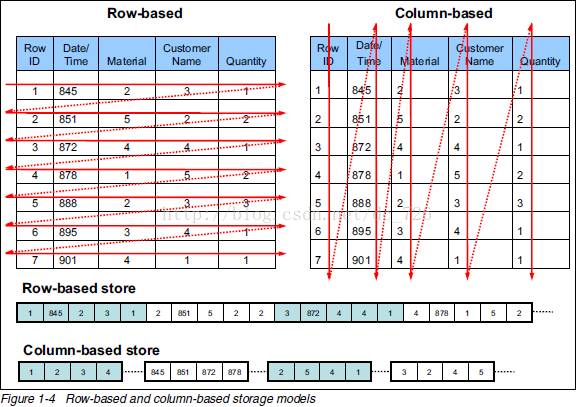
# 定义

列式存储(Column-based)是相对于传统关系型数据库的行式存储(Row-based)来说的。简单来说两者的区别就是如何组织表。

将表放入存储系统中有两种方法，而我们绝大部分是采用行存储的。行存储法是将各行放入连续的物理位置，这很像传统的记录和文件系统。列存储法是将数据按照列存储到数据库中，与行存储类似，下图是两种存储方法的图形化解释。



应用行式存储的数据库系统称为行式数据库，同理应用列式存储的数据库系统称为列式数据库。随着列式数据库的发展，传统的行式数据库加入了列式存储的支持，形成具有两种存储方式的数据库系统。

传统的关系型数据库，如Oracle、DB2、MySQL、SQL SERVER等采用行式存储法，当然传统的关系型数据库也在不断发展中。随着Oracle 12c推出了in memory组件，使得Oracle数据库具有了双模式数据存放方式，从而能够实现对混合类型应用的支持：传统的以行形式保存的数据满足OLTP应用；列形式保存的数据满足以查询为主的OLAP应用。

新兴的Hbase、HP Vertica、EMC Greenplum等分布式数据库采用列式存储，当然这些数据库也有对行式存储的支持比如HP Vertica。

随着传统关系型数据库与新兴的分布式数据库不断的发展，列式存储与行式存储会不断融合，数据库系统会呈现双模式数据存放方式，这也是商业竞争的需要。

参考：

从SQL Server看列式存储：<http://mysql.taobao.org/monthly/2022/01/03/>

# 分类

HTAP列存引擎探秘：<http://mysql.taobao.org/monthly/2022/02/02/>

## Bigtable

Google发表的bigtable。

## HBase

基于Hadoop HDFS。

## Dynamo

Amazon Dynamo。

## Cassandra

Apache Cassandra。

## HP Vertica

## Presto

## EMC Greenplum

## ClickHouse

## CStore

MySQL的列存储引擎。

参考：

<https://mp.weixin.qq.com/s/-mlPrj0tbpHQ00gL3dnuCA>

## Cstore\_fdw

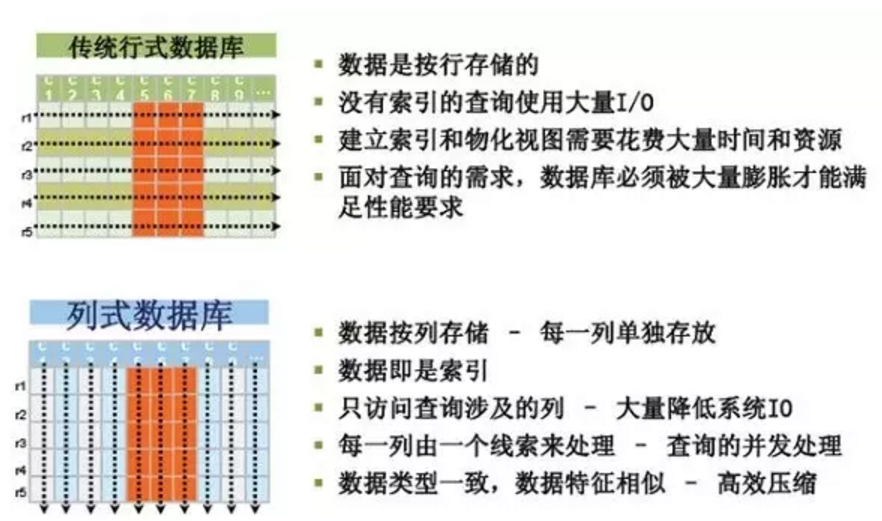
PG的列存储引擎。

参考：

<https://mp.weixin.qq.com/s/yN6NbRbMtc8d8Zu0ih9Zrw>

# 区别

## 行式数据库/列式数据库



## Key-Value/列式数据库

NoSQL可以大体上分为4个种类：Key-value、Document-Oriented、Column-Family Databases以及 Graph-Oriented Databases。

与Redis相比虽然都是存储非结构化数据，但是Redis没有主键的概念，是纯粹的key-value存储。

# 特点

为啥列存储可以大幅降低系统的I/O呢？

列式存储的主要优点之一就是可以大幅降低系统的I/O，尤其是在海量数据查询时，I/O向来是系统的主要瓶颈之一。通过下面这张图，相信大家能够彻底明白这一点。



# 应用场景

在比较了行式数据库与列式数据库之后，我们更关心的是如何根据业务场景需要选择对应的数据库系统。

适用的场景

1、日志。因为我们可以将数据储存在不同的列中，每个应用程序可以将信息写入自己的列族中。

2、博客平台。我们储存每个信息到不同的列族中。举个例子，标签可以储存在一个，类别可以在一个，而文章则在另一个。

不适用场景

1、如果我们需要ACID事务。Vassandra就不支持事务。

2、原型设计。如果我们分析Cassandra的数据结构，我们就会发现结构是基于我们期望的数据查询方式而定。在模型设计之初，我们根本不可能去预测它的查询方式，而一旦查询方式改变，我们就必须重新设计列族。

行式更适合OLTP，比如传统的基于增删改查操作的应用。列式更适合OLAP，非常适合于在数据仓库领域发挥作用，比如数据分析、海量存储和商业智能；涉及不经常更新的数据。

由于设计上的不同，列式数据库在并行查询处理和压缩上更有优势。而且数据是以列为单元存储，完全不用考虑数据建模或者说建模更简单了。要查询计算哪些列上的数据，直接读取列就行。

最后我们需要务实的指出，没有万能的数据库，列式数据库也并非万能，只不过给DBA提供了更多的选择，DBA需根据自己的应用场景自行选择。