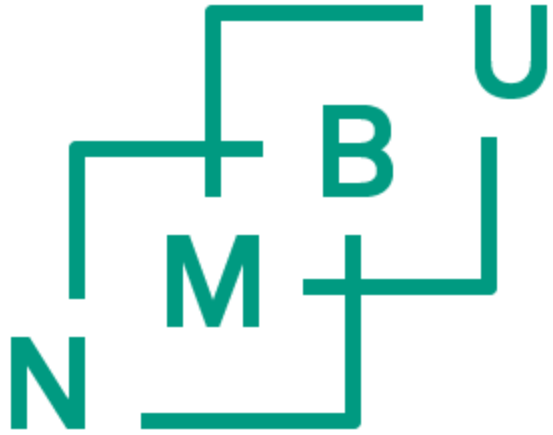


Norwegian University of Life Sciences

Datahåndtering og analyse

INF230



Outline

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

Parts of Computer

Different Storages

Storage and File Structure
File storage, Page layout, Tuple layout

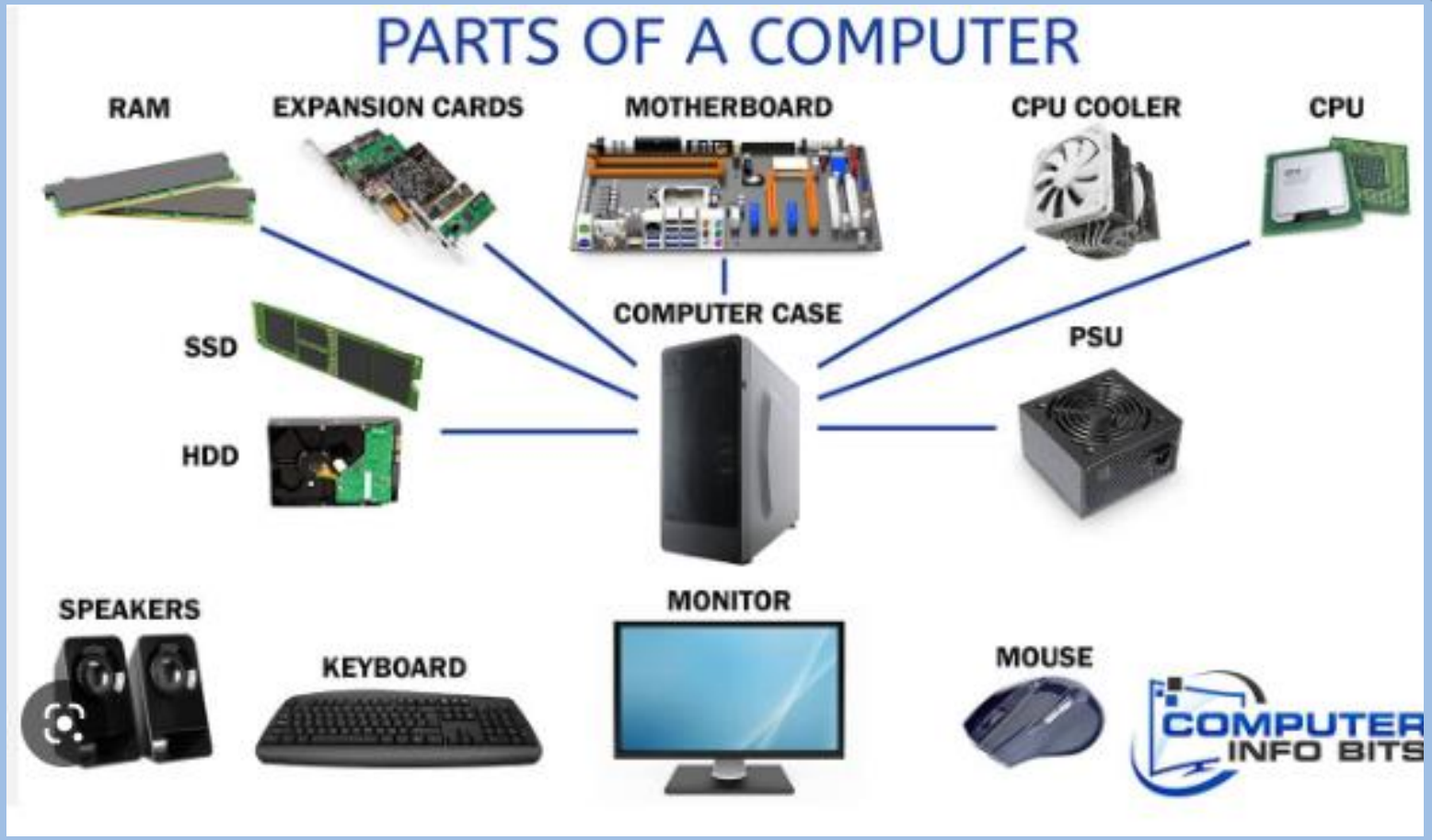
Data Storage Models
NSM and DSM

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

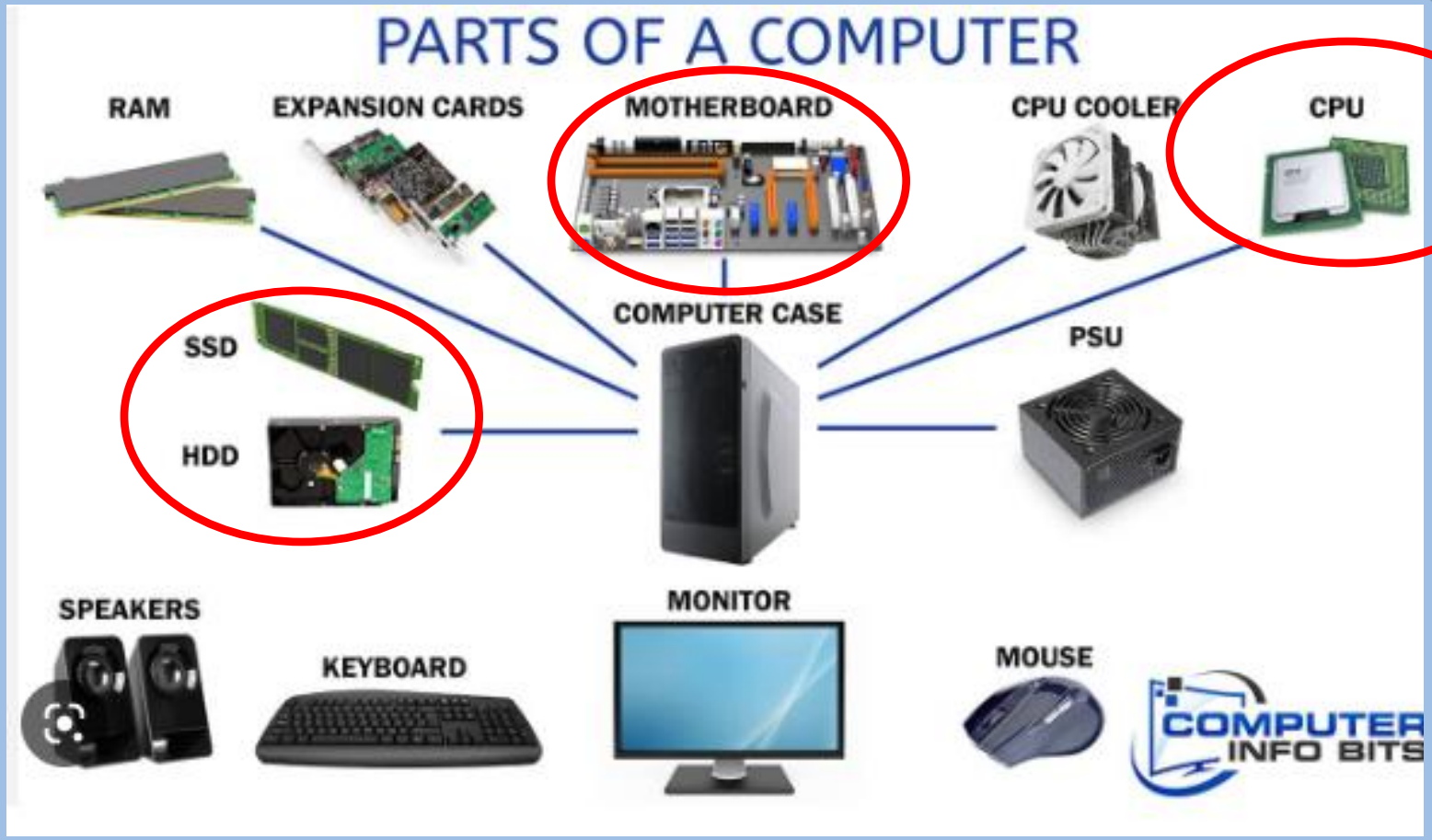
Buffer Pool Organization

Buffer Pool Optimization

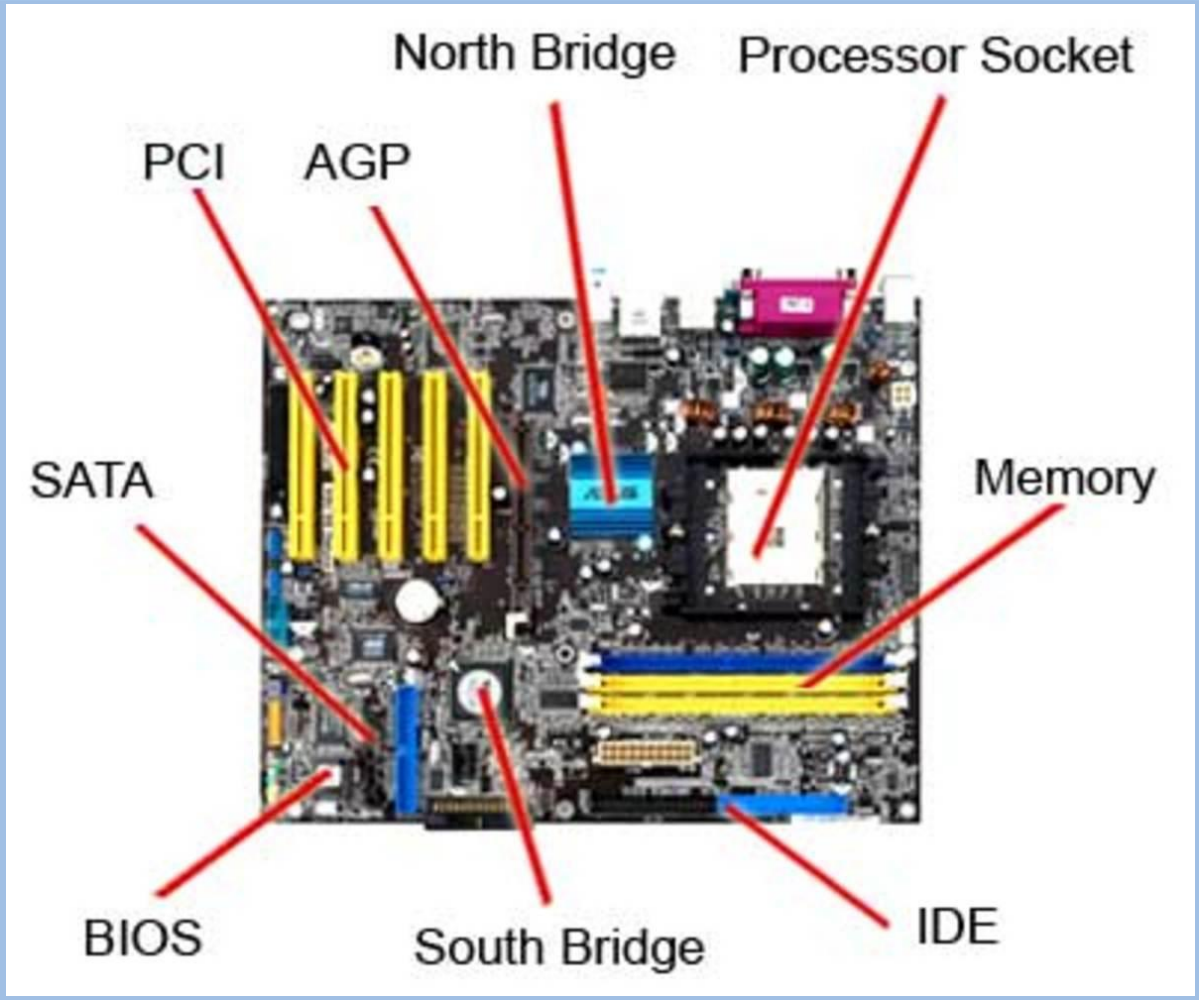
Parts of Computer



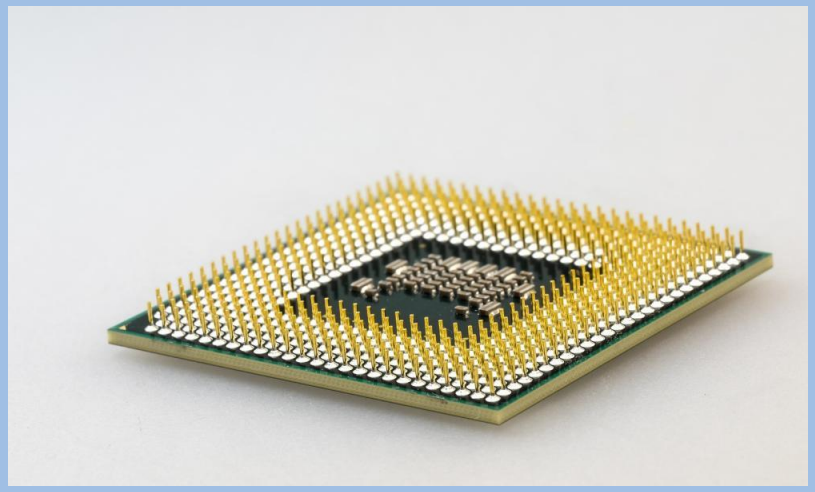
Parts of Computer



Motherboard

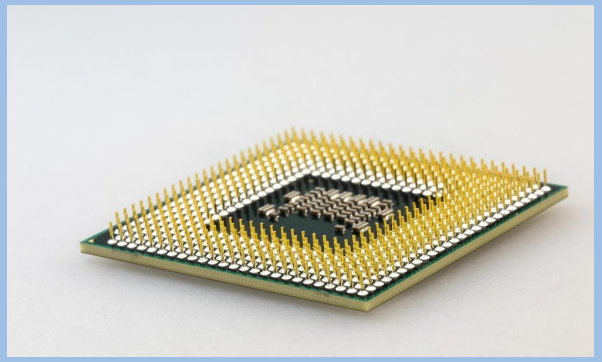


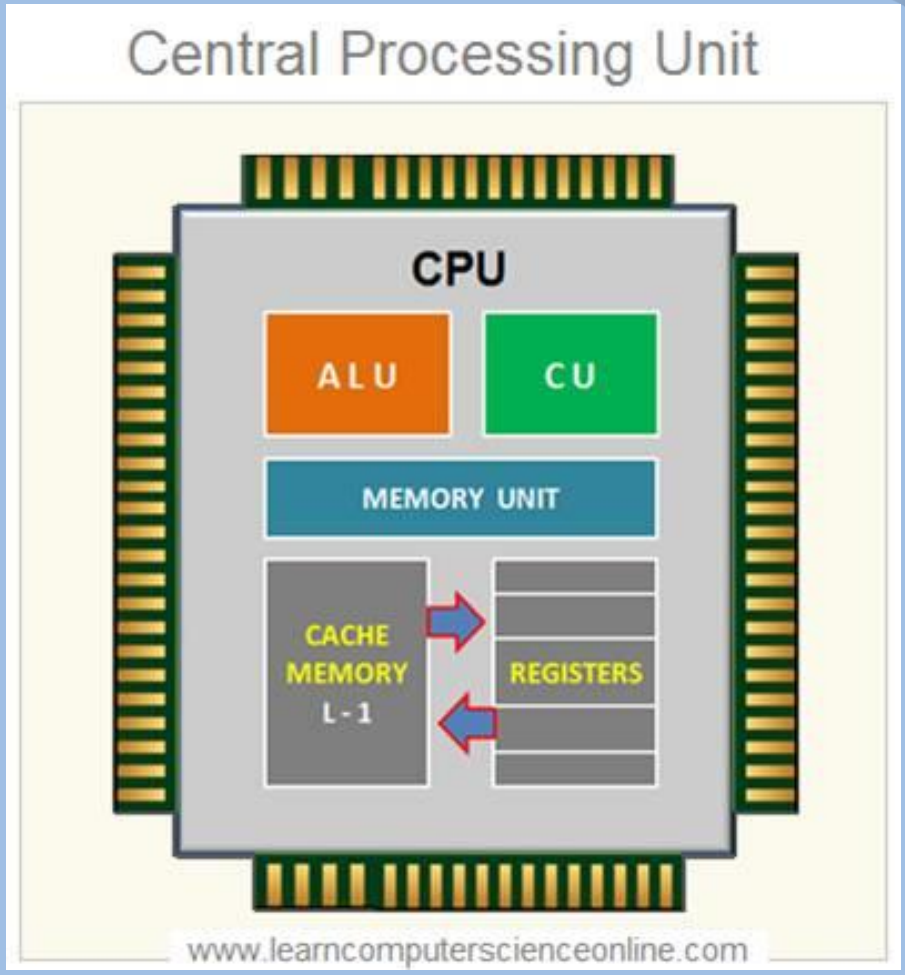
Central Processing Unit (CPU)





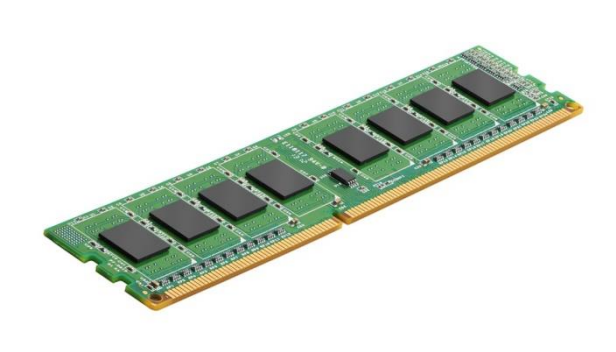
Central Processing Unit (CPU)



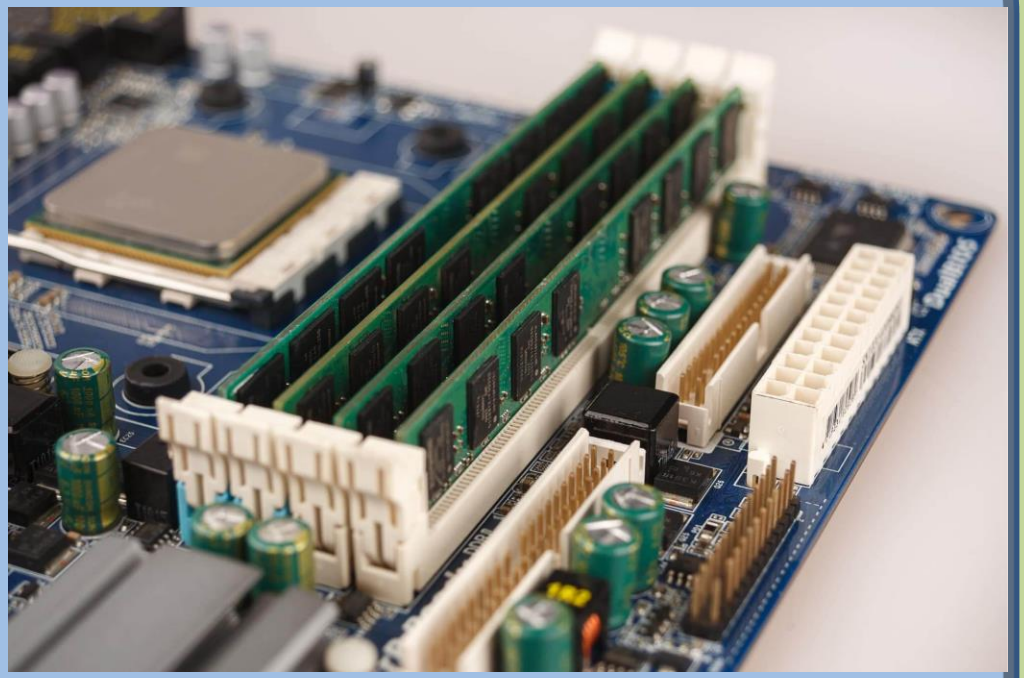


Dynamic Random Access Memory (DRAM)

DRAM



Motherboard

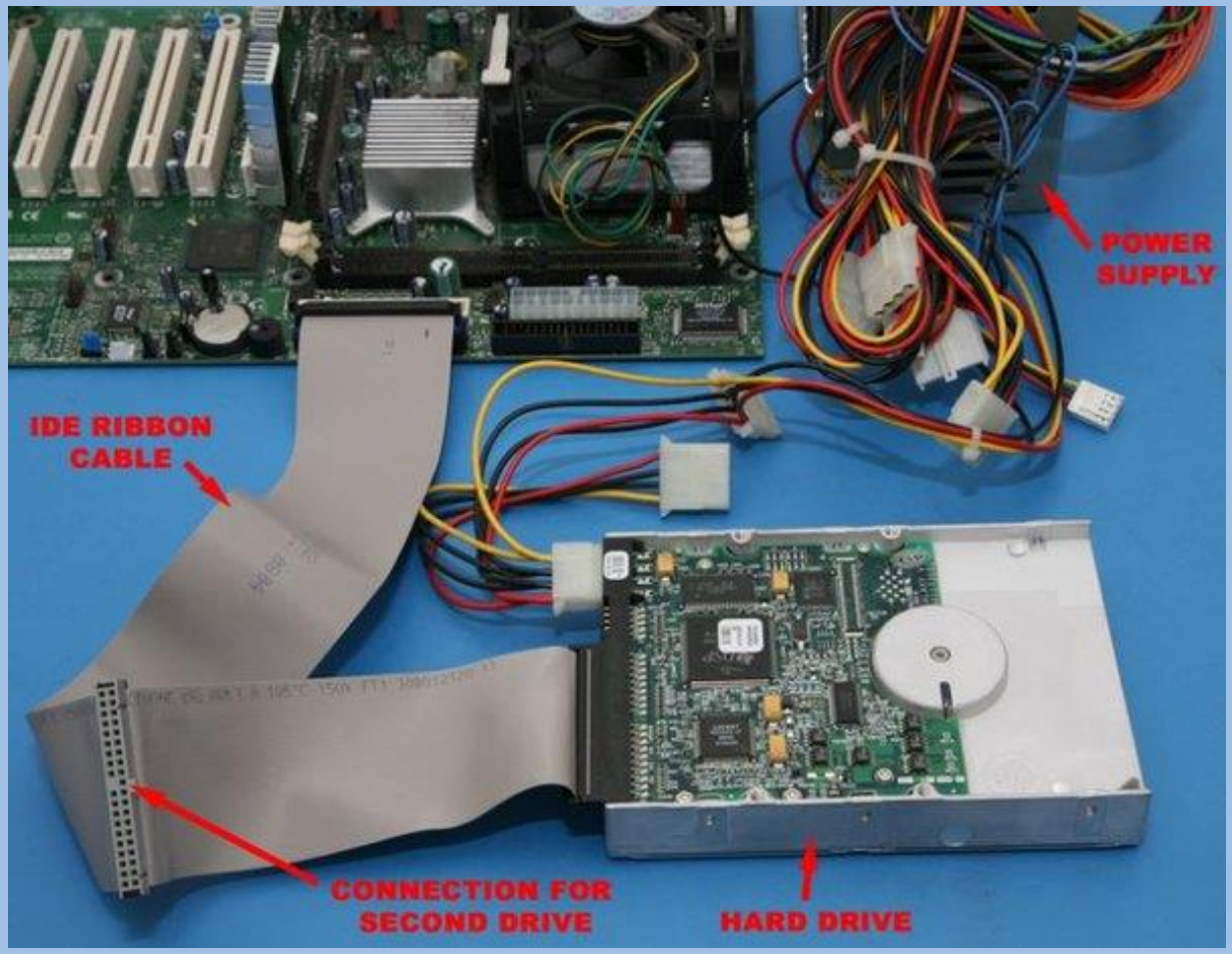


Solid-state drives memory (SSD)





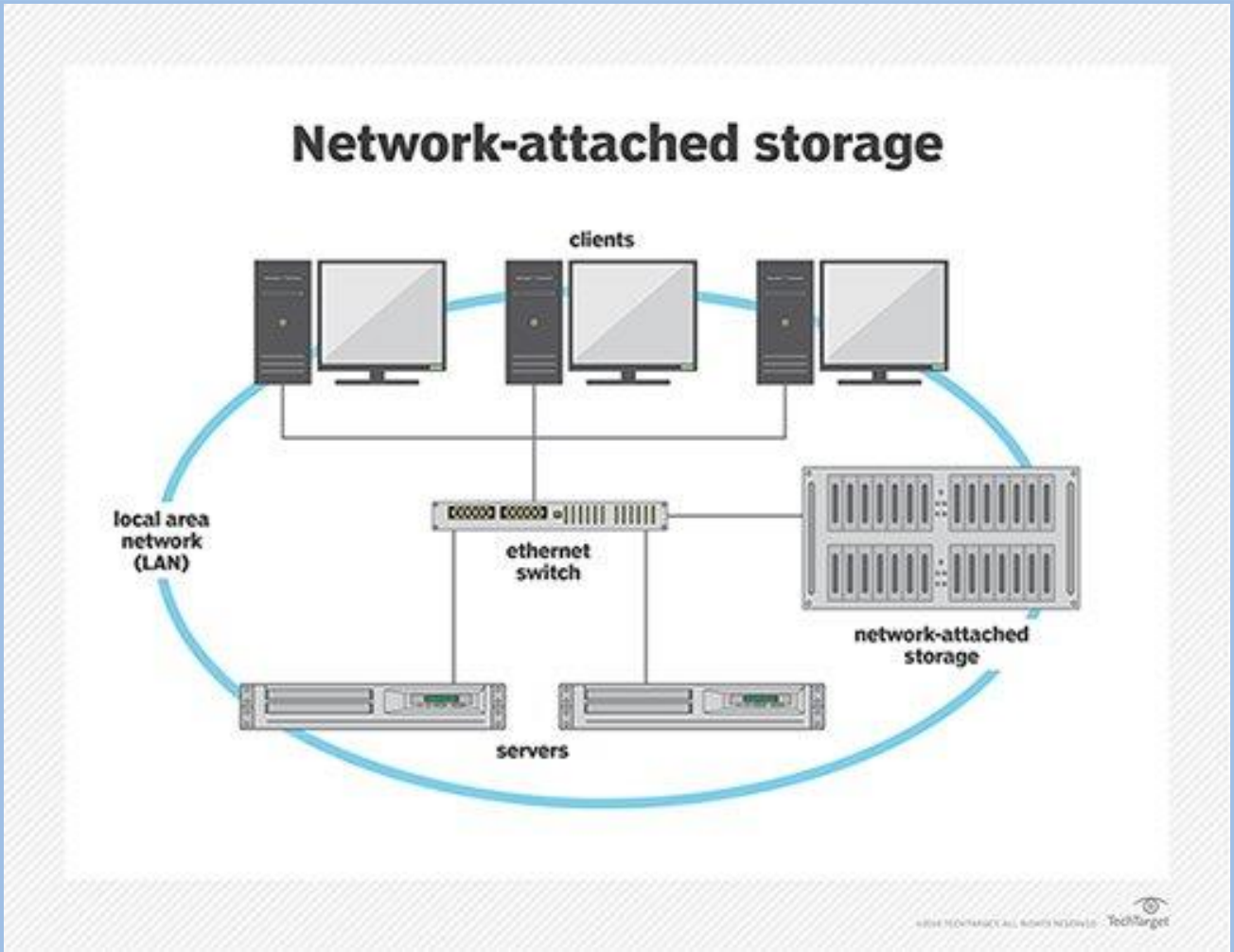
Hard Disk Drive Memory (HDD)



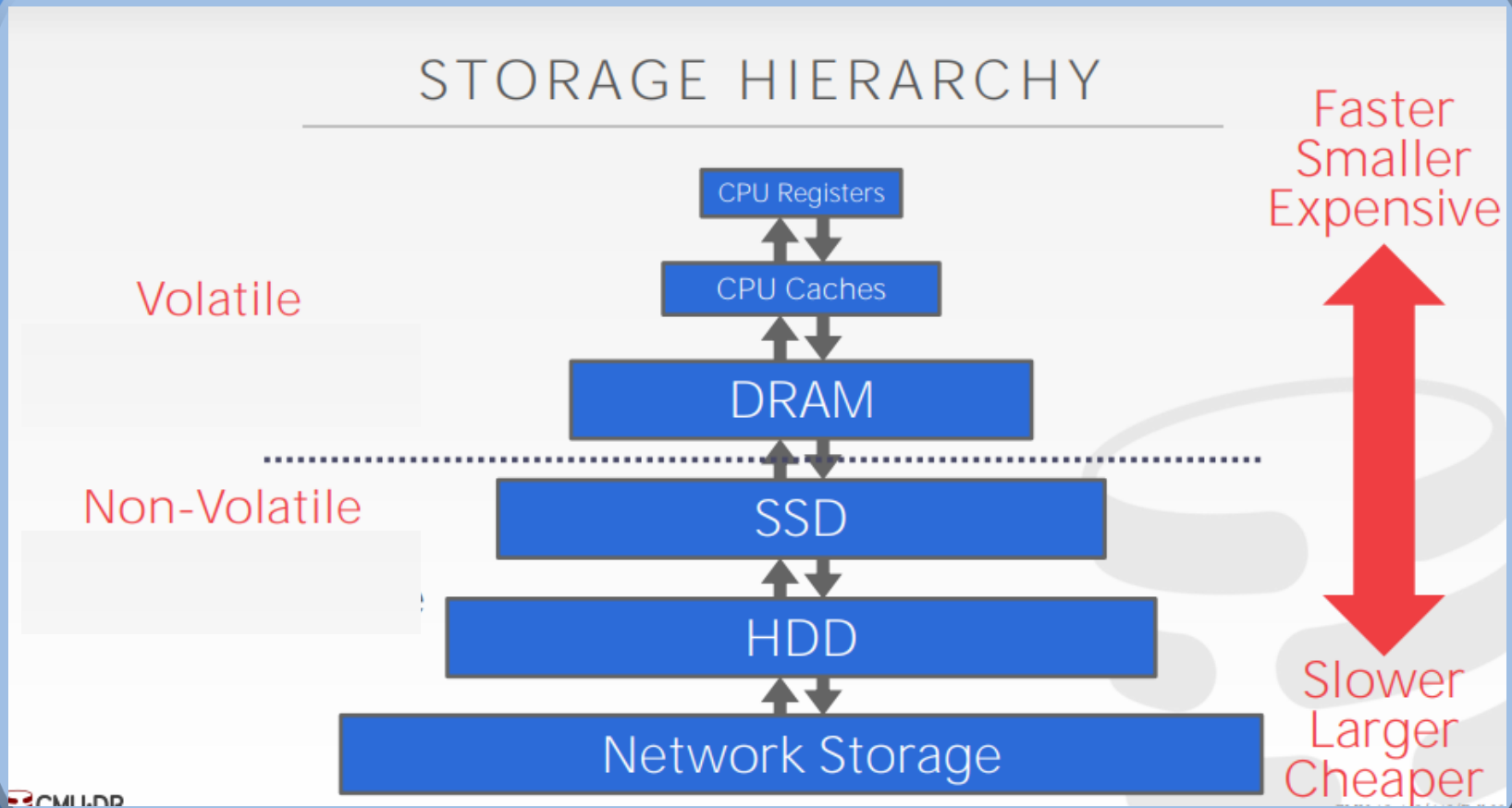
Network Storage



Network Storage



Different Storages



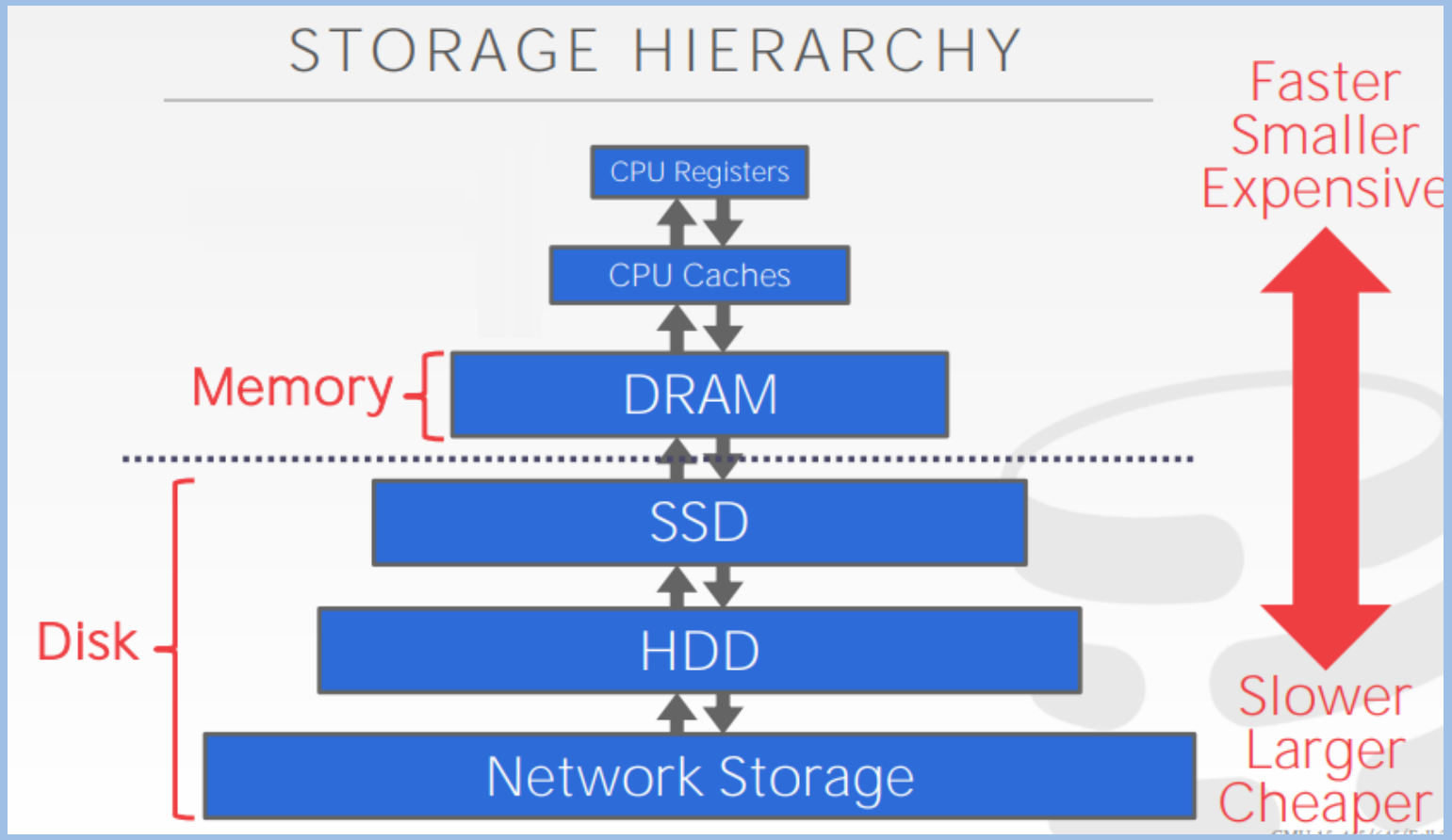
Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Different Storages

Volatile memory is a type of memory that maintains its data only while the device is powered. If the power is interrupted for any reason, the data is lost.

Non-volatile memory or non-volatile storage is a type of computer memory that can retain stored information even after power is removed.

Different Storages



Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure

DATABASE STORAGE

Problem #1: How the DBMS represents the database in files on disk. **From slides 17 to 70**

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

Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure

DISK-ORIENTED ARCHITECTURE

The DBMS assumes that the primary storage location of the database is on non-volatile disk.

The DBMS's components manage the movement of data between non-volatile and volatile storage.

To make sure that we do not lose data.

Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure

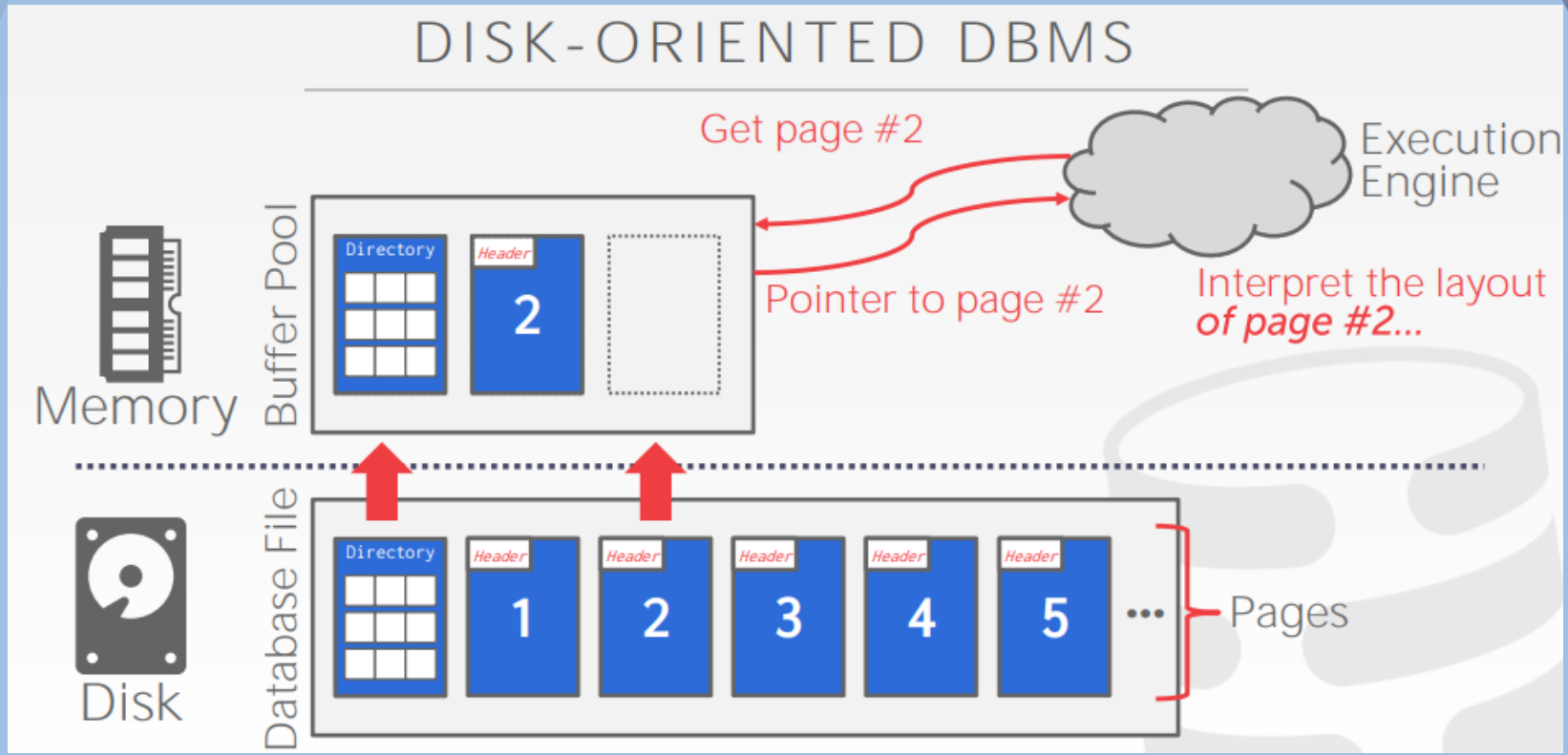
SYSTEM DESIGN GOALS

Allow the DBMS to manage databases that exceed the amount of memory available.

Reading/writing to disk is expensive, so it must be managed carefully to avoid large stalls and performance degradation.

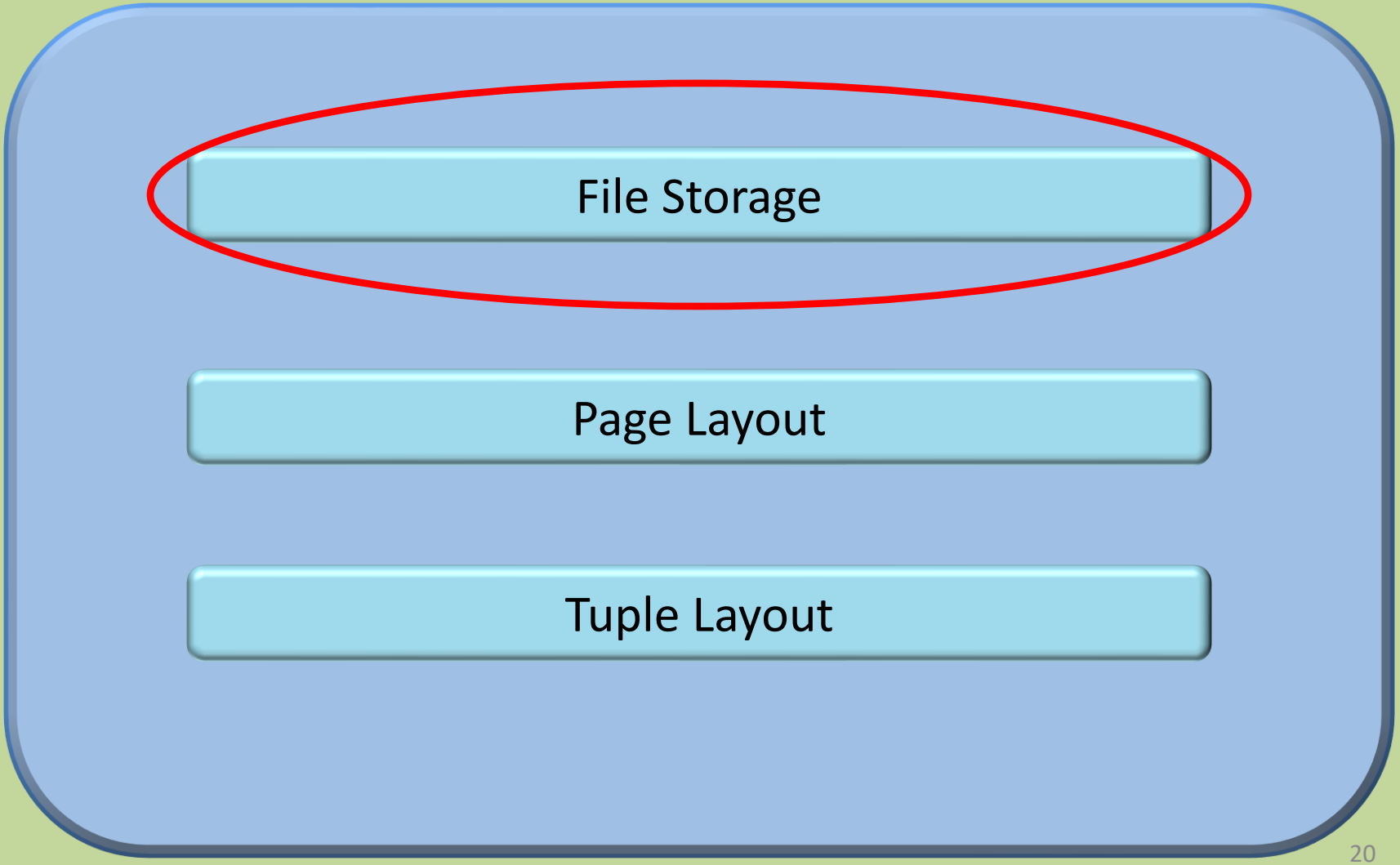
Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure



Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure



Storage and File Structure – File Storage

DATABASE PAGES

A page is a fixed-size block of data.

→ It can contain tuples, meta-data, indexes, log records…
→ Most systems do not mix page types.
→ Some systems require a page to be self-contained.

Each page is given a unique identifier.

→ The DBMS uses an indirection layer to map page ids to physical locations.

Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure – File Storage

PAGE STORAGE ARCHITECTURE

Different DBMSs manage pages in files on disk in different ways.

→ Heap File Organization
→ Sequential / Sorted File Organization
→ Hashing File Organization

At this point in the hierarchy we don't need to know anything about what is inside of the pages.

Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure – File Storage

DATABASE HEAP

A heap file is an unordered collection of pages where tuples that are stored in random order.

→ Create / Get / Write / Delete Page

→ Must also support iterating over all pages.

Need meta-data to keep track of what pages exist and which ones have free space.

Two ways to represent a heap file:

→ Linked List

→ Page Directory

Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure – File Storage

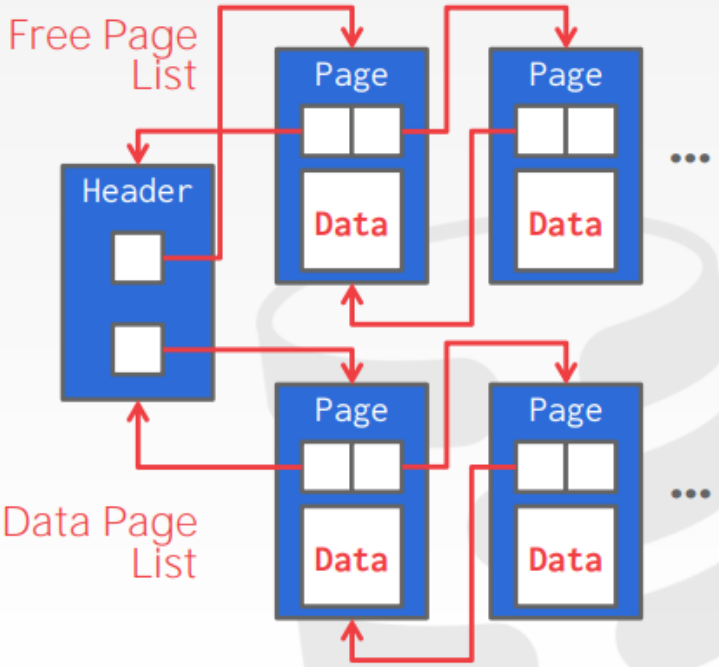
HEAP FILE: LINKED LIST

Maintain a header page at the beginning of the file that stores two pointers:

→ HEAD of the free page list.

→ HEAD of the data page list.

Each page keeps track of the number of free slots in itself.



Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

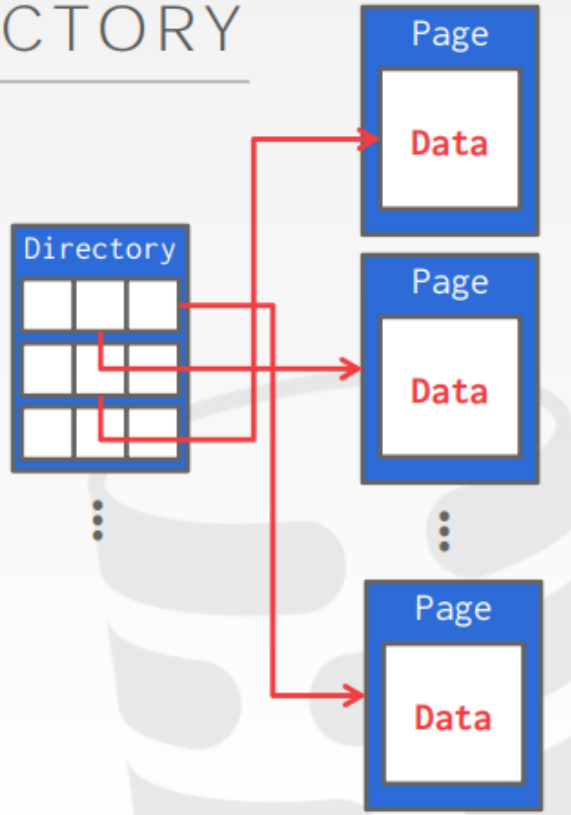
Storage and File Structure – File Storage

HEAP FILE: PAGE DIRECTORY

The DBMS maintains special pages that tracks the location of data pages in the database files.

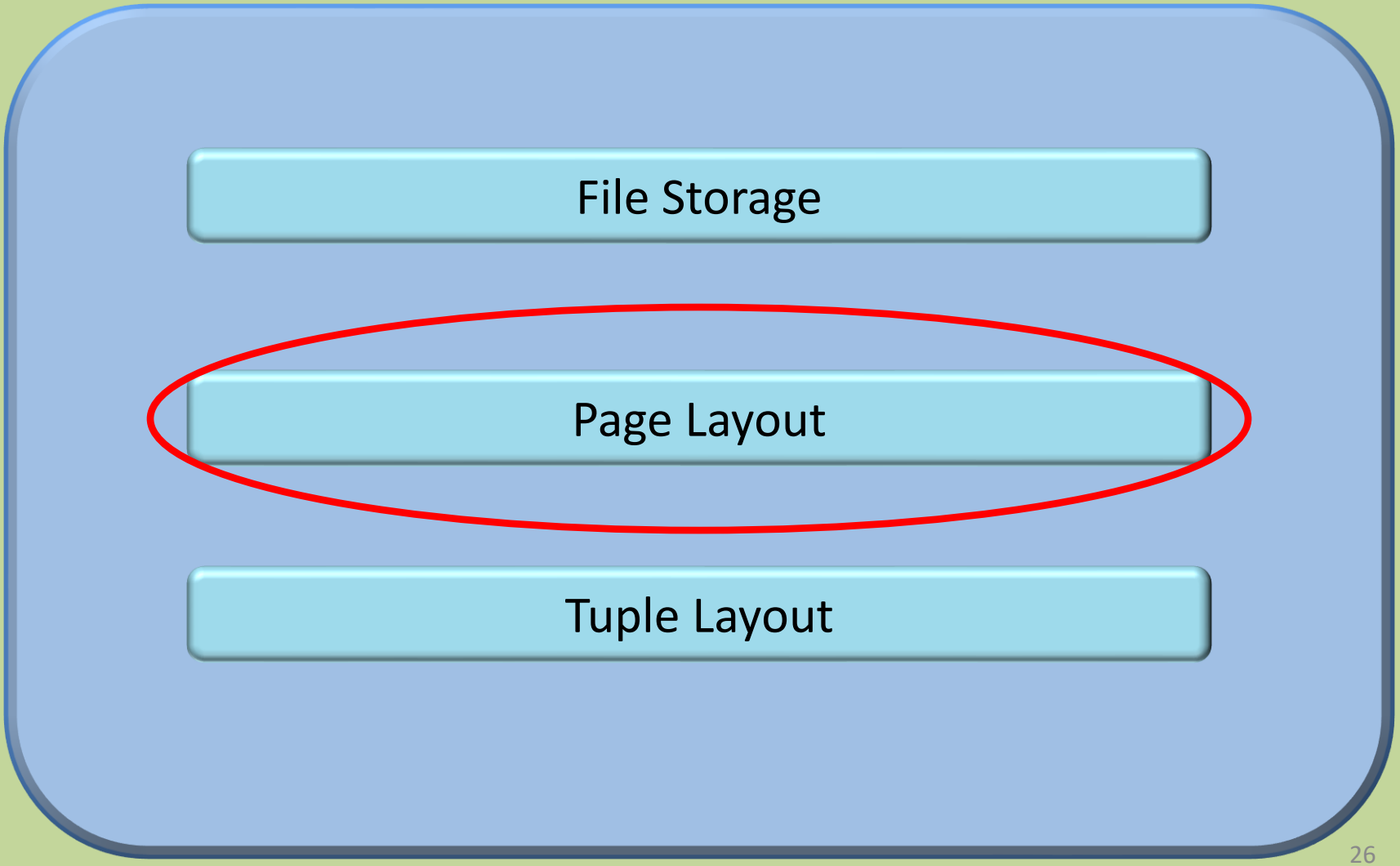
The directory also records the number of free slots per page.

The DBMS has to make sure that the directory pages are in sync with the data pages.



Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure



Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

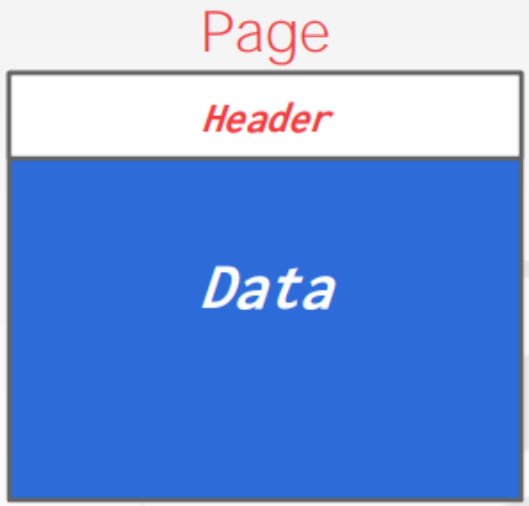
Storage and File Structure – Page Layout

PAGE HEADER

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

→ Page Size
→ Checksum
→ DBMS Version
→ Transaction Visibility
→ Compression Information

Some systems require pages to be self-contained (e.g., Oracle).



Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure – Page Layout

Page Size:

Indicates the size of the page (e.g., 4KB, 8KB, 16KB).
Helps DBMS know where one page ends and next begins.

Checksum:

It is a verification value computed from page contents to detect data corruption during read/write or storage operation.

DBMS Version:

Information about which database version created/last modified the page to ensures compatibility when upgrading DBMS versions. It also prevents older DBMS from misinterpreting newer page formats.

Storage and File Structure – Page Layout

Transaction Visibility:

It tracks which transactions can see data on this page if multiple transactions are running. This is important for maintaining isolation between different transactions. For example,
Transaction A might be reading data and transaction B might be modifying the same data.

Compression Information:

This information indicates if page data is compressed. It also shows which compression algorithm is used.

Storage and File Structure – Page Layout

A Self-Contained Page:

A self-contained page is a database page that contains all the information needed to interpret its contents independently, without requiring references to other pages.

A self-contained page can be read and understood without requiring other pages. This page has no dependencies on other pages for basic interpretation.

Storage and File Structure – Page Layout

PAGE LAYOUT

For any page storage architecture, we now need to understand how to organize the data stored inside of the page.

→ We are still assuming that we are only storing tuples.

Two approaches:

→ Tuple-oriented

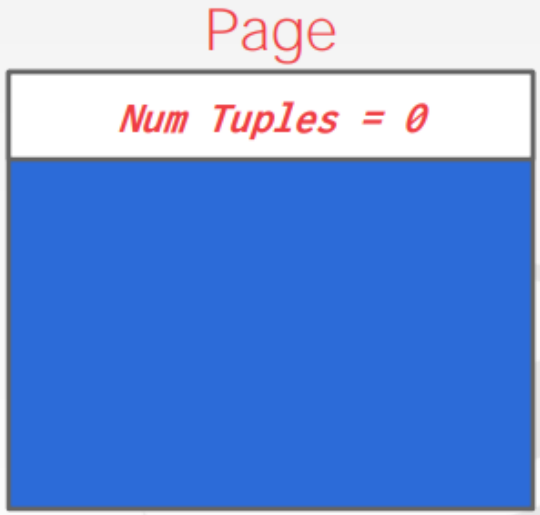
→ Log-structured

Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure – Page Layout

TUPLE STORAGE

How to store tuples in a page?



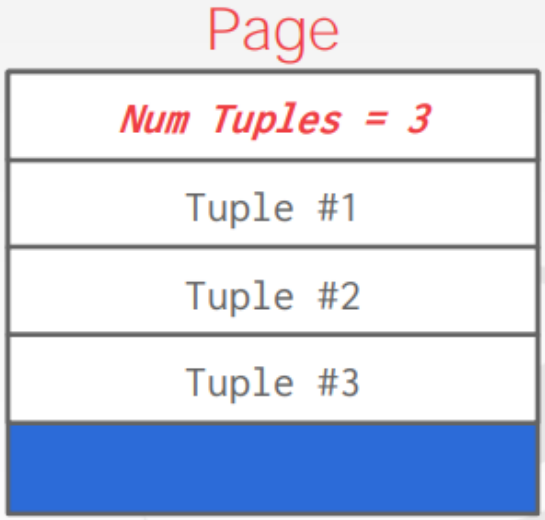
Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure – Page Layout

TUPLE STORAGE

How to store tuples in a page?

Strawman Idea: Keep track of the number of tuples in a page and then just append a new tuple to the end.



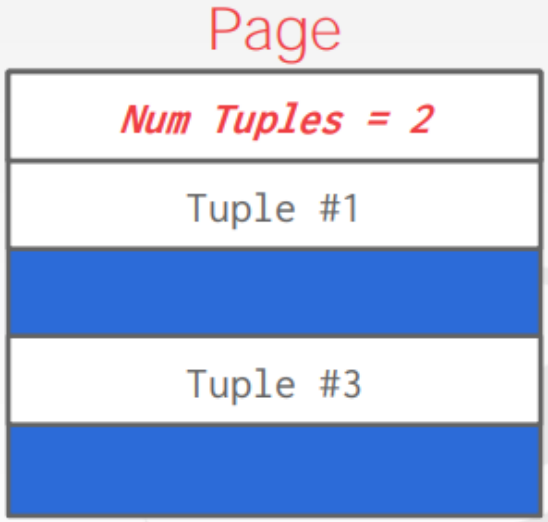
Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure – Page Layout

TUPLE STORAGE

How to store tuples in a page?

Strawman Idea: Keep track of the number of tuples in a page and then just append a new tuple to the end.
→ What happens if we delete a tuple?



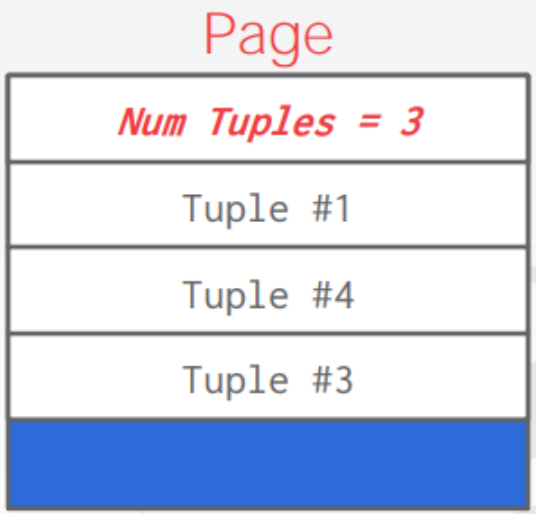
Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure – Page Layout

TUPLE STORAGE

How to store tuples in a page?

Strawman Idea: Keep track of the number of tuples in a page and then just append a new tuple to the end.
→ What happens if we delete a tuple?



Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure – Page Layout

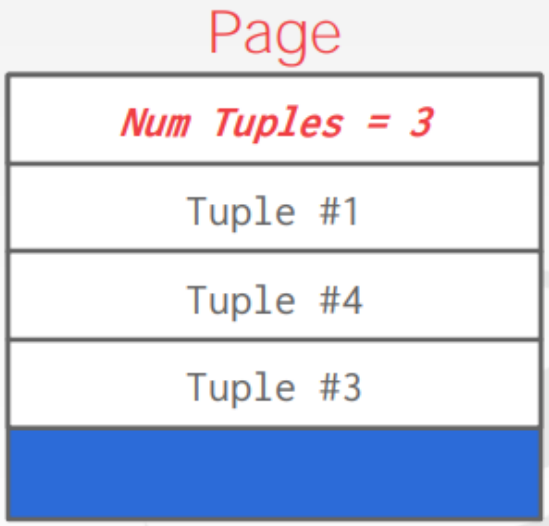
TUPLE STORAGE

How to store tuples in a page?

Strawman Idea: Keep track of the number of tuples in a page and then just append a new tuple to the end.

→ What happens if we delete a tuple?

→ What happens if we have a variable-length attribute?



Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure – Page Layout

SLOTTED PAGES

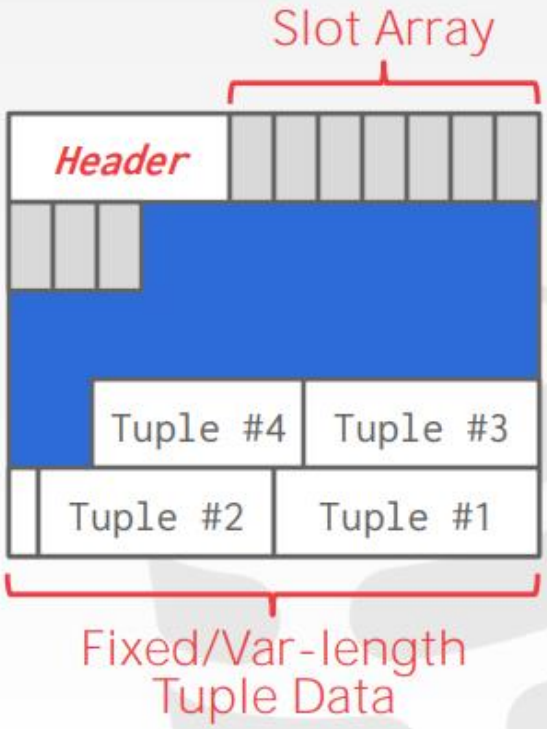
The most common layout scheme is called slotted pages.

The slot array maps "slots" to the tuples' starting position offsets.

The header keeps track of:

→ The # of used slots

→ The offset of the starting location of the last slot used.



Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure – Page Layout

SLOTTED PAGES

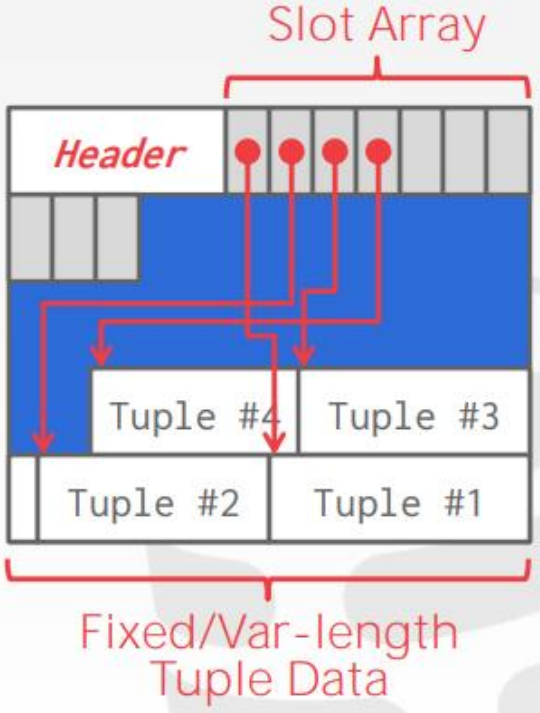
The most common layout scheme is called slotted pages.

The slot array maps "slots" to the tuples' starting position offsets.

The header keeps track of:

→ The # of used slots

→ The offset of the starting location of the last slot used.



Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

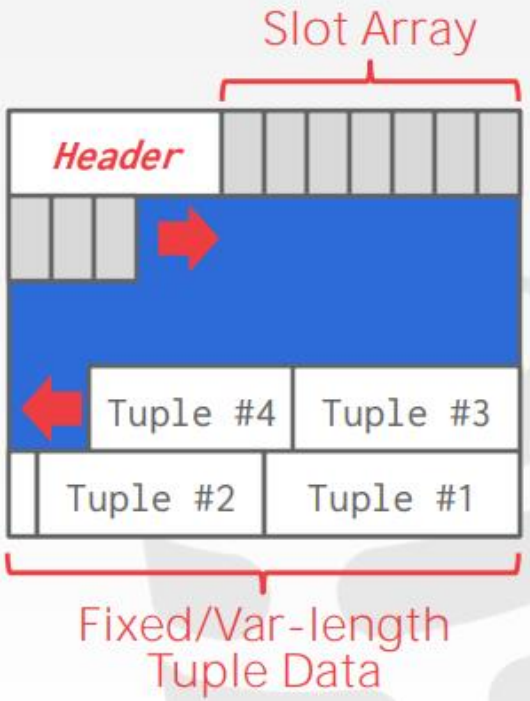
Storage and File Structure – Page Layout

SLOTTED PAGES

The most common layout scheme is called slotted pages.

The slot array maps "slots" to the tuples' starting position offsets.

The header keeps track of:
→ The # of used slots
→ The offset of the starting location of the last slot used.



Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure

File Storage

Page Layout

Tuple Layout

Storage and File Structure – Tuple Layout

TUPLE LAYOUT

A tuple is essentially a sequence of bytes.

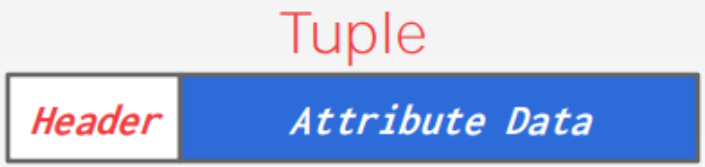
It's the job of the DBMS to interpret those bytes into attribute types and values.

Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure – Tuple Layout

TUPLE HEADER

Each tuple is prefixed with a header that contains meta-data about it.



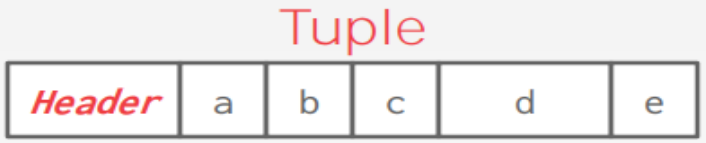
Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure – Tuple Layout

TUPLE DATA

Attributes are typically stored in the order that you specify them when you create the table.

This is done for software engineering reasons.




```
CREATE TABLE foo (
  a INT PRIMARY KEY,
  b INT NOT NULL,
  c INT,
  d DOUBLE,
  e FLOAT
);
```


Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Data Storage Models

N-ary Storage Model (NSM)

Vs

Decomposition Storage Model (DSM)

Data Storage Models

Online Transaction Processing (OLTP)

Vs

Online Analytical Processing (OLAP)

Data Storage Models
Online Transaction Processing (OLTP)

Simple Queries that read and update a small amount of data that is related to a single entity in the database.

This is usually the kind of application that people build first.

Every time we build a new application: a new website, a new mobile application. We are building an OLTP application.

We are getting new information from outside world and injecting it into our database.

Data Storage Models
Online Transaction Processing (OLTP)

We are doing the same operation over and over again. The example is the Amazon Store Front. The website we go to buy stuff is an OLTP application.

I am adding stuff to my cart, I am making purchases, I am updating my account information.

A lot of customers are updating the data but each customer is updating only a small amount of data. A query for a customer is only updating small information.

We are getting new information from outside world and injecting it into our database.

Data Storage Models
Online Transaction Processing (OLTP)

Optimized for transactional superiority instead of data analysis.

A joint bank account of wife and husband.

Both try to withdraw money at the same time.



Data Storage Models
Online Analytical Processing (OLAP)

Complex queries that read large portions of the database spanning multiple entities.

You execute these queries on the data you have collected from you OLTP applications.

Its data analysis not just data processing.

A company may compare the sales in January and February with the sales of the same months for another store in different location which has a separate database.

Data Storage Models
OLTP vs OLAP



OLTP characteristics:

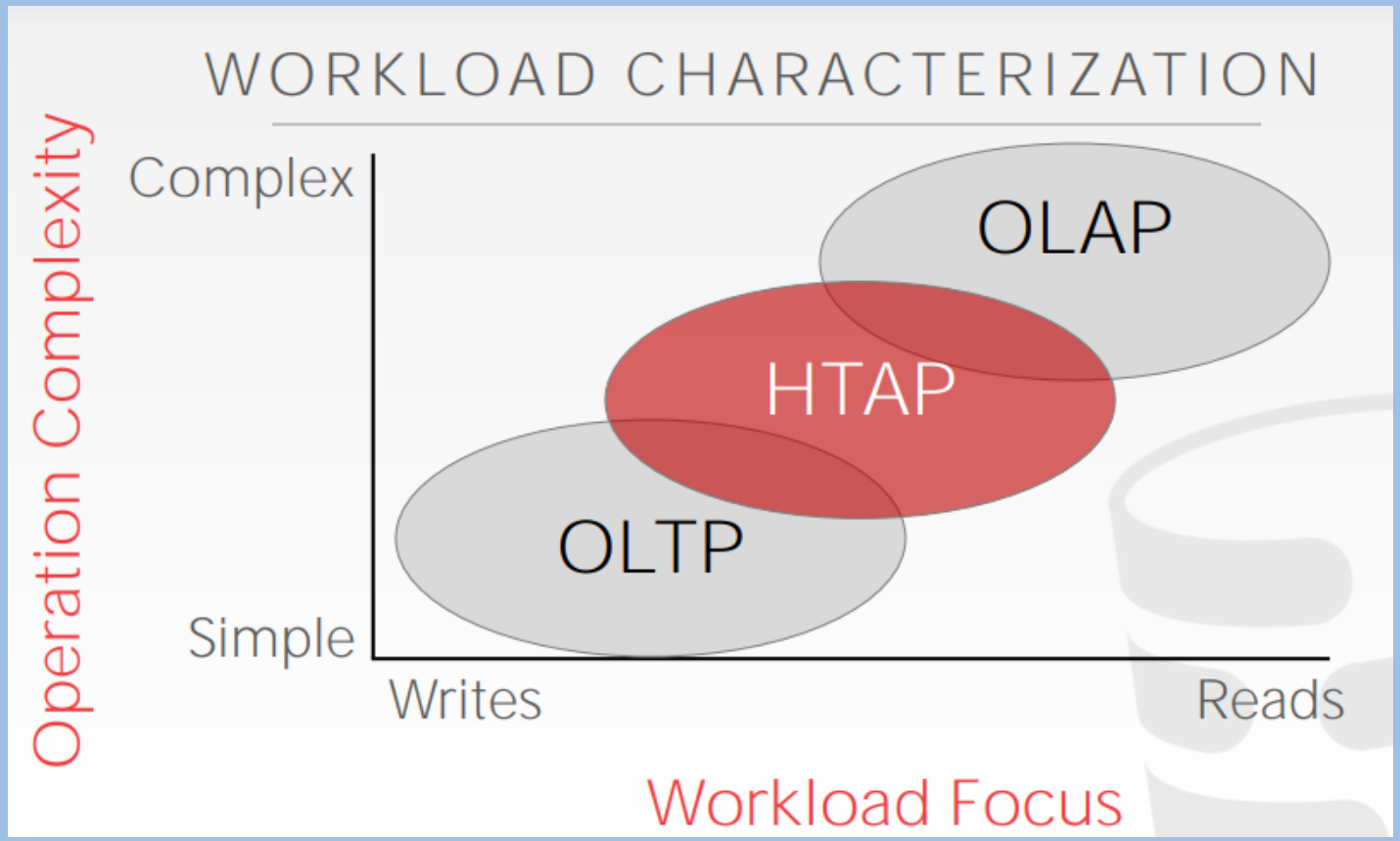
- Many transactions
- Latency sensitive
- Small payloads
- Balanced read/write or Heavy write workloads

OLAP characteristics:

- Few transactions
- Throughput sensitive
- Large (return) payloads
- Heavy read workloads (including full table scans)

https://thecustomizewindows.com/2021/12/what-is-online-transaction-processing/

Data Storage Models
OLTP vs OLAP



Hybrid transaction/analytical processing (**HTAP**) "breaks the wall" between **transaction processing** and analytics.

Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Data Storage Models
N-ary Storage Model (NSM)

DATA STORAGE MODELS

The DBMS can store tuples in different ways that are better for either OLTP or OLAP workloads.

We have been assuming the **n-ary storage model** (aka "row storage") so far this semester.

Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Data Storage Models
N-ary Storage Model (NSM)

N-ARY STORAGE MODEL (NSM)

The DBMS stores all attributes for a single tuple contiguously in a page.

Ideal for OLTP workloads where queries tend to operate only on an individual entity and insert-heavy workloads.

Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Data Storage Models
N-ary Storage Model (NSM)

N-ARY STORAGE MODEL (NSM)

The DBMS stores all attributes for a single tuple contiguously in a page.



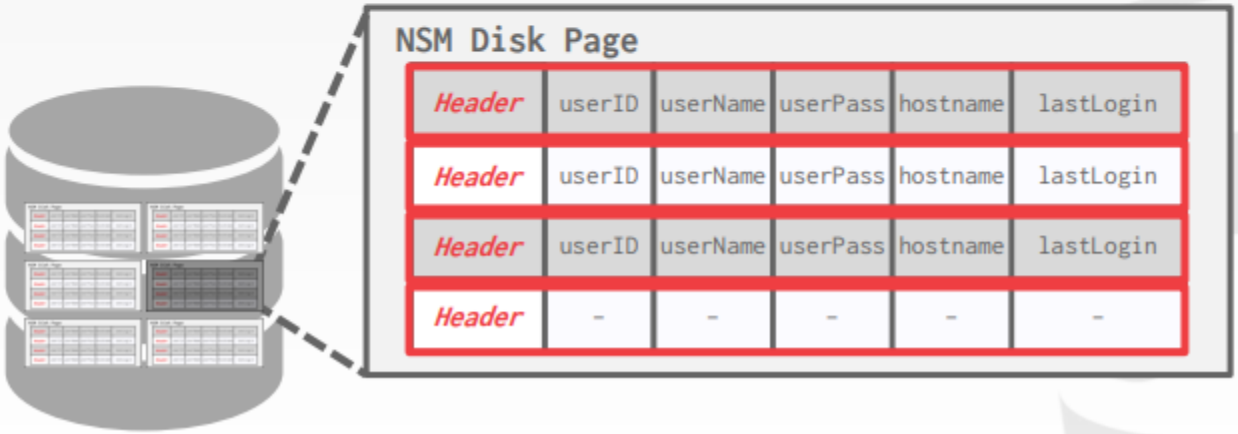
Header	userID	userName	userPass	hostname	lastLogin	←Tuple #1
Header	userID	userName	userPass	hostname	lastLogin	←Tuple #2
Header	userID	userName	userPass	hostname	lastLogin	←Tuple #3
Header	-	-	-	-	-	←Tuple #4

Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Data Storage Models
N-ary Storage Model (NSM)

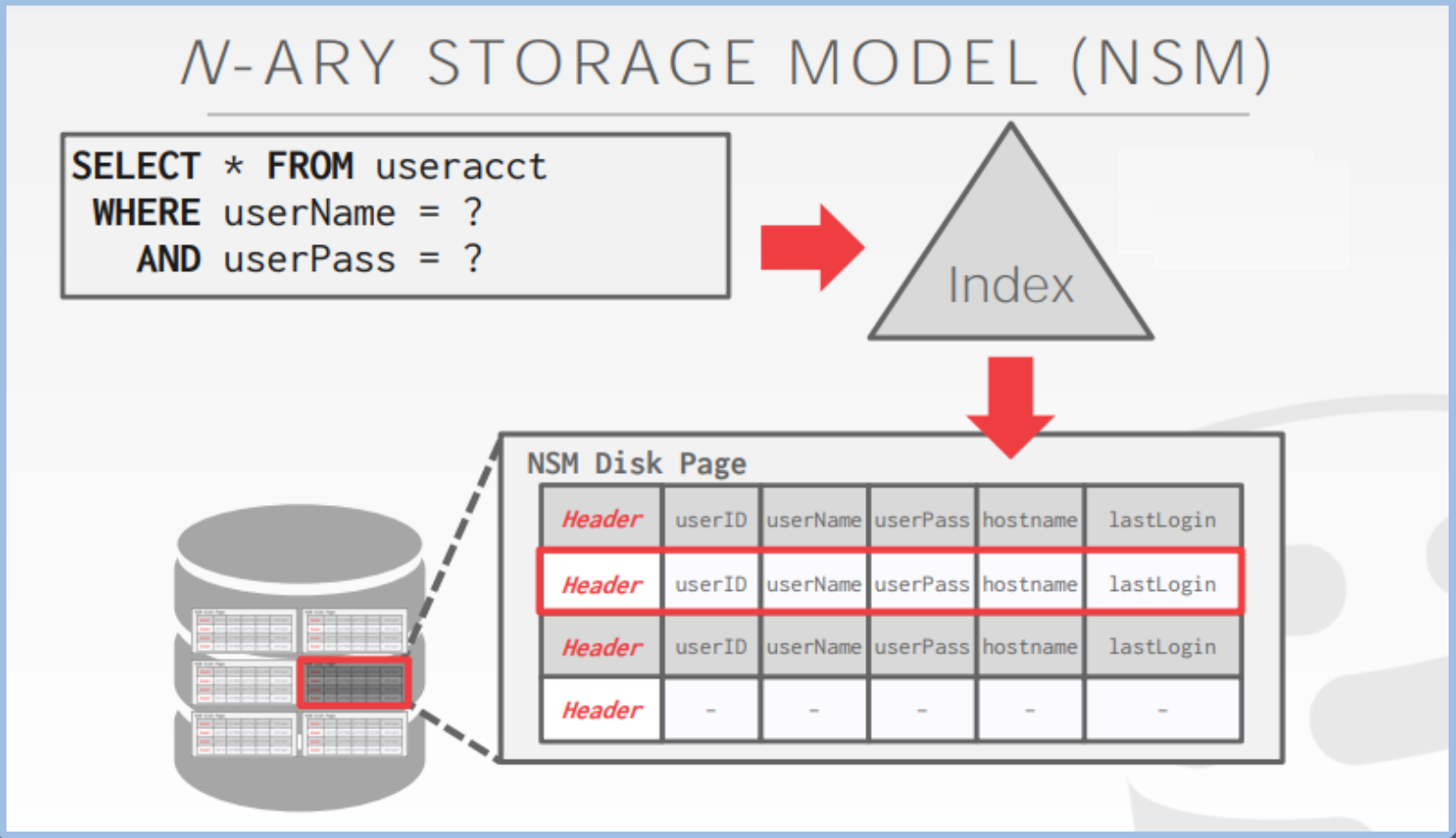
N-ARY STORAGE MODEL (NSM)

The DBMS stores all attributes for a single tuple contiguously in a page.



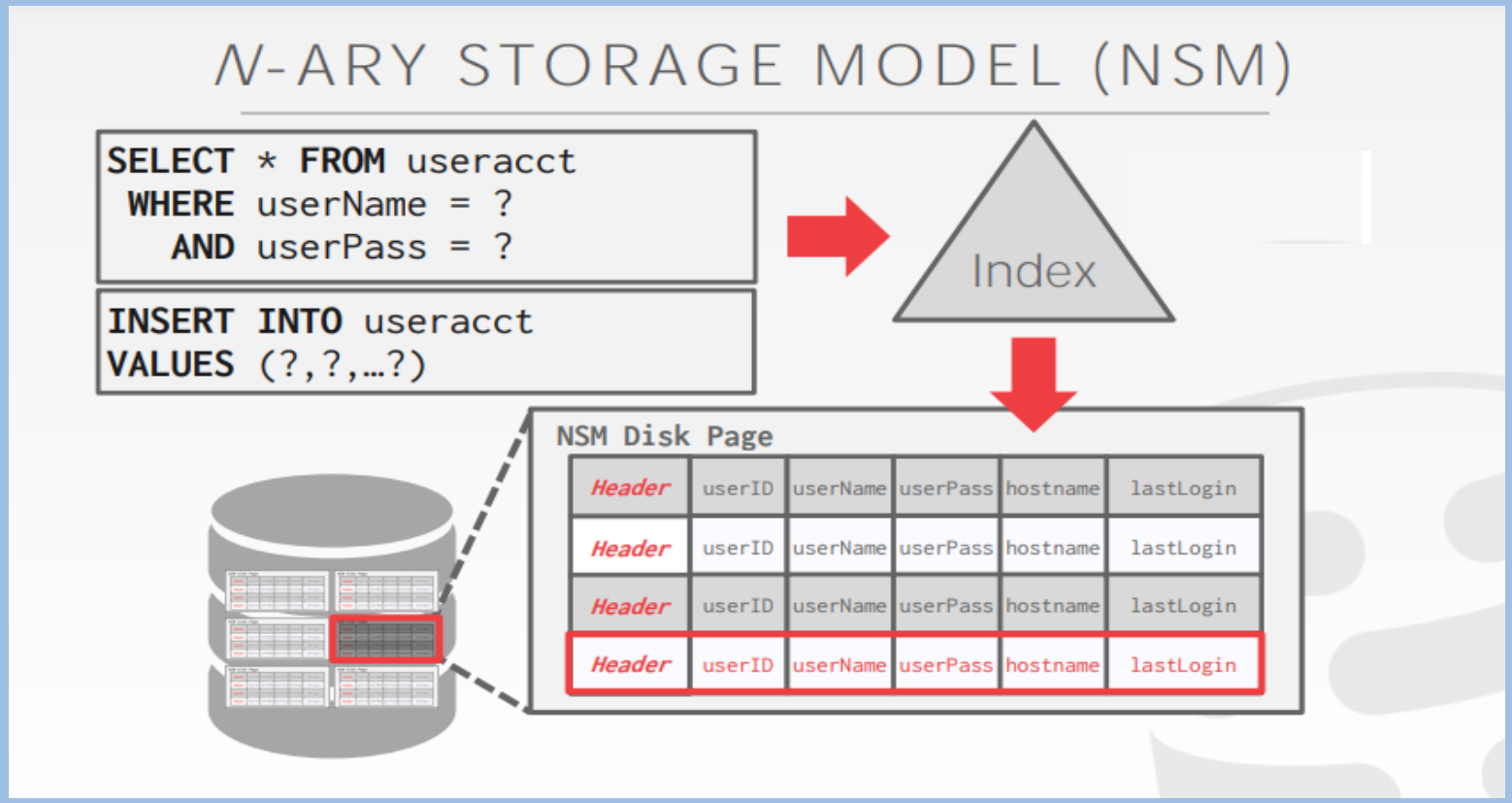
Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Data Storage Models
N-ary Storage Model (NSM)



Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Data Storage Models
N-ary Storage Model (NSM)

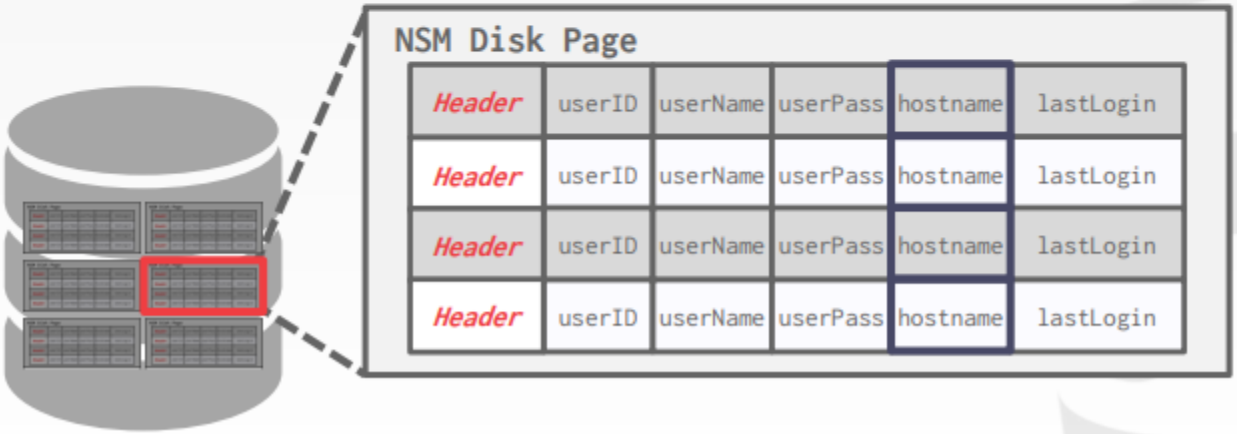


Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Data Storage Models
N-ary Storage Model (NSM)

N-ARY STORAGE MODEL (NSM)

```
SELECT COUNT(U.lastLogin),
       EXTRACT(month FROM U.lastLogin) AS month
  FROM useracct AS U
 WHERE U.hostname LIKE '%.gov'
 GROUP BY EXTRACT(month FROM U.lastLogin)
```

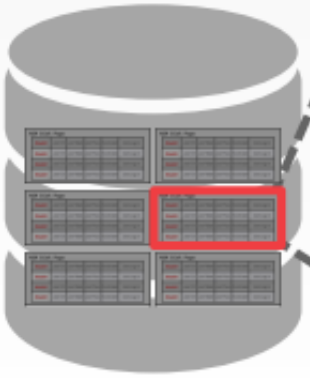


Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Data Storage Models
N-ary Storage Model (NSM)

N-ARY STORAGE MODEL (NSM)

```
SELECT COUNT(U.lastLogin),
       EXTRACT(month FROM U.lastLogin) AS month
  FROM useracct AS U
 WHERE U.hostname LIKE '%.gov'
 GROUP BY EXTRACT(month FROM U.lastLogin)
```

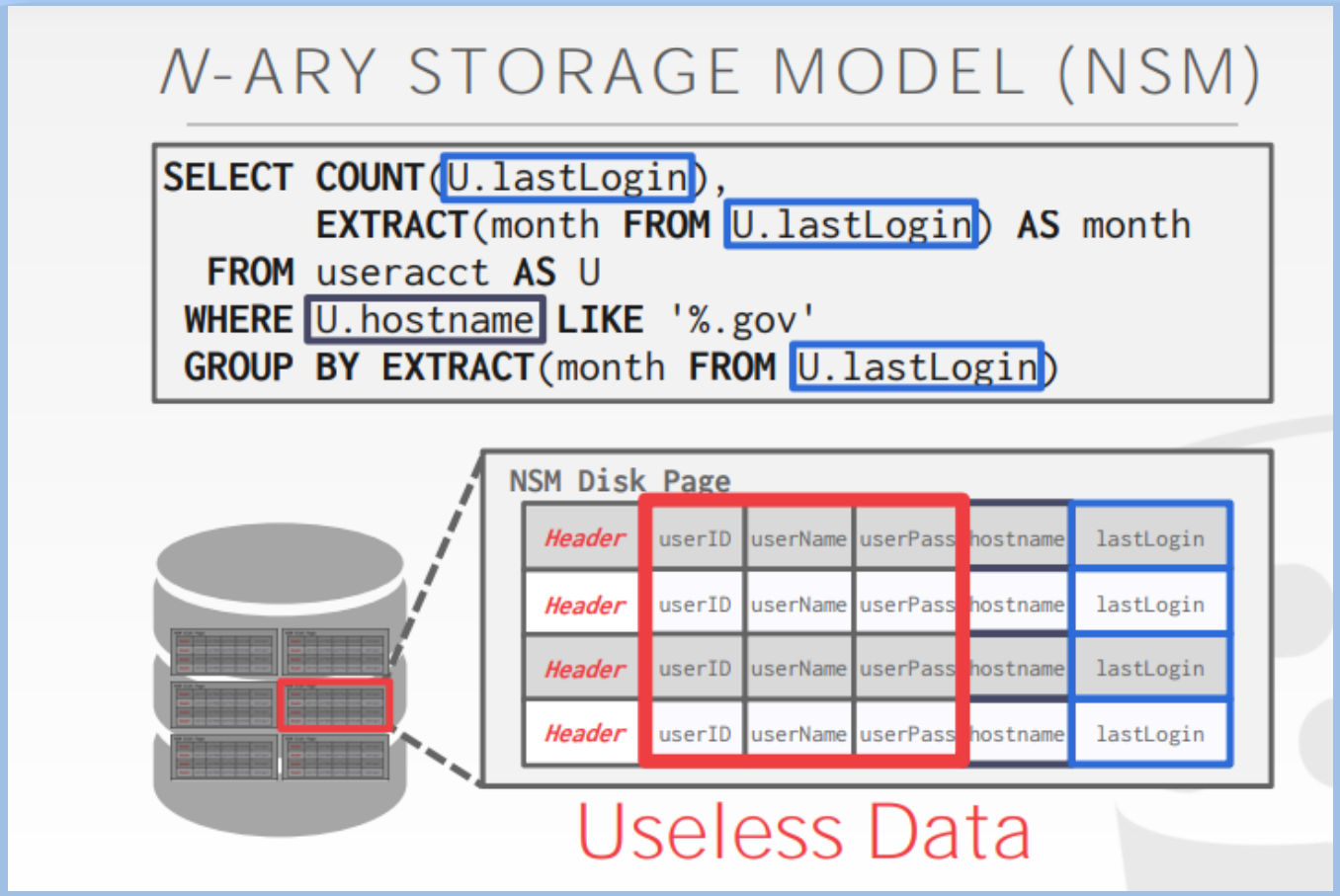


NSM Disk Page

Header	userID	userName	userPass	hostname	lastLogin
Header	userID	userName	userPass	hostname	lastLogin
Header	userID	userName	userPass	hostname	lastLogin
Header	userID	userName	userPass	hostname	lastLogin

Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Data Storage Models
N-ary Storage Model (NSM)



The system will also transfer useless data from disk to memory. There is no way to do it separately.

Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Data Storage Models
N-ary Storage Model (NSM)

N-ARY STORAGE MODEL

Advantages

→ Fast inserts, updates, and deletes.
→ Good for queries that need the entire tuple.

Disadvantages

→ Not good for scanning large portions of the table and/or a subset of the attributes.

Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Data Storage Models
Decomposition Storage Model (DSM)

DECOMPOSITION STORAGE MODEL (DSM)

The DBMS stores the values of a single attribute for all tuples contiguously in a page.
→ Also known as a "column store".

Ideal for OLAP workloads where read-only queries perform large scans over a subset of the table's attributes.

Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Data Storage Models
Decomposition Storage Model (DSM)

DECOMPOSITION STORAGE MODEL (DSM)

The DBMS stores the values of a single attribute for all tuples contiguously in a page.
→ Also known as a "column store".



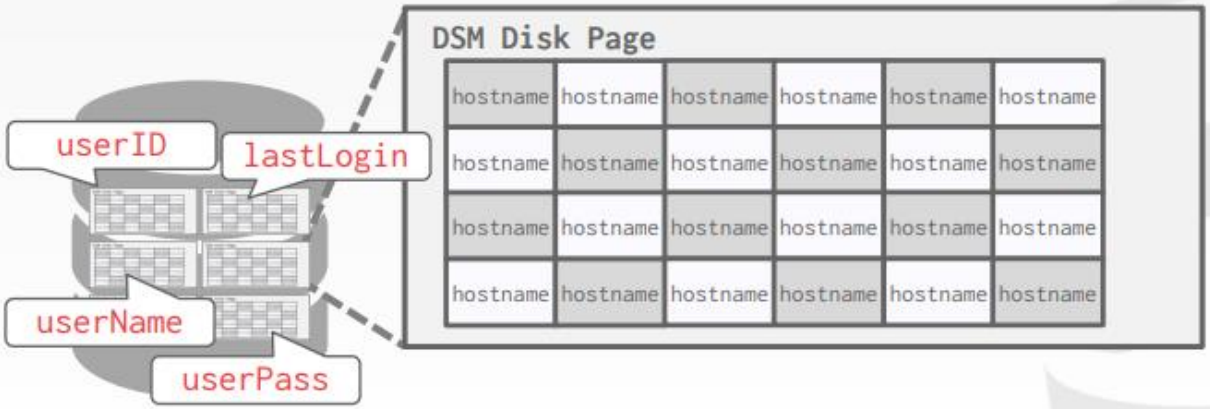
Header	userID	userName	userPass	hostname	lastLogin
Header	userID	userName	userPass	hostname	lastLogin
Header	userID	userName	userPass	hostname	lastLogin
Header	userID	userName	userPass	hostname	lastLogin

Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Data Storage Models
Decomposition Storage Model (DSM)

DECOMPOSITION STORAGE MODEL (DSM)

The DBMS stores the values of a single attribute for all tuples contiguously in a page.
→ Also known as a "column store".



Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Data Storage Models
Decomposition Storage Model (DSM)

DECOMPOSITION STORAGE MODEL (DSM)

```
SELECT COUNT(U.lastLogin),
       EXTRACT(month FROM U.lastLogin) AS month
  FROM useracct AS U
 WHERE U.hostname LIKE '%.gov'
 GROUP BY EXTRACT(month FROM U.lastLogin)
```

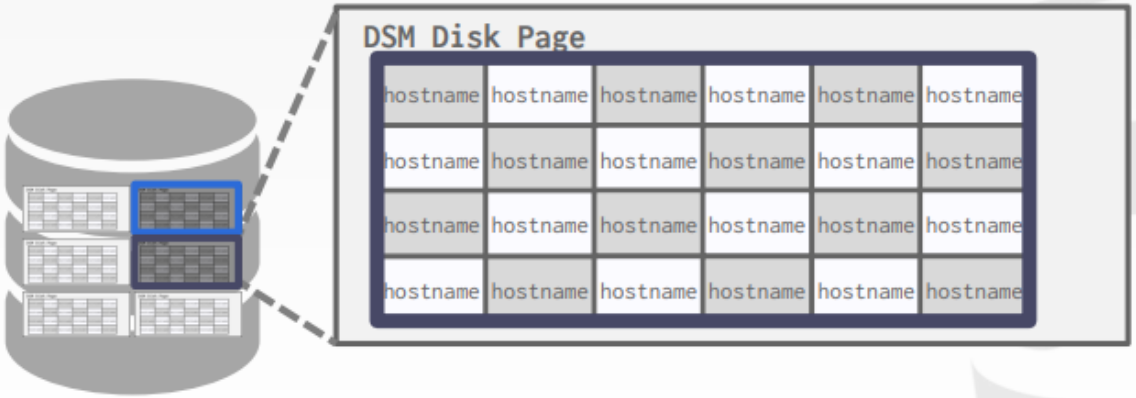


Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Data Storage Models
Decomposition Storage Model (DSM)

DECOMPOSITION STORAGE MODEL (DSM)

```
SELECT COUNT(U.lastLogin),
       EXTRACT(month FROM U.lastLogin) AS month
  FROM useracct AS U
 WHERE U.hostname LIKE '%.gov'
 GROUP BY EXTRACT(month FROM U.lastLogin)
```



Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure

DATABASE STORAGE

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

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

From slides 71 to end.

Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure

DATABASE STORAGE

Spatial Control:

→ Where to write pages on disk.

→ The goal is to keep pages that are used together often as physically close together as possible on disk.

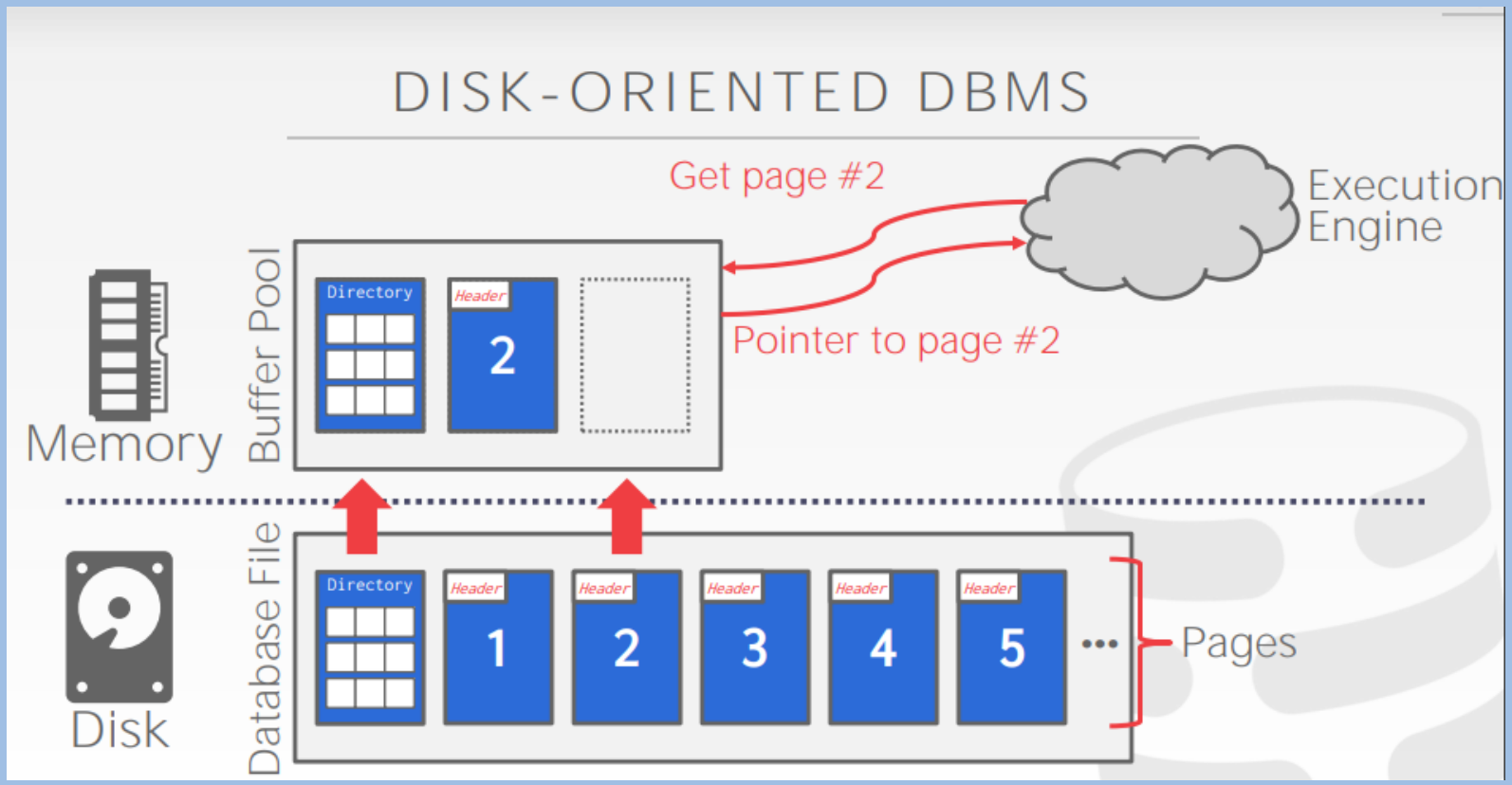
Temporal Control:

→ When to read pages into memory, and when to write them to disk.

→ The goal is minimize the number of stalls from having to read data from disk.

Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure



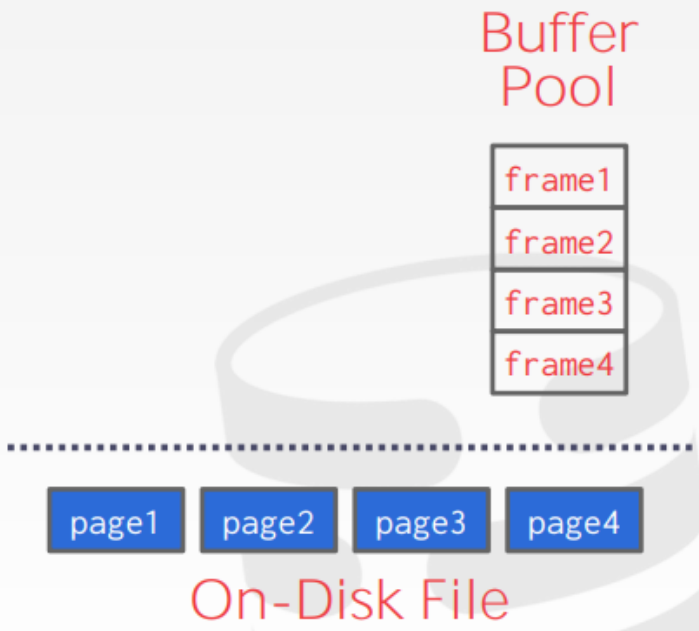
Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure
Buffer Pool Organization

BUFFER POOL ORGANIZATION

Memory region organized as an array of fixed-size pages.
An array entry is called a **frame**.

When the DBMS requests a page, an exact copy is placed into one of these frames.



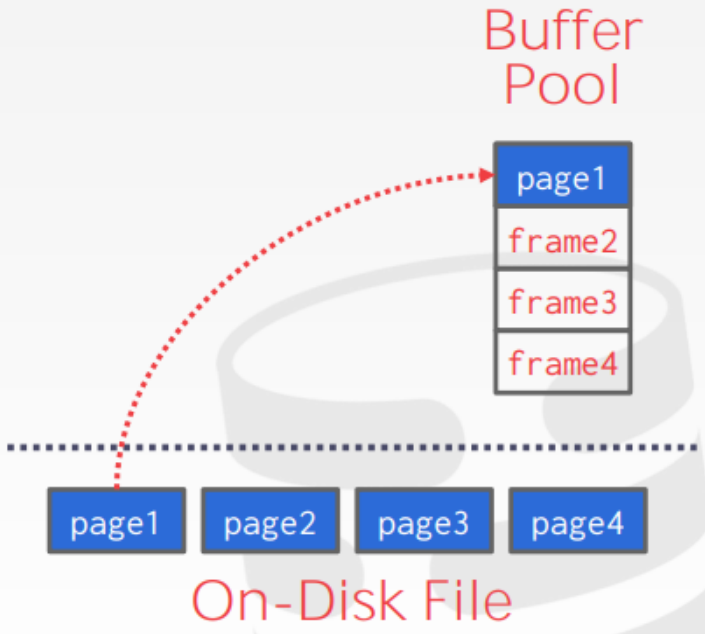
Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure
Buffer Pool Organization

BUFFER POOL ORGANIZATION

Memory region organized as an array of fixed-size pages.
An array entry is called a **frame**.

When the DBMS requests a page, an exact copy is placed into one of these frames.



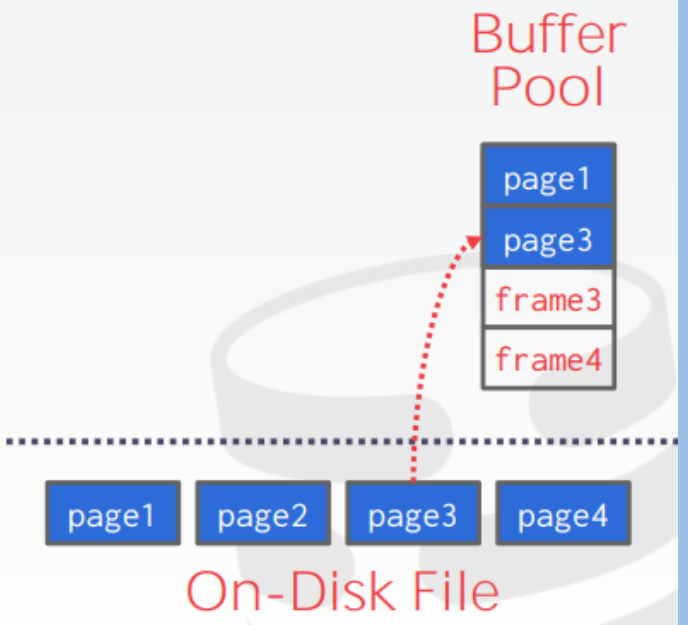
Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure
Buffer Pool Organization

BUFFER POOL ORGANIZATION

Memory region organized as an array of fixed-size pages.
An array entry is called a **frame**.

When the DBMS requests a page, an exact copy is placed into one of these frames.



The pages in the buffer pool are not in the same order as they are on the disk.
Page table will take care of the tracking and mapping.

Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

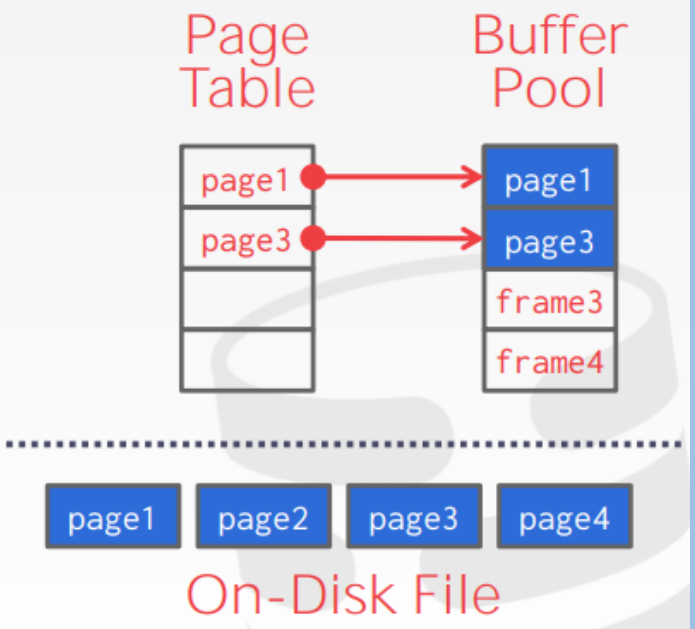
Storage and File Structure
Buffer Pool Organization

BUFFER POOL META-DATA

The **page table** keeps track of pages that are currently in memory.

Also maintains additional meta-data per page:
→ **Dirty Flag**

Dirty flag is a bit that shows whether the page has been modified since it has been fetched from desk.
Did some query or transaction made change to it while being in memory.



Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure
Buffer Pool Organization

MULTIPLE BUFFER POOLS

The DBMS does not always have a single buffer pool for the entire system.

→ Multiple buffer pool instances
→ Per-database buffer pool

Helps reduce latch contention and improve locality.

Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure
Buffer Pool Organization

Multiple buffer pools in memory help reduce latch contention and improve locality.

Multiple buffer pools help databases perform better by solving two key problems. First, they **reduce latch contention** - instead of all transactions competing for one big buffer pool, transactions are distributed across multiple pools, allowing more parallel operations.

Second, they **improve locality** by keeping related data together - for example, one pool can store index pages while another stores data pages, which reduces cache pollution and ensures frequently accessed data stays in memory longer.

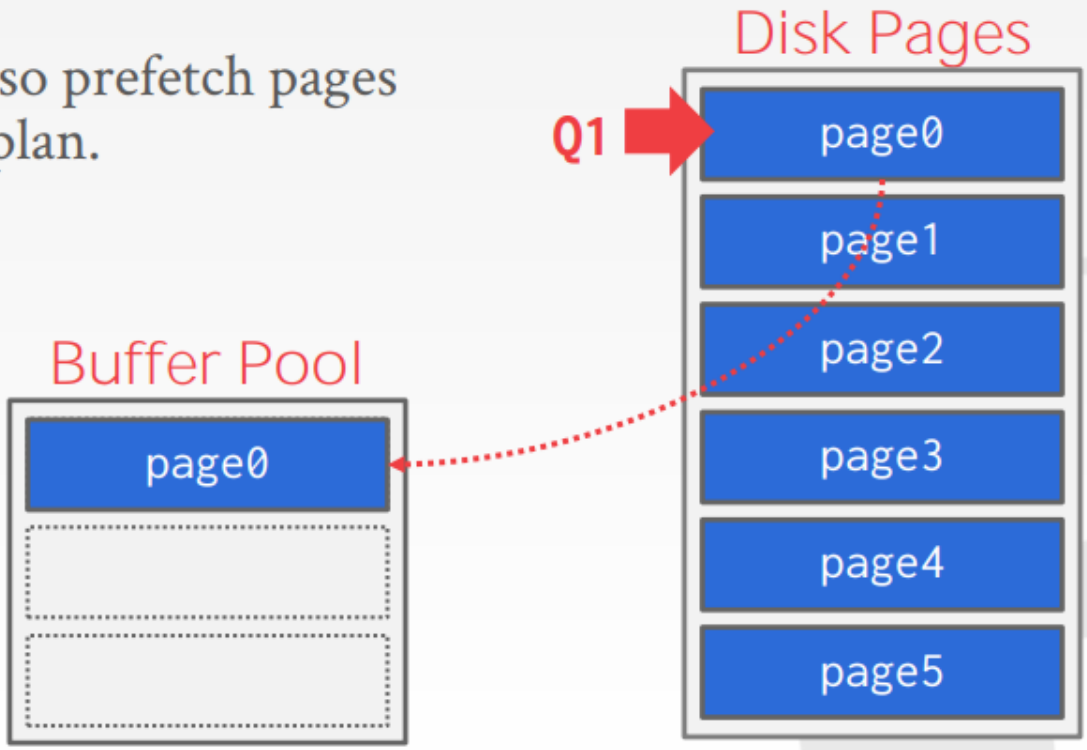
This partitioning approach is like having multiple checkout lanes at a store instead of one: it reduces waiting time (contention) and you can optimize each lane for different purposes (locality), resulting in better overall operation.

Storage and File Structure
Buffer Pool Optimization

PRE-FETCHING

The DBMS can also prefetch pages based on a query plan.

→ Sequential Scans
→ Index Scans



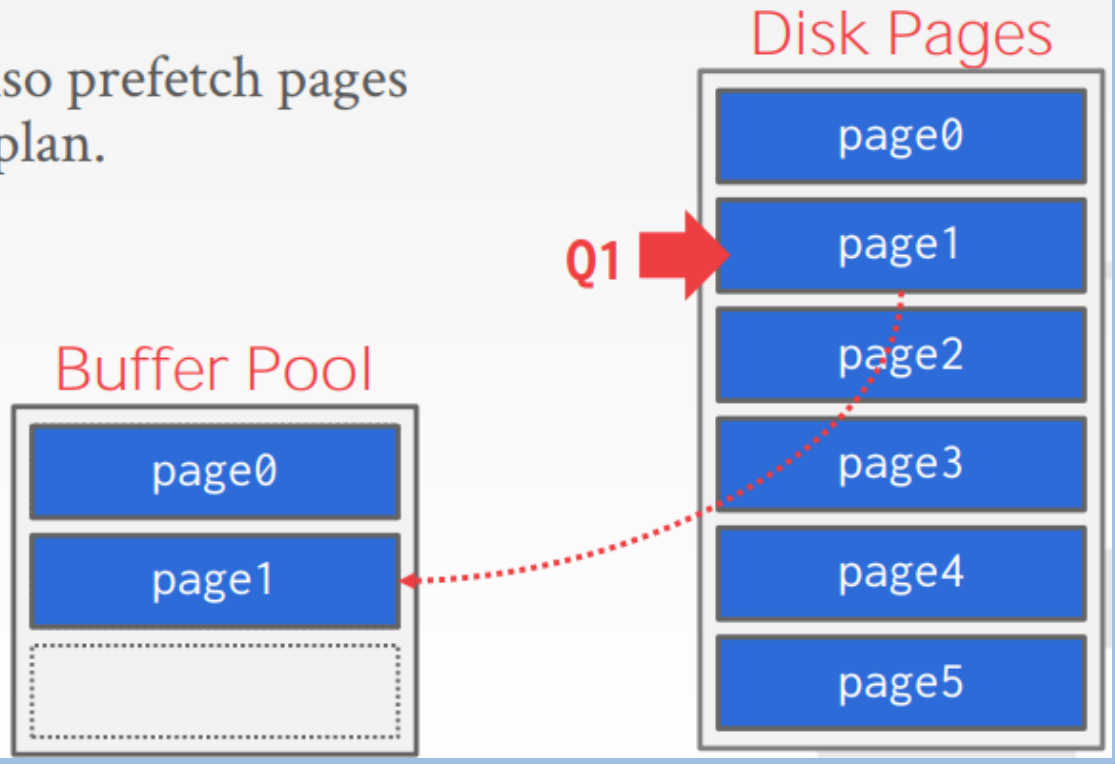
Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure
Buffer Pool Optimization

PRE-FETCHING

The DBMS can also prefetch pages based on a query plan.

→ Sequential Scans
→ Index Scans

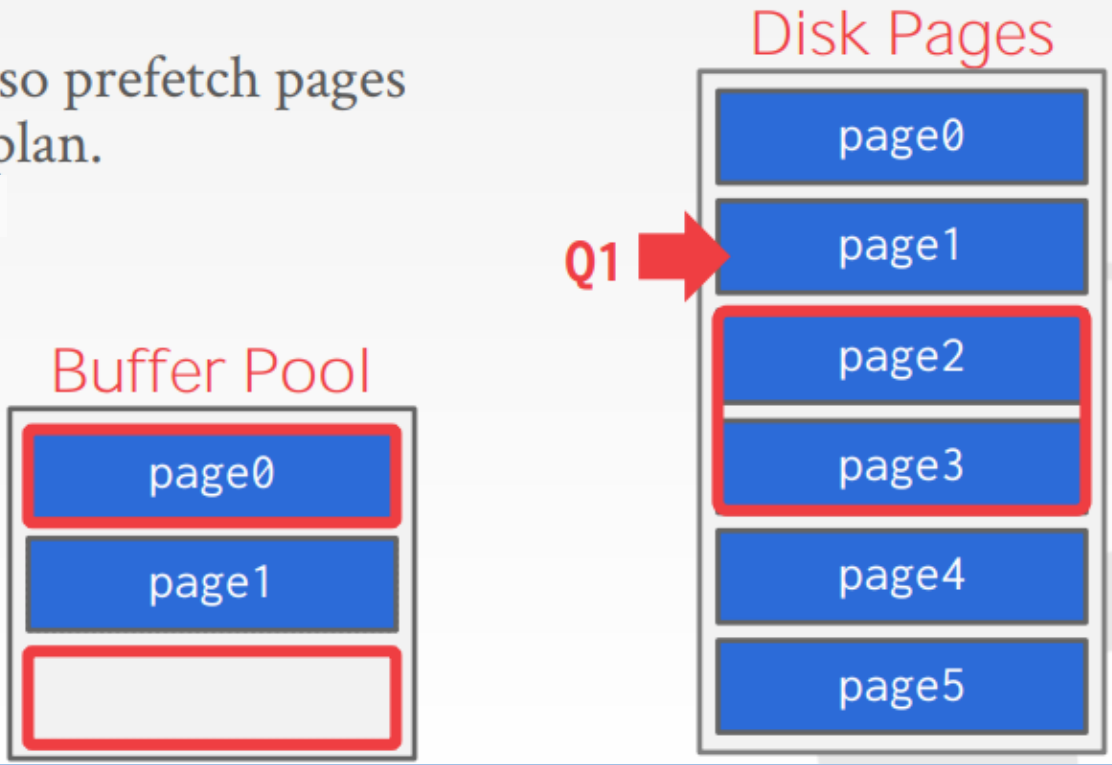


Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure
Buffer Pool Optimization

PRE-FETCHING

The DBMS can also prefetch pages based on a query plan.

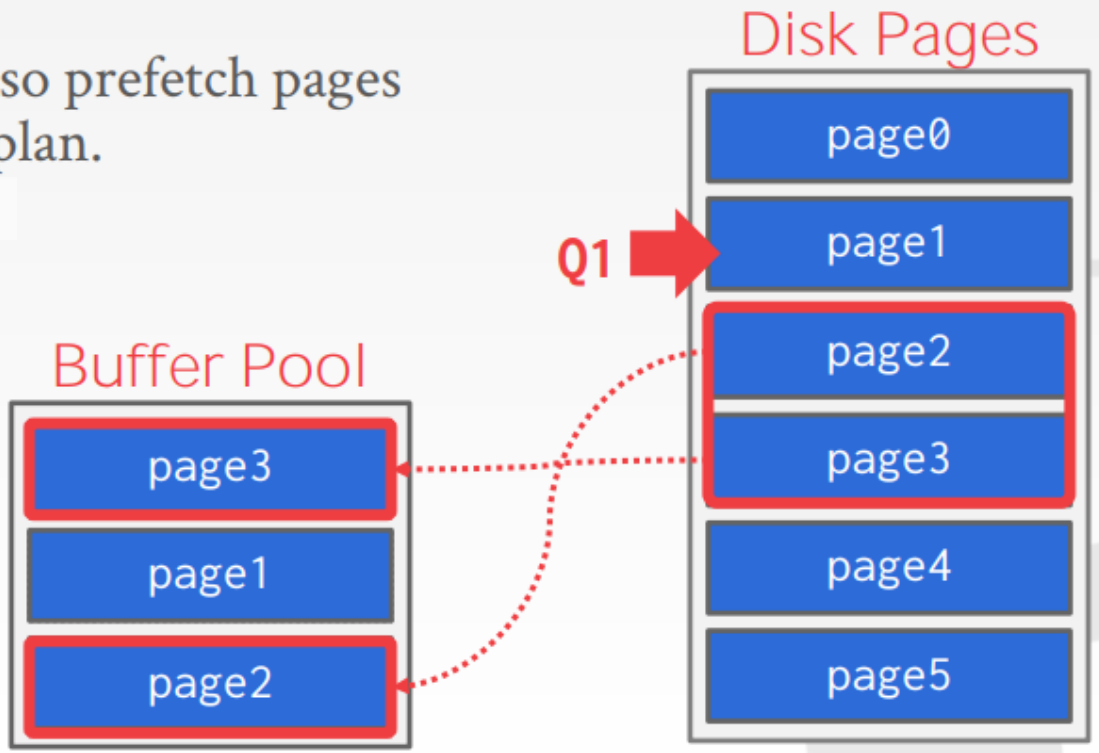


Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure
Buffer Pool Optimization

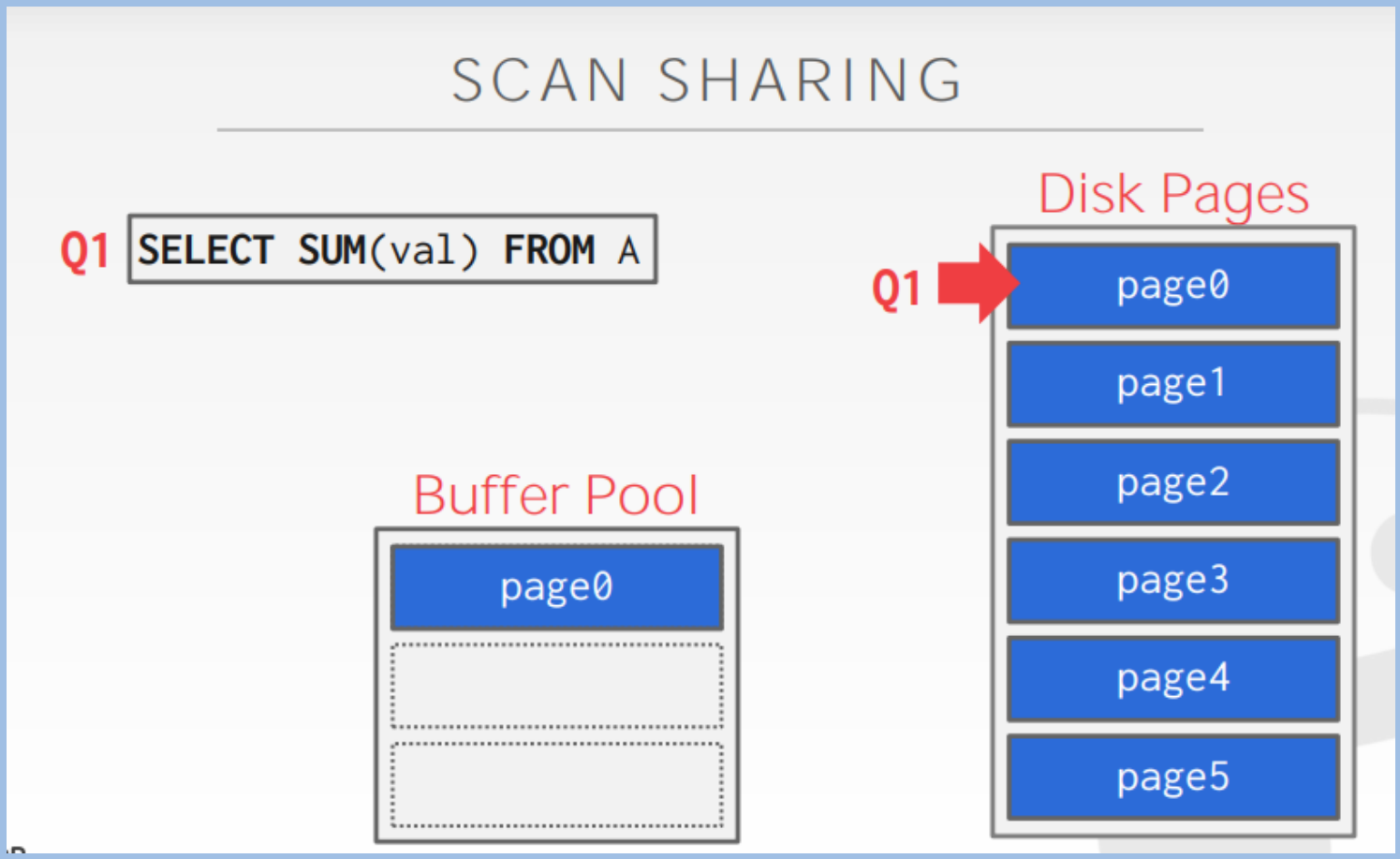
PRE-FETCHING

The DBMS can also prefetch pages based on a query plan.



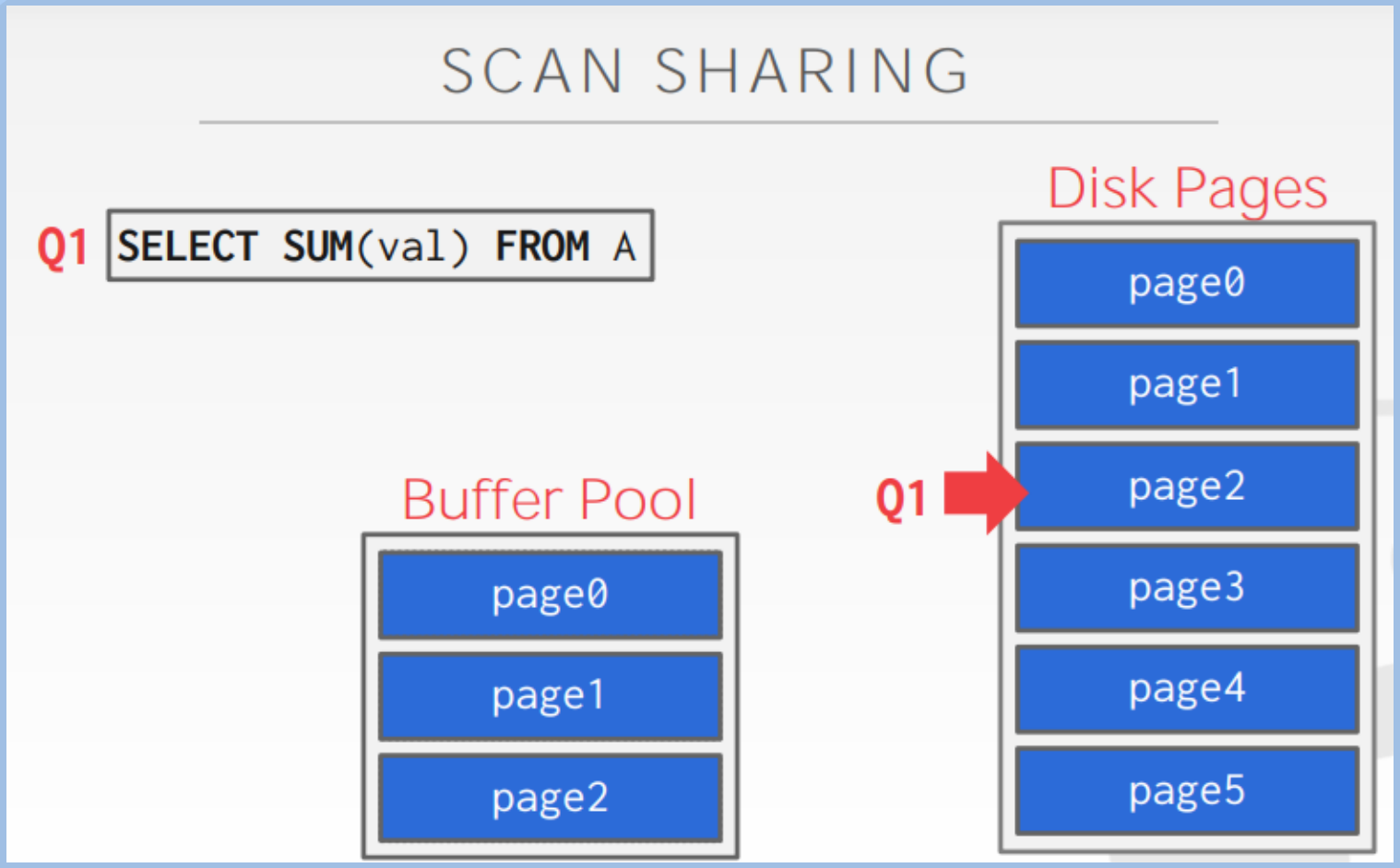
Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure
Buffer Pool Optimization



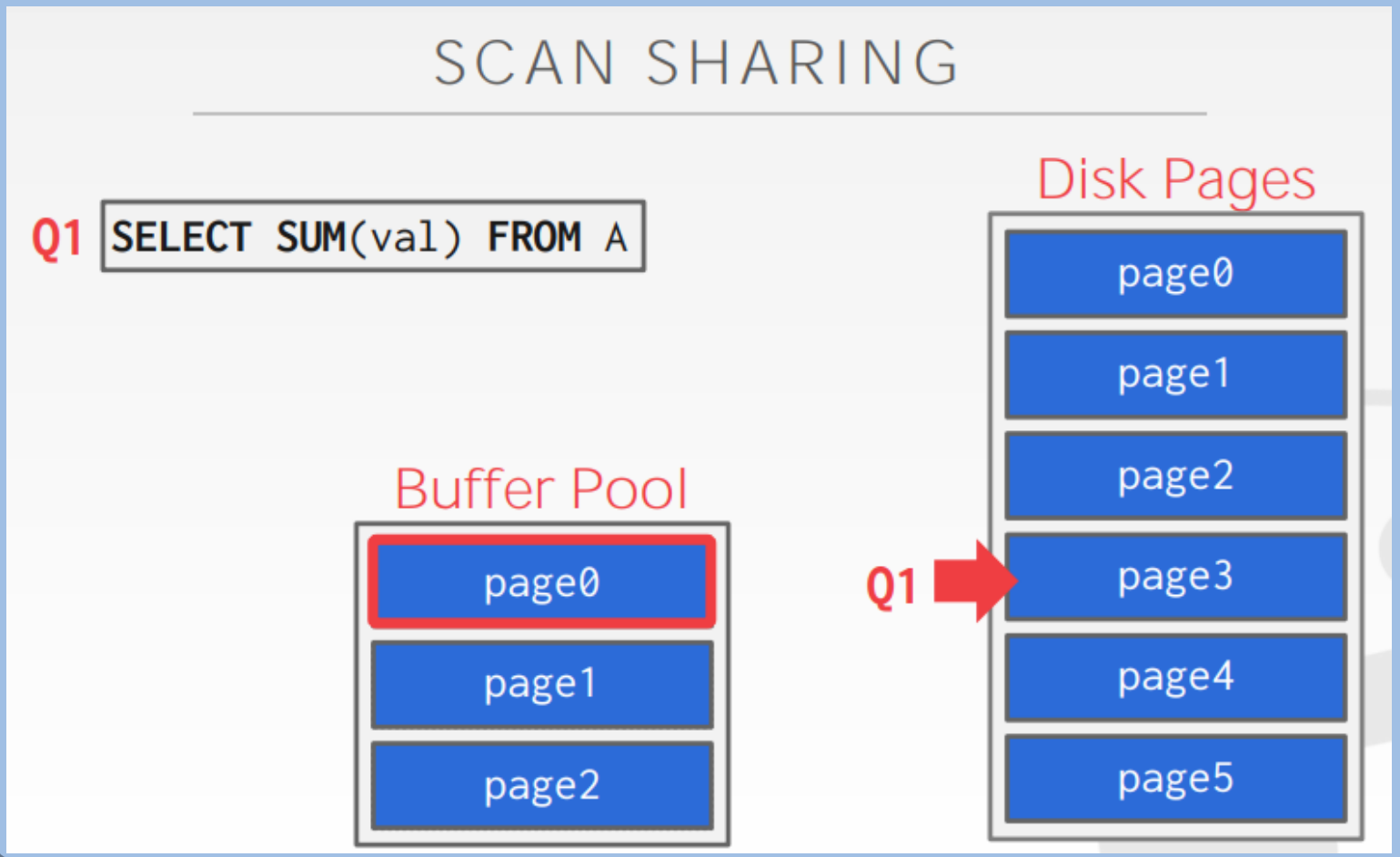
Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure
Buffer Pool Optimization



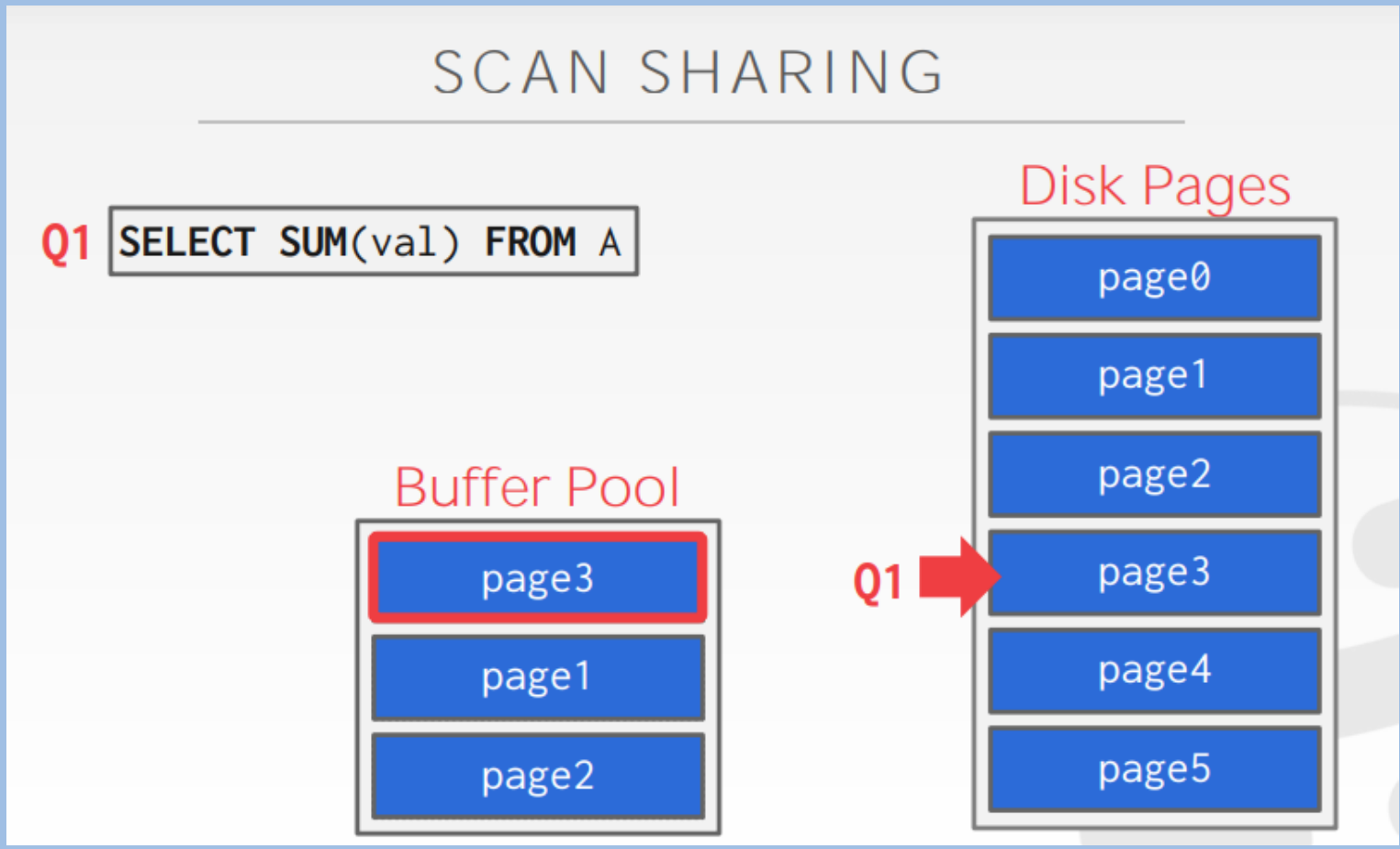
Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure
Buffer Pool Optimization



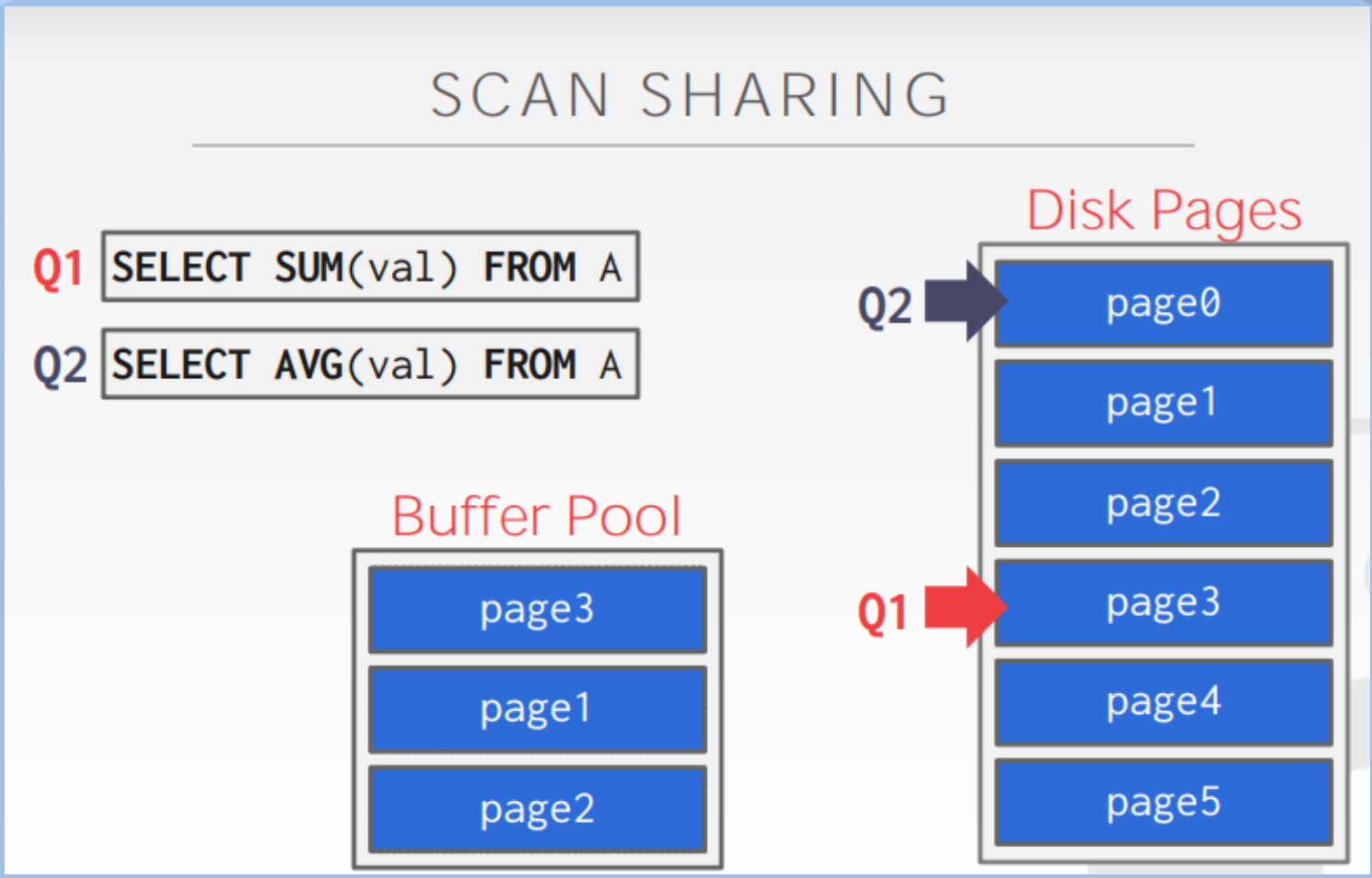
Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure
Buffer Pool Optimization



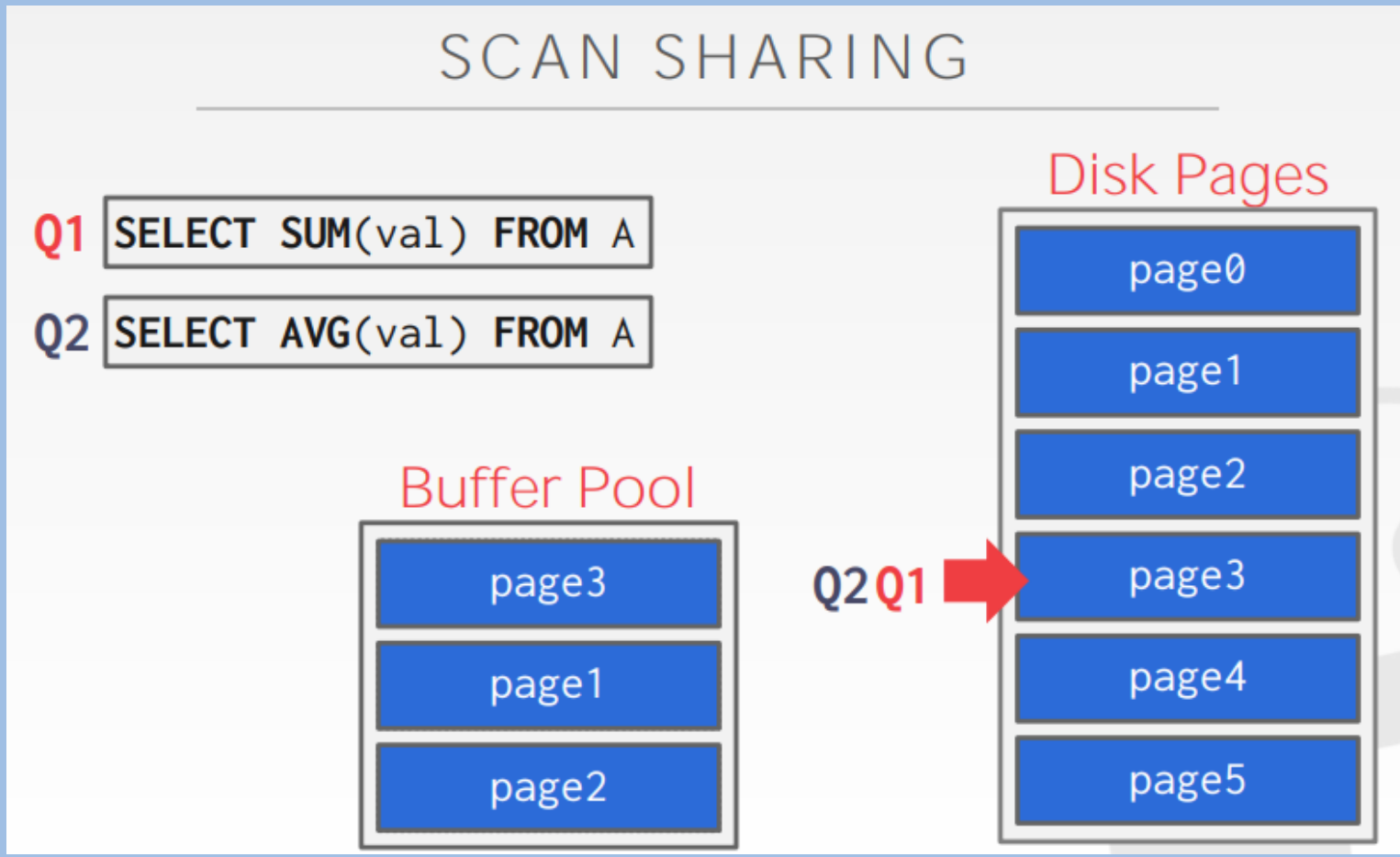
Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure
Buffer Pool Optimization



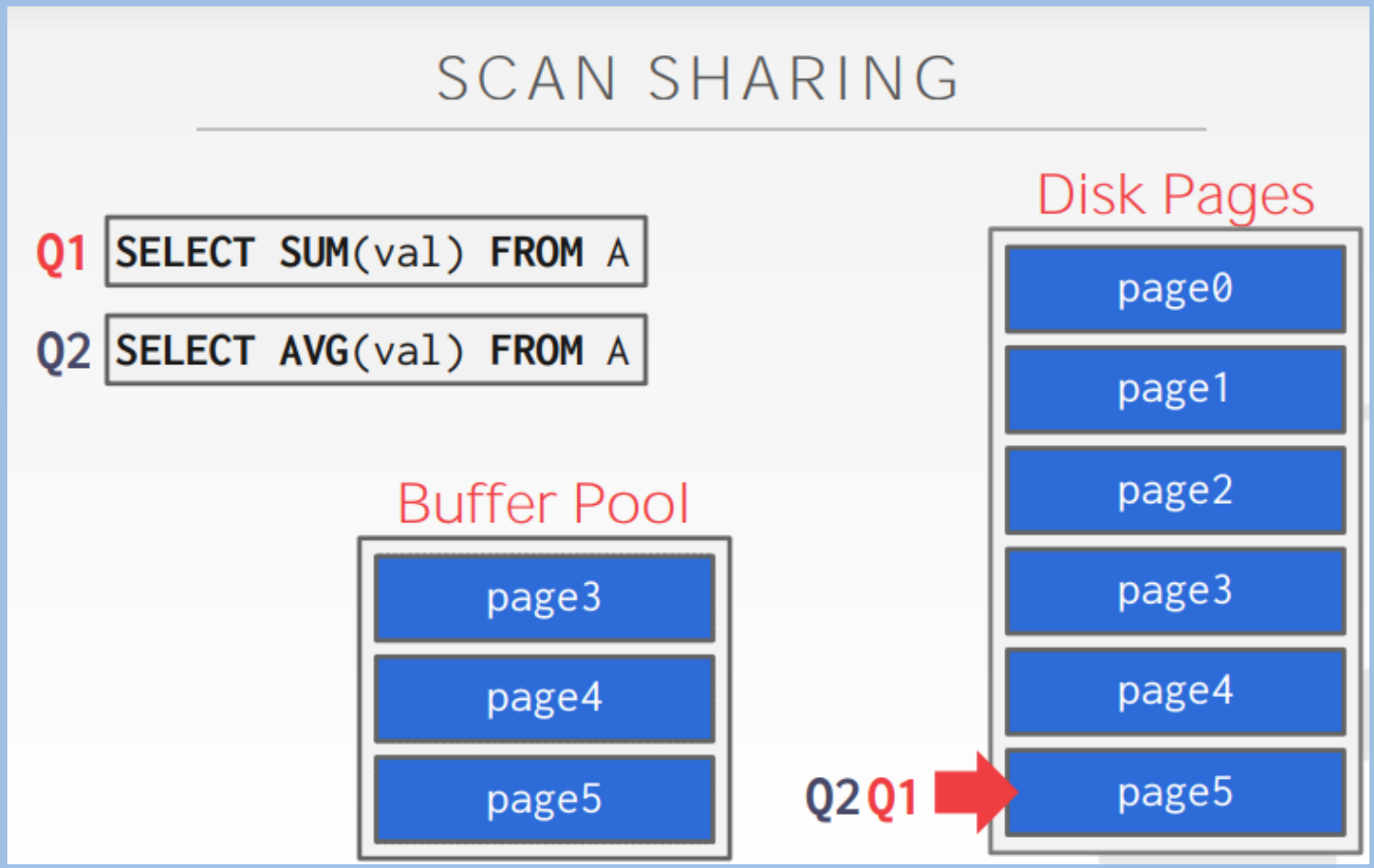
Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure
Buffer Pool Optimization



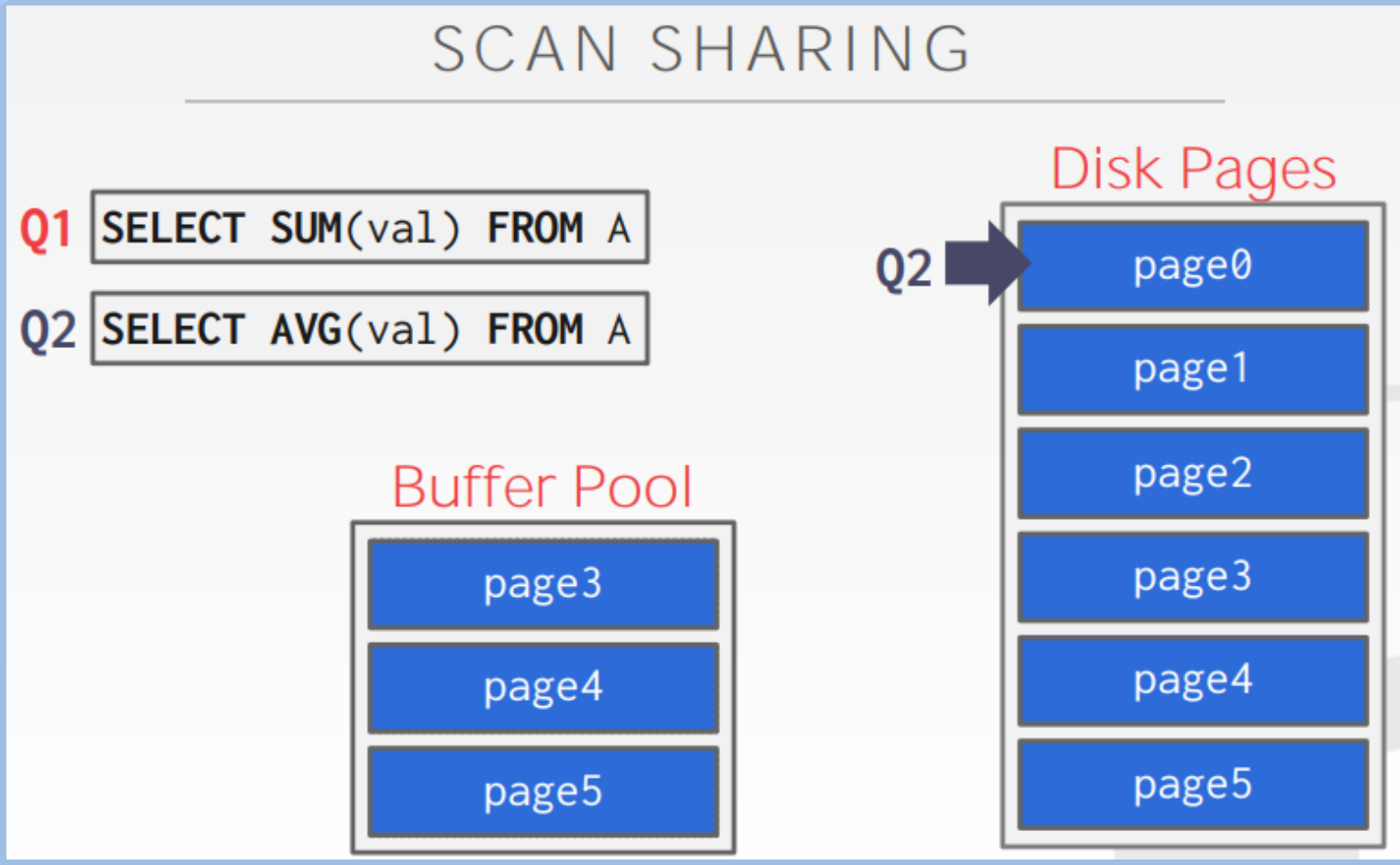
Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure
Buffer Pool Optimization



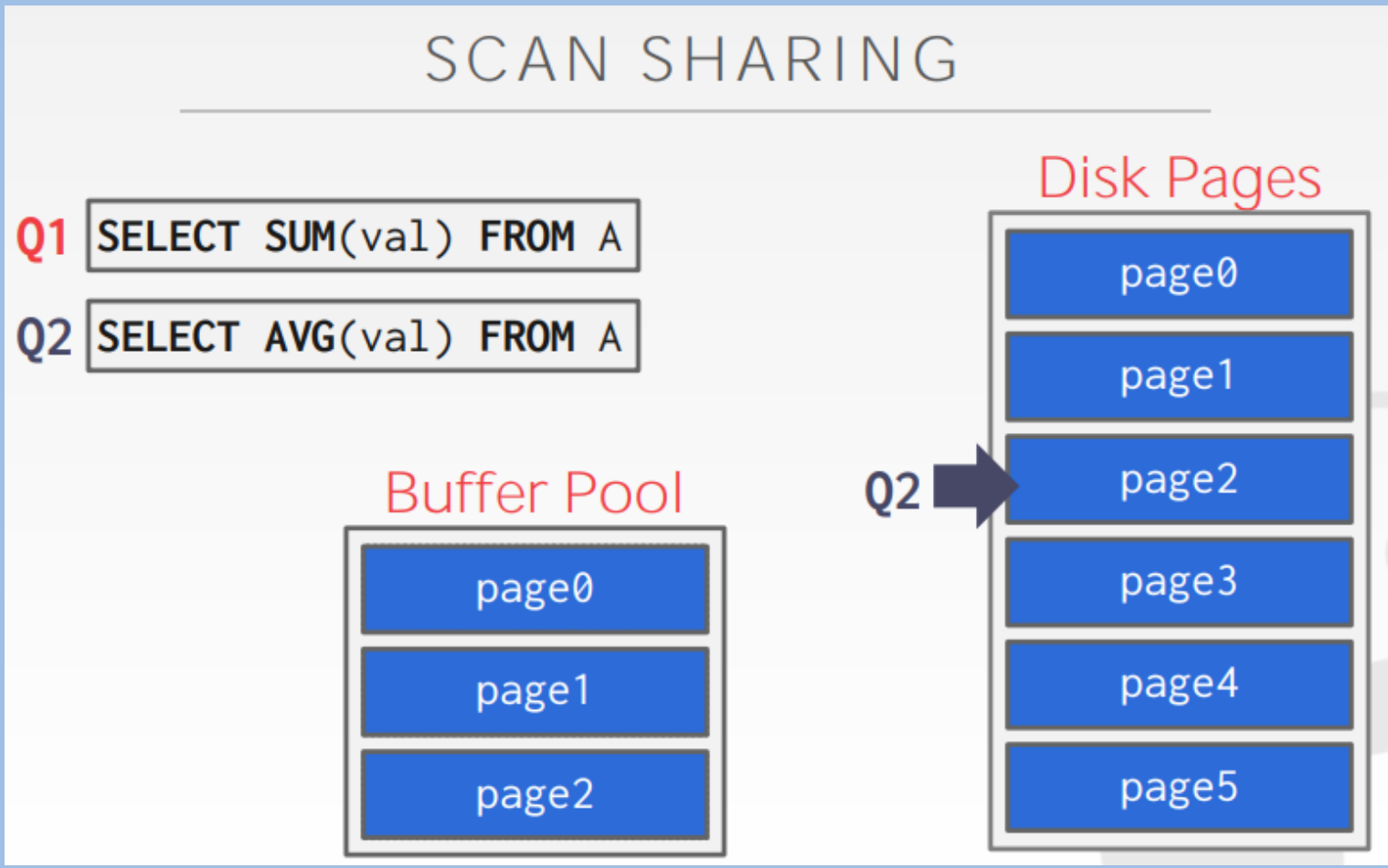
Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure
Buffer Pool Optimization



Source: Database Systems, CMU 15-445/645, https://15445.courses.cs.cmu.edu/fall2019/

Storage and File Structure
Buffer Pool Optimization



Thanks

Questions/Answers