

# 数据库系统 | Database Systems

第五课:存储-3

Lecture 5: Storage 3

Lecturer: Harbour

Date: 2021.12.06



Scan CC WeChat to Join the Community添加CC好友,接受进群邀请

Welcome to follow the GitHub repo

欢迎关注我们的代码仓库

https://github.com/cnosdatabase/cnosdb





# 计划教学内容



必讲

DB/DBMS

关系模型与关系代数

数据库存储

散列索引(哈希)

 $\mathbf{B}$ + $\mathbf{M}$ 

查询处理

并发控制

如果有兴趣

SQL

查询优化

恢复系统

分布式OLTP/OLAP

# 00 本节大纲



- 数据表示
- OLTP/OLAP/HTAP
- NSM/DSM



- ▶ 元组里的数据本质上就是字节数组,DBMS知道如何去解释这些字节到真正的属性值。
- ➤ 数据表示模式是DBMS如何通过字节去存储值。



#### Integers (整数)

- 使用IEEE-754标准的C/C++类型去编码整数。
- 例子: INTEGER, BIGINT, SMALLINT, TINYINT



#### Variable Precision Numbers (浮点数)

- **使用**IEEE-754标准的C/C++类型去编码
- 操作浮点数的速度比操作固定精度的数字的速度要快。
- 例子: FLOAT, REAL



#### Fixed-Point Precision Numbers(固定精度数字)

- 这些是具有任意精度和小数位数的数字类型。
- **当数据精确性要求很高的**时候可以使用这种类型,但**DBMS会花**费更高的代价去进行计算。
- 例子: NUMERIC, DECIMAL



#### Variable-Length Data(变长数据)

- 他们表示任意长度的数据类型,
- **大多数DBMS不允**许一个元组大小超过单个页大小。
- 也有一些系统会存储大数据在额外的文件上,并在元组中记录文件指针。
- 例子: VARCHAR, VARBINARY, TEXT, BLOB



#### 大数据值

- 当我们想要保存的东西因为体积太大而无法放在一个单个page,我们需要面对这个情况。
  - ▶ 溢出页存储
  - ▶ 外部文件存储



#### Dates and Times (日期和时间)

- **日期**/时间的表示因系统的不同而不同。**从unix**时代开始,时间通常表示为单位时间(**微**/**毫)秒**(时间戳)。
- 例子: TIME, DATE, TIMESTAM



#### System Catalogs(系统目录)

- DBMS 在其内部目录中存储有关数据库的元数据。
  - ▶ 表、列、索引、视图
  - ▶ 用户、权限
  - ▶ 内部统计

- **几乎每个** DBMS 都将其数据库目录存储在自身中。
  - 围绕元组包装对象抽象。
  - ▶ 引导目录表的专用代码。

#### **OLTP**



#### OLTP: Online Transaction Processing(在线事务处理)

• OLTP工作类型的特点是快速、短时间运行的操作,一次操作单个实体的简单查询和重复查询。

#### **OLAP**



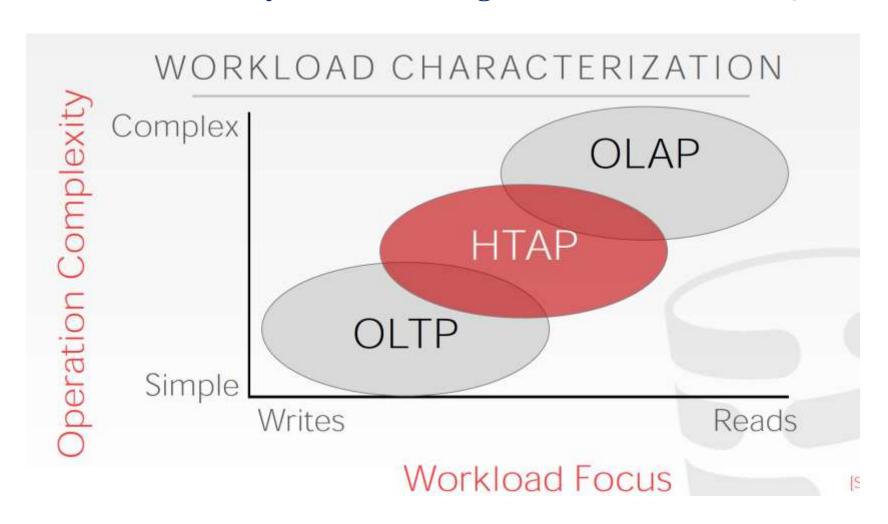
#### OLAP: Online Analytical Processing(在线分析处理)

• OLAP工作类型的特点是长时间、复杂的查询,对数据库大量数据的读取。在OLAP工作类型中,数据库系统专注于分析,并从OLTP数据库中获取新数据(一般为同步链路)。

#### HTAP



#### HTAP: Hybrid Transaction + Analytical Processing(混合事务分析处理)

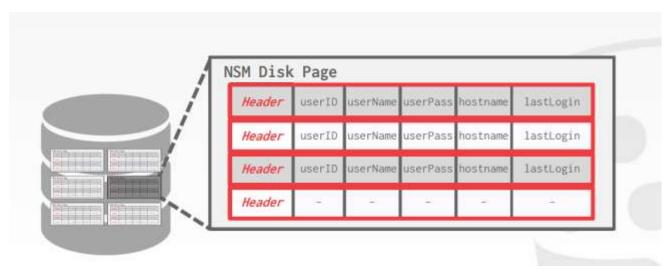


#### **NSM**



# DBMS将一个元组的所有属性连续的存储在单个页中,所以NSM又称为"行存"。

- 优点:
  - ▶ 快速插入、更新和删除。
  - > 对需要整个元组的查询友好。
- 缺点:
  - 对大范围的扫描和查找一部分属性不友好,因为在查询中可能会获取不必要的数据来污染缓冲池。



#### **DSM**



# DBMS将一个单独的属性(列)连续的存在一个块中。因此,这也被称为"列存"。

#### 优点:

- ▶ 减少查询过程中浪费的数据,因为DBMS只读取查询所需的数据(制度去查询所需的属性)。
- ▶ 由于对同一属性的所有值连续存储,因此可以获得更好的压缩性能(或使用特定的压缩)。

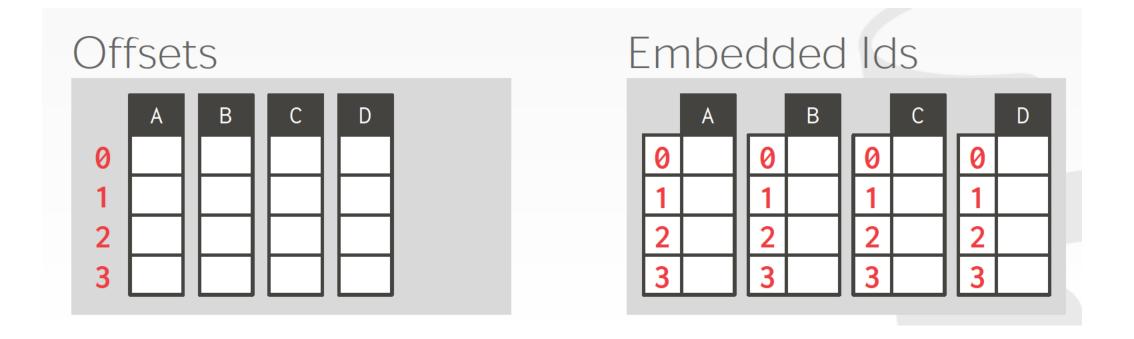
#### 缺点:

▶ 由于每个元组被存在不同的地方,因此对文件的点查、插入、更新和删除不友好。

#### **DSM**



- 固定长度的偏移
- 嵌入式元组id





# Q&A



Scan CC WeChat to Join the Community 添加CC好友,接受进群邀请

Welcome to follow the GitHub repo 欢迎关注我们的代码仓库 https://github.com/cnosdatabase/cnosdb



