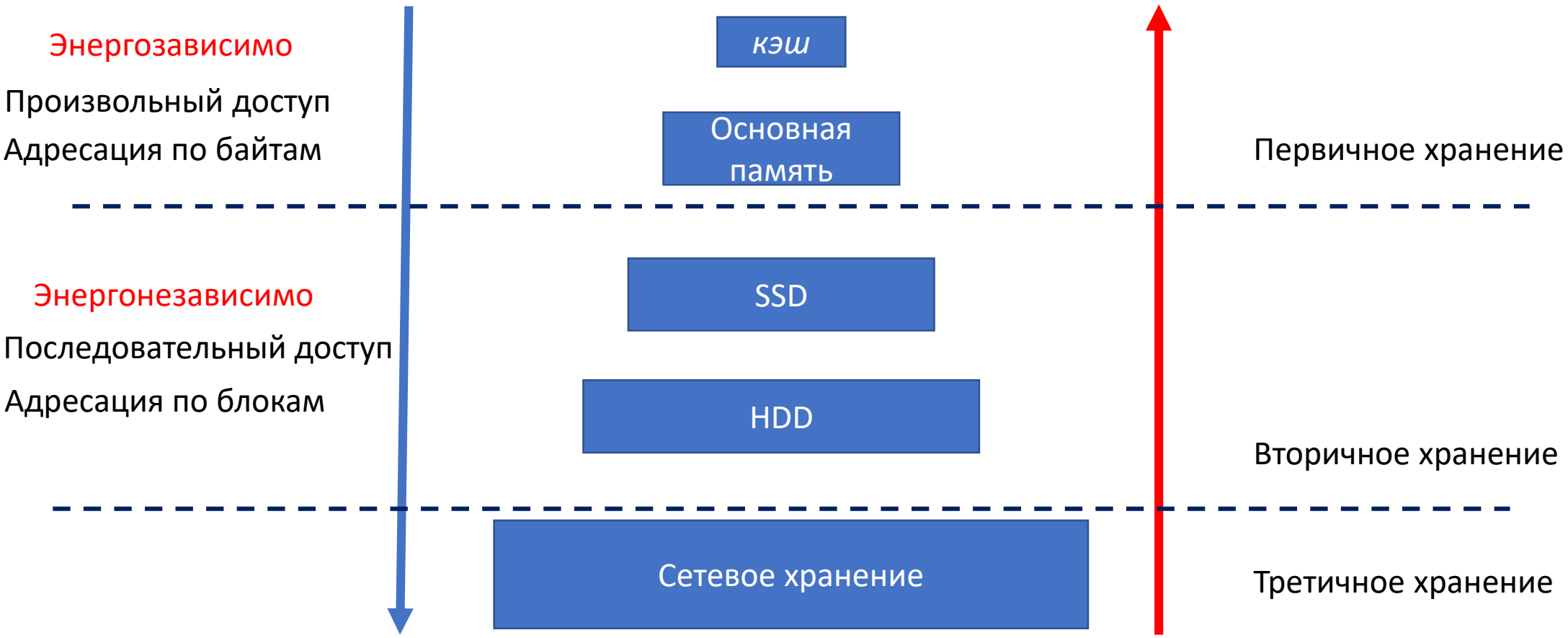
Лекция 4

Методы организации хранения данных

Классификация типов хранения

- Разделяется на
  - Энергозависимая память. Данные пропадают при выключении.
  - Энерогонезависимая память. Данные сохраняются даже при выключении.
- Факторы, влияющие на выбора типа хранения:
  - Скорость доступа
  - Стоимость за единицу данных
  - Надежность

Иерархия типов хранения



Время доступа

- 0.5 нс – L1 кеш
- 7 нс – L2 кеш
- 100 нс – Память
- 100.000 нс – SSD + 3-4 порядка
- 10.000.000 нс – HDD + 5-6 порядков

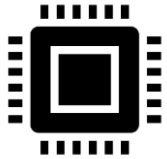
По стоимости даже примерно тяжело привести статистику, но зависимость обратная.

Задача БД

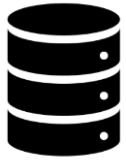
Необходимо «симулировать», что мы можем хранить всю БД в памяти.

Так как чтение и запись на диск довольно затратные операции, необходимо эффективно управлять этим для избегания больших ожиданий и потерь в производительности.

Дискоориентированные СУБД

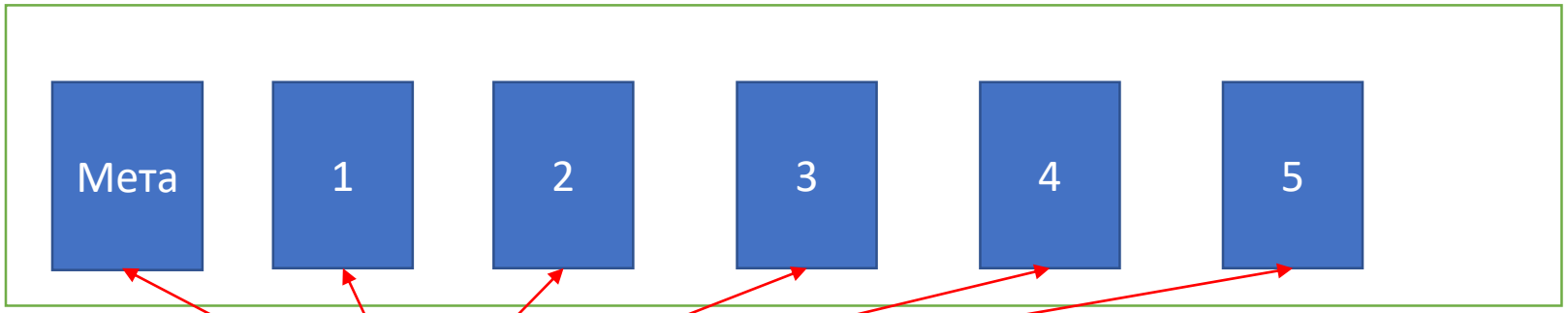


Память



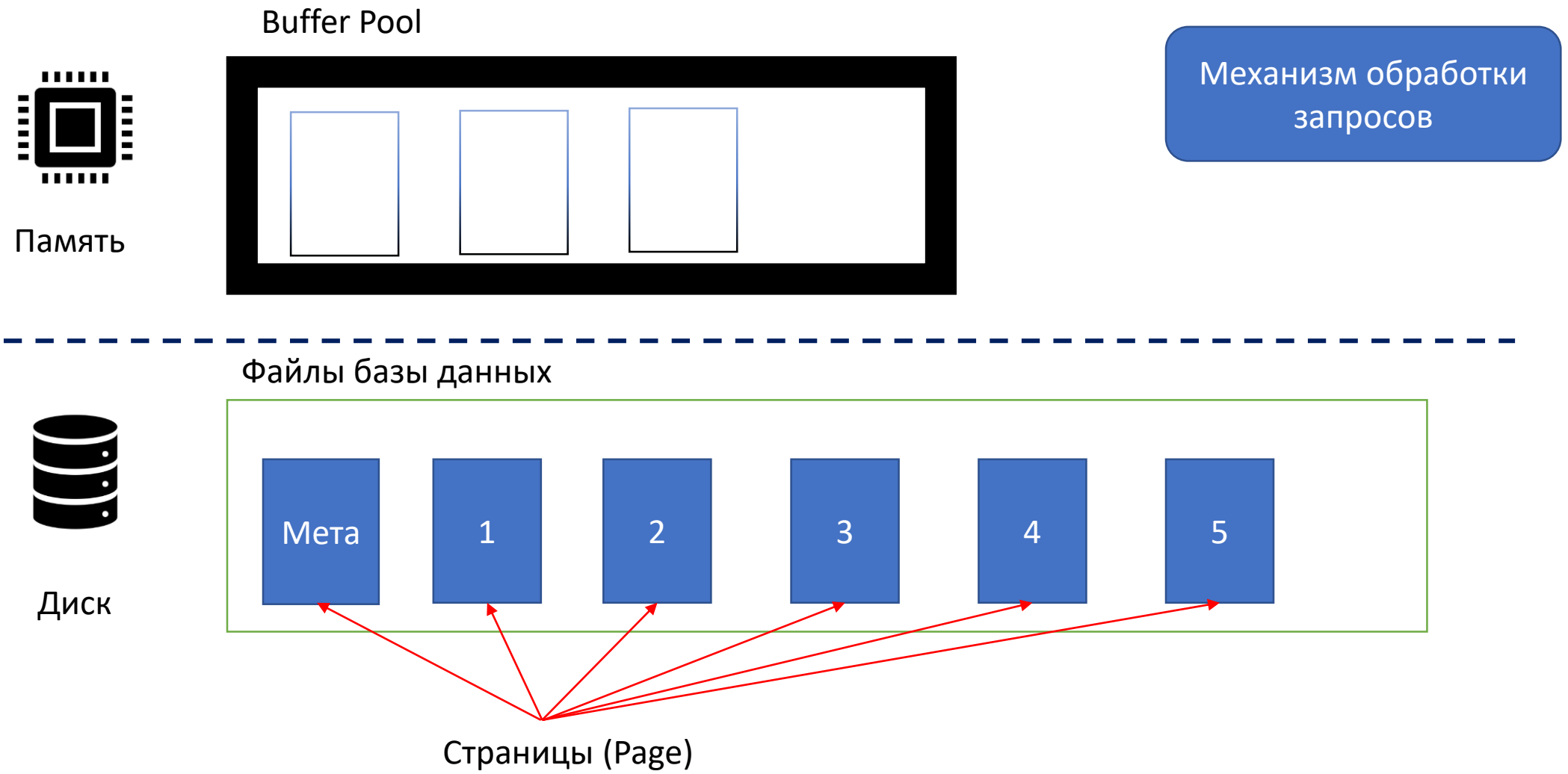
Диск

Файлы базы данных

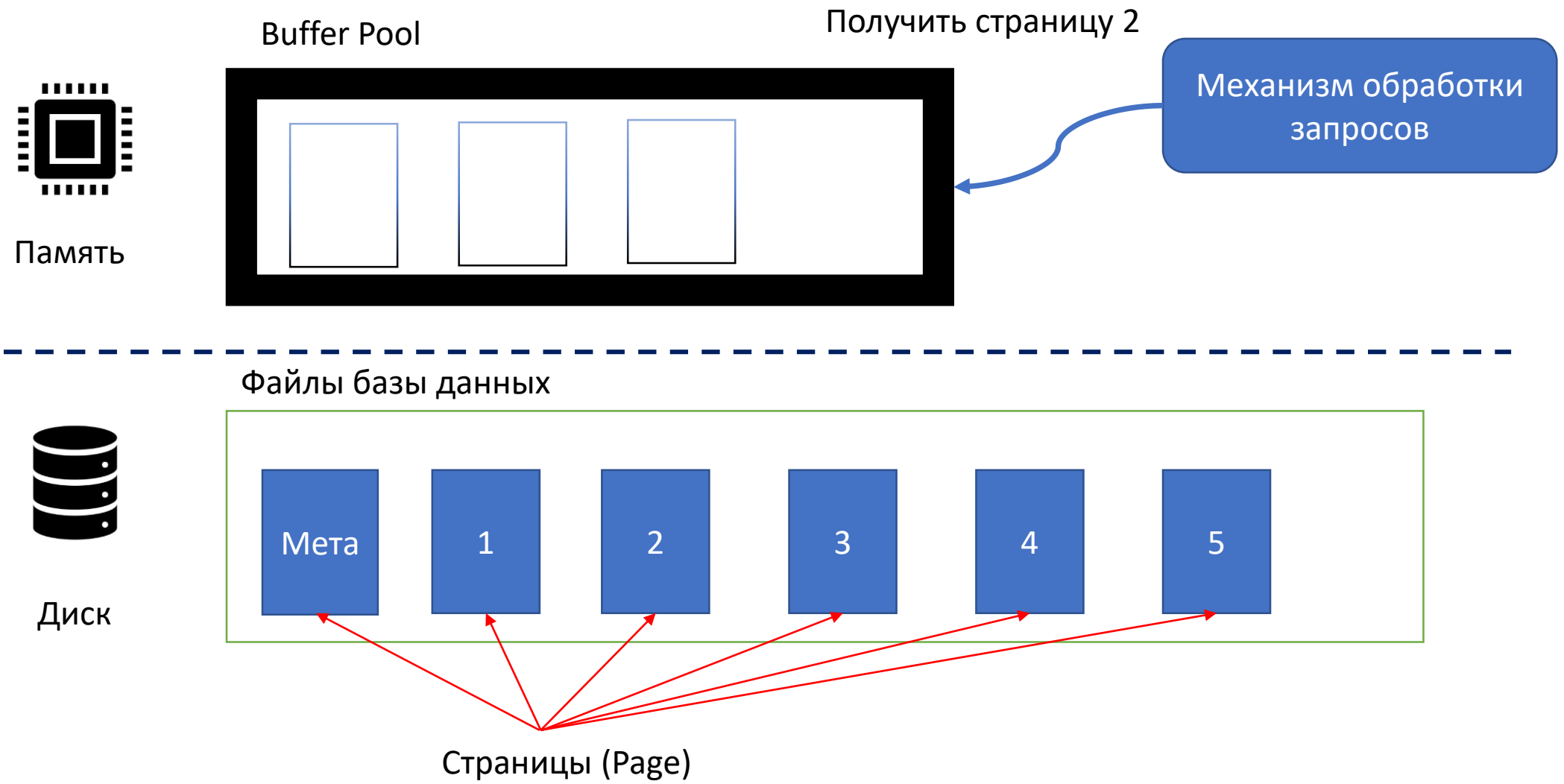


Страницы (Page)

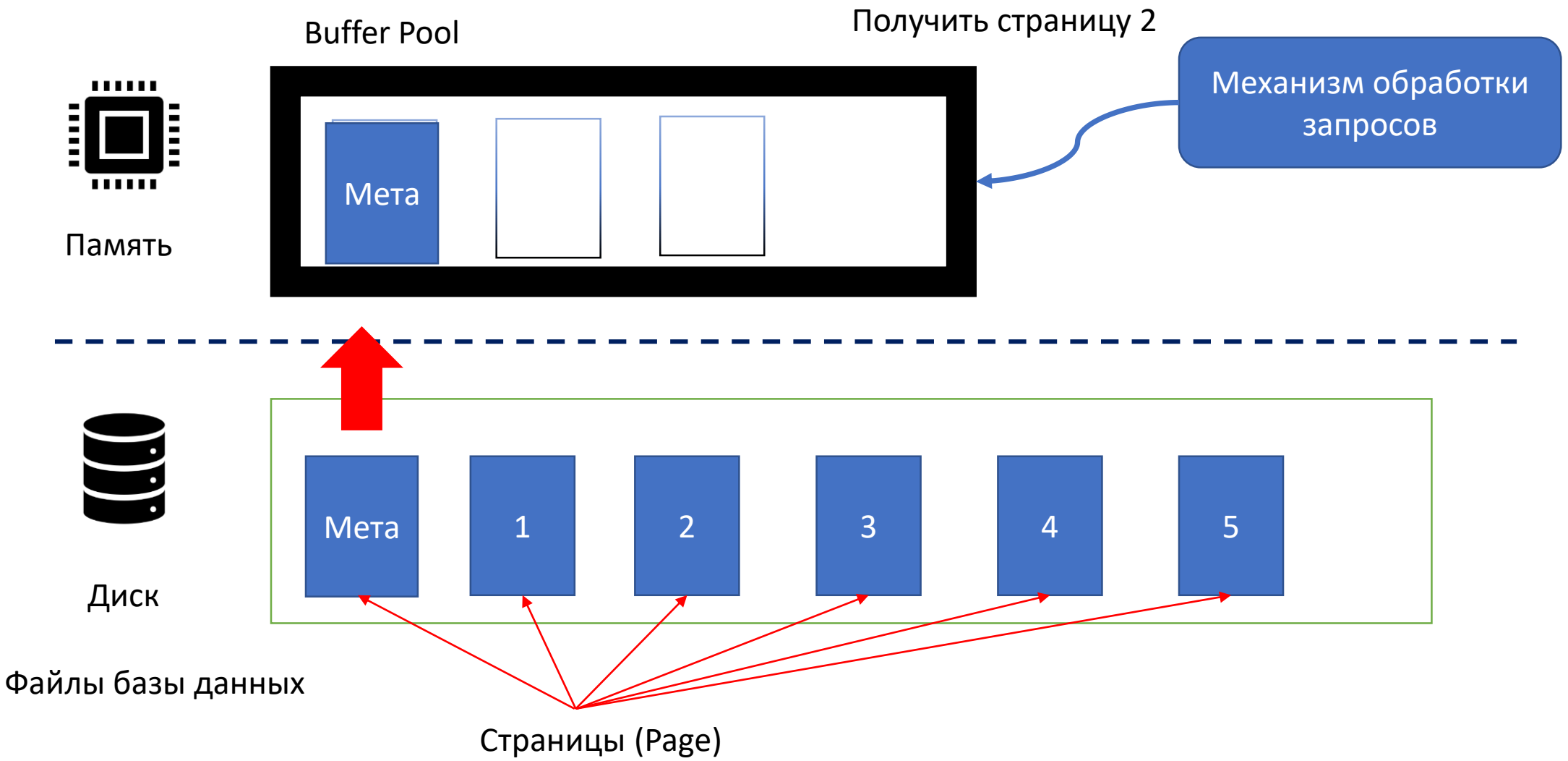
Дискоориентированные СУБД



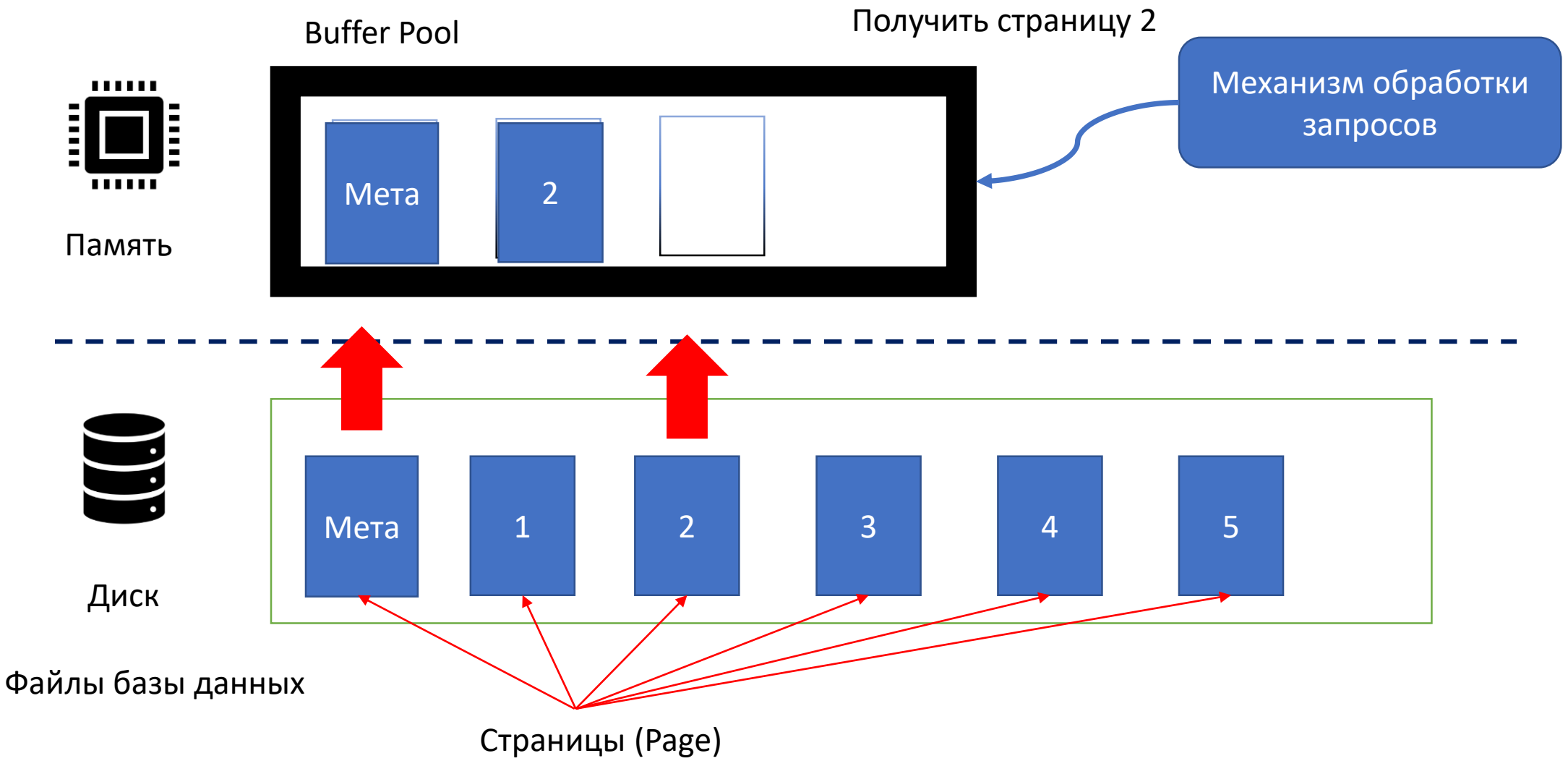
Дискоориентированные СУБД



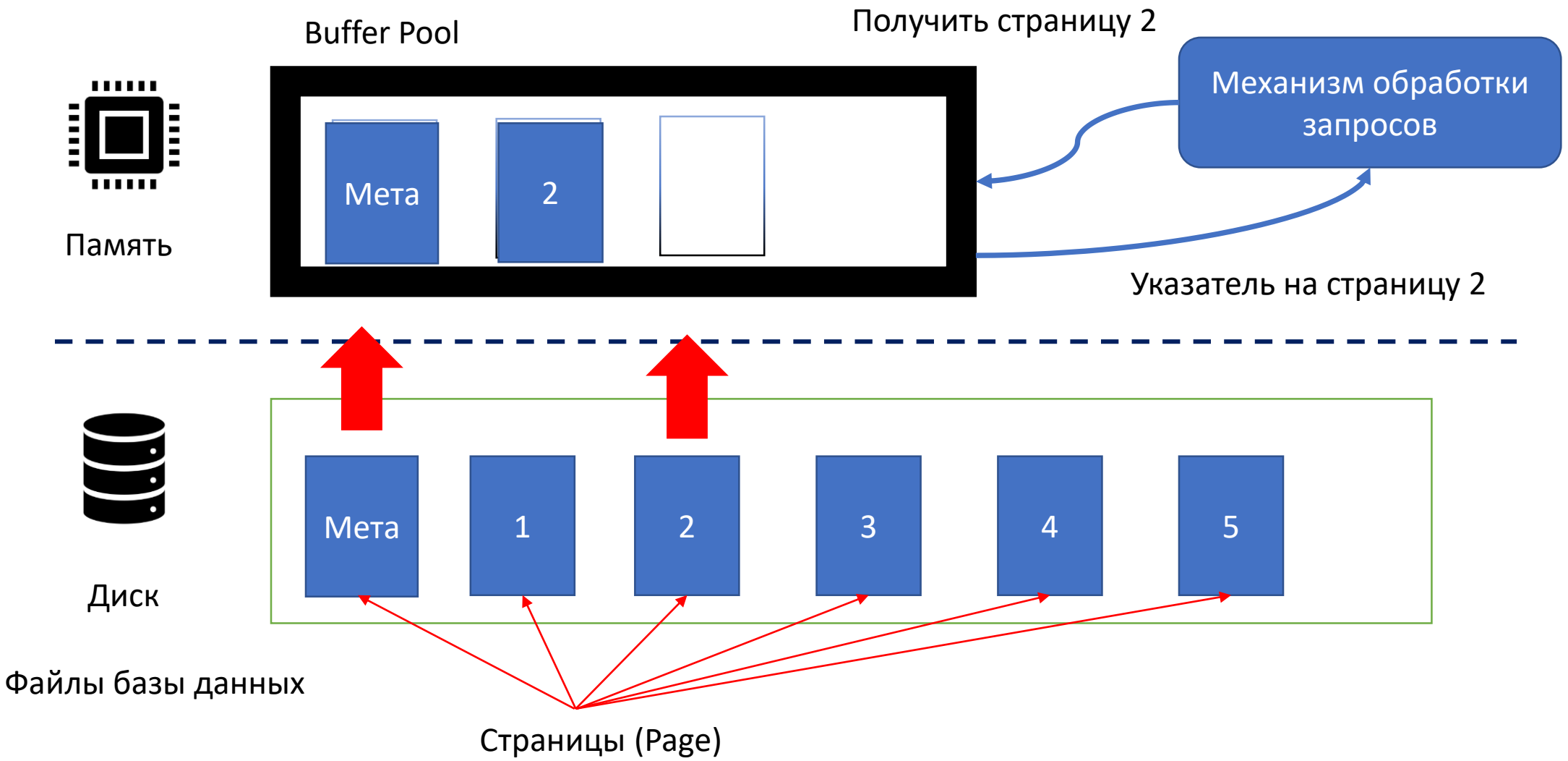
Дискоориентированные СУБД



Дискоориентированные СУБД



Дискоориентированные СУБД



Почему не использовать ОС?

СУБД почти всегда предполагает самостоятельный контроль для работы с памятью и обычно работает лучше чем ОС

- Правильный порядок записи «грязных» страниц на диск
- Специфичный предзабор страниц
- Политика замены буфера
- Управление процессами/потоками

Хранилище для СУБД

- Как СУБД хранит базу данных в файле на диске
- Как СУБД управляет памятью и осуществляет перенос данных туда-обратно

Хранилище для СУБД

- Как СУБД хранит базу данных в файле на диске
  - Хранение файлов
  - Расположение внутри страницы
  - Расположение внутри кортежа
- Как СУБД управляет памятью и осуществляет перенос данных туда-обратно

Хранение файлов

СУБД хранит базу данных как один или несколько файлов на диске.

Операционная система ничего не знает о содержимом в данных файлах.

Менеджер хранилища

- Менеджер хранилища (storage manager) отвечает за управление файлами данных.
  - Наиболее продвинутые иногда делают собственное планирование чтения и записи для локализации страниц
- Менеджер хранилища организует файлы как коллекцию страниц
  - Происходит отслеживание чтения/записи для страниц
  - Происходит отслеживание свободного места

Страницы (Page)

- Страница – блок данных **фиксированного** размера (обычно 8Кб)
  - Содержит в себе практически всю информации о базе данных (кортежи, метаданные, индексы, записи логов, и т.д.)
  - Большинство СУБД не смешивают типы страниц
  - Часть СУБД требуют от страниц автономности (внутри страницы хранится вся информация, требуемая для обработки страницы)

Страницы (Page)

- Каждой странице присвоен уникальный идентификатор
- СУБД использует дополнительные структуры для связи идентификаторов и физических адресов.

Архитектура хранения страниц

Различные СУБД могут сохранять страницы на диск разными способами:

- Организация файлов в виде «кучи»
- Последовательная/отсортированная организация файлов
- Организация файлов с помощью хешей

Организация файла в виде «кучи»

Файл в виде «куча» – неупорядоченная коллекция страниц, в которой кортежи хранятся в случайном порядке

- Необходимо иметь определенный API для управления страницами: Создание / Получение / Запись / Удаление Страниц
- Необходимо поддерживать итерационную обработку по страницам

Организация файла в виде «кучи»

Требуются метаданные для отслеживания существующих страниц и какие из них содержат в себе свободное место

Существует 2 способа:

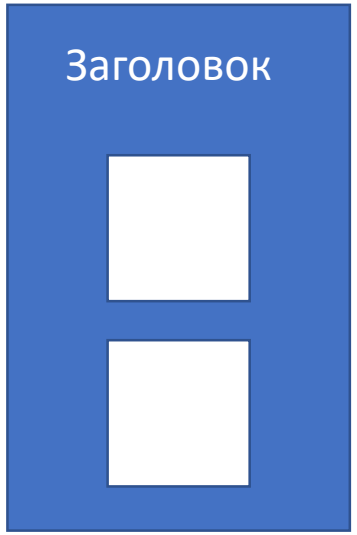
- Связные списки
- Директория страниц

Файл в виде «кучи». Связный список

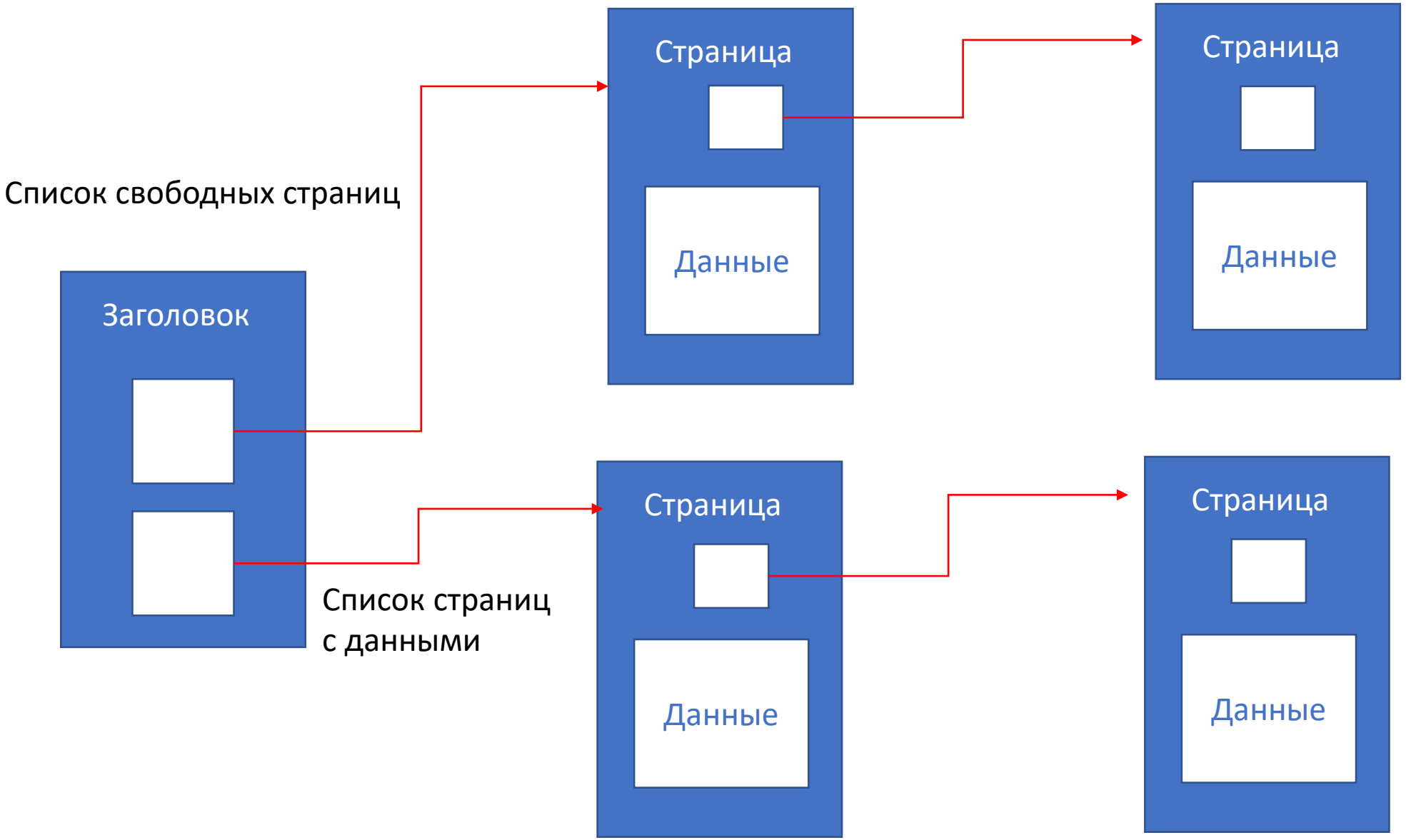
Содержит в себе заголовок указателей

в начале файла:

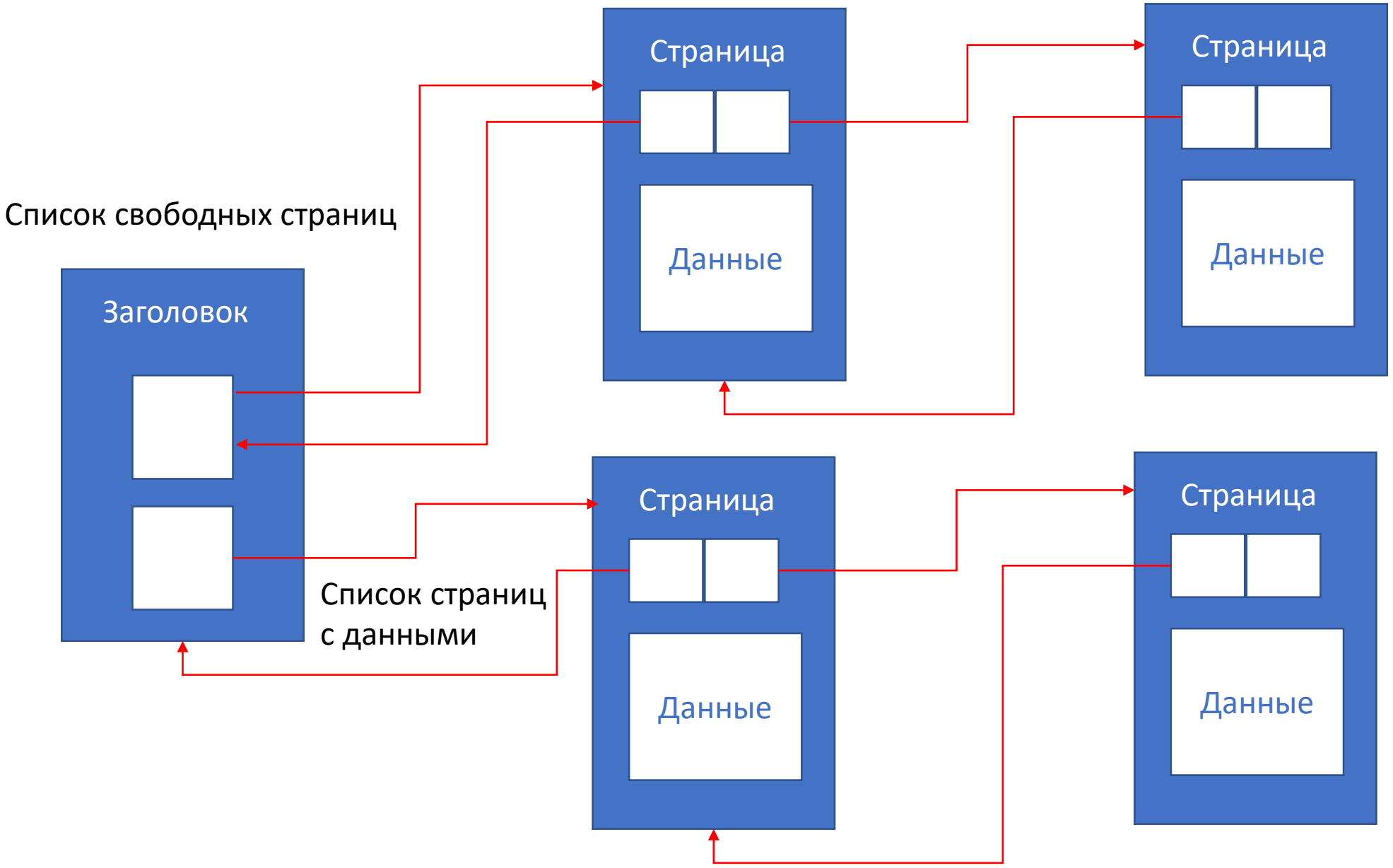
- Первый элемент списка свободных страниц
- Первый элемент списка страниц с данными



Файл в виде «кучи». Связный список



Файл в виде «кучи». Связный список



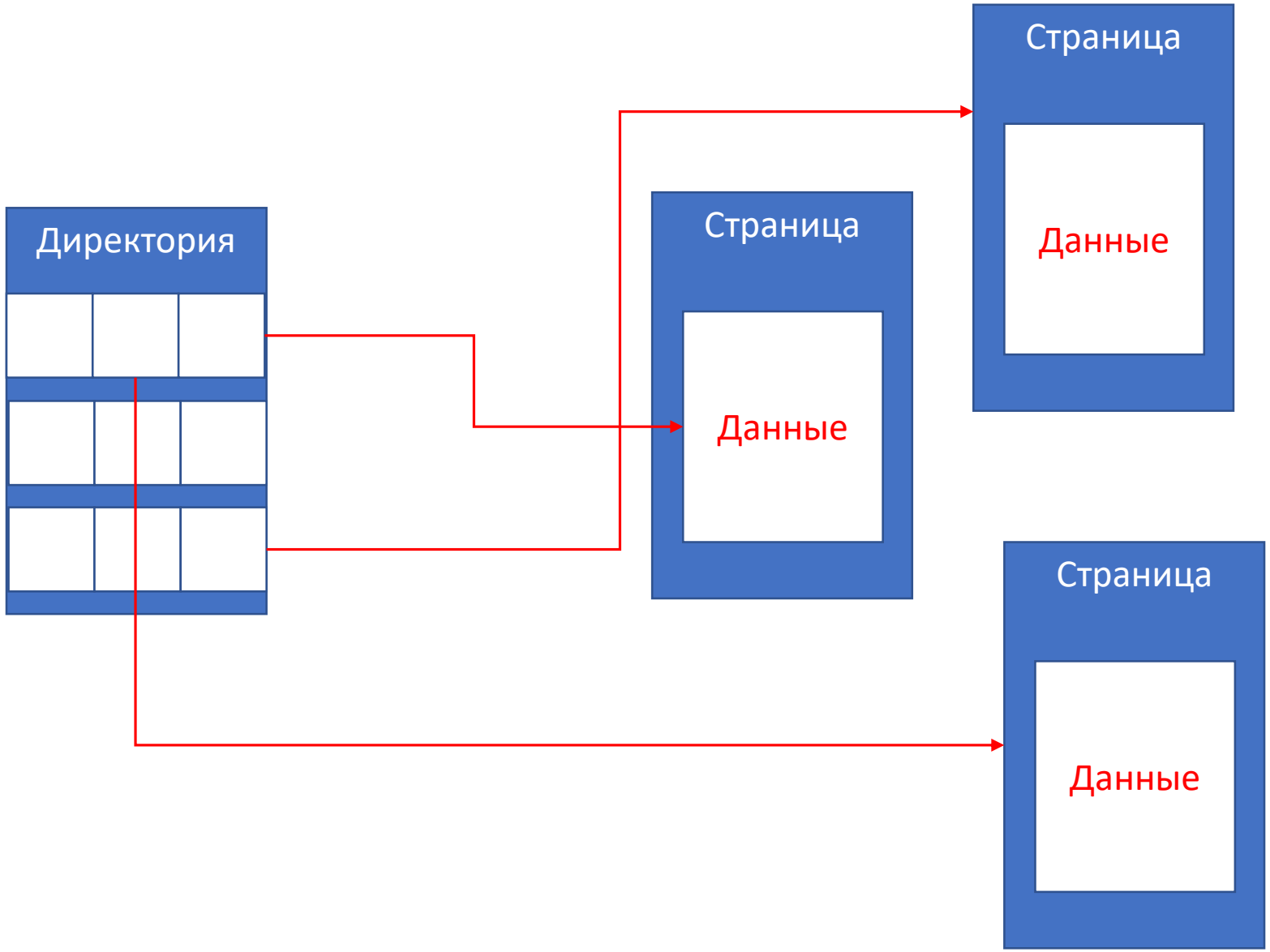
Файл в виде «кучи». Директория страниц

СУБД содержит специальные страницы, которые отслеживают местоположение страниц данных в файлах базы данных.

Также директория фиксирует количество свободных слотов на страницу.

СУБД должна убедиться, что страницы директории синхронизированы со страницам данных.

Файл в виде «кучи». Директория страниц



Заголовок страницы

На каждой странице есть заголовок с метаданными о содержании страницы

- Размер страницы
- Checksum
- Версия СУБД
- Информация о транзакциях и компрессии

Часть СУБД требуют от страниц автономности

Организация страницы

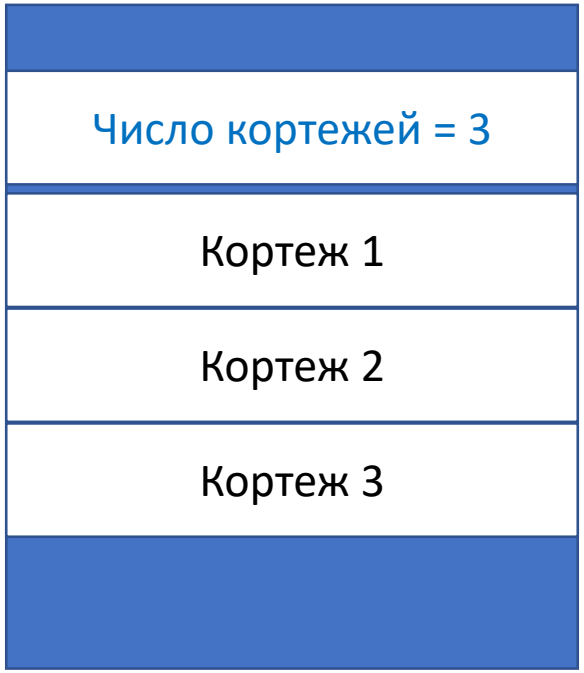
- Хранение кортежей
- Хранение лога изменений

Хранение кортежей

Как хранить кортежи?

Прямолинейный вариант:

Отслеживать количество кортежей на странице и добавлять кортеж в конец.



Хранение кортежей

Как хранить кортежи?

Прямолинейный вариант:

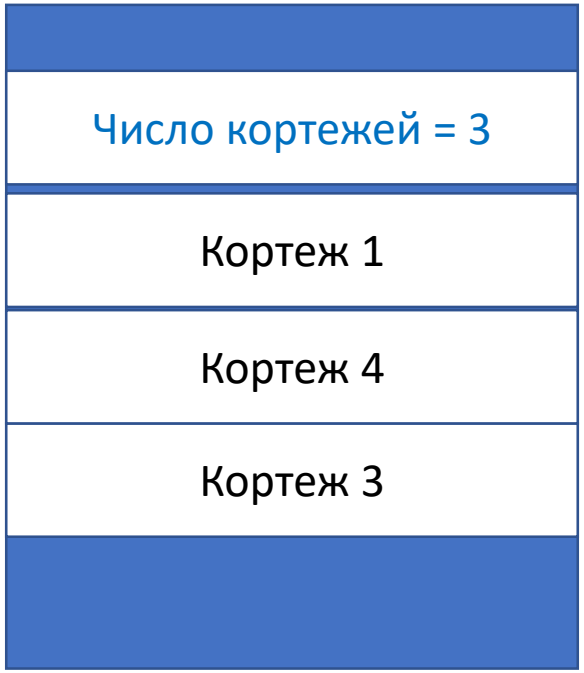
Отслеживать количество кортежей на странице и добавлять кортеж в конец.

Хранение кортежей

Как хранить кортежи?

Прямолинейный вариант:

Отслеживать количество кортежей на странице и добавлять кортеж в конец.



Комбинированные страницы

Slotted pages – наиболее частая организация страницы

Массив «слотов» связывает слот и начальное смещение кортежа

Заголовок содержит информацию о:

- Количество используемых слотов
- Смещение начальной локации последнего слота

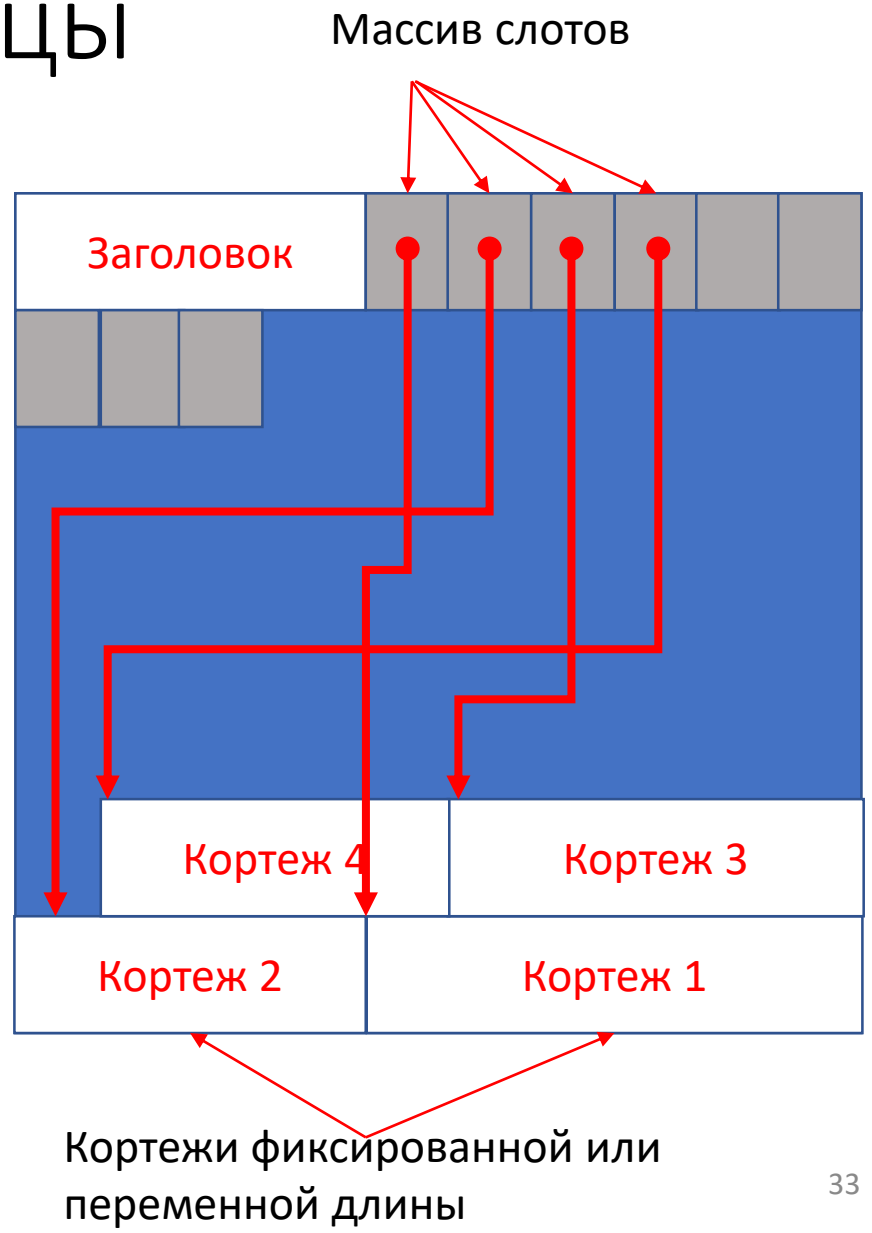
Комбинированные страницы

Slotted pages – наиболее частая организация страницы

Массив «слотов» связывает слот и начальное смещение кортежа

Заголовок содержит информацию о:

- Количество используемых слотов
- Смещение начальной локации последнего слота



Описание кортежа

Физически кортеж – это последовательность байтов.

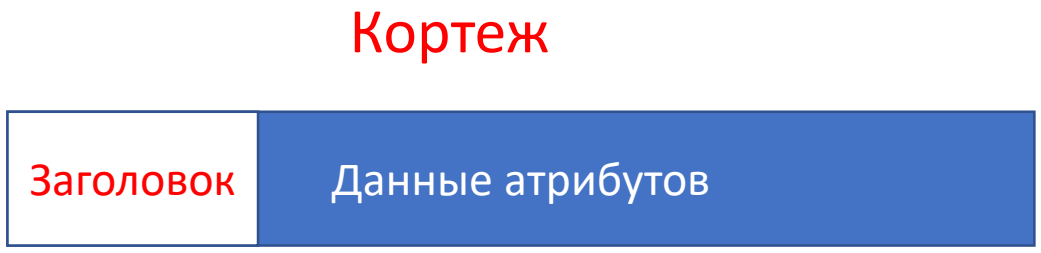
Задача СУБД – интерпретировать байты в атрибуты и их значения.

Заголовок кортежа

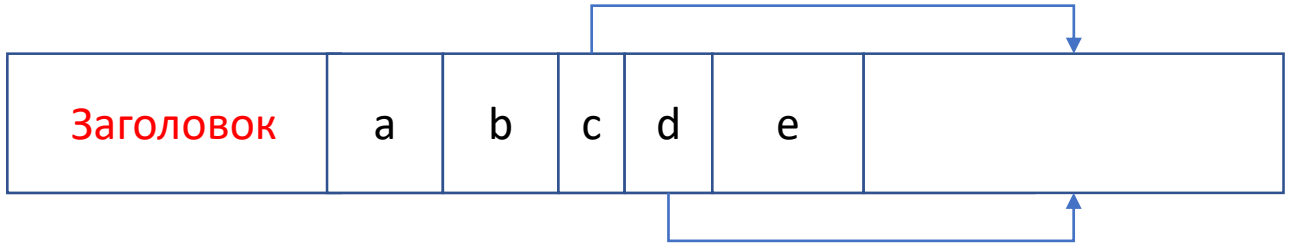
Каждый кортеж содержит в себе метаданные

- Зона видимости
- Битовая карта для NULL значений

Обычно метаданные о схеме хранить не требуется



Данные кортежа



Обычно атрибуты хранятся в порядке, который установлен для создания таблиц.

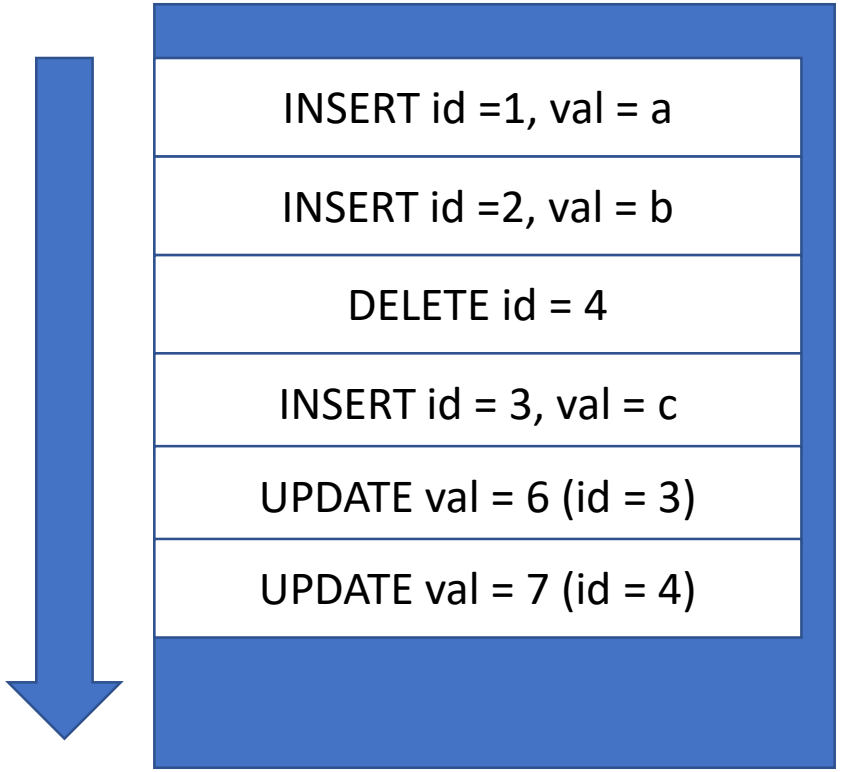
Часть СУБД позволяет осуществлять реорганизацию атрибутов.

```
CREATE TABLE test (
  a INT PRIMARY KEY,
  b INT NOT NULL,
  c VARCHAR(10),
  d VARCHAR(20),
  e FLOAT
);
```

Организация файлов в виде логов

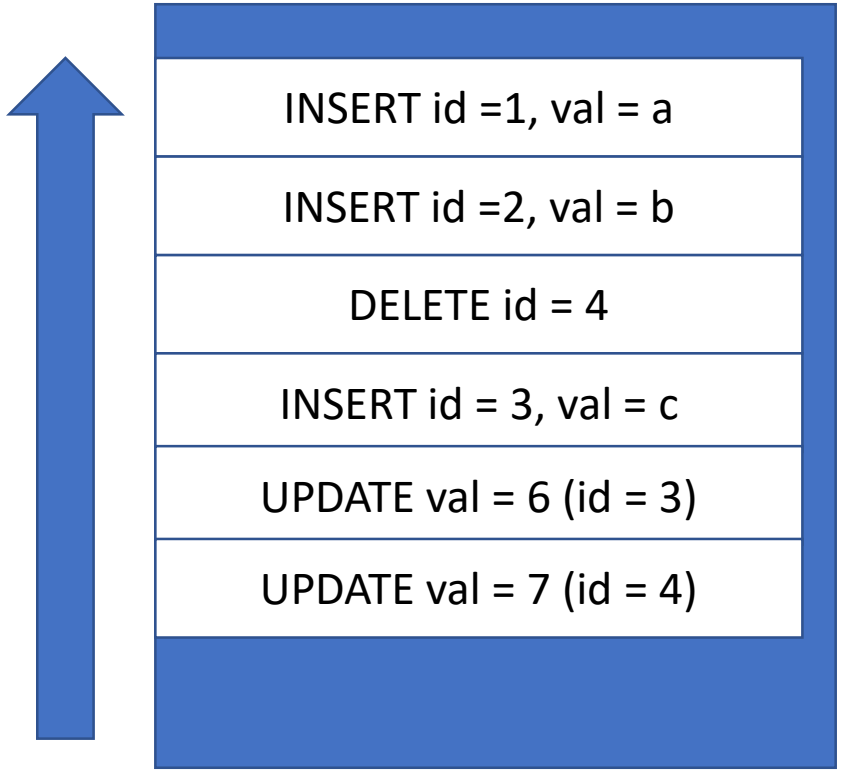
Вместо хранения кортежа в странице, СУБД хранит только изменения.

- При вставке описывается весь кортеж
- Удаление маркирует кортеж как удаленный
- Обновление содержит в себе только дельту изменений



Организация файлов в виде логов

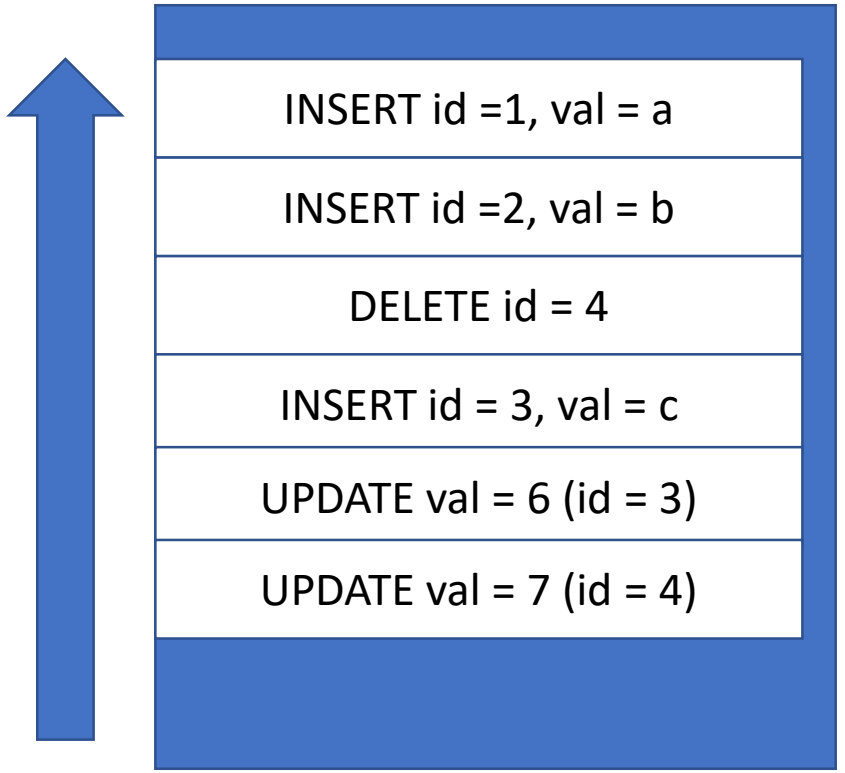
Чтобы считать запись СУБД требуется считать лог в обратном порядке и собрать запись



Организация файлов в виде логов

Чтобы считать запись СУБД требуется считать лог в обратном порядке и собрать запись

Для оптимизации можно построить индексы для перехода по локациям лога



Представление данных

- INTEGER/BIGINT/SMALLINT/TINYINT
  - C/C++ представление
- FLOAT/REAL и NUMERIC/DECIMAL
  - Стандарт IEEE-754 / Числа с фиксированной длиной
- VARCHAR/VARBINARY/TEXT/BLOB
  - Заголовок с информацией о длине, затем байты данных
- TIME/DATE/TIMESTAMP
  - 32/64 bit целые числа секунд или микросекунд с Unix epoch

Большие значения

Большинство СУБД не позволяют кортежу превышать размер страницы.

Для хранения значений больших чем страница, СУБД использует дополнительные страницы переполнения (overflow).

Буферный пул

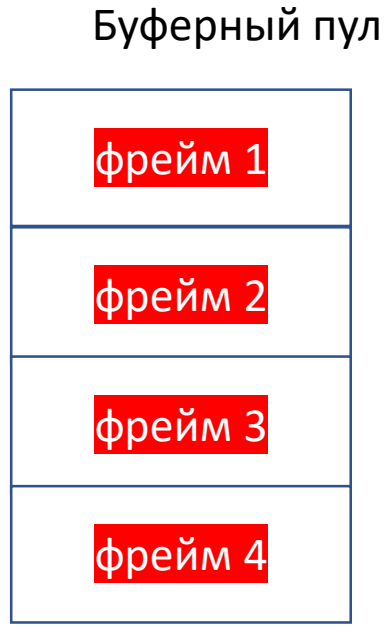
Буферный пул

Буферный пул – область памяти, организованная как массив страниц фиксированного размера.

Элемент массива называется фреймом (frame).

Когда СУБД запрашивает страницу, точная копия находится в одном или нескольких фреймах.

Буферный пул



Страница 1

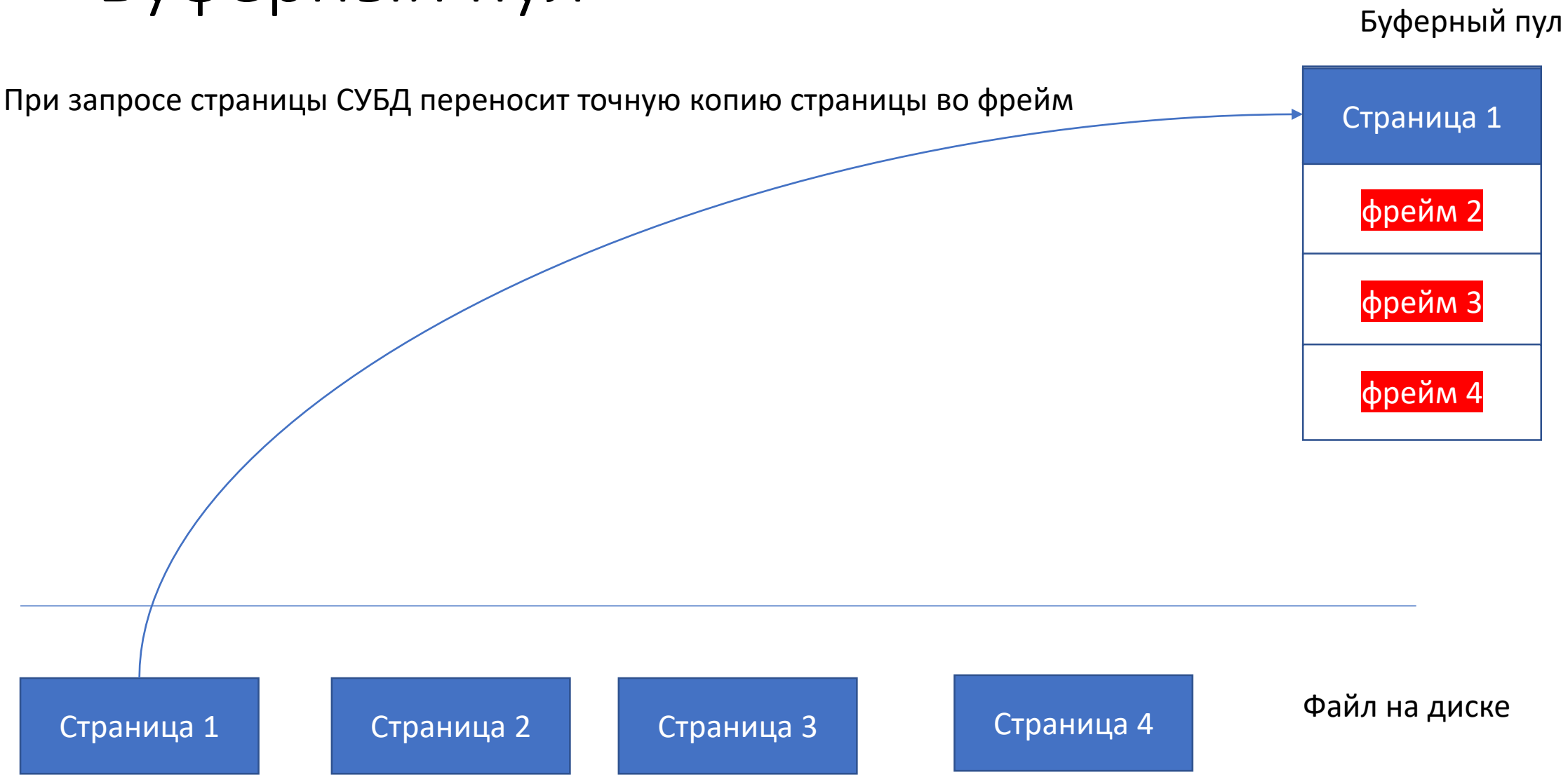
Страница 2

Страница 3

Страница 4

Файл на диске

Буферный пул



Буферный пул

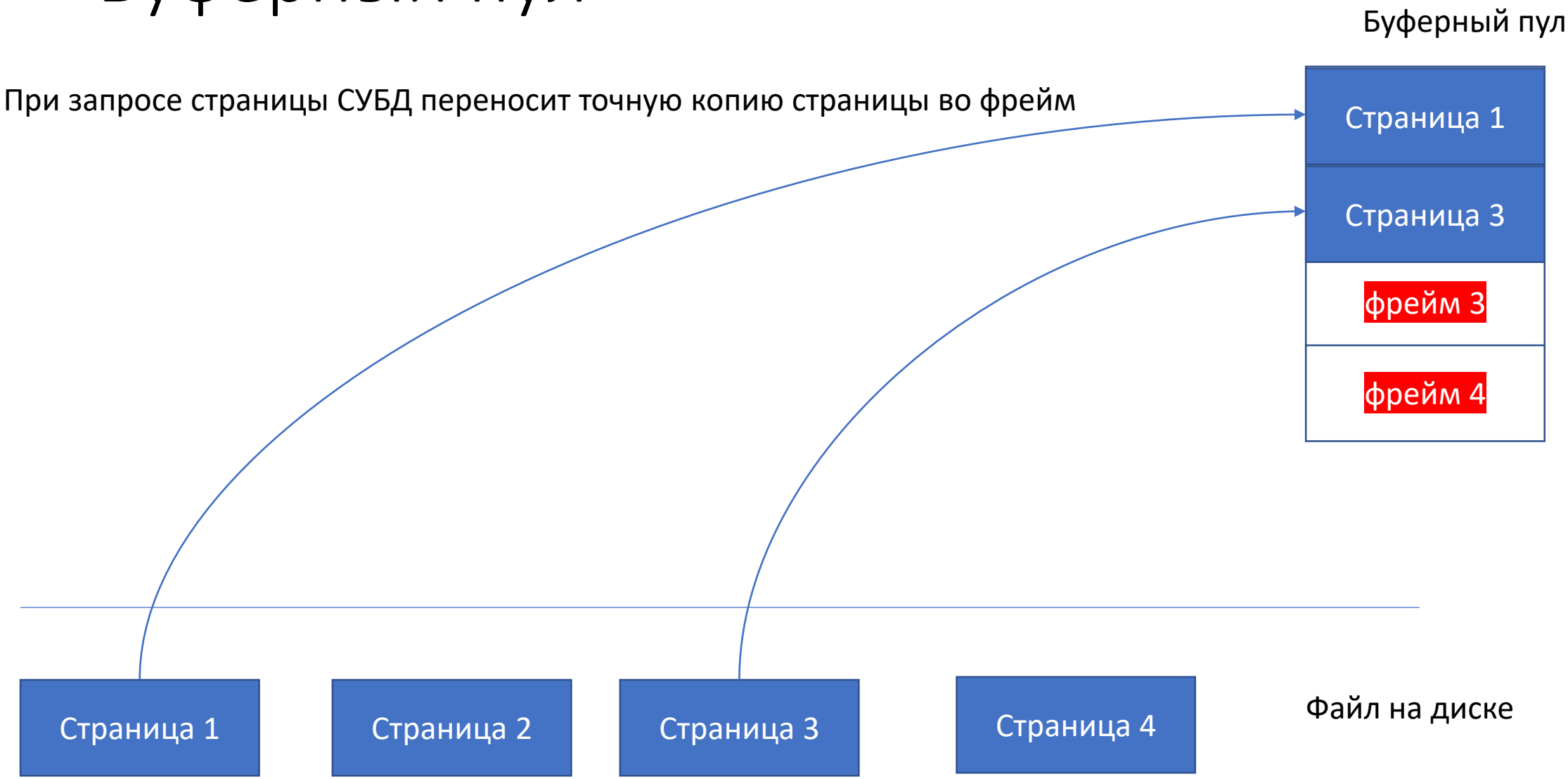
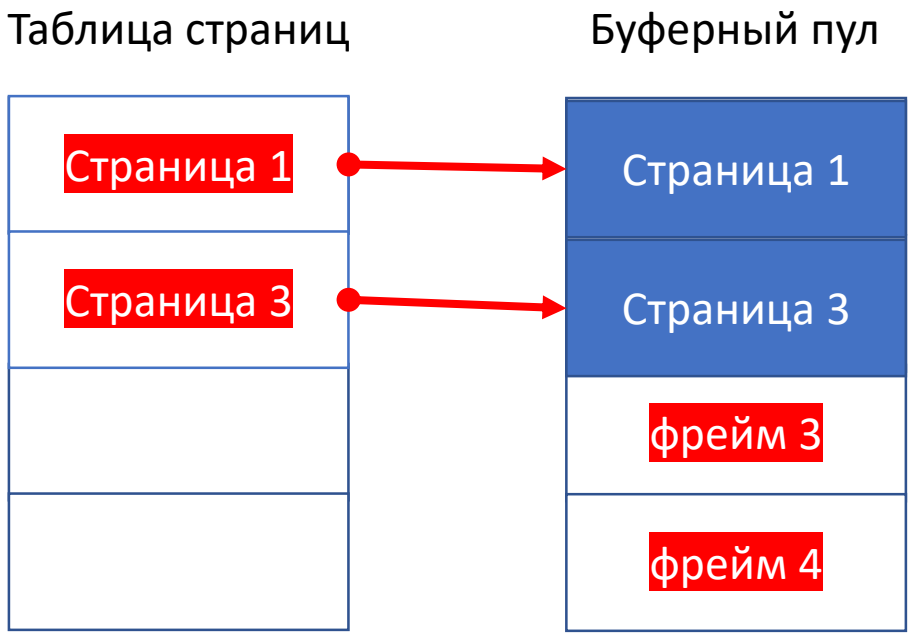


Таблица страниц (Page table)

Таблица страниц отслеживает страницы, которые сейчас находятся в памяти

В них также содержится информация о странице:

- Грязный флаг (Dirty flag)
  - Бит, указывающий были ли какие-то изменения на данной странице
- Счетчик ссылок/защелка (pin).
  - Если происходит какое-то действие, то данный фрейм нельзя считывать



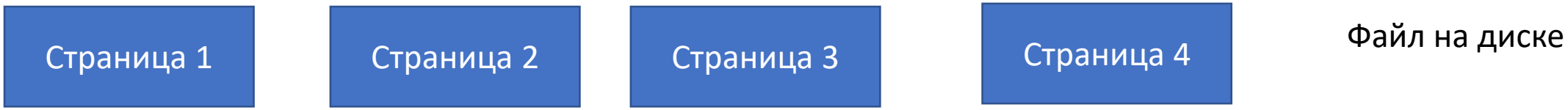
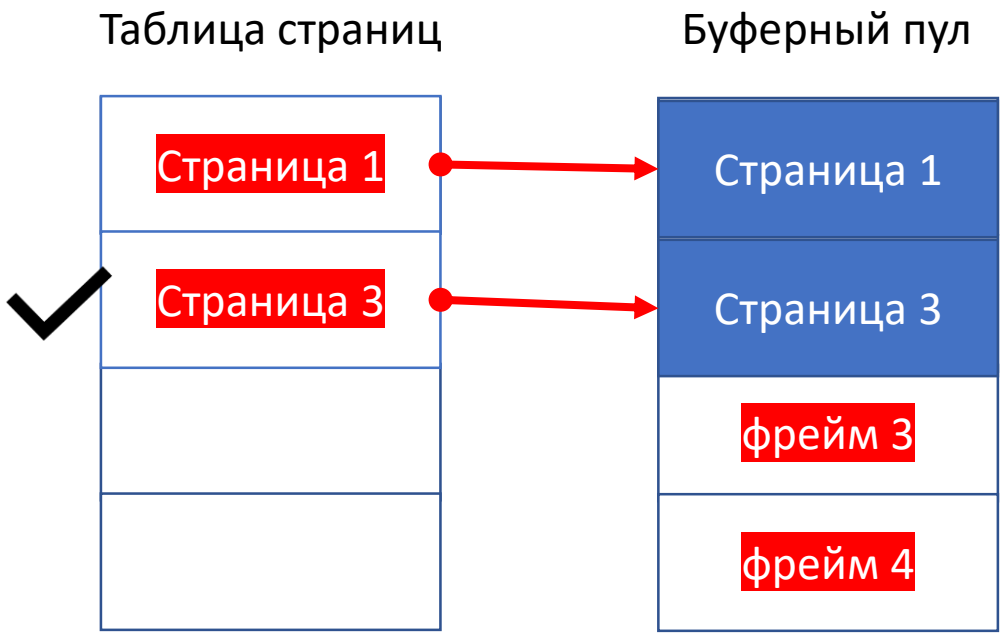


Таблица страниц (Page table)

Таблица страниц отслеживает страницы, которые сейчас находятся в памяти

В них также содержится информация о странице:

- Грязный флаг (Dirty flag)
  - Бит, указывающий были ли какие-то изменения на данной странице
- Счетчик ссылок/защелка (pin).
  - Если происходит какое-то действие, то данный фрейм нельзя считывать



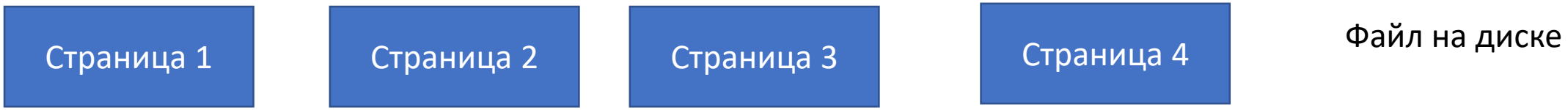
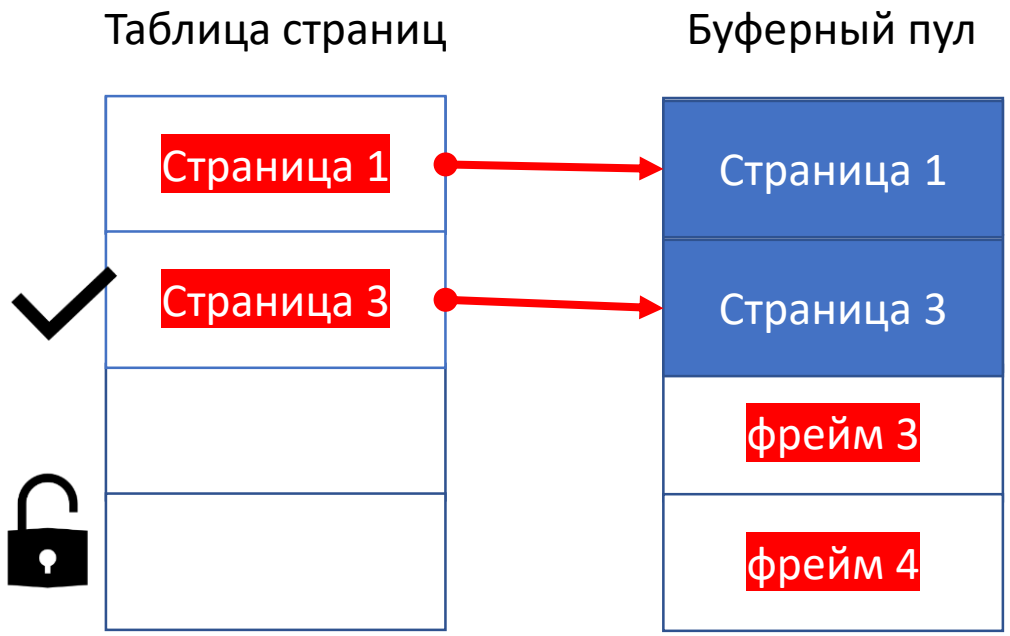


Таблица страниц (Page table)

При считывании новой страницы на диске изначально осуществляется аллоцирование места в таблице страниц, затем считывание в буферный пул.



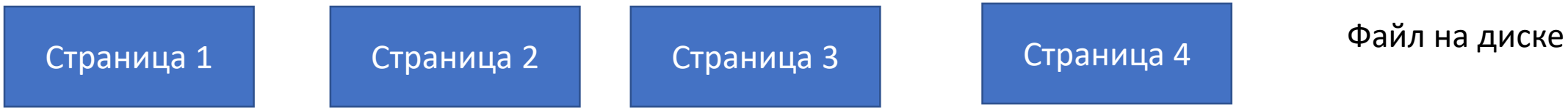


Таблица страниц (Page table)

При считывании новой страницы на диске изначально осуществляется аллоцирование места в таблице страниц, затем считывание в буферный пул.

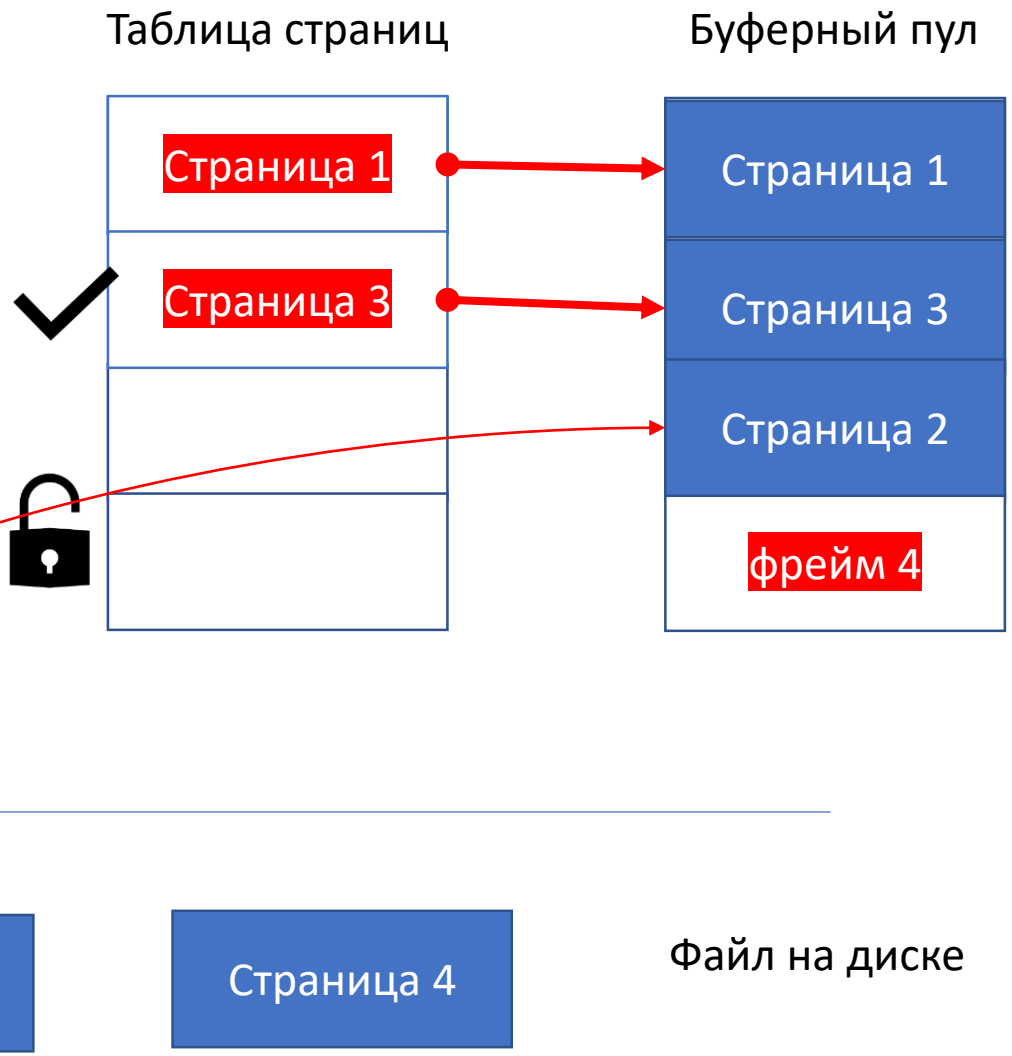
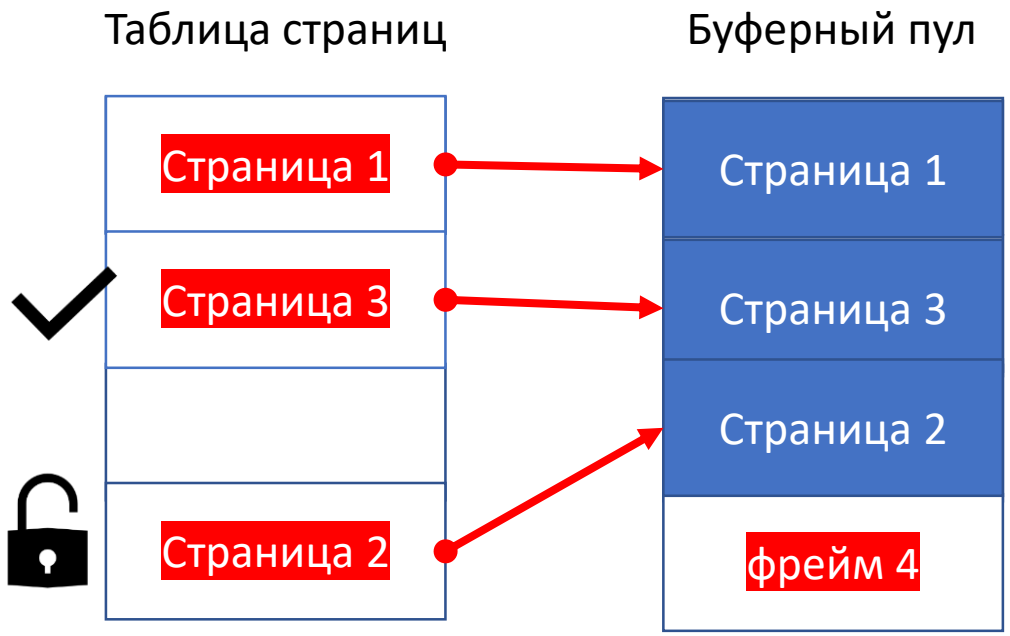
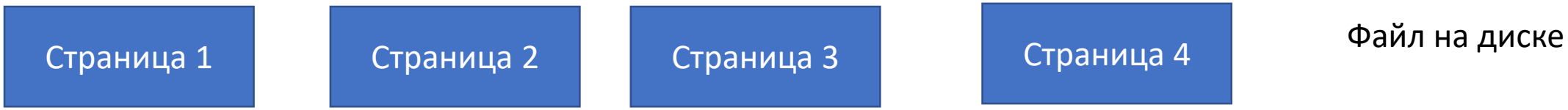


Таблица страниц (Page table)

При считывании новой страницы на диске изначально осуществляется аллоцирование места в таблице страниц, затем считывание в буферный пул.





Блокировки и защелки

- Блокировка
  - Высокоуровневая логическая единица. Осуществляет блокировку логического контента от других транзакций
  - Выполняется в процессе работы транзакции
  - Должна быть осуществлена возможность отката изменений
- Защелки
  - Защищает критические секции внутренних структур СУБД от других потоков
  - Выполняется в процессе работы операций
  - Не требуется возможность отката изменений

Таблица страниц и директория страниц

- Директория страниц – связь между идентификаторами страницы и расположением страницы в файлах базы данных
  - все изменения должны быть записаны на диск, для восстановления и перегрузки
- Таблица страниц – связь между идентификаторами страницы и копией страницы во фреймах буферного пула
  - данные структуры хранятся в памяти и их хранение на диске не требуется

Политика выделения памяти

- Глобальная
  - принимаются решения для всех активный транзакций

- Локальная
  - выделение фреймов для конкретной транзакции без учета поведения параллельных транзакций
  - требуется поддержка общих страниц

Оптимизация работы буферного пула

- Несколько буферных пулов
- Предзабор (pre-fetching)
- Совместный поиск

Несколько буферных пулов

- У СУБД далеко не всегда есть единый буферный пул для всей системы
  - Несколько экземпляров буферного пула
  - Буферный пул на базу данных
  - Буферный пул по типу страниц

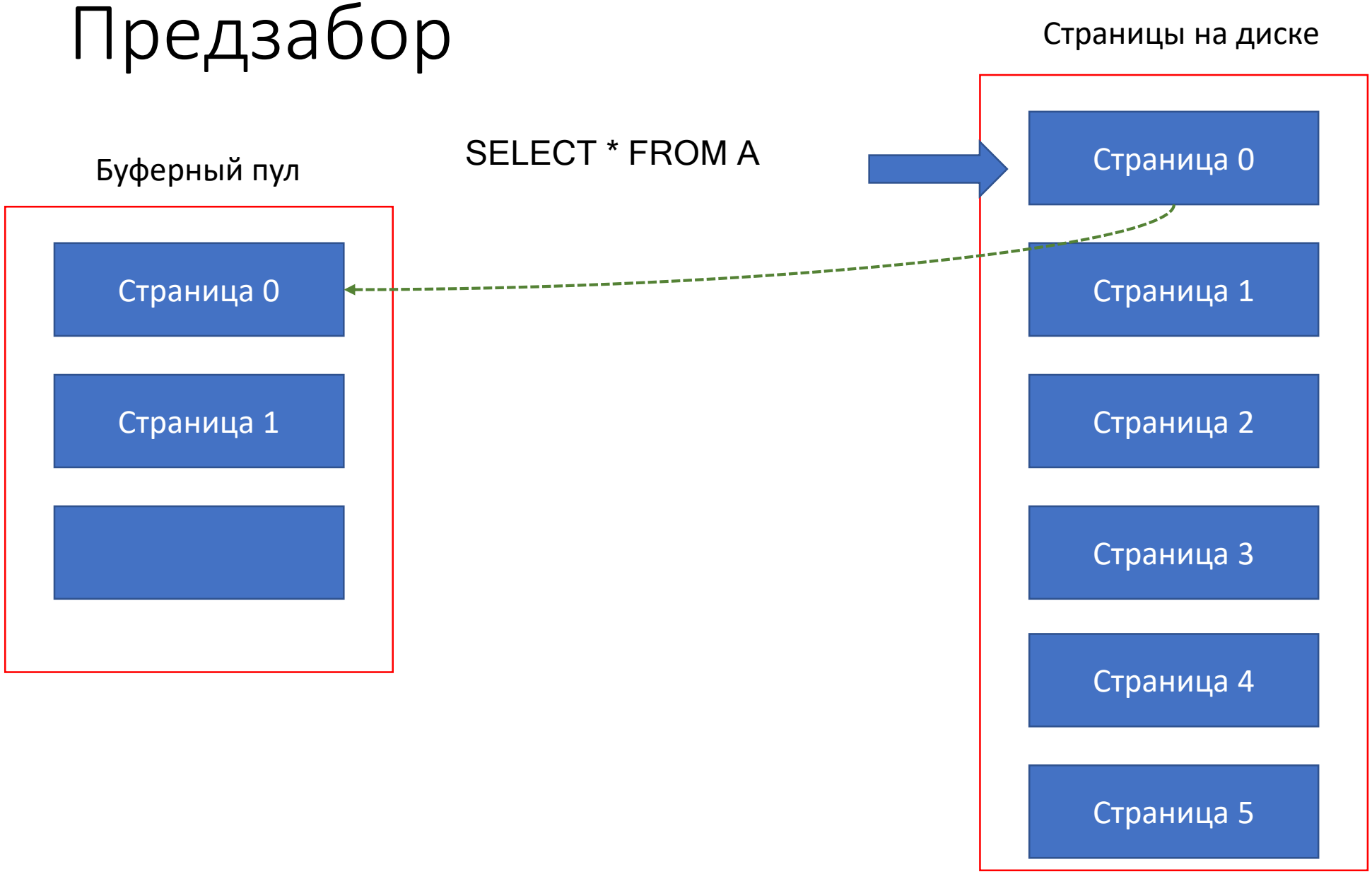
Несколько буферных пулов позволяет улучшить локальность и конкуренцию «защелок».

Предзабор

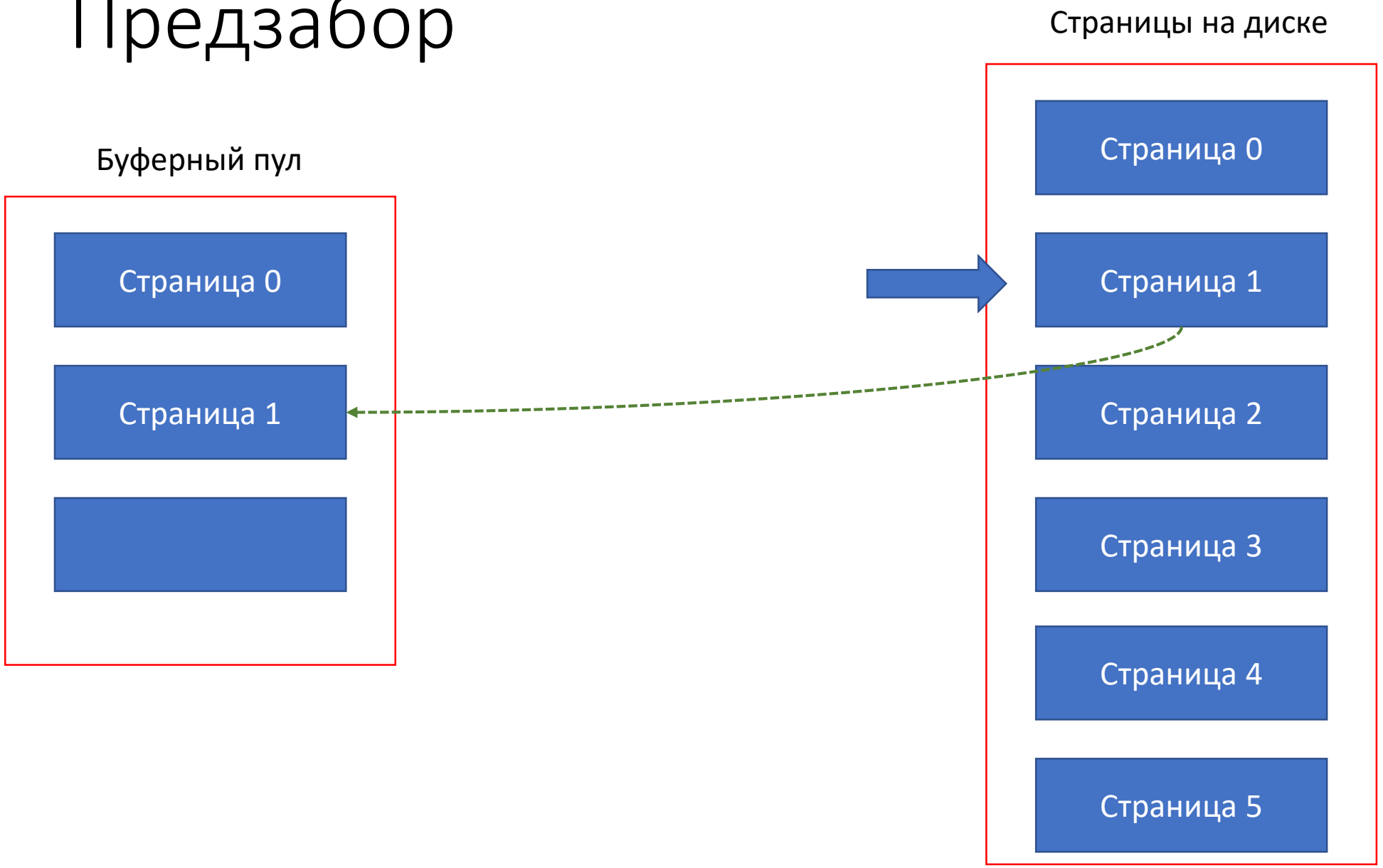
СУБД может осуществлять предзабор страниц, основанных на плане запроса

- Последовательный доступ
- Доступ по индексам

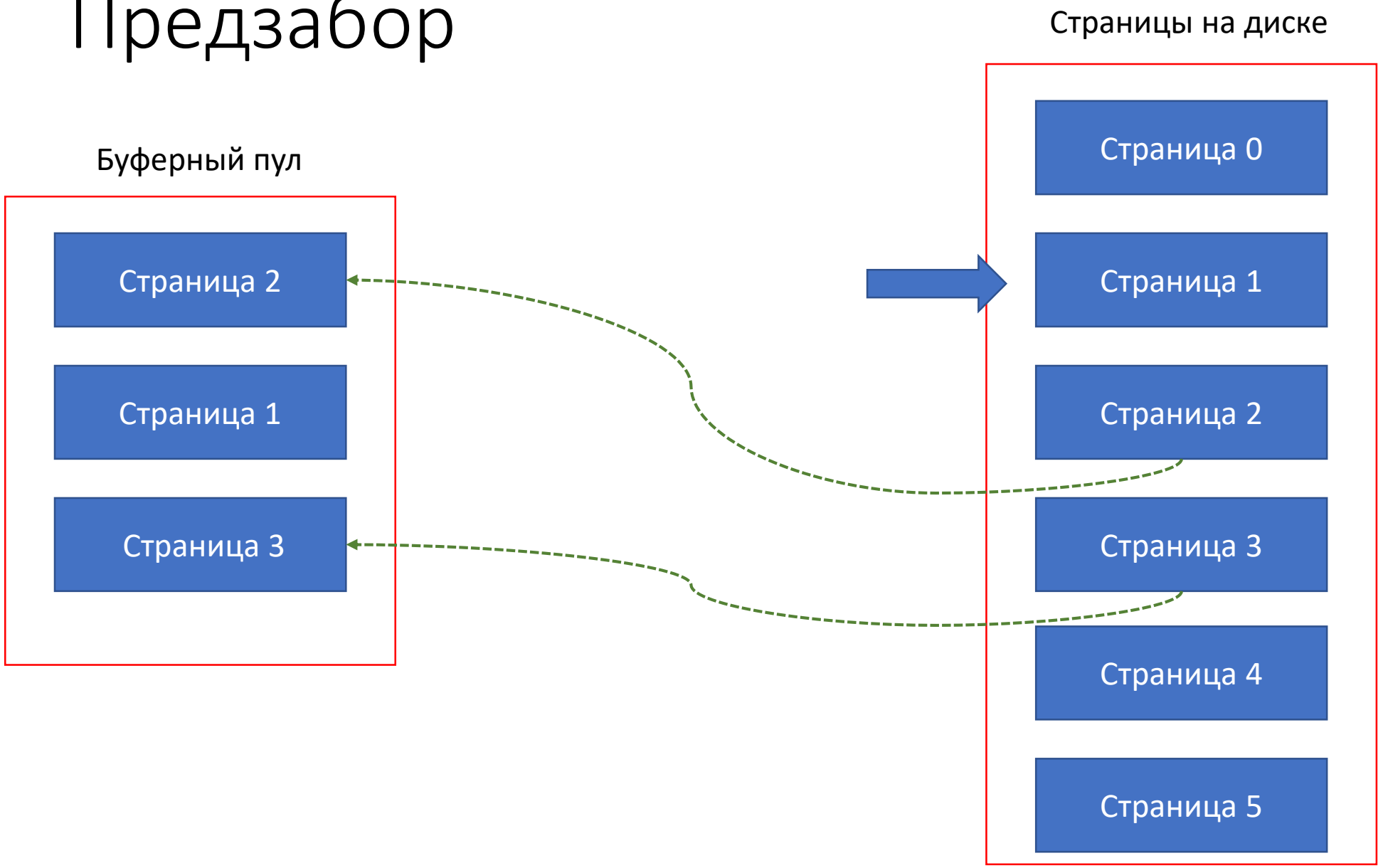
Предзабор


Страницы на диске
SELECT * FROM A
Буферный пул
Страница 0
Страница 1
Страница 0
Страница 1
Страница 2
Страница 3
Страница 4
Страница 5

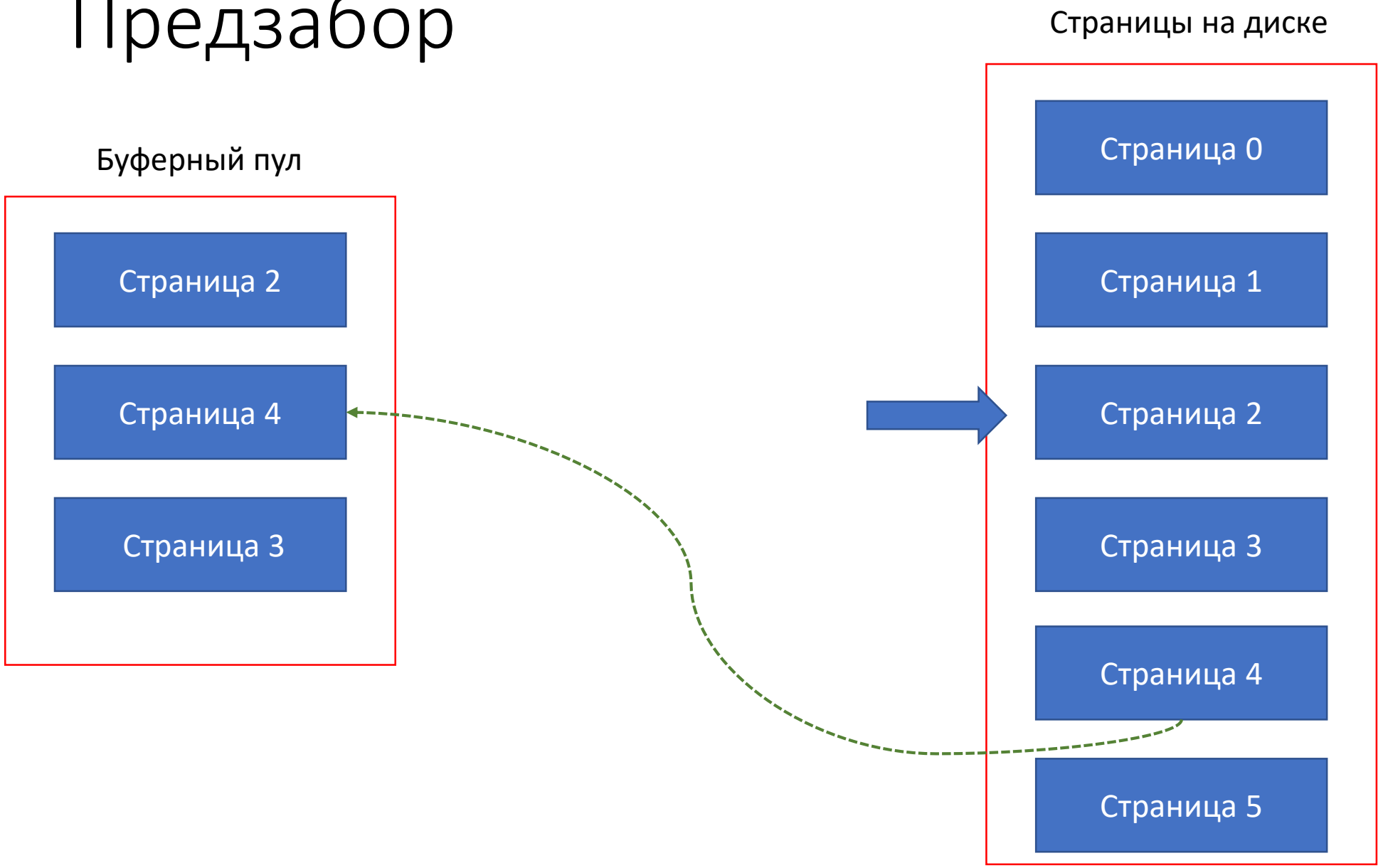
Предзабор


Страницы на диске
Страница 0
Страница 1
Страница 2
Страница 3
Страница 4
Страница 5
Буферный пул
Страница 0
Страница 1

Предзабор


Страницы на диске
Страница 0
Страница 1
Страница 2
Страница 3
Страница 4
Страница 5
Буферный пул
Страница 2
Страница 1
Страница 3

Предзабор


Страницы на диске
Страница 0
Страница 1
Страница 2
Страница 3
Страница 4
Страница 5
Буферный пул
Страница 2
Страница 4
Страница 3

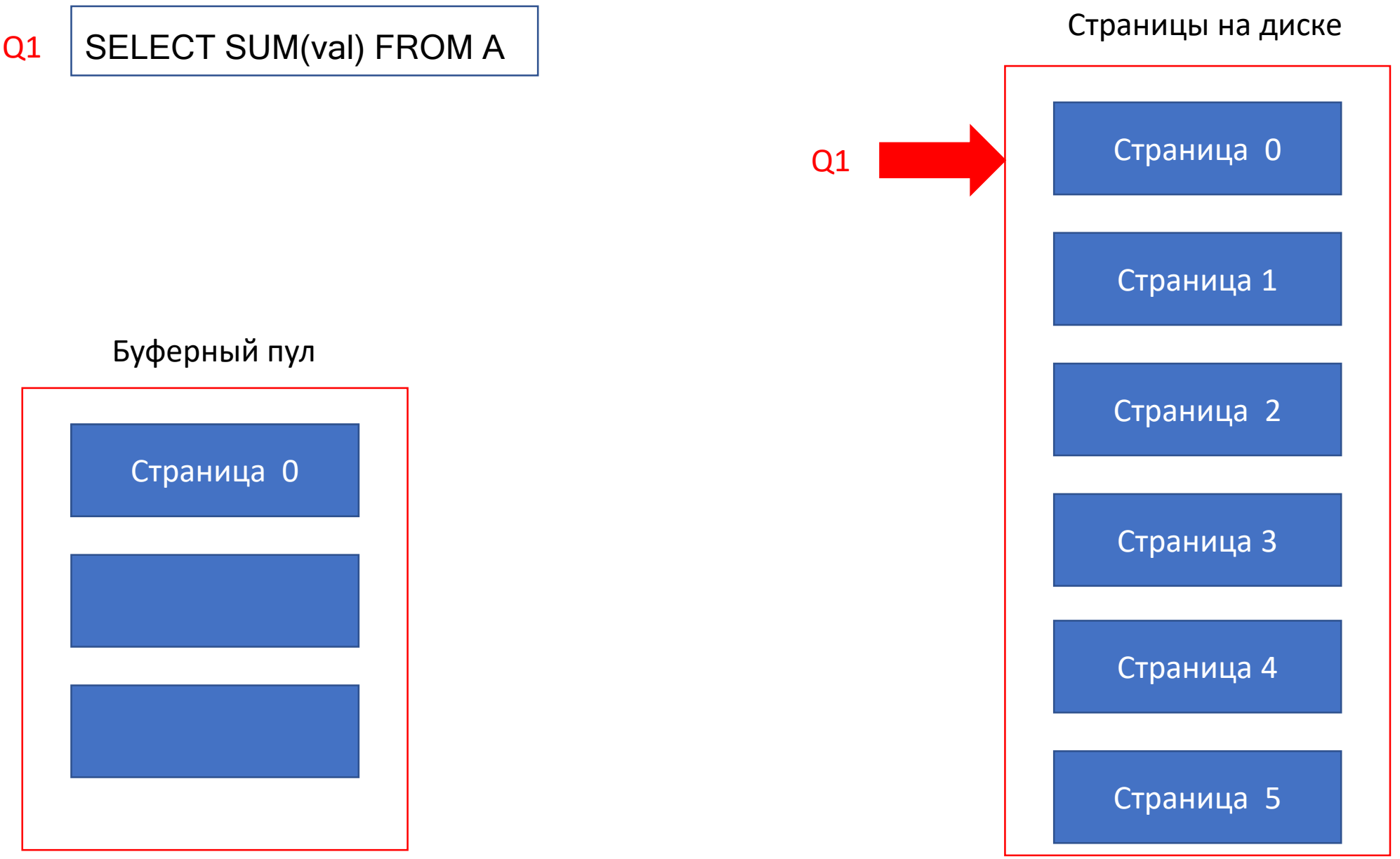
Совместный обход

- Запросы могут переиспользовать данные, полученные с диска, для вычисления операций
  - Данный вариант отличается от кеширования результата, когда мы фактически сохраняем результат выполнения операции, а затем периодически его переиспользуем.
- Несколько запросов могу использовать один и тот же курсор* в процессе чтения таблицы
  - Запросы не должны быть теми же самыми
  - Существует вариант, когда совместно переиспользуются промежуточные результаты

*Курсор — ссылка на контекстную область памяти.

Совместный обход

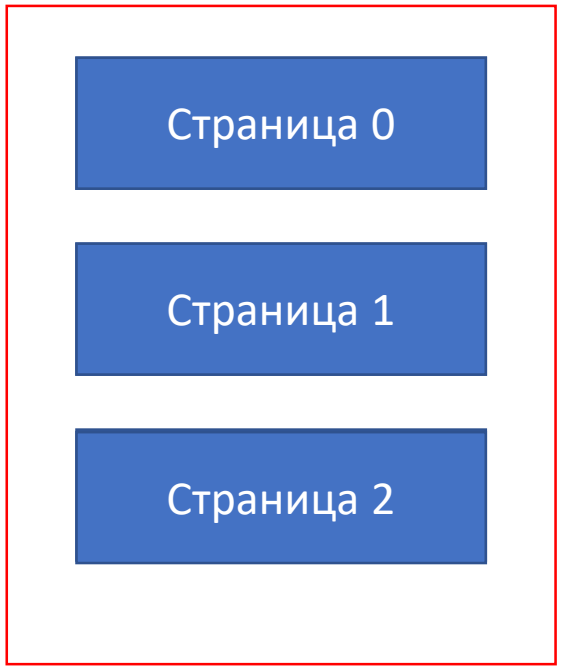
Q1 SELECT SUM(val) FROM A



Совместный обход

Q1 SELECT SUM(val) FROM A

Буферный пул



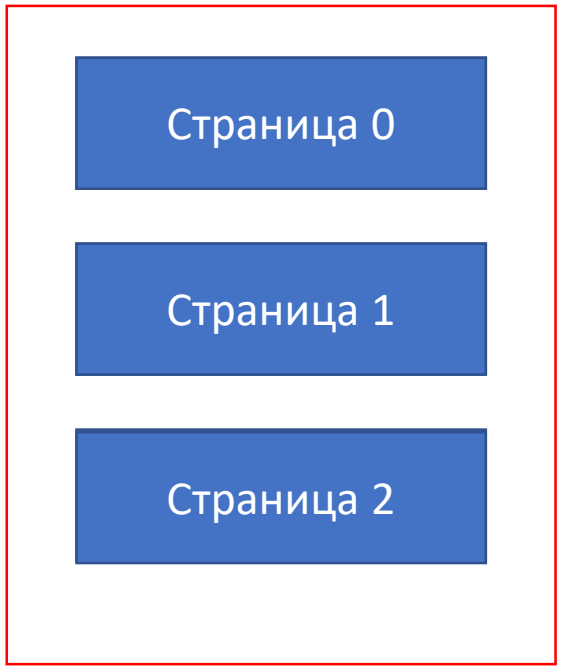
Страницы на диске

Совместный обход

Q1 SELECT SUM(val) FROM A

Q2 SELECT AVG(val) FROM A

Буферный пул



Q2 Q1

Страницы на диске

Совместный обход

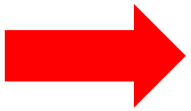
Q1 SELECT SUM(val) FROM A

Q2 SELECT AVG(val) FROM A

Буферный пул

Страницы на диске

Q2 Q1



Совместный обход

Q1 SELECT SUM(val) FROM A

Q2 SELECT AVG(val) FROM A

Страницы на диске

- Страница 0
- Страница 1
- Страница 2
- Страница 3
- Страница 4
- Страница 5

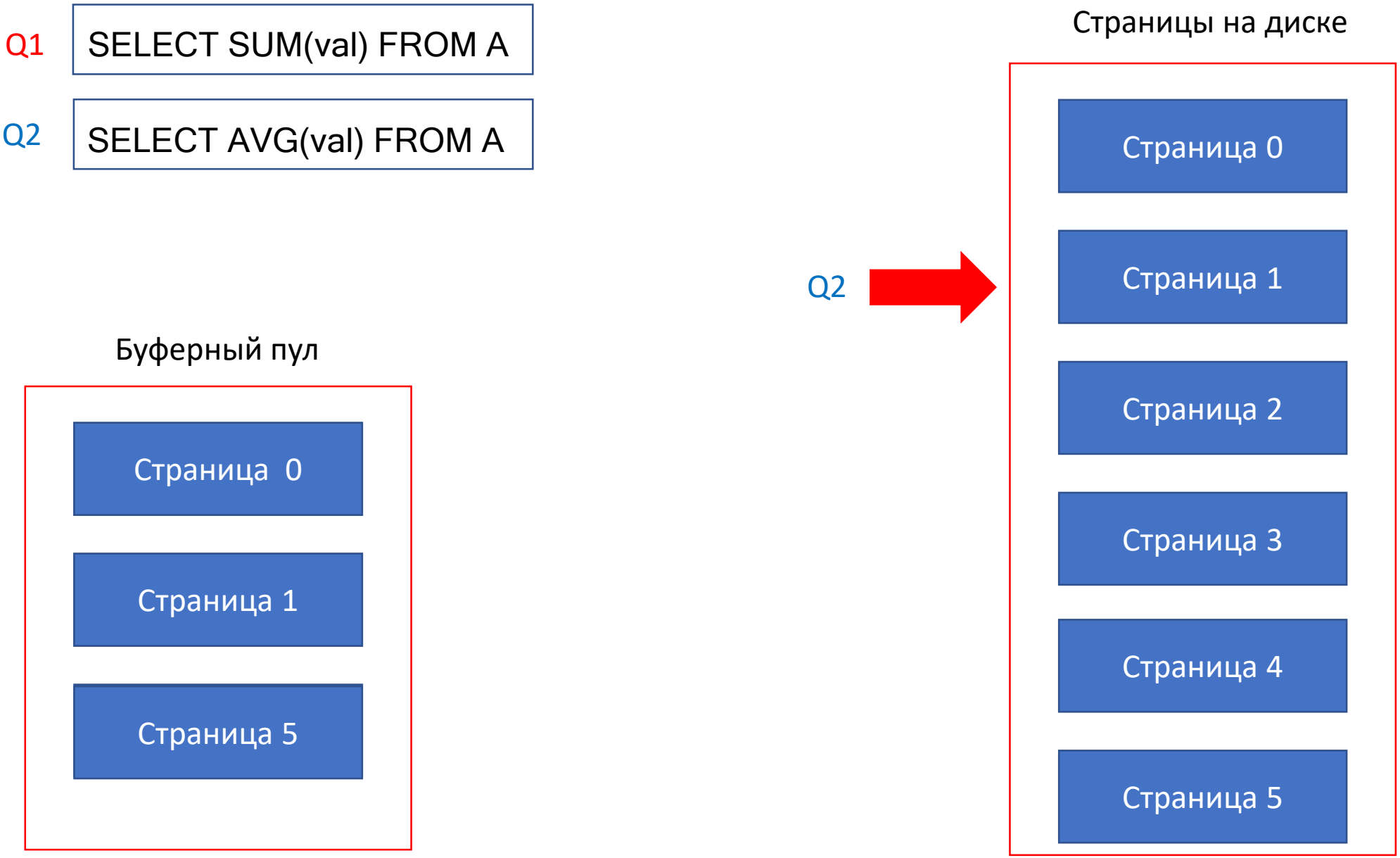
Буферный пул

- Страница 0
- Страница 1
- Страница 2

Совместный обход

Q1 SELECT SUM(val) FROM A

Q2 SELECT AVG(val) FROM A



Совместный обход

- Поддерживается IBM DB2 и SQL Server
- Oracle поддерживает только совместные курсоры для одинаковых запросов
- PostgreSQL содержит в себе структуры, позволяющие подобные операции

Политика замены буфера

Если СУБД требуется заменить один из фреймов для освобождения место новому фрейму, то необходимо выбрать как страницу требуется выбросить из буферного пула.

Основные свойства:

- Корректность
- Точность
- Скорость
- Накладные расходы на метаданных

FIFO

First in, first out

В FIFO содержится очередь идентификаторов страниц в порядке возрастания, добавляя страницы в конец очереди.

Когда буферный пул заполнен, берется элемент с начала очереди и выбрасывается.

Главный недостаток – отслеживается только первый вход; нет никакой информации о том, что страница забиралась еще раз.

LRU

Least Recently Used. Аналогично существует очередь из ID, однако при переиспользовании страницы она помещается в конец очереди.

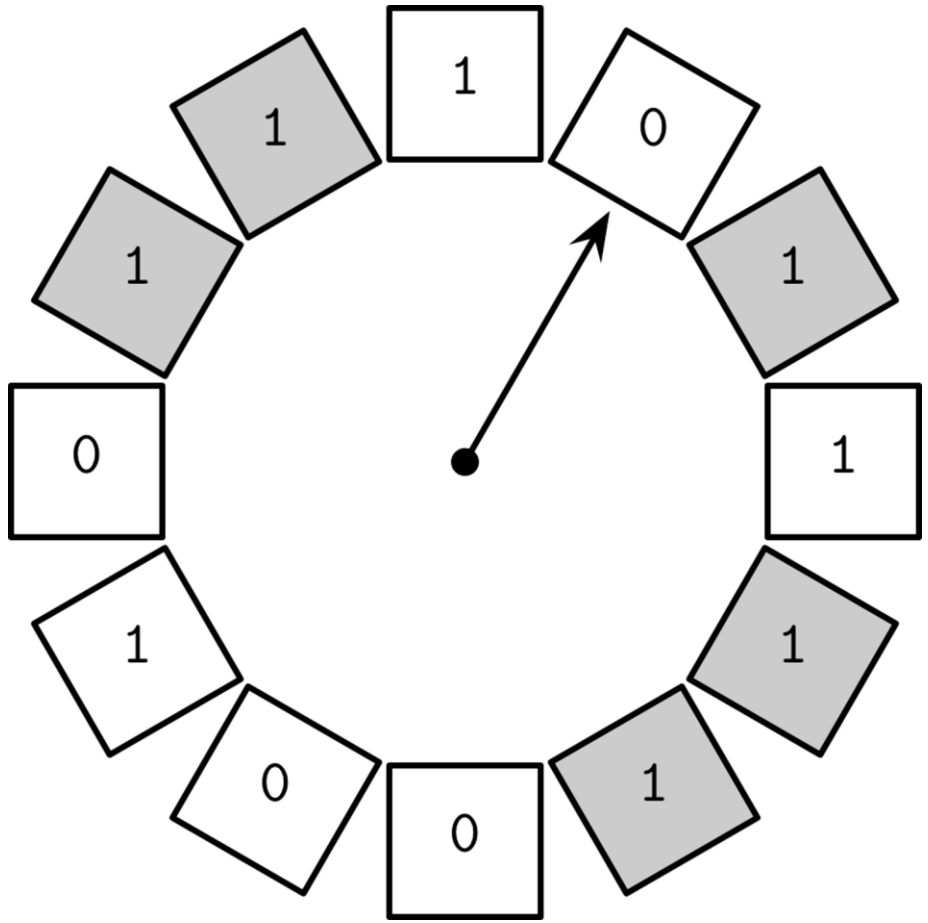
Недостаток – расходы на обновление ссылочности и перелинковку узлов очереди.

CLOCK

LRU алгоритмы могут быть довольно точными, но не всегда оптимально быстрыми. Алгоритм CLOCK используется, как альтернатива LRU.

CLOCK структура содержит ссылки на страницы и связанные с этими страницами биты в циклическом буфере.

CLOCK



Каждый раз, когда требуется страница, ее бит доступа становится 1. Алгоритм работает, обходя циклически следующим образом:

- Если бит доступа – 1, и на страницу нет ссылок, то в бит пишется 0 и осуществляется просмотр следующей страницы.
- Если бит доступа – 0, то страница становится *кандидатом* и планируется на выброс из буфера.
- Если на страницу есть ссылка, то бит остается неизменным.

LFU

Least Frequently Used. Вместо отслеживания считывания страницы с диска, отслеживает события ссылки на страницу.

Сортировка происходит по частоте использования страницы.