Введение в реляционные базы данных

Лекция 3: Операции над отношениями и Хранение

Артем Толканев

October 02, 2024

На этой лекции

- Продолжим рассматривать базовые операции SQL
- Начнем изучать хранение данных

Языки работы с СУБД

- **Data Query Language (DQL)**
- Data Manipulation Language (DML)
- Data Definition Language (DDL)
- Data Control Language (DCL)

И не только

```
SELECT * FROM book
```

book_id	name	theme_id	publishing_house_id	year_of_publishing	language_id
200	Война и Мир 1,2 том	1	11	1 998	1
201	Война и Мир 3,4 том	1	11	1 998	1
202	Othello	2	12	2 005	2
203	Курс аналитической геометрии	5	13	2 005	1
204	Обломов	1	14	2 005	1
205	Капитанская Дочка	1	15	2 005	1
206	Общая физика	5	13	2 005	1
207	Дубровский	1	15	2 005	1
208	Анна Каренина	1	11	2 005	1
209	1984	2	12	2 005	2
210	Мартин Иден	2	16	2 005	1
211	Сердца Трех	2	16	2 007	1
212	Белый Клык	2	16	2 007	1
213	Три Сестры	1	14	2 007	1
214	Русские сказки	4	14	2 007	1
215	Курс аналитической геометрии	5	13	2 006	1

Управление выводом данных

ORDER BY

Выводит кортежи в соответствии с сортировкой значений одного или нескольких атрибутов этих кортежей.

```
SELECT
            l.language
            ,b.name
      FROM
            book b
            INNER JOIN
            language l ON l.language_id = b.language_id
      ORDER BY
            l.language
```

language	name
английский	Война и Мир 1,2 том
английский	Война и Мир 3,4 том
французский	Othello
английский	Курс аналитической геометрии
английский	Обломов
английский	Капитанская Дочка
английский	Общая физика
английский	Дубровский
английский	Анна Каренина

ORDER BY

Выводит кортежи в соответствии с сортировкой значений одного или нескольких атрибутов этих кортежей.

```
SELECT
        l.language
        ,b.name
    FROM
        book b
        INNER JOIN
        language l ON l.language_id = b.language_id
    ORDER BY
        l.language
```

language	name
английский	Война и Мир 1,2 том
английский	Война и Мир 3,4 том
английский	Анна Каренина
английский	Курс аналитической геометрии
английский	Обломов
английский	Капитанская Дочка
английский	Общая физика
английский	Дубровский
французский	Othello

language	name
английский	Война и Мир 1,2 том
английский	Война и Мир 3,4 том
французский	Othello
английский	Курс аналитической геометрии
английский	Обломов
английский	Капитанская Дочка
английский	Общая физика
английский	Дубровский
английский	Анна Каренина

language	name
английский	Общая физика
английский	Война и Мир 3,4 том
английский	Анна Каренина
английский	Курс аналитической геометрии
английский	Обломов
английский	Капитанская Дочка
английский	Война и Мир 1,2 том
английский	Дубровский
французский	Othello

ртировкой

ибутов этих

```
SE
        l.language
        ,b.name
FROM
        book b
        INNER JOIN
        language l ON l.language_id = b.language_id
ORDER BY
        l.language
```

language	name
английский	Война и Мир 1,2 том
английский	Война и Мир 3,4 том
английский	Анна Каренина
английский	Курс аналитической геометрии
английский	Обломов
английский	Капитанская Дочка
английский	Общая физика
английский	Дубровский
французский	Othello

language	name
английский	Война и Мир 1,2 том
английский	Война и Мир 3,4 том
французский	Othello
английский	Курс аналитической геометрии
английский	Обломов
английский	Капитанская Дочка
английский	Общая физика
английский	Дубровский
английский	Анна Каренина

ORDER BY

Выводит кортежи в соответствии с сортировкой значений одного или нескольких атрибутов этих кортежей.

```
SELECT
        l.language
        ,b.name
        ,l.language_id * b.book_id AS k
    FROM
        book b
        INNER JOIN
        language l ON l.language_id = b.language_id
    ORDER BY
        l.language_id * b.book_id DESC
```

language	name	k
французский	Othello	404
английский	Курс аналитической геометрии	215
английский	Анна Каренина	208
английский	Дубровский	207
английский	Общая физика	206
английский	Капитанская Дочка	205
английский	Обломов	204
английский	Война и Мир 3,4 том	201
английский	Война и Мир 1,2 том	200

LIMIT\TOP\FETCH

OFFSET

```
SELECT * FROM book ORDER BY book_id LIMIT 2 OFFSET 4
SELECT * FROM book ORDER BY book_id OFFSET 4 ROWS FETCH NEXT 2 ROWS ONLY
```

book_id	name	theme_id	publishing_house_id	year_of_publishing	language_id
204	Обломов	1	14	2 005	1
205	Капитанская Дочка	1	15	2 005	1

Оконные функции

Выполняет вычисление на множестве кортежей, которые относятся к текущему кортежу, без сворачивания их в один кортеж. (типа GROUP BY без группировки)

```
SELECT
        count(*) OVER ()
    FROM
        book b
```

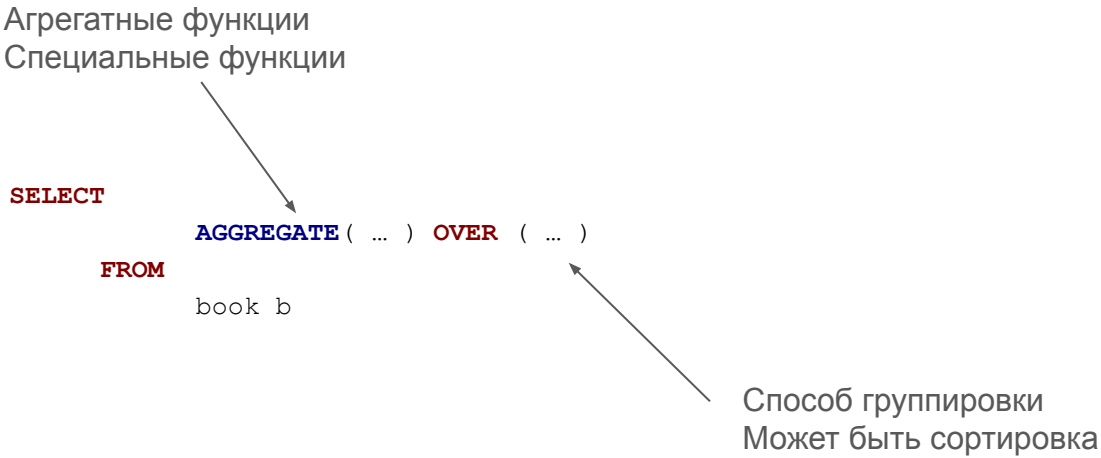
book_id	name	theme_id	publishing_house_id	year_of_publishing	language_id
200	Война и Мир 1,2 том	1	11	1 998	1
201	Война и Мир 3,4 том	1	11	1 998	1
202	Othello	2	12	2 005	2
203	Курс аналитической геометрии	5	13	2 005	1
204	Обломов	1	14	2 005	1
205	Капитанская Дочка	1	15	2 005	1
206	Общая физика	5	13	2 005	1
207	Дубровский	1	15	2 005	1
208	Анна Каренина	1	11	2 005	1
209	1984	2	12	2 005	2
210	Мартин Иден	2	16	2 005	1
211	Сердца Трех	2	16	2 007	1
212	Белый Клык	2	16	2 007	1
213	Три Сестры	1	14	2 007	1
214	Русские сказки	4	14	2 007	1
215	Курс аналитической геометрии	5	13	2 006	1

Выполняет вычисление на множестве кортежей, которые относятся к текущему кортежу, без сворачивания их в один кортеж. (типа GROUP BY без группировки)

```
SELECT
        count(*) OVER ()
    FROM
        book b
```

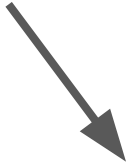
count
16
16
16
16
16
16
16
16
16
16
16
16
16
16
16
16

Выполняет вычисление на множестве кортежей, которые относятся к текущему кортежу, без сворачивания их в один кортеж. (типа GROUP BY без группировки)



language	name
английский	Война и Мир 1,2 том
английский	Война и Мир 3,4 том
французский	Othello
английский	Курс аналитической геометрии
английский	Обломов
английский	Капитанская Дочка
английский	Общая физика
английский	Дубровский
английский	Анна Каренина
русский	NULL
английский	Курс аналитической геометрии
немецкий	NULL

```
SELECT language,count(DISTINCT name)
FROM (
      SELECT
                  l.language
                  ,b.name
            FROM
                  book b
                  RIGHT JOIN
                  language l ON
l.language_id = b.language_id
                  ) AS t
      GROUP BY language
```



language	count
английский	13
французский	2
русский	0
немецкий	0

language	name
английский	Война и Мир 1,2 том
английский	Война и Мир 3,4 том
французский	Othello
английский	Курс аналитической геометрии
английский	Обломов
английский	Капитанская Дочка
английский	Общая физика
английский	Дубровский
английский	Анна Каренина
русский	NULL
английский	Курс аналитической геометрии
немецкий	NULL

```
SELECT language,count(DISTINCT name)
FROM (
      SELECT
                  l.language
                  ,b.name
            FROM
                  book b
                  RIGHT JOIN
                  language l ON
l.language_id = b.language_id
                  ) AS t
      GROUP BY language
```

```
SELECT
      language
      ,count(name) OVER (PARTITION BY
language) as c
FROM (
      SELECT DISTINCT
                  l.language
                  ,b.name
            FROM
                  book b
                  RIGHT JOIN
                  language l ON l.language_
= b.language_id
                  ) AS t
```

language	c
английский	13
английский	13
французский	2
английский	13
английский	13
английский	13
английский	13
английский	13
английский	13
русский	0
английский	13
немецкий	0

language	count
английский	13
французский	2
русский	0
немецкий	0

language	name
английский	Война и Мир 1,2 том
английский	Война и Мир 3,4 том
французский	Othello
английский	Курс аналитической геометрии
английский	Обломов
английский	Капитанская Дочка
английский	Общая физика
английский	Дубровский
английский	Анна Каренина
русский	NULL
английский	Курс аналитической геометрии
немецкий	NULL

```
SELECT
      t.language
      ,count(name) OVER (PARTITION BY
t.language) AS c
      ,ROW_NUMBER() OVER () AS rn
FROM (
      SELECT DISTINCT
                  l.language
                  ,b.name
            FROM
                  book b
                  RIGHT JOIN
                  language l ON l.language_id
= b.language_id
                  ) AS t
```

language	c	rn
английский	13	1
английский	13	2
французский	2	3
английский	13	4
английский	13	5
английский	13	6
английский	13	7
английский	13	8
английский	13	9
русский	0	10
английский	13	11
немецкий	0	12

```sql
SELECT
	*
	,ROW_NUMBER() OVER (ORDER BY language_id DESC ) as rn
	FROM "language" l
```

language_id	language	rn
0	русский	4
1	английский	3
2	французский	2
3	немецкий	1

```
SELECT
      t.language
      ,count(name) OVER (PARTITION BY t.language) AS c
      ,ROW_NUMBER() OVER () AS rn
FROM (
      SELECT DISTINCT
      l.language ,b.name
      FROM
      book b RIGHT JOIN
      language l ON l.language_id = b.language_id) AS t
```

language	c	rn
английский	13	1
английский	13	2
французский	2	3
английский	13	4
английский	13	5
английский	13	6
английский	13	7
английский	13	8
английский	13	9
русский	0	10
английский	13	11
немецкий	0	12

```
SELECT
      t.language
      ,count(name) OVER (PARTITION BY t.language) AS c
      ,ROW_NUMBER() OVER (PARTITION BY t.language) AS rn
FROM (
      SELECT DISTINCT
      l.language ,b.name
      FROM
      book b RIGHT JOIN
      language l ON l.language_id = b.language_id) AS t
```

language	c	rn
английский	13	1
английский	13	2
французский	2	1
английский	13	3
английский	13	4
английский	13	5
английский	13	6
английский	13	7
английский	13	8
русский	0	1
английский	13	9
немецкий	0	1

```sql
SELECT
      t.language
      ,count(name) OVER (PARTITION BY t.language) AS c
      ,ROW_NUMBER() OVER (PARTITION BY t.language) AS rn
FROM (
      SELECT DISTINCT
      l.language ,b.name
      FROM
      book b RIGHT JOIN
      language l ON l.language_id = b.language_id) AS t
```

But there's no guarantee that the same rows will be returned if you run the query again. If you are really after three arbitrary rows, it might be a good idea to add an ORDER BY clause with the expression (SELECT NULL) to let people know that your choice is intentional and not an oversight. Here's how your query would look.

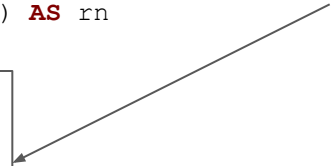
© Training Kit (Exam 70-461) Querying Microsoft SQL Server 2012 (MCSA)

Вложенные запросы

Прокидываем результат запроса внутрь другого запроса

```
SELECT
      t.language
      ,count(name) OVER (PARTITION BY t.language) AS c
      ,ROW_NUMBER() OVER (PARTITION BY t.language) AS rn
FROM (
      SELECT DISTINCT
      l.language ,b.name
      FROM
      book b RIGHT JOIN
      language l ON l.language_id = b.language_id) AS t
```

подзапрос



girafe ai

Прокидываем результат запроса внутрь другого запроса

language_id	language
0	русский
1	английский
2	французский
3	немецкий

language_id	language
1	английский
2	французский

```
SELECT
      *
      FROM "language" l
```

```
SELECT * FROM "language" l
      WHERE
            l.language_id  IN (SELECT language_id FROM book b)
;
```

подзапрос

```
SELECT l.language_id,l.language FROM
language l
WHERE
            l.language_id  = ANY (SELECT language_id FROM book b)
;
SELECT l.language_id,l.language FROM
language l
WHERE
            EXISTS(
SELECT 1 FROM book b WHERE language_id = l.language_id)
```

Прокидываем результат запроса внутрь другого запроса

language_id	language
0	русский
1	английский
2	французский
3	немецкий

language_id	language
3	немецкий
0	русский

```
SELECT
      *
      FROM "language" l


 SELECT * FROM "language" l
       WHERE
             l.language_id NOT IN(SELECT language_id FROM book b)
 ;

 SELECT l.language_id,l.language FROM
 language l
 WHERE
             l.language_id  <> ALL (SELECT language_id FROM book b)
 ;
 SELECT l.language_id,l.language FROM
 language l
 WHERE
            NOT EXISTS(
SELECT 1 FROM book b WHERE language_id = l.language_id)
```

подзапрос

Прокидываем результат запроса внутрь другого запроса

language_id	language	v
0	русский	10
1	английский	11
2	французский	12
3	немецкий	13

```
SELECT
        *
    FROM
    language l
    CROSS JOIN LATERAL (SELECT l.language_id + 10 AS v) AS t
```

подзапрос

language_id	language	v
0	русский	0
1	английский	13
2	французский	2
3	немецкий	0

```
SELECT
        *
    FROM
    language l
    CROSS JOIN LATERAL (
        SELECT count(DISTINCT b."name") AS v
            FROM
                book b
            WHERE
                b.language_id = l.language_id) AS t
```

JOIN LATERAL = APPLY в TSQL

Прокидываем результат запроса внутрь другого запроса

```
SELECT
        *
        ,(SELECT count(DISTINCT b.name) FROM book b WHERE b.language_id = l.language_id) AS v
FROM
        language l
```

подзапрос

language_id	language	v
0	русский	0
1	английский	13
2	французский	2
3	немецкий	0

```
SELECT
        *
FROM
language l
CROSS JOIN LATERAL (
        SELECT count(DISTINCT b."name") AS v
                FROM
                        book b
                WHERE
                        b.language_id = l.language_id) AS t
```

JOIN LATERAL = APPLY в TSQL

CTE (common table expression (обобщенное табличное выражение))

Определяет временный результат запроса, который впоследствии может быть использован в другой части запроса.

```
WITH t_language AS (
SELECT
      *
      FROM
      language l
)
SELECT
      *
      FROM
      t_language
```

language_id	language
0	русский
1	английский
2	французский
3	немецкий

Определяет временный результат запроса, который впоследствии может быть использован в другой части запроса.

```
WITH t_language AS (
SELECT
      *
      FROM
      language l
)
SELECT
      *
      FROM
      t_language
```

language_id	language
0	русский
1	английский
2	французский
3	немецкий

language_id	lang
0	русский
1	английский
2	французский
3	немецкий

```
WITH
      t_language(
            language_id
            ,lang)
AS (
SELECT
      *
      FROM
      language l
)
SELECT
      *
      FROM
      t_language
```


Определяет временный результат запроса, который впоследствии может быть использован в другой части запроса.

MySQL, MSSQL

```
WWITH
      t_language(
            lang
            ,lang)
AS (
SELECT
      *
      FROM
      language l
)
SELECT
      *
      FROM
      t_language
```

return: SQL Error

lang	lang
0	русский
1	английский
2	французский
3	немецкий

Postgres

```
WITH
      t_language(
            lang
            ,lang)
AS (
SELECT
      *
      FROM
      language l
)
SELECT
      *
      FROM
      t_language
```


Задача: вывести в столбце буквы английского алфавита в верхнем регистре

???

Задача: вывести в столбце буквы английского алфавита в верхнем регистре

???

```
SELECT 'A'
UNION
SELECT 'B'
...
```


Задача: вывести в столбце буквы английского алфавита в верхнем регистре

???

```
WITH t AS (
SELECT 1 AS n
UNION ALL
SELECT 2
),c AS (
SELECT
            ascii('A') - 1 + ROW_NUMBER() OVER () AS n
      FROM  t t1  CROSS JOIN  t t2  CROSS JOIN  t t3  CROSS JOIN
      t t4  CROSS JOIN  t t5
)
SELECT
            CHR(n::int)
      FROM
            c
      WHERE
            c.n <= ascii('Z')
```


Задача: вывести в столбце буквы английского алфавита в верхнем регистре

```
WITH RECURSIVE t AS (
SELECT
      65 AS n
UNION ALL
SELECT
      n + 1 AS n
FROM t
WHERE
      n < 65 + (ascii('Z') -
ascii('A'))
)
SELECT CHR(n) FROM t
```

Postgres

```
WITH t AS (
SELECT
      65 AS n
UNION ALL
SELECT
      n + 1 AS n
FROM t
WHERE
      n < 65 + (ascii('Z') -
ascii('A'))
)
SELECT CHAR(n) FROM t
```

MS SQL

Можем рекурсивно работать с целым отношением

```
WITH RECURSIVE t AS (
SELECT
      1 AS n ,l.language_id, l.language
FROM language l
UNION ALL
SELECT
      n + 1 AS n ,l.language_id, l.language
FROM t
INNER JOIN language l
ON t.language_id < l.language_id
)
SELECT * FROM t
```

Postgres

Хранение

Реляционные базы данных подразумевают хранение данных на энергонезависимой памяти.

Хранение

Реляционные базы данных подразумевают хранение данных на энергонезависимой памяти.

СУБД отвечает за нас передвижением данных между энергозависимой (ОЗУ) и энергонезависимой памятью.

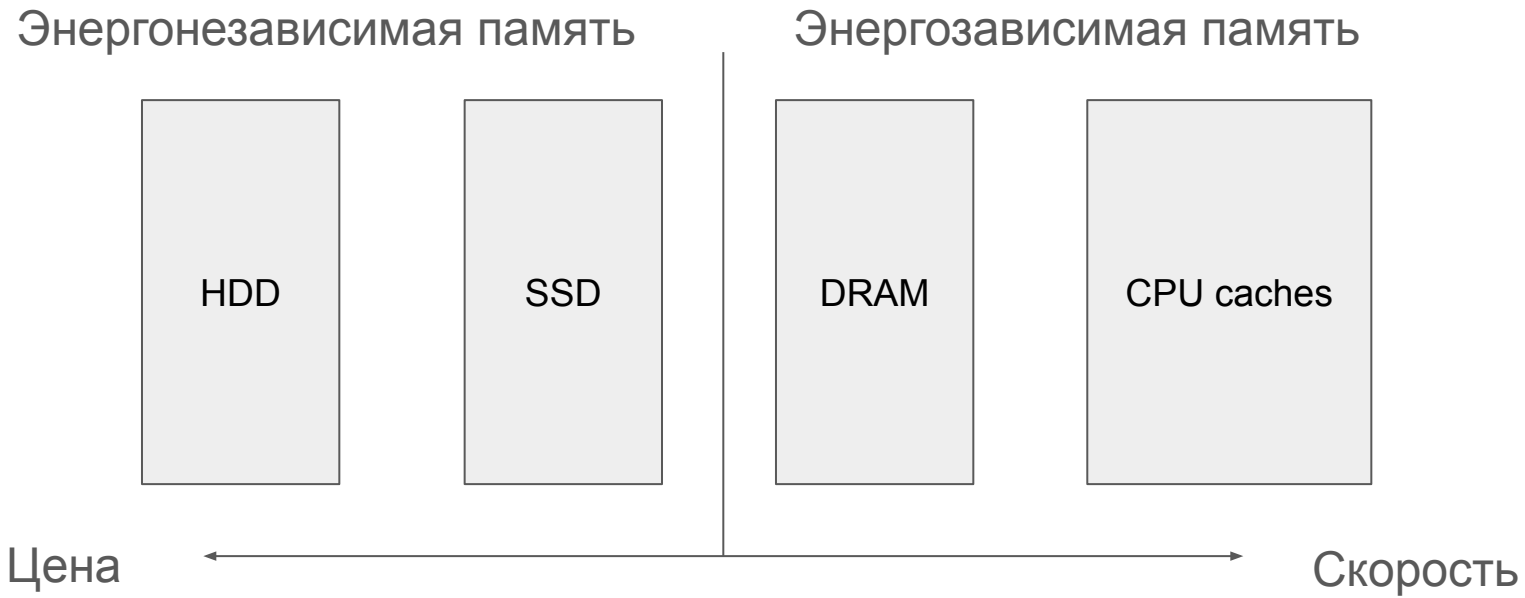
Хранение

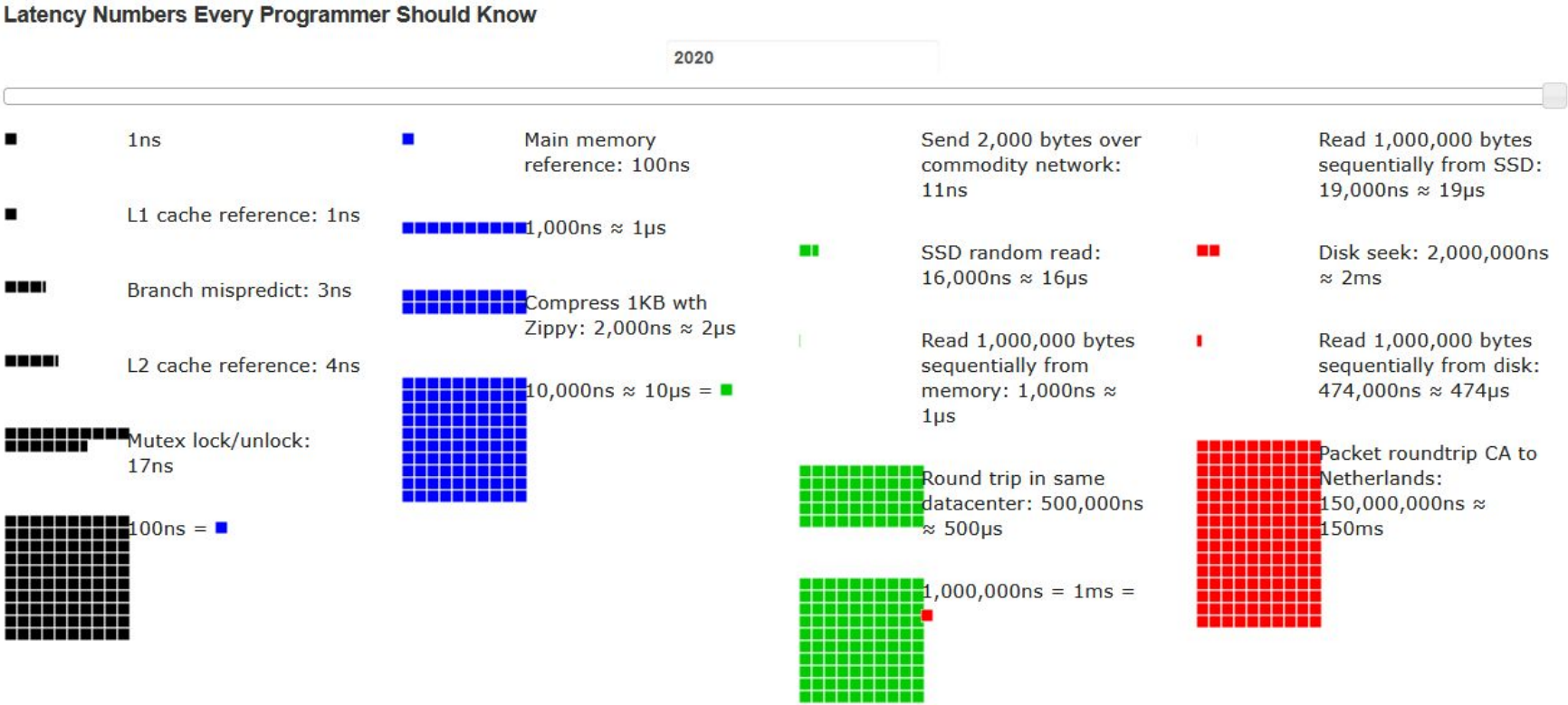
Реляционные базы данных подразумевают хранение данных на энергонезависимой памяти.

СУБД отвечает за нас передвижением данных между энергозависимой (ОЗУ) и энергонезависимой памятью.

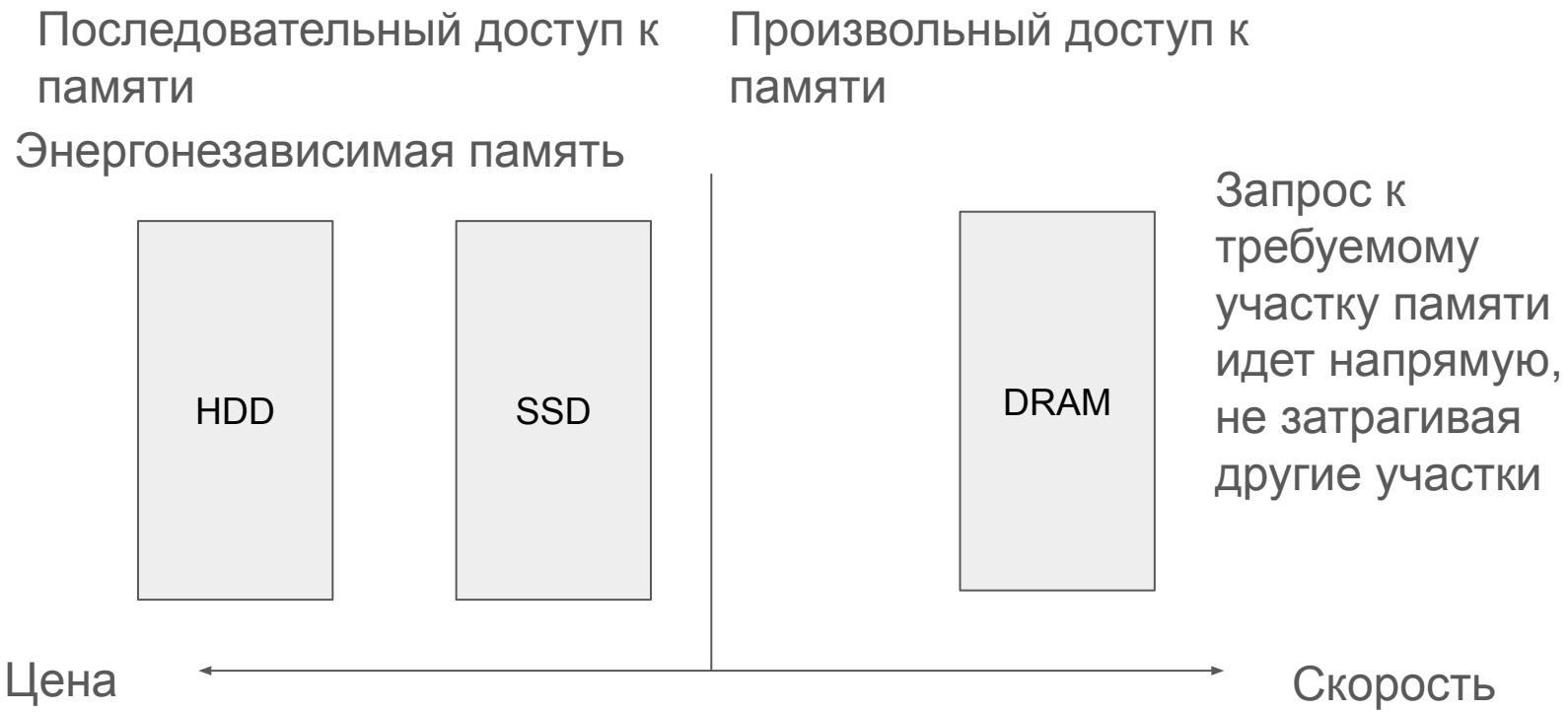
Возникает вопрос: если мы оперируем с данными в ОЗУ, то как мы можем отвечать за то, что они корректно переносятся на диск?

Иерархия хранения данных концептуально:




Latency Numbers Every Programmer Should Know
2020
1ns
L1 cache reference: 1ns
Branch mispredict: 3ns
L2 cache reference: 4ns
Mutex lock/unlock: 17ns
100ns =
Main memory reference: 100ns
1,000ns ≈ 1µs
Compress 1KB wth Zippy: 2,000ns ≈ 2µs
10,000ns ≈ 10µs =
Send 2,000 bytes over commodity network: 11ns
SSD random read: 16,000ns ≈ 16µs
Read 1,000,000 bytes sequentially from memory: 1,000ns ≈ 1µs
Round trip in same datacenter: 500,000ns ≈ 500µs
1,000,000ns = 1ms =
Read 1,000,000 bytes sequentially from SSD: 19,000ns ≈ 19µs
Disk seek: 2,000,000ns ≈ 2ms
Read 1,000,000 bytes sequentially from disk: 474,000ns ≈ 474µs
Packet roundtrip CA to Netherlands: 150,000,000ns ≈ 150ms

Иерархия хранения данных концептуально:

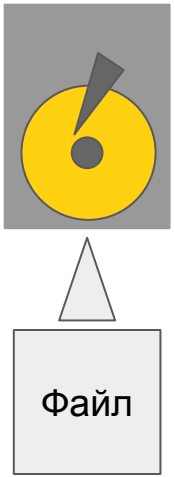


СУБД организовано так, что
она пытается максимизировать последовательный доступ к памяти

уменьшить количество записей в случайные участки памяти, так чтобы данные хранились в непрерывных блоках

HDD

Хранение СУБД
На диске присутствует файл базы данных (или несколько)



Хранение СУБД
На диске присутствует файл базы данных (или несколько)

Файл базы данных состоит из нескольких частей, с которыми работает СУБД

Хранение СУБД
На диске присутствует файл базы данных (или несколько)

Файл базы данных состоит из нескольких частей, с которыми работает СУБД

ОЗУ

Работа с СУБД

Buffer pool
Buffer pool manager
Cache manager

ОЗУ

Работа с СУБД

Buffer pool
Buffer pool manager
Cache manager

Работа с СУБД

Блок СУБД для получения данных

ОЗУ

Дай мне страницу 1, чтобы выполнить запрос

HDD

Работа с СУБД

Блок СУБД для получения данных

Дай мне страницу 1, чтобы выполнить запрос

ОЗУ

HDD

Работа с СУБД

Блок СУБД для получения данных

Дай мне страницу 1, чтобы выполнить запрос

ОЗУ

Файл метаданных о самом файле		

HDD

Файл метаданных о самом файле	Страница 1	Страница 2	Страница 3

Работа с СУБД

Блок СУБД для получения данных

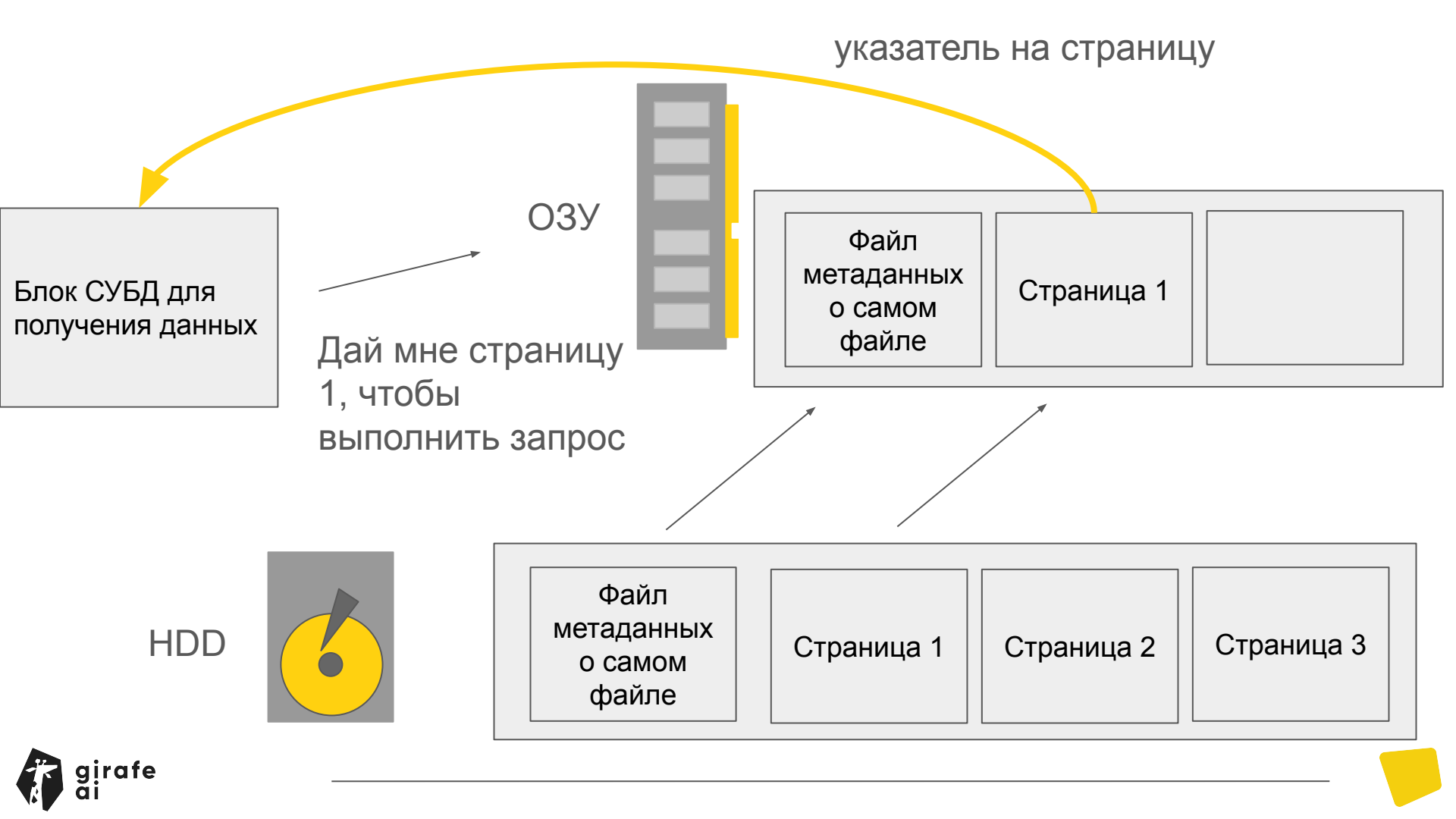
Дай мне страницу 1, чтобы выполнить запрос

ОЗУ

Файл метаданных о самом файле	Страница 1	

HDD

Файл метаданных о самом файле	Страница 1	Страница 2	Страница 3


указатель на страницу
Блок СУБД для получения данных
ОЗУ
Дай мне страницу 1, чтобы выполнить запрос
Файл метаданных о самом файле
Страница 1
HDD
Файл метаданных о самом файле
Страница 1
Страница 2
Страница 3


указатель на страницу
ОЗУ
Блок СУБД для получения данных
Файл метаданных о самом файле
Страница 1
Дай мне страницу 1, чтобы выполнить запрос
что-то делает
HDD
Файл метаданных о самом файле
Страница 1
Страница 2
Страница 3

указатель на страницу

Блок СУБД для получения данных

ОЗУ

Файл метаданных о самом файле

Страница 1

Дай мне страницу 1, чтобы выполнить запрос

Я ХОЧУ ОБНОВИТЬ СТРАНИЦУ 1

HDD

Файл метаданных о самом файле

Страница 1

Страница 2

Страница 3

girafe ai


указатель на страницу
ОЗУ
Блок СУБД для получения данных
Файл метаданных о самом файле
Страница 1
Дай мне страницу 1, чтобы выполнить запрос
Я ХОЧУ ОБНОВИТЬ СТРАНИЦУ 1
HDD
Файл метаданных о самом файле
Страница 1
Страница 2
Страница 3


указатель на страницу
ОЗУ
Блок СУБД для получения данных
Дай мне страницу 1, чтобы выполнить запрос
Файл метаданных о самом файле
Страница 1
Я ХОЧУ ОБНОВИТЬ СТРАНИЦУ 1
HDD
Файл метаданных о самом файле
Страница 1
Страница 2
Страница 3

Хранение СУБД
На диске присутствует файл базы данных (или несколько)

Файл базы данных состоит из нескольких частей, с которыми работает СУБД

СУБД хранит базу данных в одном или нескольких файлах

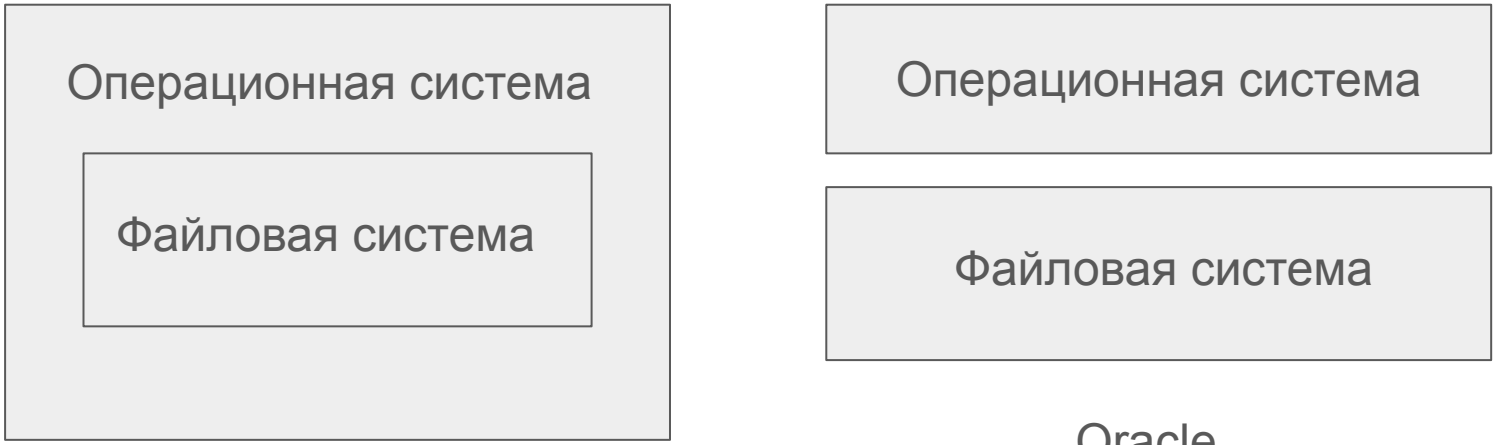
Она не знает о содержании этих файлов

Другие СУБД тоже не знают о содержании файлов

СУБД хранит базу данных в одном или нескольких файлах

Она не знает о содержании этих файлов

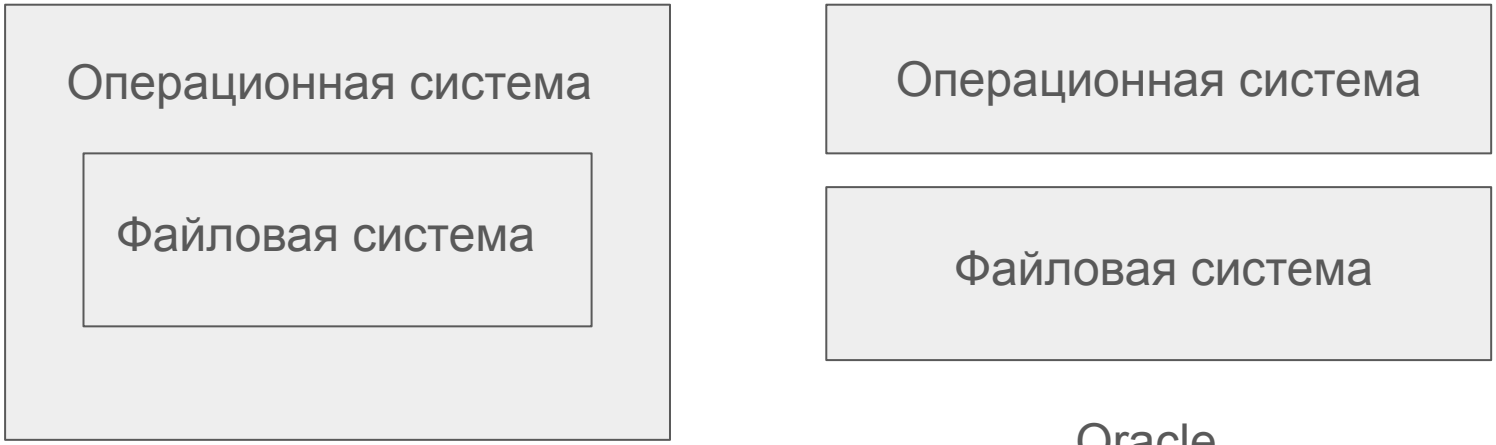
Другие СУБД тоже не знают о содержании файлов



Oracle

У СУБД для управления файлами есть специальный файловый менеджер

Он занимается организацией хранения файлов БД и страниц в файлах



Страницы памяти - блок данных фиксированного размера

В ней хранятся разные объекты памяти

У каждой страницы есть свой уникальный ID

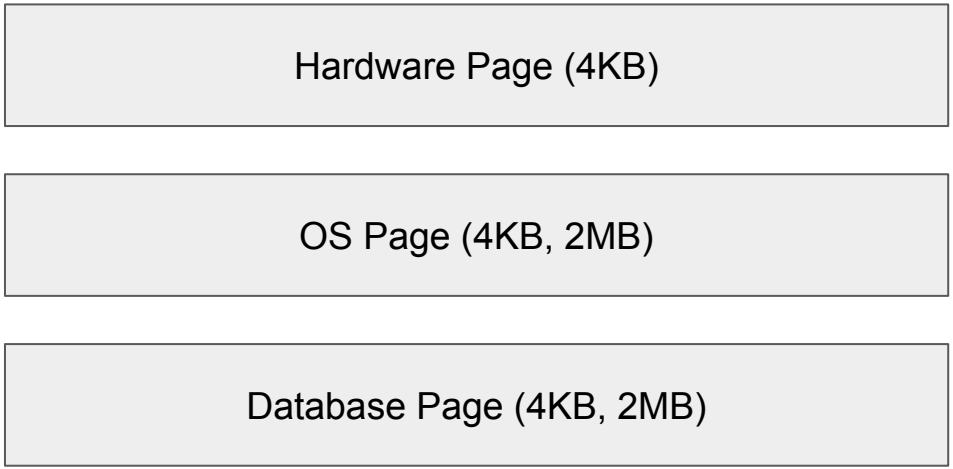
Описание того что хранится в странице дано в самой странице

Есть несколько вариантов понимая страниц в БД

Hardware Page (4KB)

Наибольший объем памяти,
который может быть Атомарно

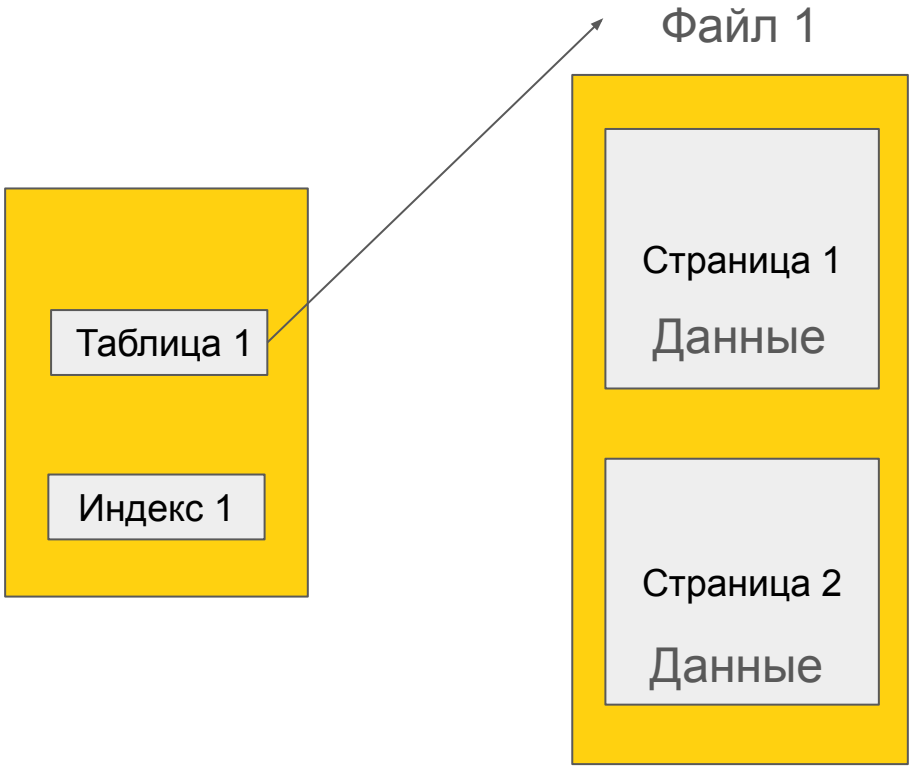
Есть несколько вариантов понимая страниц, которые вы можете увидеть



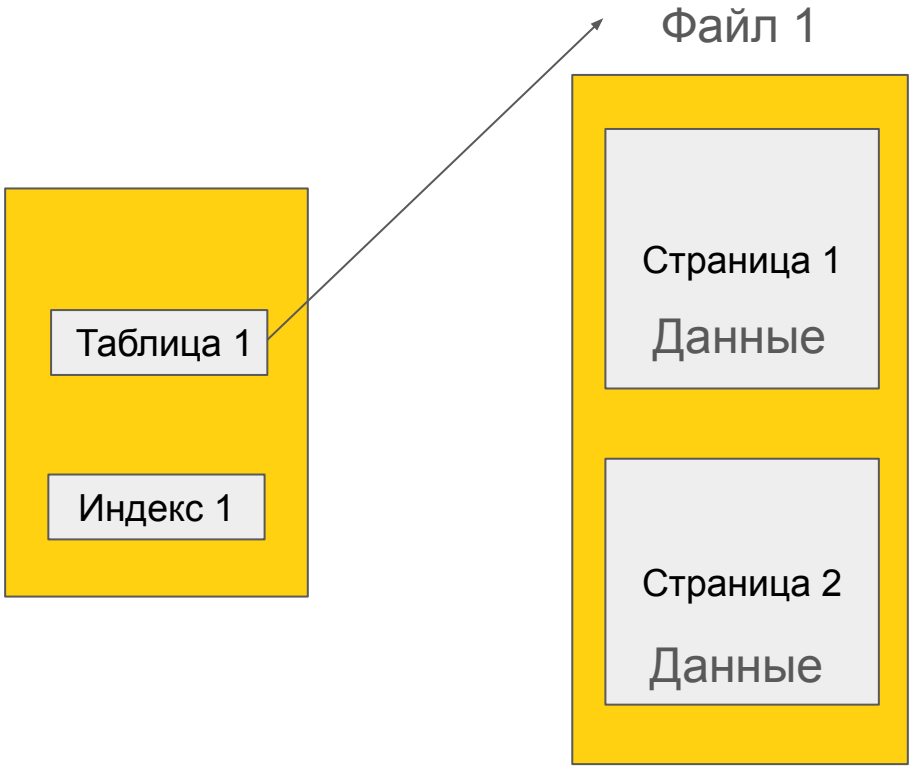
MySQL 16KB

MS SQL, Postgres 8KB

СУБД поддерживает специальную страницу, которая содержит информацию о других страницах в файлах БД.



СУБД поддерживает специальную страницу, которая содержит информацию о других страницах в файлах БД.



Количество свободного места на страницу
Список пустых страниц
Тип страницы