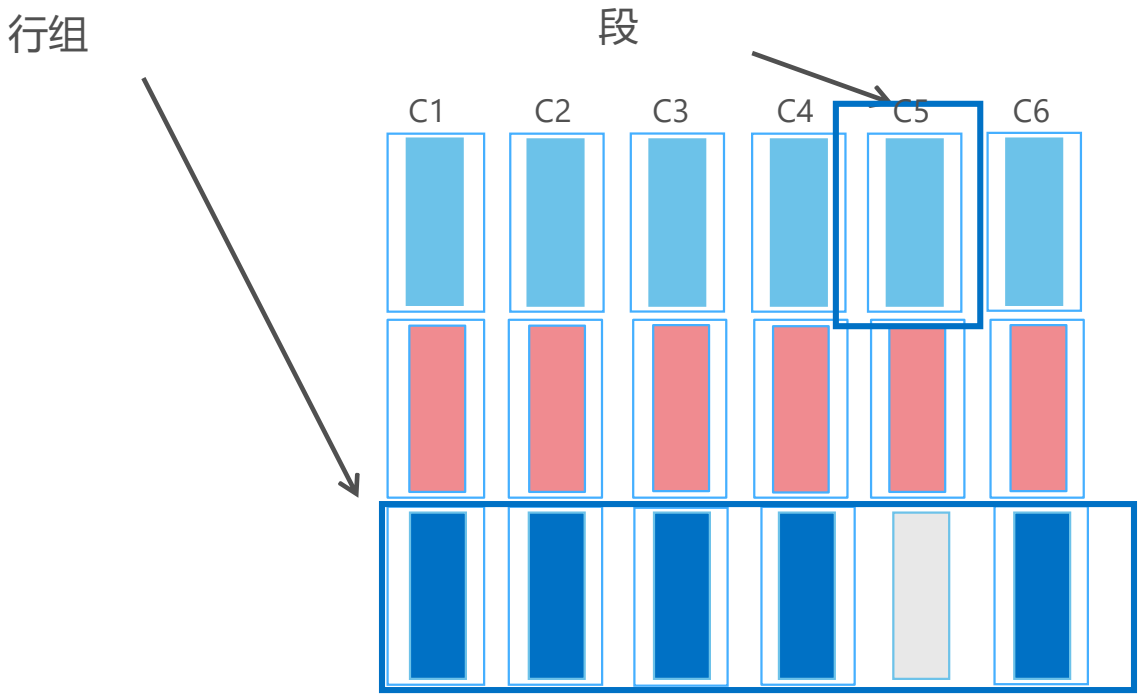


- **提升的压缩:**
  相同类型的数据压缩更好
- **减少的I/O:**
  仅获取所需要的列
- **提升的性能:**
  - 多数数据放在内存中
  - 为CPU的使用率优化

数据按列存储



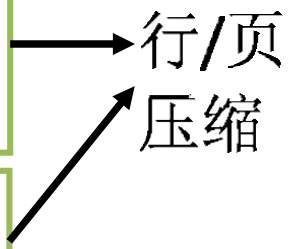
列存储索引: 术语



- 行组
  - 每个组行(通常100万行)
- 段
  - 段是来自行组内的数据列
- 每个段一起压缩并且存储于物理介质上
- 段是从磁盘和内存中传输的单元

列存储索引：例子

OrderDateKey	ProductKey	StoreKey	RegionKey	Quantity	SalesAmount
20101107	106	01	1	6	30.00
20101107	103	04	2	1	17.00
20101107	109	04	2	2	20.00
20101107	103	03	2	1	17.00
20101107	106	05	3	4	20.00
20101108	106	02	1	5	25.00
20101108	102	02	1	1	14.00
20101108	106	03	2	5	25.00
20101108	109	01	1	1	10.00
20101109	106	04	2	4	20.00
20101109	106	04	2	5	25.00
20101109	103	01	1	1	17.00



第1部: 水平分区(创建行组)

OrderDateKey	ProductKey	StoreKey	RegionKey	Quantity	SalesAmount
20101107	106	01	1	6	30.00
20101107	103	04	2	1	17.00
20101107	109	04	2	2	20.00
20101107	103	03	2	1	17.00
20101107	106	05	3	4	20.00
20101108	106	02	1	5	25.00

~1M 行

OrderDateKey	ProductKey	StoreKey	RegionKey	Quantity	SalesAmount
20101108	102	02	1	1	14.00
20101108	106	03	2	5	25.00
20101108	109	01	1	1	10.00
20101109	106	04	2	4	20.00
20101109	106	04	2	5	25.00
20101109	103	01	1	1	17.00

第2步: 垂直分区(创建段)

OrderDateKey	ProductKey	StoreKey	RegionKey	Quantity	SalesAmount
20101107	106	01	1	6	30.00
20101107	103	04	2	1	17.00
20101107	109	04	2	2	20.00
20101107	103	03	2	1	17.00
20101107	106	05	3	4	20.00
20101108	106	02	1	5	25.00

OrderDateKey	ProductKey	StoreKey	RegionKey	Quantity	**SalesAmount**
20101108	102	02	1	1	**14.00**
20101108	106	03	2	5	**25.00**
20101108	109	01	1	1	**10.00**
20101109	106	04	2	4	**20.00**
20101109	106	04	2	5	**25.00**
20101109	103	01	1	1	**17.00**

第3步：压缩每个段

OrderDateKey
20101107
20101107
20101107
20101107
20101107
20101108

ProductKey
106
103
109
103
106
106

StoreKey
01
04
04
03
05
02

RegionKey
1
2
2
2
3
1

Quantity
6
1
2
1
4
5

SalesAmount
30.00
17.00
20.00
17.00
20.00
25.00

OrderDateKey
20101108
20101108
20101108
20101109
20101109
20101109

ProductKey
102
106
109
106
106
103

StoreKey
02
03
01
04
04
01

RegionKey
1
2
1
2
2
1

Quantity
1
5
1
4
5
1

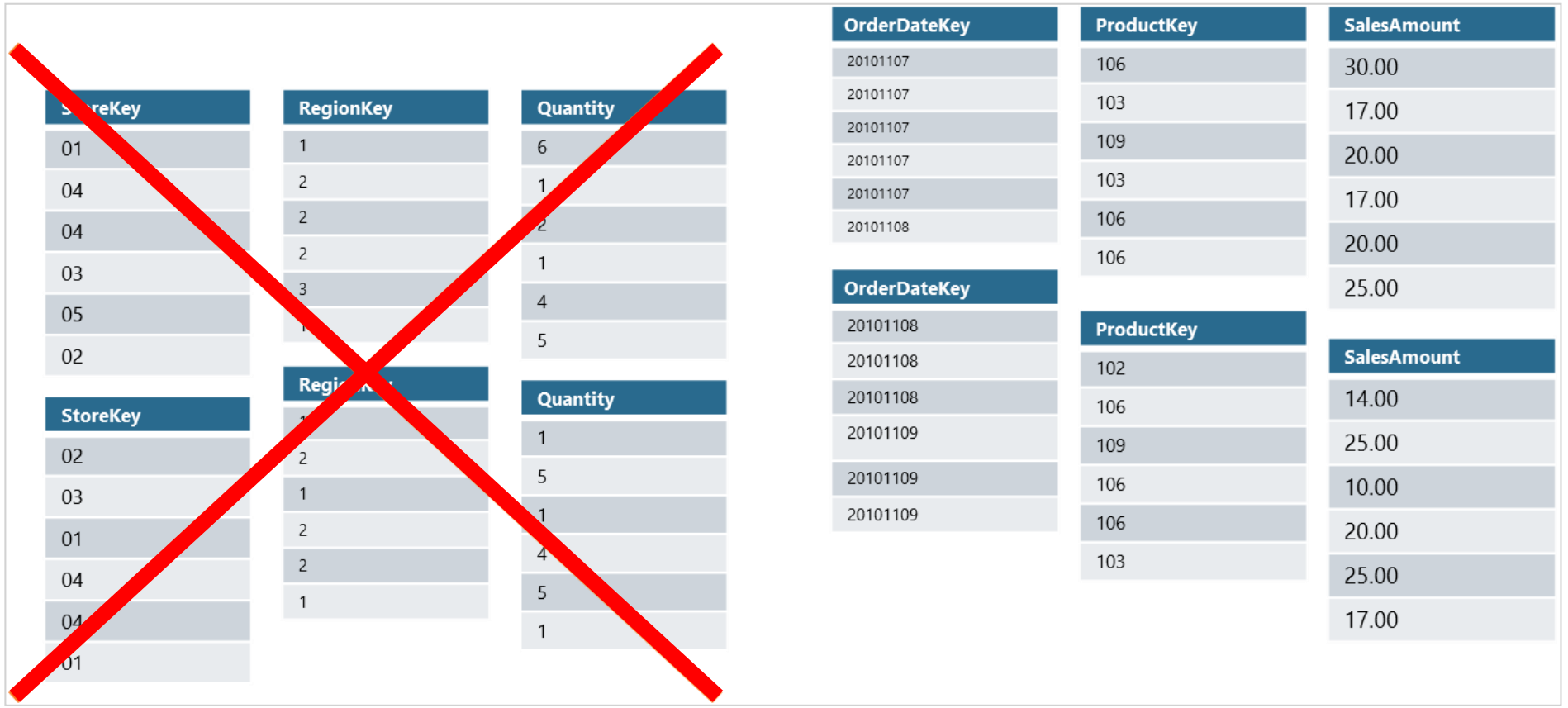
SalesAmount
14.00
25.00
10.00
20.00
25.00
17.00

一些段比其他压缩更多

*编码和重新排序不显示

获取需要的列
淘汰段

```
SELECT ProductKey, SUM (SalesAmount)
FROM SalesTable
WHERE OrderDateKey < 20101108
```



reKey
01
04
04
03
05
02

StoreKey
02
03
01
04
04
01

RegionKey
1
2
2
2
3
1

Regi
2
1
2
2
1

Quantity
6
1
2
1
4
5

Quantity
1
5
1
4
5
1

OrderDateKey
20101107
20101107
20101107
20101107
20101107
20101108

ProductKey
106
103
109
103
106
106

SalesAmount
30.00
17.00
20.00
17.00
20.00
25.00

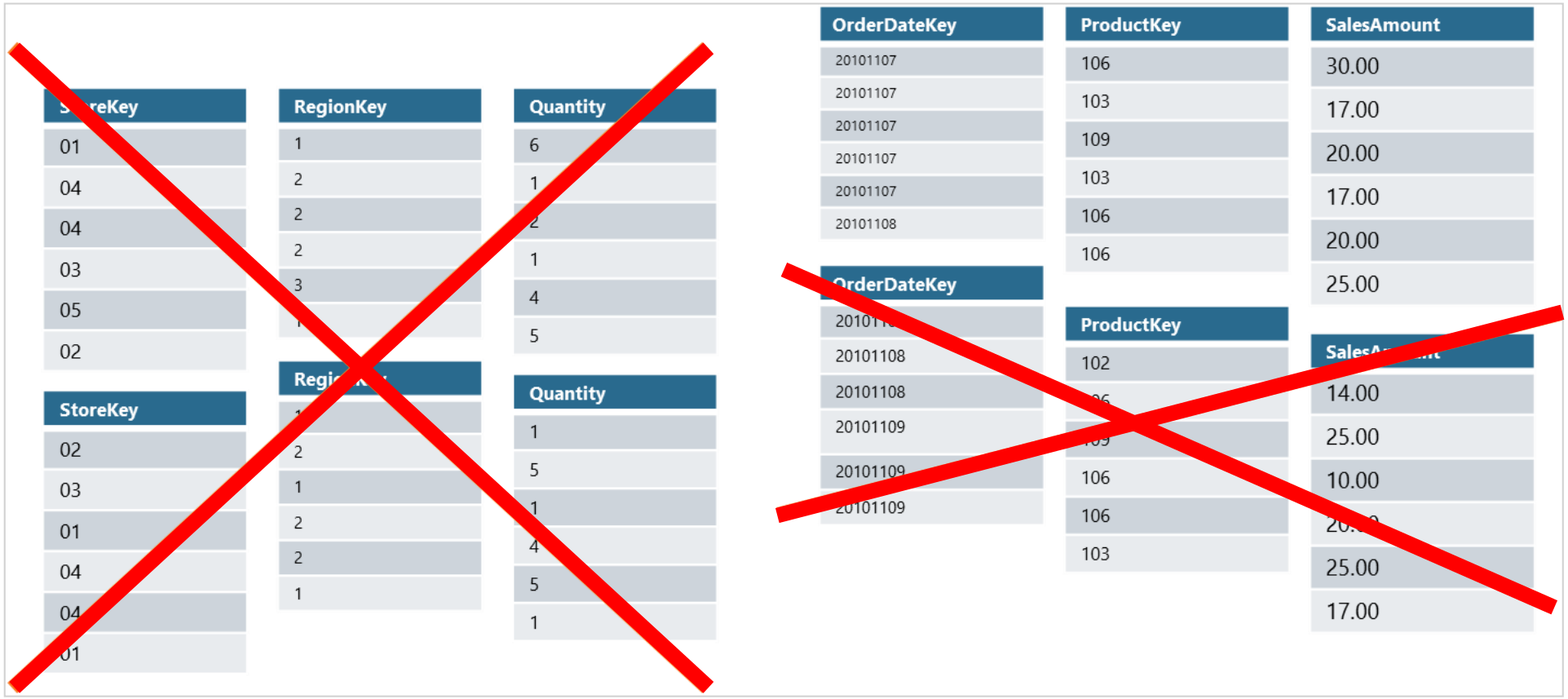
OrderDateKey
20101108
20101108
20101108
20101109
20101109
20101109

ProductKey
102
106
109
106
106
103

SalesAmount
14.00
25.00
10.00
20.00
25.00
17.00

获取需要的列
淘汰段

```
SELECT ProductKey, SUM (SalesAmount)
FROM SalesTable
WHERE OrderDateKey < 20101108
```



reKey
01
04
04
03
05
02

RegionKey
1
2
2
2
3
1

Quantity
6
1
2
1
4
5

StoreKey
02
03
01
04
04
01

Regi
1
2
1
2
2
1

Quantity
1
5
1
4
5
1

OrderDateKey
20101107
20101107
20101107
20101107
20101107
20101108

ProductKey
106
103
109
103
106
106

SalesAmount
30.00
17.00
20.00
17.00
20.00
25.00

OrderDateKey
20101
20101108
20101108
20101109
20101109
20101109

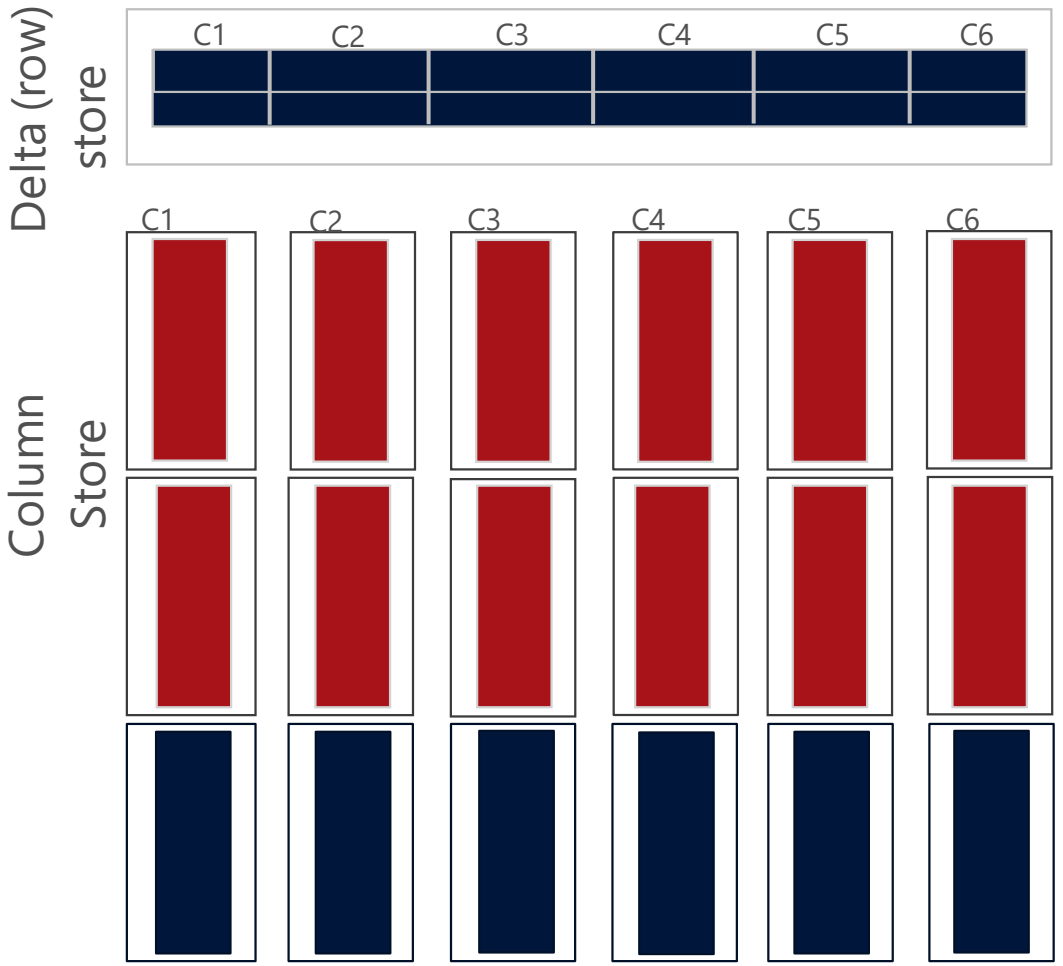
ProductKey
102
106
106
106
103

Sales
14.00
25.00
10.00
20.
25.00
17.00

SQL Server 2014列存储索引: 聚集的& 可更新的

- 快速的执行数据仓库查询
  - 速度提升10倍或更多
- 不再需要保存表
  - 节约了空间
- 可以更新，插入，删除数据
  - 更简单的管理
- 不需要其他索引
  - 节省空间和简化管理
- 支持更多的数据类型

可更新的列存储索引



- 表由列存储和行存储组成 DML
- （更新，删除，插入）操作利用差异（行）存储
- INSERT 值
  - 插入到delta 存储
- DELETE
  - 逻辑操作
  - 物理上的数据移除是在执行REBUILD 操作后执行
- UPDATE
  - 先DELETE 后INSERT.
- BULK INSERT
  - 如果 batch < 100k, 插入到delta 存储, 否则直接到列存储
- SELECT
  - 从列和行存储中联合数据- UNION 操作.

- “Tuple mover” 把数据转换成列格式，一旦段存储满(一百万行)
- REORGANIZE 强制tuple mover 执行

对比空间使用
101 百万行表+ 索引空间