CMU15445

Introduce to Database System

Lecture 03

Database

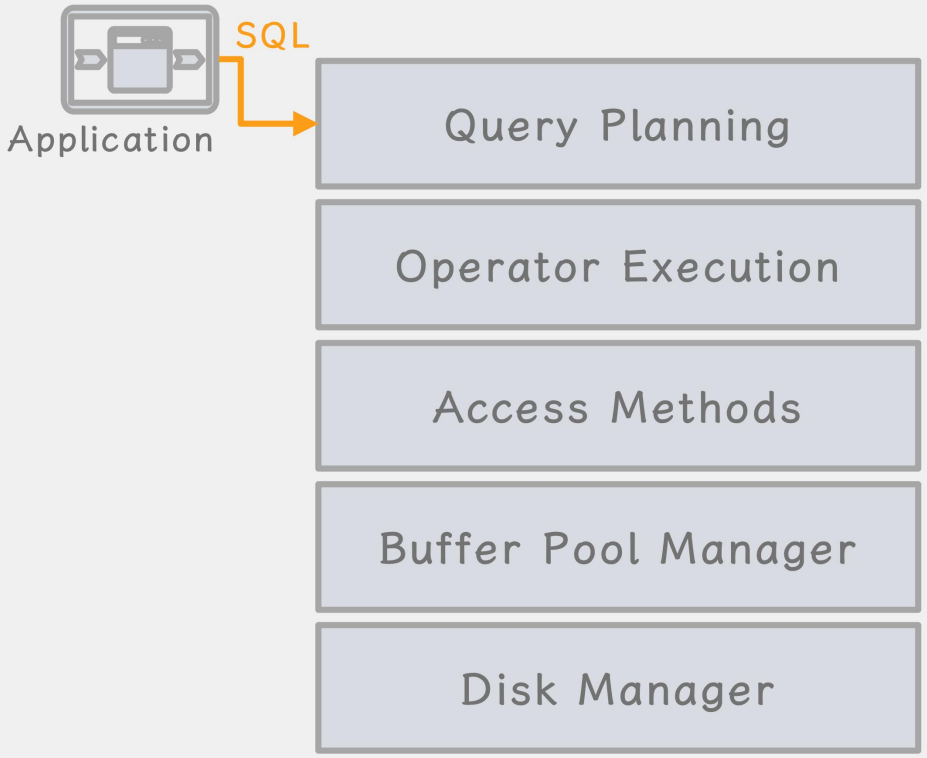
Storage Part 1

QingZhiLiangCheng

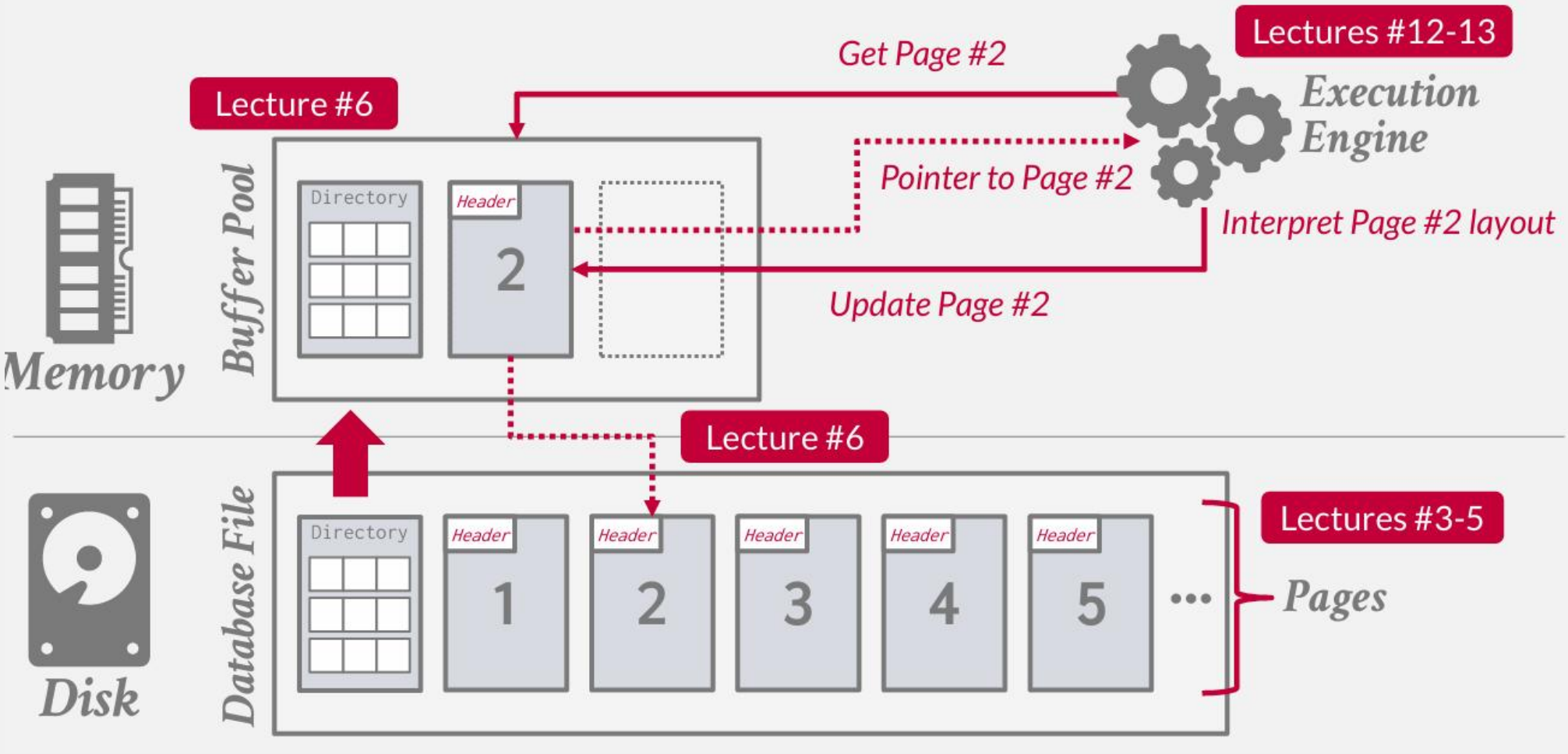
Overview

- Lectuer 01: Course Intro & Relational Model
- Lecture 02: Modern SQL

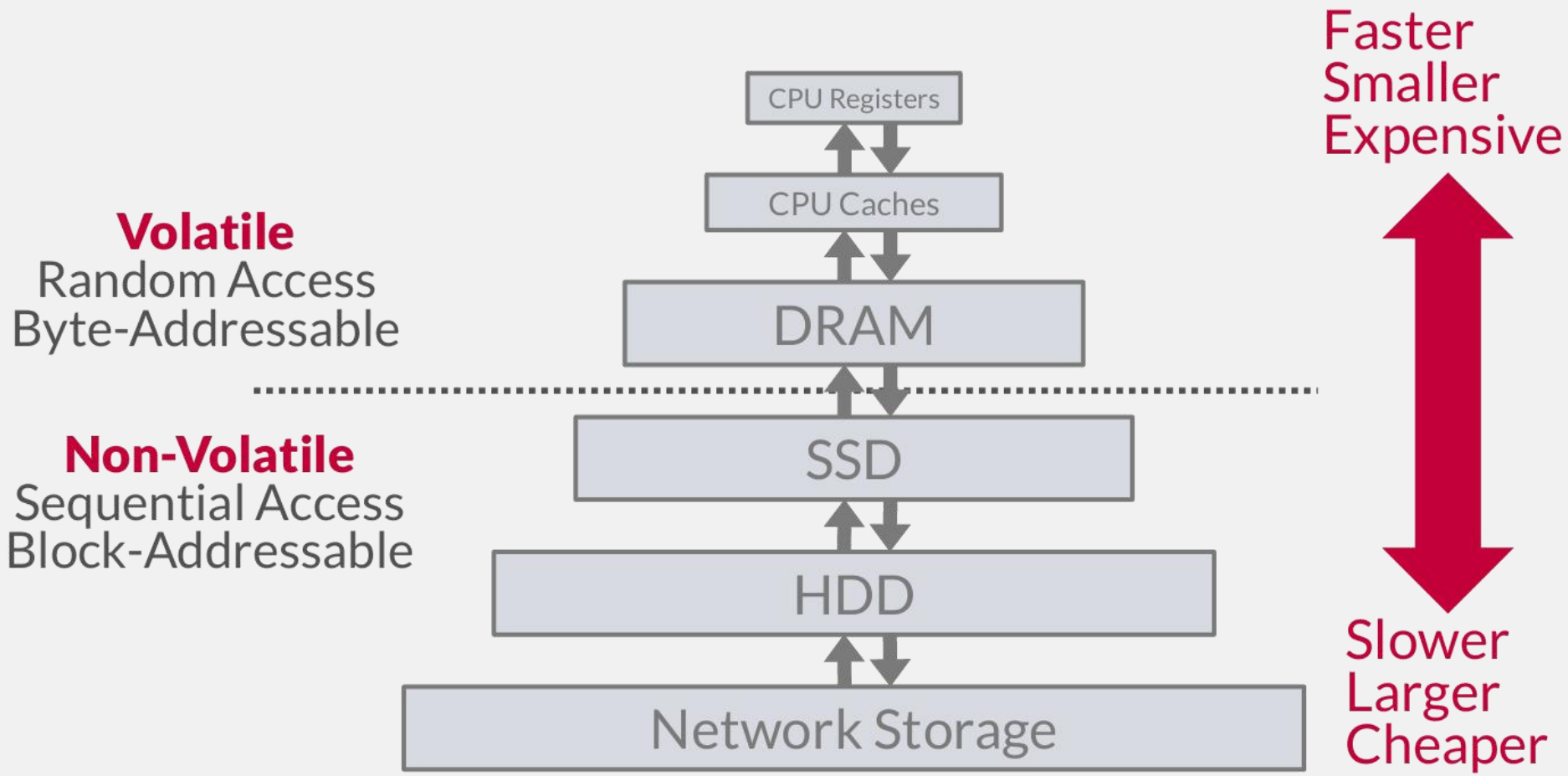
Overview


SQL
Application
Query Planning
Operator Execution
Access Methods
Buffer Pool Manager
Disk Manager

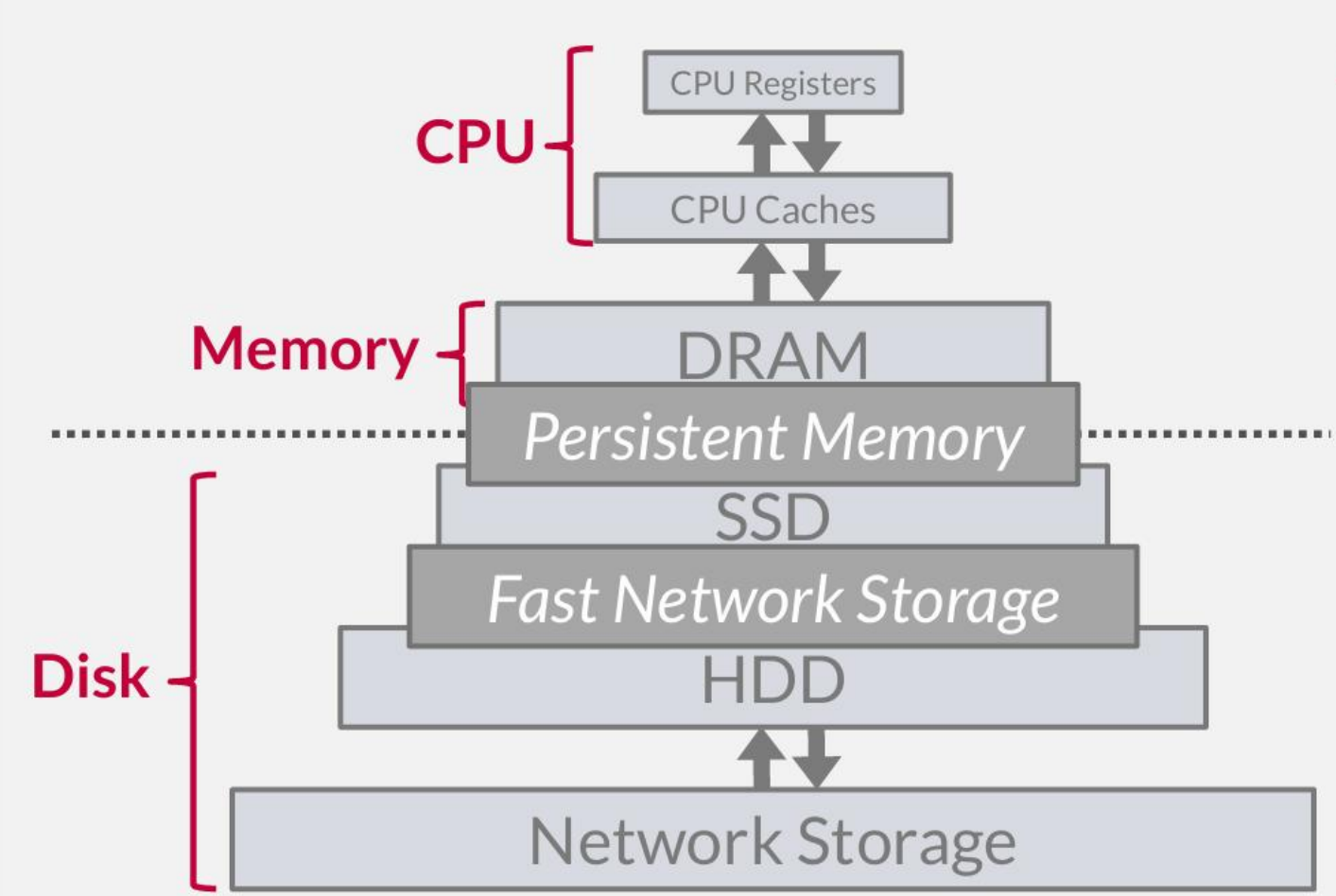
Disk-Oriented DBMS



Storage Hierarchy



Storage Hierarchy



Access Time


1 ns L1 Cache Ref
1 sec
4 ns L2 Cache Ref
4 sec
100 ns DRAM
100 sec
16,000 ns SSD
4.4 hours
2,000,000 ns HDD
3.3 weeks
~50,000,000 ns Network Storage
1.5 years
1,000,000,000 ns Tape Archives
31.7 years

Sequential vs Random Access

- database system is going to prefer sequential access over random access.
    - HDD
    - SSD
- Our goal:我们希望营造一种假象—— 正在完全使用内存的方式操作数据库

like virtual memory?

But we don't want to do virtual memory for the OS

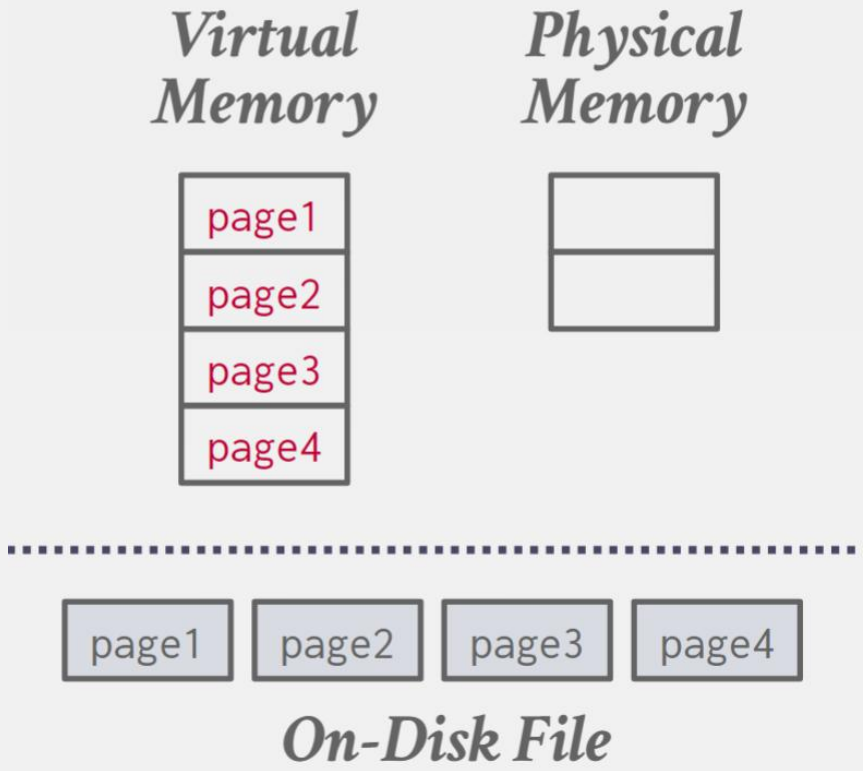
So why not use the OS?

- memory mapping(mmap)

可以将磁盘的内容映射到进程的虚拟内存中

the OS is responsible for deciding is the thing you need in memory or not.

the OS does all the management of the data moving the data back and forth for us.



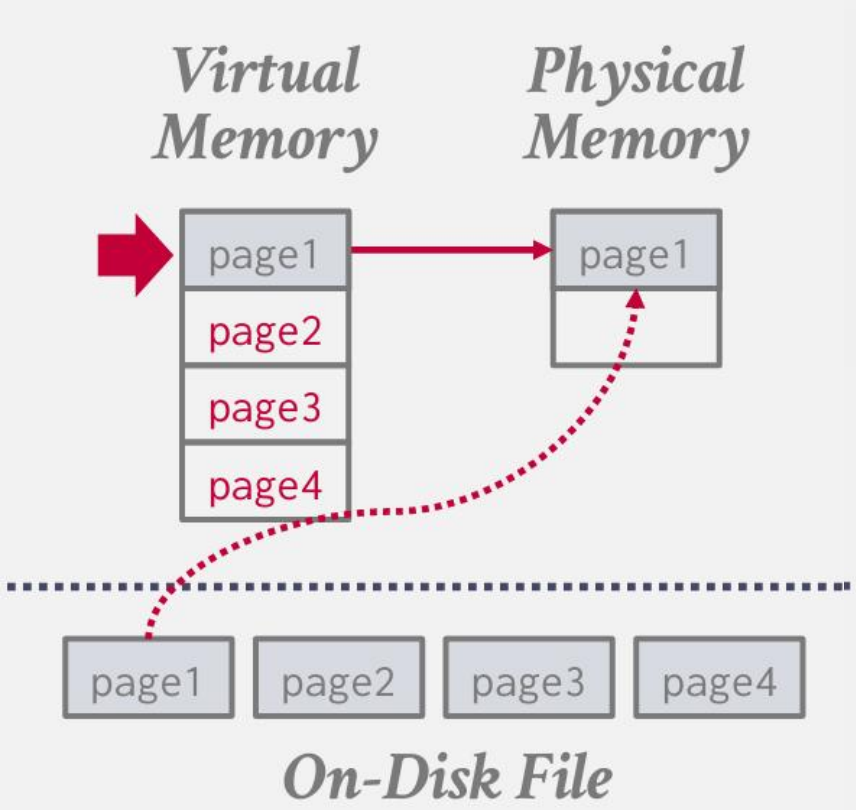
So why not use the OS?

- memory mapping(mmap)

可以将磁盘的内容映射到进程的虚拟内存中

the OS is responsible for deciding is the thing you need in memory or not.

the OS does all the management of the data moving the data back and forth for us.



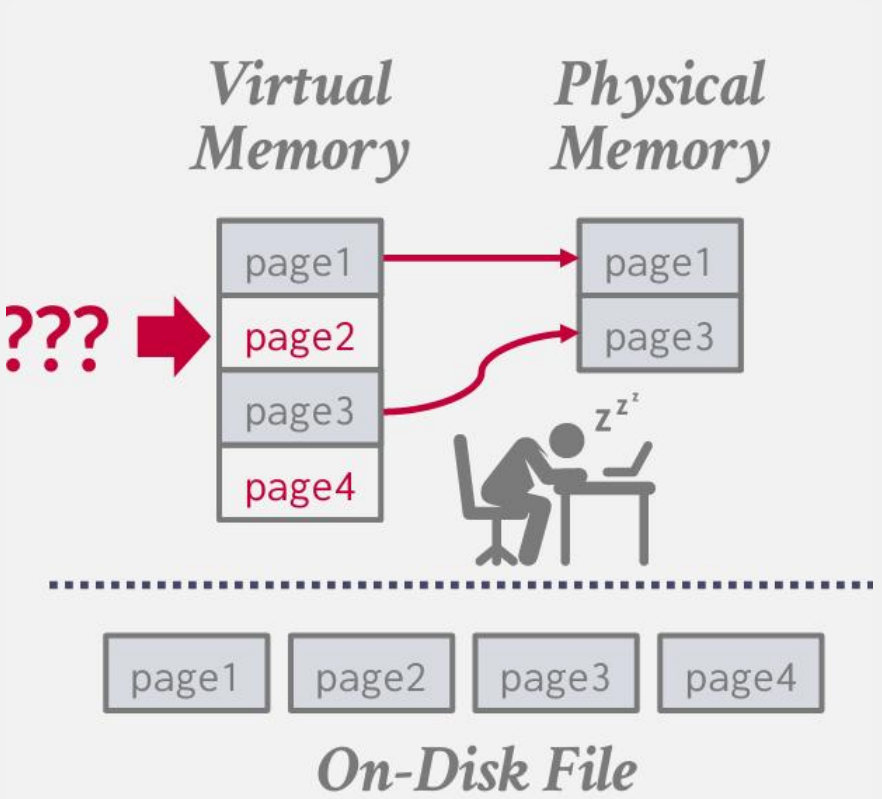
So why not use the OS?

- memory mapping(mmap)

可以将磁盘的内容映射到进程的虚拟内存中

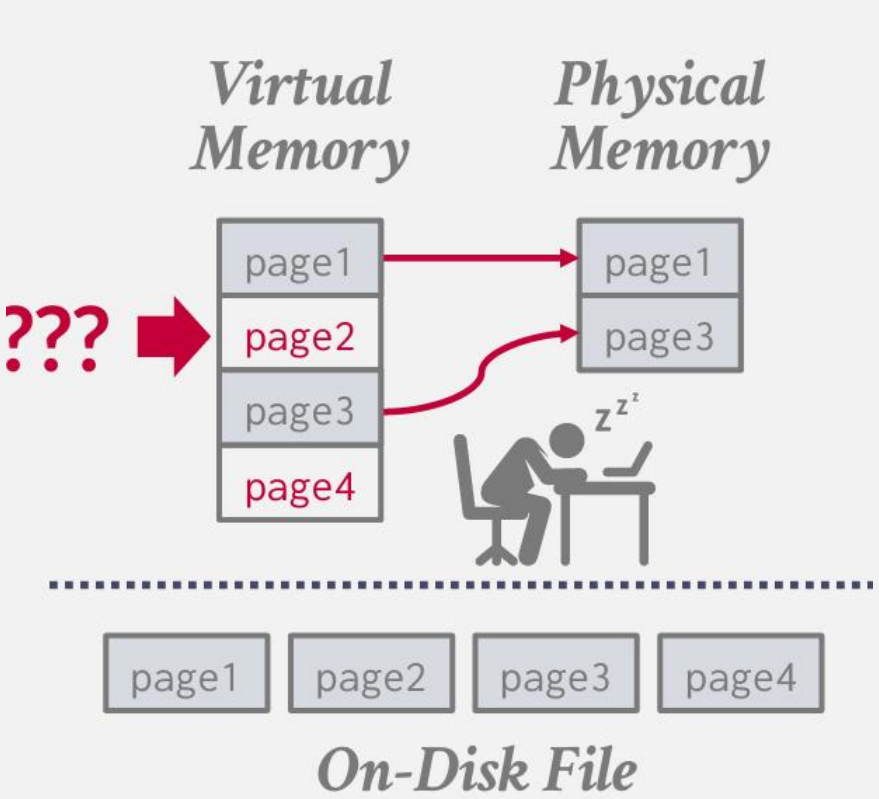
the OS is responsible for deciding is the thing you need in memory or not.

the OS does all the management of the data moving the data back and forth for us.



So why not use the OS?

- OS can flush dirty pages at any time.
- DBMS doesn't know which pages are in memory.
- 没法获得错误代码 只能取回中断



Why not use the OS?

https://db.cs.cmu.edu/mmap-cidr2022/

Database Storage

- Problem 1: How the DBMS represents the database files on disk.

- Problem 2: How the DBMS manages its memory and moves data back-and-forth from disk.

Today's Agenda

- File Storage
- Page Layout
- Tuple Layout

Today's Agenda

- File Storage
- Page Layout
- Tuple Layout

File Storage

- DBMS 将数据库作为一个或多个文件存储在磁盘上
  - 通常采用专有格式
  - OS 对这些文件的内容 一无所知
  - SQLite, DuckDB 单文件数据库
  - Postgres, MySQL 多文件数据库

Storage Manager

- 数据库系统中负责维护和协调不同文件的部分
- 系统中与硬件或存储设备通信的组件
- 多数系统都会维护自己的Disk Manager 决定按某种顺序读取哪些页面
- 数据库文件被分割成page

Database Pages

- Page is a fixed-size block of data.
    - tuples, meata-data, indexes, log records...
    - self-contained.
- Each page is given a unique identifier.
    - page ID

Database Pages

- three different notions of “pages” in a DBMS:
  - Hardware Page
  - OS Page
  - Database Page
- 或许越大越好 因为访存最大化
- 但越大写入操作的性能或许也越差

Page Storage Architecture

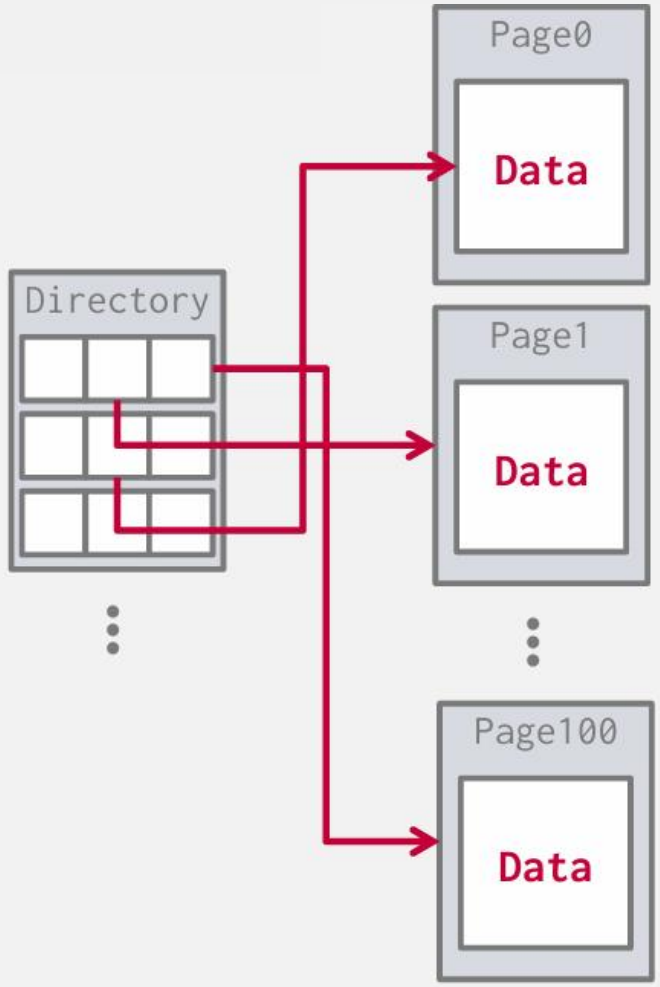
- Different DBMSs manage page in files on disk in different ways
  - Heap File Organization
  - Tree File Organization
  - Sequential / Sorted File Organization
  - Hashing File Organization

Heap File

- heap fine is an unordered collection of pages with tuples that are stored in random order.

- Storage Manager 提供一些API: 创建, 获取, 写入, 修改， 迭代器

Heap File: Page Directory



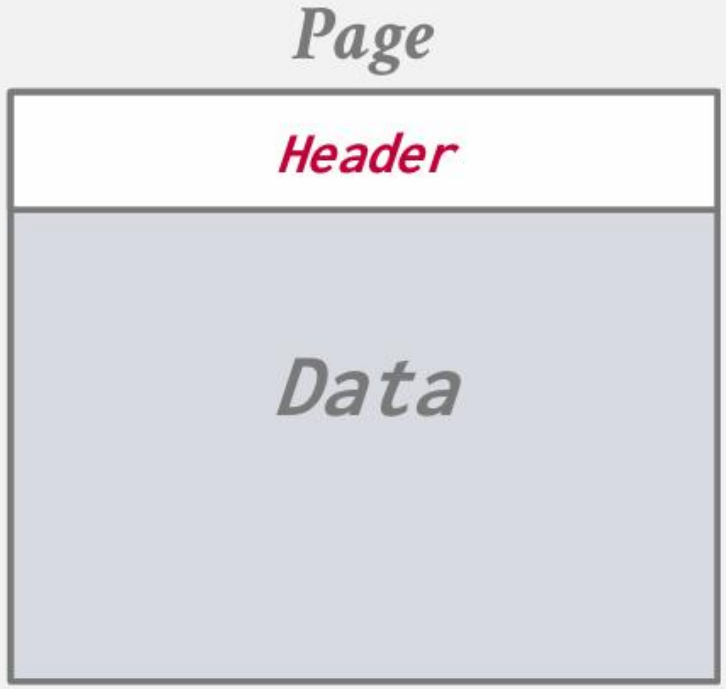
Today's Agenda

- File Storage
- Page Layout
- Tuple Layout

Page Layout

Every page contains a header of meta-data about the page's contents.

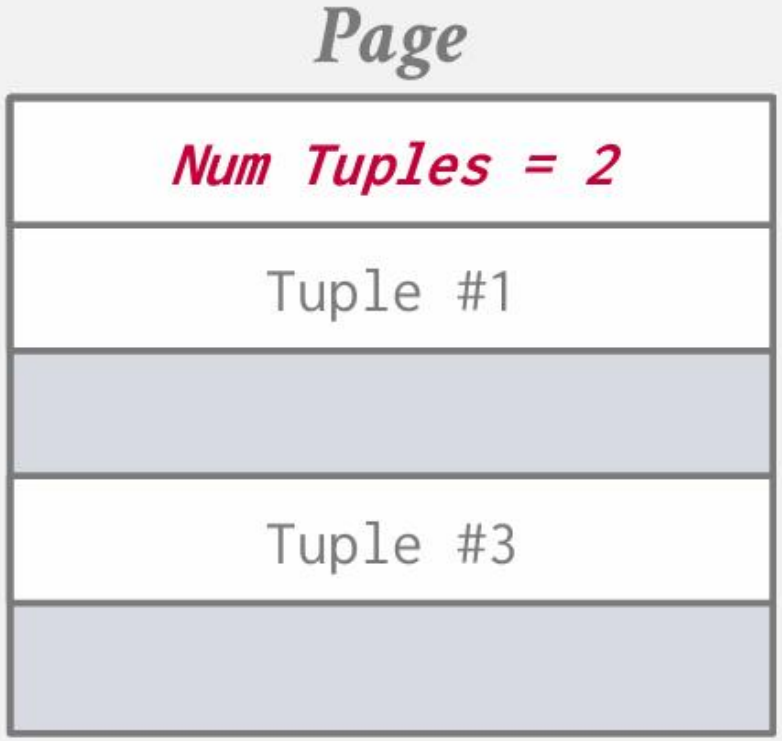
- Page Size
- Checksum
- DBMS version
- Schema Information
- Data Summary
- ...

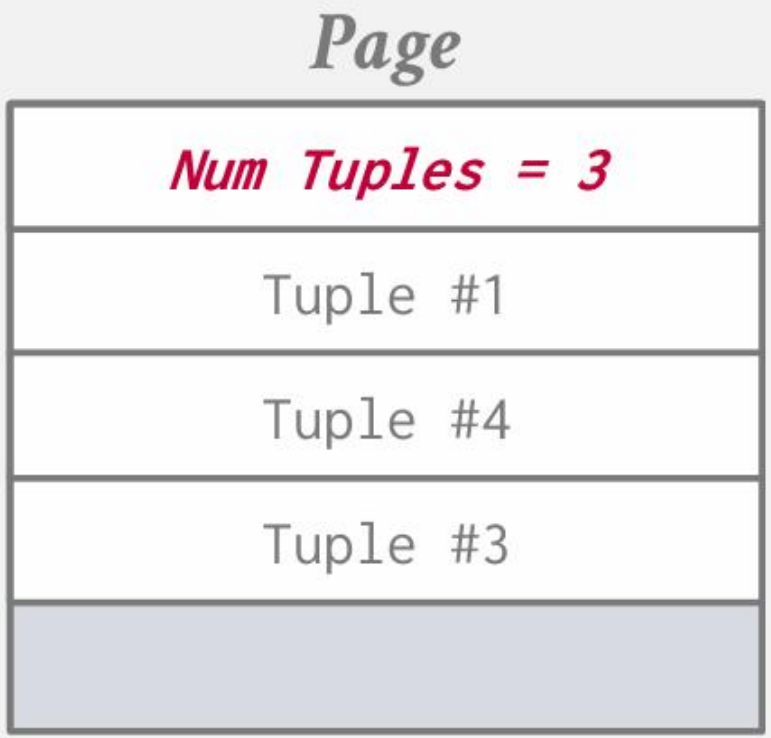


Page Layout

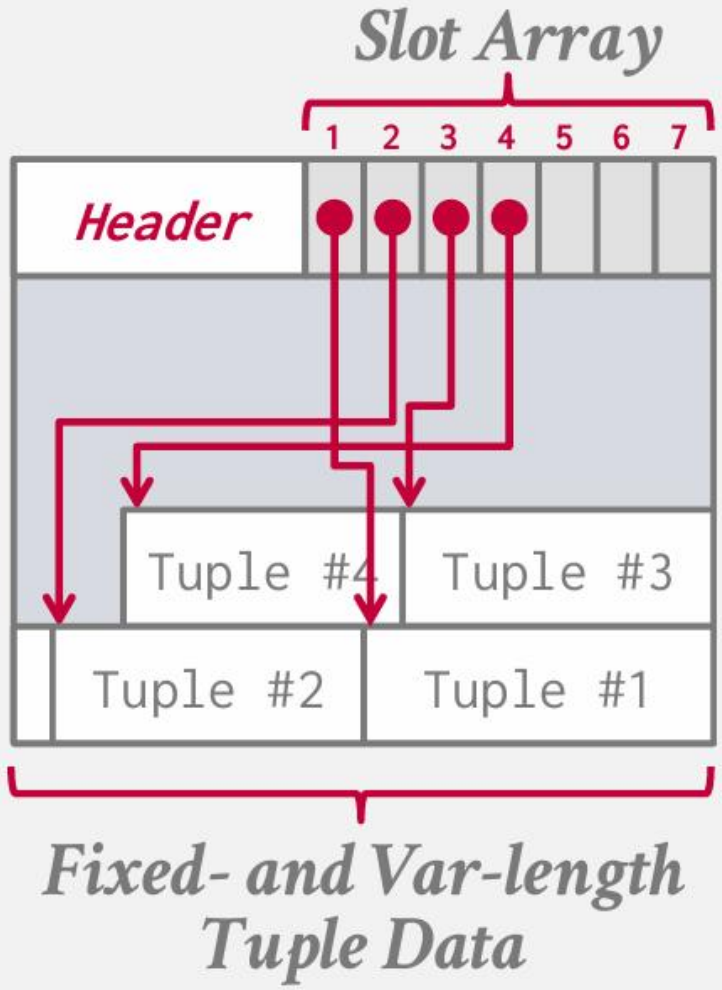
- How to organize the data inside of the page?
    - we are still assuming that wa are only storing tuples in a row-oriented storage model.
- Approach 1: Tuple-oriented Storage
- Approach 2: Log-structed Storage
- Approach 3: Index-organized Storage

Tuple-oriented Storage

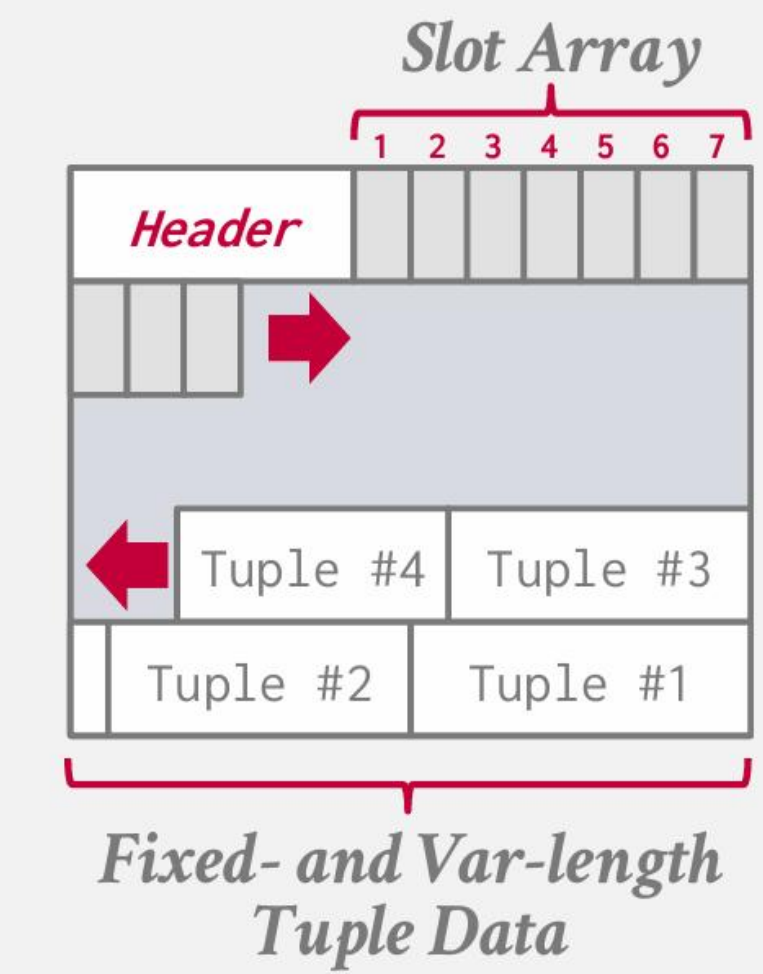




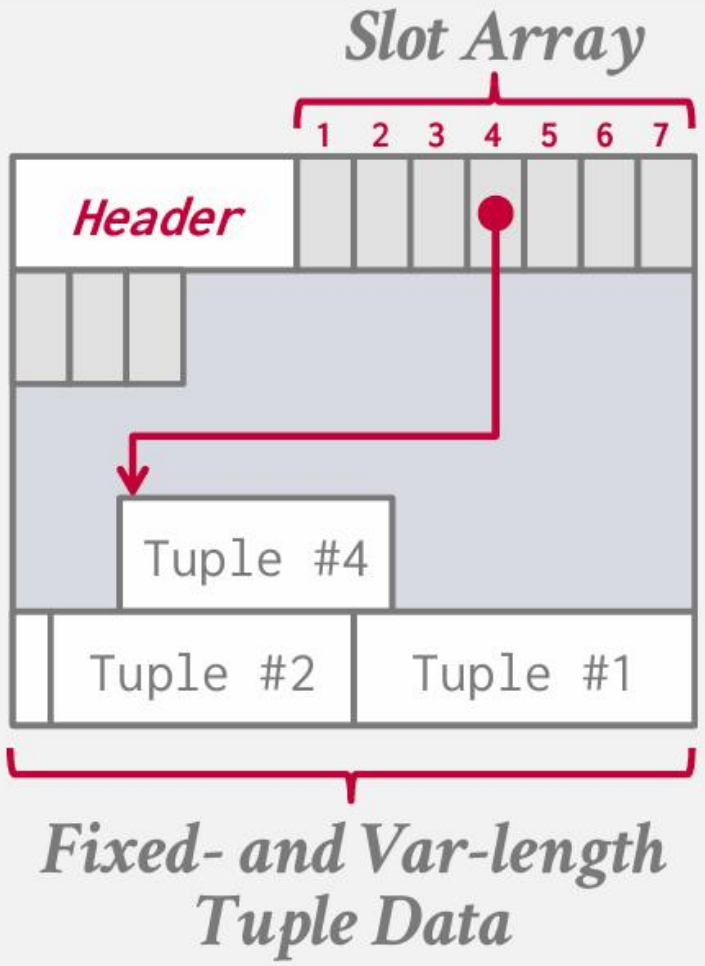
Slotted Pages



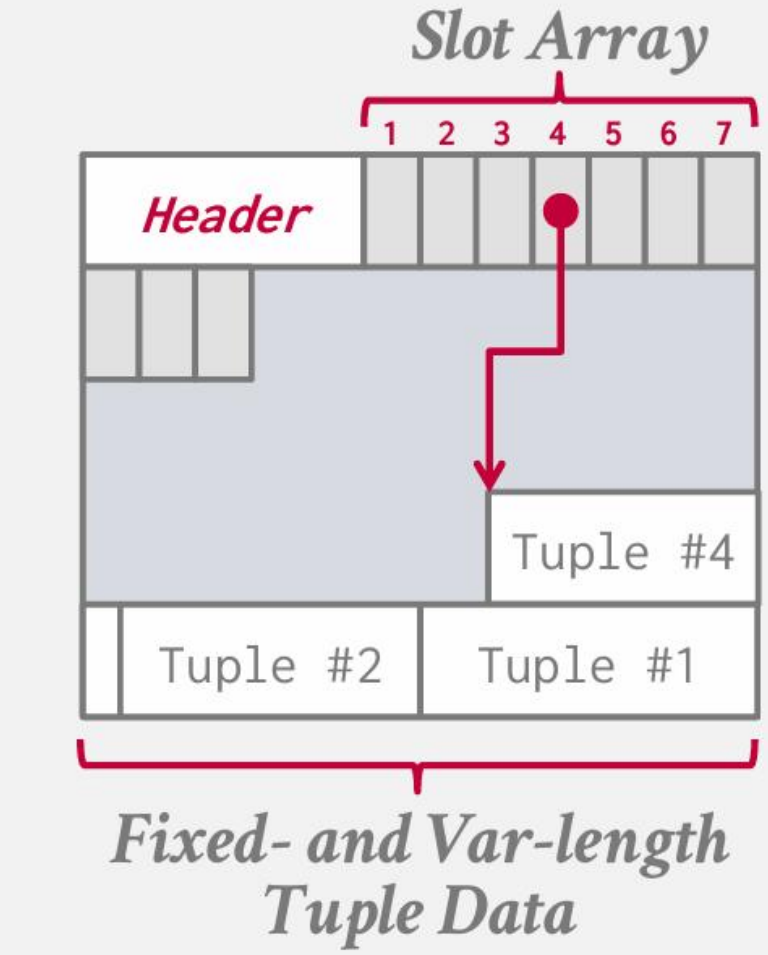
Slotted Pages



Slotted Pages



Slotted Pages



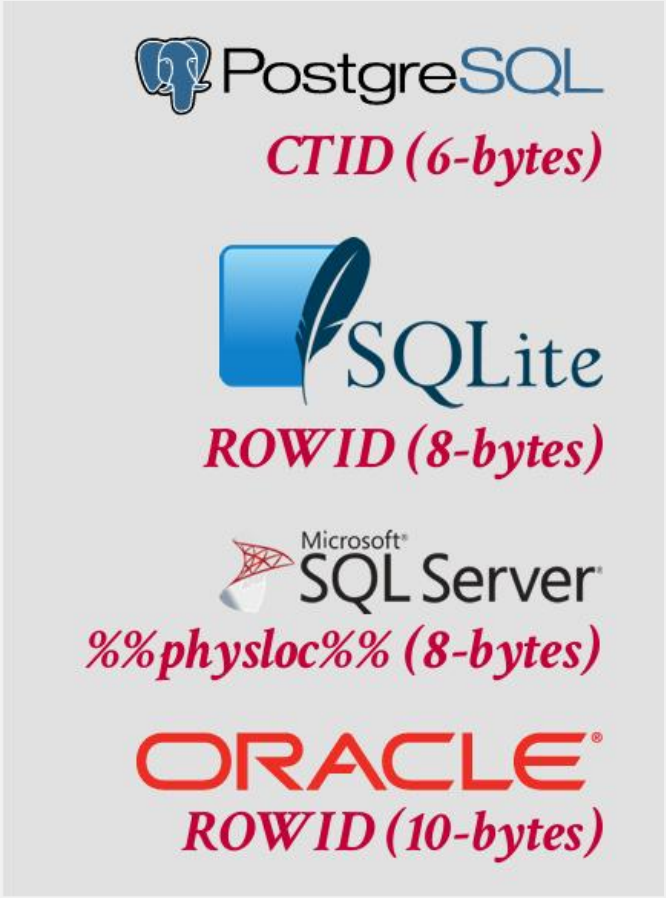
Record IDs

- Tuple 的唯一标识

-本质上是一种物理位置 File Id, Page Id, Slot

- 大多数DBMS并未在tuple中存储id

- SQLite -- ROWID



Demo

https://onecompiler.com/