

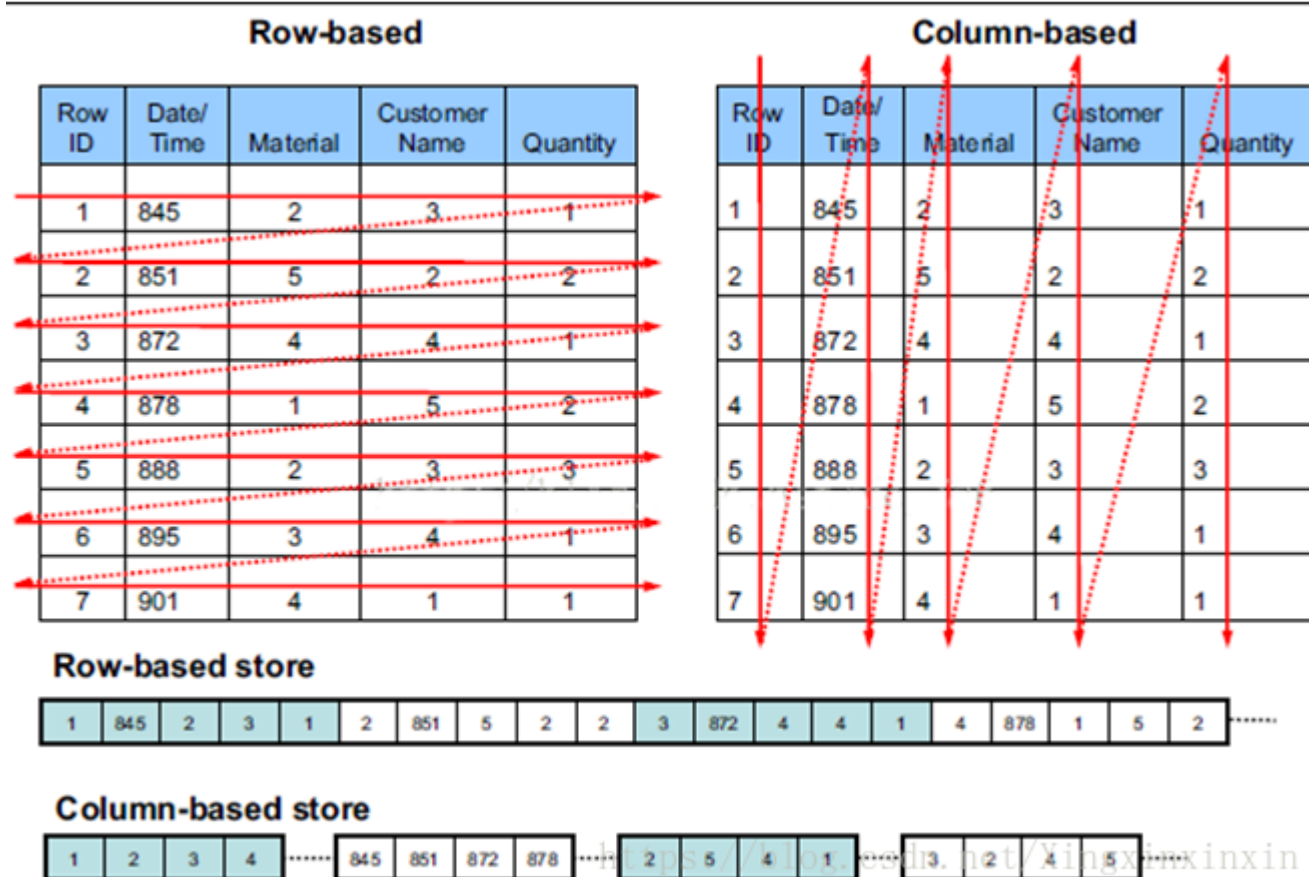
The background of the slide features a dynamic splash of blue liquid, possibly water or ink, against a white background. The splash originates from the top left and bottom right corners, with a large, billowing shape in the center-left and a more delicate, trailing shape in the top right. The liquid has a glossy, translucent appearance with varying shades of blue.

# 数据仓库的列存储

# 什么是列存储

- 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 storage)来说的。简单来说两者的区别就是如何组织表:
- Row-based storage stores a table in a sequence of rows.
- Column-based storage stores a table in a sequence of columns.

# 什么是列存储



# 应用场景

- **OLAP**。一个OLAP类型的查询可能需要访问几百万甚至几十亿个数据行，且该查询往往只关心少数几个数据列。例如，查询今年销量最高的前**20**个商品，这个查询只关心三个数据列：时间（**date**）、商品（**item**）以及销售量（**sales amount**）。列式数据库只需要读取存储着“时间、商品、销量”的数据列，而行式数据库需要读取所有的数据列。

# 列存储优点

- 同一个数据列的数据类型相同，数据相似性高，列式数据库压缩性比行存储好。
- 查询单列只需扫描单列，不必加载整行数据
- 缓存命中率提高，这是因为同一列的数据被高度压缩，常用的**Page**被频繁访问而变得异常活跃，**Buffer Manager**把活跃的数据页缓存到内存中，不常用的**Page**被换出（**Page Out**）

# 如何使用列存储

- 创建列存储表:

```
testDB=# create table test_column_ao (  
testDB(#          id bigint  
testDB(#          ,name varchar(128)  
testDB(#          ,value varchar(128))  
testDB=# with (appendonly=true,ORIENTATION=column,compresslevel=5)  
testDB=# distributed by (id);  
CREATE TABLE
```

# 列存储与行存储查询比较

- 创建表结构相同的普通表和列存表。

```
create table r_table (  
id int,  
col1 int,col2 int,col3 int,col4 int,col5 int,col6 int,col7 int,col8 int,col9 int,col10 int,  
col11 varchar(10),col12 varchar(10),col13 varchar(10),col14 varchar(10),col15 varchar(10),  
col16 varchar(10),col17 varchar(10),col18 varchar(10),col19 varchar(10),col20 varchar(10),  
col21 varchar(20),col22 varchar(20),col23 varchar(20),col24 varchar(20),col25 varchar(20),  
col26 varchar(20),col27 varchar(20),col28 varchar(20),col29 varchar(20),col30 varchar(20),  
col31 varchar(30),col32 varchar(30),col33 varchar(30),col34 varchar(30),col35 varchar(30),  
col36 varchar(30),col37 varchar(30),col38 varchar(30),col39 varchar(30),col40 varchar(30),  
col41 varchar(40),col42 varchar(40),col43 varchar(40),col44 varchar(40),col45 varchar(40),  
col46 varchar(40),col47 varchar(40),col48 varchar(40),col49 varchar(40),col50 varchar(40),  
col51 varchar(50),col52 varchar(55),col53 varchar(60),col54 varchar(65),col55 varchar(128),  
col56 date,col57 text,col58 timestamp,col59 varchar(125),col60 bigint)  
with (appendonly=true,compresslevel =5)  
distributed by (col11);  
  
create table c_table (  
id int,  
col1 int,col2 int,col3 int,col4 int,col5 int,col6 int,col7 int,col8 int,col9 int,col10 int,  
col11 varchar(10),col12 varchar(10),col13 varchar(10),col14 varchar(10),col15 varchar(10),  
col16 varchar(10),col17 varchar(10),col18 varchar(10),col19 varchar(10),col20 varchar(10),  
col21 varchar(20),col22 varchar(20),col23 varchar(20),col24 varchar(20),col25 varchar(20),  
col26 varchar(20),col27 varchar(20),col28 varchar(20),col29 varchar(20),col30 varchar(20),  
col31 varchar(30),col32 varchar(30),col33 varchar(30),col34 varchar(30),col35 varchar(30),  
col36 varchar(30),col37 varchar(30),col38 varchar(30),col39 varchar(30),col40 varchar(30),  
col41 varchar(40),col42 varchar(40),col43 varchar(40),col44 varchar(40),col45 varchar(40),  
col46 varchar(40),col47 varchar(40),col48 varchar(40),col49 varchar(40),col50 varchar(40),  
col51 varchar(50),col52 varchar(55),col53 varchar(60),col54 varchar(65),col55 varchar(128),  
col56 date,col57 text,col58 timestamp,col59 varchar(125),col60 bigint)  
with (appendonly=true,ORIENTATION=column,compresslevel =5)  
distributed by (col11);
```



# 列存储与行存储查询比较

- copy导入数据。

```
test2=# \copy r_table from '/home/cl/columnstore/data.dat' with delimiter ',';
```

```
test2=# \copy c_table from '/home/cl/columnstore/data.dat' with delimiter ',';
```

- Select \* from r\_table limit 1;查看数据



# 列存储与行存储查询比较

- 比较存储大小

`Select pg_relation_size('table_name');`

- 查询10个字段消耗时间
- 查询20个字段消耗时间
- 查询50个字段消耗时间
- 查询所有字段消耗时间
- 通过count具体字段可使测试字段被扫描
  - `SELECT count(col1),count(col2)... from tablename;`