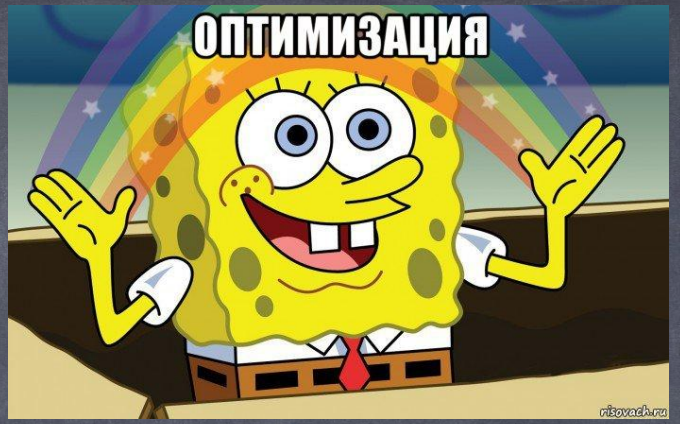


SQL

Оптимизация

3

Содержание

1) Что такое оптимизация и её типы
2) Оптимизация средствами БД
3) Оптимизация запросов
4) Средства анализа запросов
5) Примеры запросов

Цель лекции

Цели Лекции:

- Рассмотреть основные методы повышения производительности запросов
- Рассмотреть примеры правильного написания запросов
- Познакомить со средствами анализа запросов

Оптимизация

Что же такое оптимизация в глобальном смысле?

Оптимизация - процесс максимизации выгодных характеристик

Например:

- Время выполнения работ
- Затраты
- Качество
- И что то среднее («Оптимальное»)…

Оптимизация

Классификация возможных видов оптимизации в БД:

- Средствами БД
- Запросов

Оптимизация средствами БД

Что можно сделать средствами БД?

1) Работа с индексами

index
2
3
4
5
12
13
14
16
17
19

org_id	org_name
5	Выставочный центр СО РАН
3	ГНЦ Вектор
16	ГПНТБ
14	Геофизическая служба СО РАН
12	Издательство СО РАН
2	Институт автоматики и электрометрии СО РАН
17	Институт археологии и этнографии СО РАН
4	Институт вычислительных технологий СО РАН
13	Институт геологии и минералогии СО РАН
19	Институт горного дела СО РАН

Оптимизация средствами БД

Что можно сделать средствами БД?

2) Работа с кэшем

```
SHOW VARIABLES LIKE '%query_cache%';
```

session_variables

SHOW VARIABLES LIKE '%query_cache%' | Enter a SQL expre

	Variable_name	Value
1	have_query_cache	YES
2	query_cache_limit	1048576
3	query_cache_min_res_unit	4096
4	query_cache_size	16777216
5	query_cache_type	OFF
6	query_cache_wlock_invalidate	OFF

Оптимизация средствами БД

Что можно сделать средствами БД?

3) Хинтинг Индексов

```
tbl_name [[AS] alias] [index_hint_list]

index_hint_list:
    index_hint [index_hint] ...

index_hint:
    USE {INDEX|KEY}
      [FOR {JOIN|ORDER BY|GROUP BY}] ([index_list])
  | IGNORE {INDEX|KEY}
      [FOR {JOIN|ORDER BY|GROUP BY}] (index_list)
  | FORCE {INDEX|KEY}
      [FOR {JOIN|ORDER BY|GROUP BY}] (index_list)

index_list:
    index_name [, index_name] ...
```

Оптимизация средствами БД

Что можно сделать средствами БД?

4) Партиционирование таблиц (partitioning)

```
CREATE TABLE orders_range (
customer_surname VARCHAR(30),
store_id INT,
salesperson_id INT,
order_date DATE,
note VARCHAR(500)
) ENGINE = MYISAM
PARTITION BY RANGE( YEAR(order_date) ) (
PARTITION p_old VALUES LESS THAN(2008),
PARTITION p_2008 VALUES LESS THAN(2009),
PARTITION p_2009 VALUES LESS THAN(MAXVALUE)
);
```

Оптимизация средствами БД

Что можно сделать средствами БД?

5) Работа с Сессиями пользователей и Блокировкой таблиц

Оптимизация средствами БД

Что можно сделать средствами БД?

6) Дополнительные приятные мелочи

- Грамотное использование ключей (ID)
- Использование полей с типом NOT NULL
- Использование фиксированной длины значений
- А также все остальные логичные вещи…

Оптимизация запросов

1) Стараемся НЕ использовать SELECT * …

Больше – Дольше ☺️

```
select * from
    (select @t:=@t+1 as ROWNUM1 , T.* from employees.employees T, (SELECT @t:=0) AS foo) as a1 where a1.ROWNUM1 BETWEEN 5 and 10;
```

employees

select * from (select @t:=@t+1 as ROWNUM1 , T.* from employees.em | Enter a SQL expression to filter results (use Ctrl+Space)

	ROWNUM1	emp_no	birth_date	first_name	last_name	gender	hire_date
1	5	10,005	1955-01-21	Kyoichi	Maliniak	M	1989-09-12
2	6	10,006	1953-04-20	Anneke	Preusig	F	1989-06-02
3	7	10,007	1957-05-23	Tzvetan	Zielinski	F	1989-02-10
4	8	10,008	1958-02-19	Saniya	Kalloufi	M	1994-09-15
5	9	10,009	1952-04-19	Sumant	Peac	F	1985-02-18
6	10	10,010	1963-06-01	Duangkaew	Piveteau	F	1989-08-24

Save Cancel Script Record Panels Grid Text 6 row(s) fetched - 543ms

Оптимизация запросов

Попытка 1

```
select a1.ROWNUM1, a1.first_name, a1.last_name from
    (select @t:=@t+1 as ROWNUM1 , T.* from employees.employees T, (SELECT @t:=0) AS foo) as a1 where a1.ROWNUM1 BETWEEN 5 and 10;
```

employees

select a1.ROWNUM1, a1.first_name, a1.last_name from (select @t:=@t+ | Enter a SQL expression to filter results (use Ctrl+Space)

	123 ROWNUM1	ABC first_name	ABC last_name
1	5	Kyoichi	Maliniak
2	6	Anneke	Preusig
3	7	Tzvetan	Zielinski
4	8	Saniya	Kalloufi
5	9	Sumant	Peac
6	10	Duangkaew	Piveteau

Save Cancel Script Record Panels Grid Text 6 row(s) fetched - 502ms

Оптимизация запросов

Попытка 2

```
select * from
    (select @t:=@t+1 as ROWNUM1 , T.first_name, T.last_name from employees.employees T, (SELECT @t:=0) AS foo) as a1 where a1.ROWNUM1 BETWEEN 5 and 10;
```

employees

select * from (select @t:=@t+1 as ROWNUM1 , T.first_name, T.last_nam | Enter a SQL expression to filter results (use Ctrl+Space)

	123 ROWNUM1	ABC first_name	ABC last_name
1	5	Kyoichi	Maliniak
2	6	Anneke	Preusig
3	7	Tzvetan	Zielinski
4	8	Saniya	Kalloufi
5	9	Sumant	Peac
6	10	Duangkaew	Piveteau

Save | Cancel | Script | Record | Panels | Grid | Text | 6 row(s) fetched - 168ms

Оптимизация запросов

2) Используйте LIMIT если необходим НЕ полный список результатов

```
select emp_no, first_name, last_name
from employees where first_name like 'Mar%';
```

select emp_no, first_name, last_name from employees wh | Enter a SQL expression to filter results (use Ctrl+Space)

	emp_no	first_name	last_name
1	10,011	Mary	Sluis
2	10,069	Margareta	Bierman
3	10,109	Mariusz	Prampolini
4	10,137	Maren	Hutton
5	10,144	Marla	Brendel
6	10,196	Marc	Hellwagner
7	10,232	Marko	Auria
8	10,249	Marie	Boreale
9	10,275	Marek	Luck

Save | Cancel | Script | Record | Panels | Grid | Text | 200 row(s) fetched - 4ms

```
select emp_no, first_name, last_name
from employees where first_name like 'Mar%' limit 5;
```

select emp_no, first_name, last_name from employees wh | Enter a SQL expression to filter results (use Ctrl+Space)

	emp_no	first_name	last_name
1	10,011	Mary	Sluis
2	10,069	Margareta	Bierman
3	10,109	Mariusz	Prampolini
4	10,137	Maren	Hutton
5	10,144	Marla	Brendel

Save | Cancel | Script | Record | Panels | Grid | Text | 5 row(s) fetched - 2ms

Оптимизация запросов

3) Использование пользовательских переменных

\+ Сохранение промежуточных значений

\- Сложная переносимость на другие языки

```
mysql> SELECT @min_price:=MIN(price),@max_price:=MAX(price) FROM shop;
mysql> SELECT * FROM shop WHERE price=@min_price OR price=@max_price;
+---------+--------+-------+
| article | dealer | price |
+---------+--------+-------+
|    0003 | D      |  1.25 |
|    0004 | D      | 19.95 |
+---------+--------+-------+
```

Оптимизация запросов

4) Количество и последовательность JOIN-ов

В какой последовательности будут выполнены JOIN-ы ?

```
SELECT
  *
FROM
  Table1
INNER JOIN
  Table2 ON P1(Table1,Table2)
INNER JOIN
  Table3 ON P2(Table2,Table3)
WHERE
   P(Table1,Table2,Table3).
```

Оптимизация запросов

Последовательность JOIN-ов

```
select *
  from таблица_в_которой_10_строк
  join таблица_в_которой_1000_строк
  join таблица_в_которой_1000000_строк
```

Оптимизация запросов

5) Старайтесь избегать большого числа запросов в циклах
(Особенно: UPDATE / DELETE / INSERT)

```
"DELETE FROM table1 WHERE user_id='$user_id';
DELETE FROM table2 WHERE user_id='$user_id';
DELETE FROM table3 WHERE user_id='$user_id';
DELETE FROM table4 WHERE user_id='$user_id';";
```

```
DELETE t1, t2 FROM t1 INNER JOIN t2 INNER JOIN t3
WHERE t1.id=t2.id AND t2.id=t3.id;
```

Оптимизация запросов

6) Используйте `EXISTS` вместо `IN` при «Тяжелых» запросах

Результат выполнения:

EXISTS намного быстрее, чем IN, когда результаты подзапроса очень велики.

IN быстрее, чем EXISTS, когда результаты суб-запроса очень малы.

```
1  SELECT column1 FROM t1 WHERE EXISTS (SELECT * FROM t2);
```

Найти тех производителей портативных компьютеров, которые также производят принтеры:

Выполнить

Консоль

```
1.  SELECT DISTINCT maker
2.  FROM Product AS lap_product
3.  WHERE type = 'laptop' AND
4.   EXISTS (SELECT maker
5.   FROM Product
6.   WHERE type = 'printer' AND
7.   maker = lap_product.maker
8.   );
```

Оптимизация запросов

7) Использование временных таблиц

Оптимизация запросов

8) Обращайте внимание на ТИП ДАННЫХ, который используете для запросов.

- Используйте ID
- Работайте меньше с Текстовыми значениями

Оптимизация запросов

Для Исследования собственного запроса можно использовать функцию **EXPLAIN**

Result

EXPLAIN select * from employees where emp_no = 10005 | Enter a SQL expression to filter results (use Ctrl+Space)

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	1	SIMPLE	employees	[NULL]	const	PRIMARY	PRIMARY	4	const	1	100.00	[NULL]

Column	JSON Name	Meaning
id	select_id	The SELECT identifier
select_type	None	The SELECT type
table	table_name	The table for the output row
partitions	partitions	The matching partitions
type	access_type	The join type
possible_keys	possible_keys	The possible indexes to choose
key	key	The index actually chosen
key_len	key_length	The length of the chosen key
ref	ref	The columns compared to the index
rows	rows	Estimate of rows to be examined
filtered	filtered	Percentage of rows filtered by table condition
Extra	None	Additional information

Оптимизация запросов

Result

EXPLAIN select * from employees where emp_no = 10005 | Enter a SQL expression to filter results (use Ctrl+Space)

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	1	SIMPLE	employees	[NULL]	const	PRIMARY	PRIMARY	4	const	1	100.00	[NULL]

Обращаем внимание:

- Type (Const / Index / All);
- Key / Possible_keys;
- Rows.

Подробный анализ инструмента:

https://habr.com/post/211022/

Оптимизация запросов

Профилирование запросов.

Команды:

SHOW PROFILE
SHOW PROFILES

Result

show profiles | Enter a SQL expression to filter results (use Ctrl+Space)

	Query_ID	Duration	Query
6	23	0.00012475	SHOW WARNINGS
7	24	0.000168	/* ApplicationName=DBeaver 5.1.2 - Main */ SET SQL_SELECT_LIMIT=DEFAULT
8	25	1.17810725	/* ApplicationName=DBeaver 5.1.2 - Main */ select * from employees ¶where CONCAT(f
9	26	0.000131	SHOW WARNINGS
10	27	0.000127	/* ApplicationName=DBeaver 5.1.2 - Main */ SET SQL_SELECT_LIMIT=200
11	28	0.00020975	SHOW WARNINGS

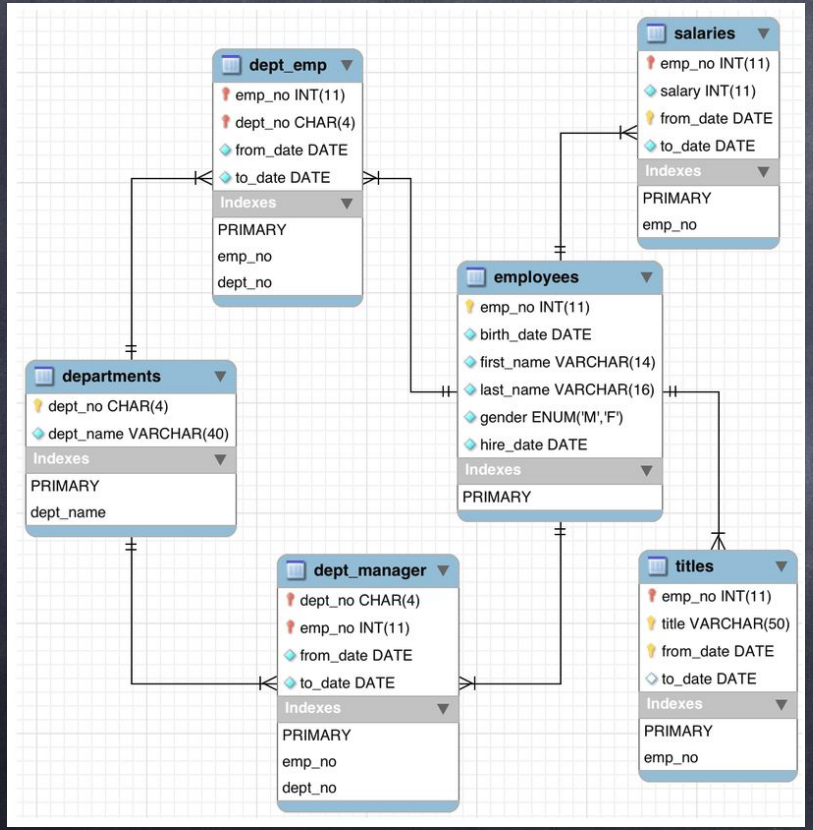
```
set profiling=1;
show profiles;
show profile for query 5;
```

Status	Duration
starting	0.000024
checking query cache for query	0.000054
Opening tables	0.000025
System lock	0.000004
Table lock	0.000036
init	0.000014
optimizing	0.000007
statistics	0.000015
preparing	0.000011
executing	0.000005
Sending data	0.000117
end	0.000008
query end	0.000003
freeing items	0.000031
storing result in query cache	0.000006
logging slow query	0.000002
cleaning up	0.000003

Подробный анализ инструмента:

https://goo.gl/UX

Примеры

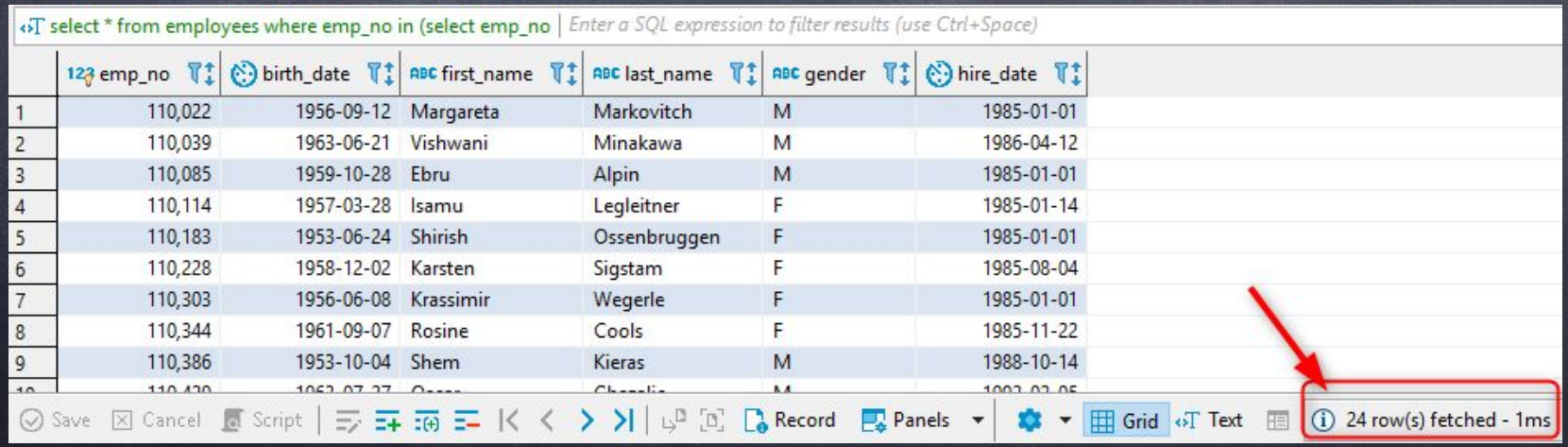


Давайте выберем работников, которые были когда либо менеджерами отделов.

Примеры

```
select * from employees
where emp_no in (select emp_no from dept_manager);
```

Конструкция с IN



Примеры

```
select * from employees as emp
    RIGHT JOIN dept_manager as man on emp.emp_no=man.emp_no;
```

Конструкция с JOIN

select * from employees as emp RIGHT JOIN dept_manage | Enter a SQL expression to filter results (use Ctrl+Space)

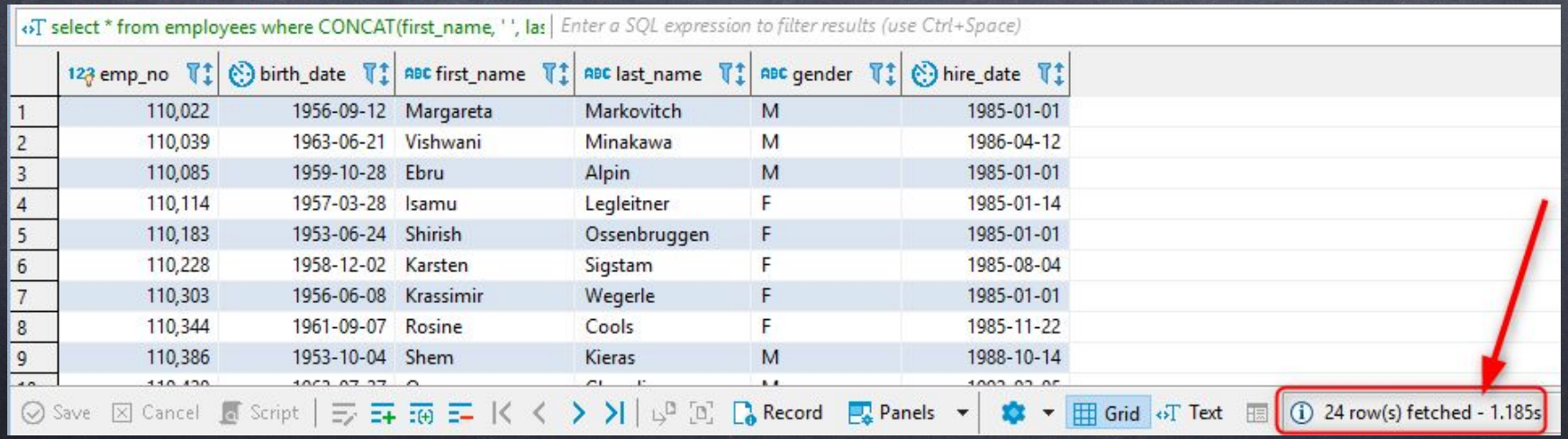
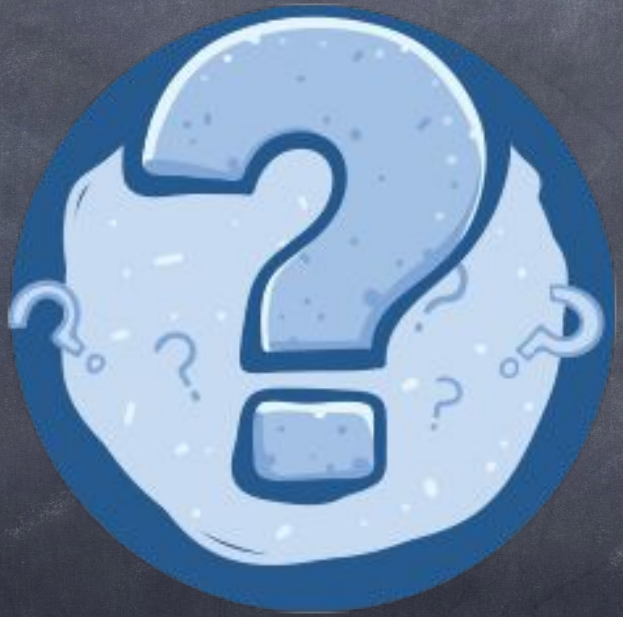
	emp_no	birth_date	first_name	last_name	gender	hire_date	emp_no	dept_no	from_date	to_date
1	110,022	1956-09-12	Margareta	Markovitch	M	1985-01-01	110,022	d001	1985-01-01	1991-10-01
2	110,039	1963-06-21	Vishwani	Minakawa	M	1986-04-12	110,039	d001	1991-10-01	9999-01-01
3	110,085	1959-10-28	Ebru	Alpin	M	1985-01-01	110,085	d002	1985-01-01	1989-12-17
4	110,114	1957-03-28	Isamu	Legleitner	F	1985-01-14	110,114	d002	1989-12-17	9999-01-01
5	110,183	1953-06-24	Shirish	Ossenbruggen	F	1985-01-01	110,183	d003	1985-01-01	1992-03-21
6	110,228	1958-12-02	Karsten	Sigstam	F	1985-08-04	110,228	d003	1992-03-21	9999-01-01
7	110,303	1956-06-08	Krassimir	Wegerle	F	1985-01-01	110,303	d004	1985-01-01	1988-09-09
8	110,344	1961-09-07	Rosine	Cools	F	1985-11-22	110,344	d004	1988-09-09	1992-08-02
9	110,386	1953-10-04	Shem	Kieras	M	1988-10-14	110,386	d004	1992-08-02	1996-08-30

Save Cancel Script Record Panels Grid Text 24 row(s) fetched - 2ms

Примеры

```
select * from employees
where CONCAT(first_name, ' ', last_name) in
    (select CONCAT(emp.first_name, ' ', emp.last_name) from dept_manager man
    JOIN employees emp on man.emp_no=emp.emp_no);
```

Конструкция с …



Вопросы