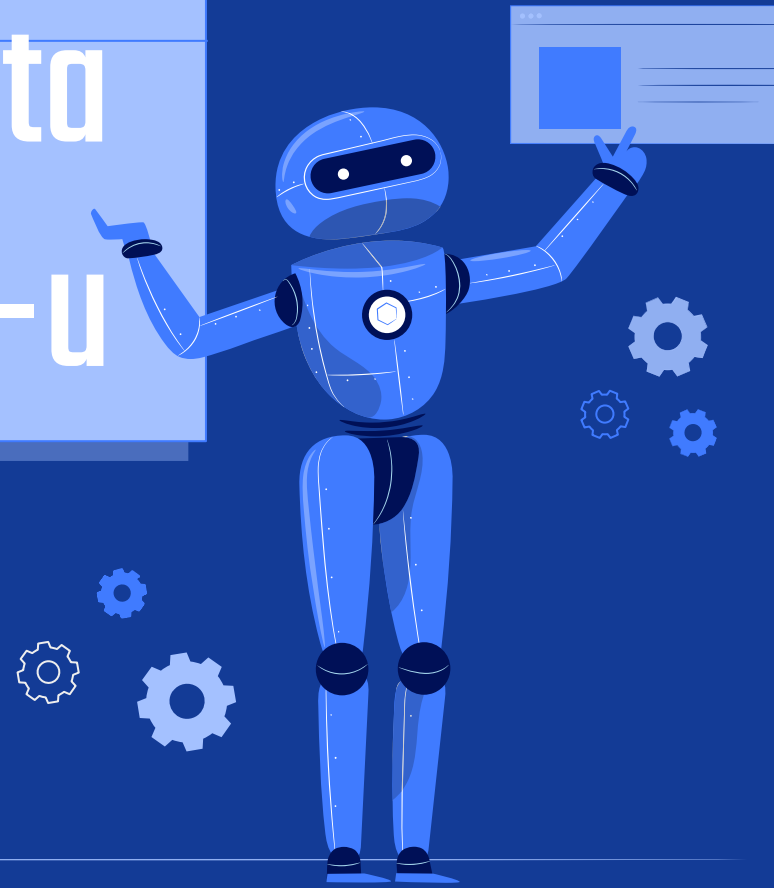
Optimizacija upita u MySQL-u

Student: Vladana Stojiljković, br.ind. 1135

Mentor: Doc. dr Aleksandar Stanimirović



Sadržaj

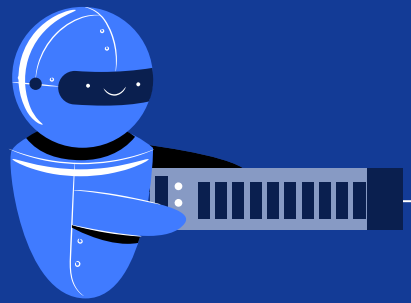
01 Uvod

02 MySQL optimizator

03 Pisanje optimizovanih upita

04 Zaključak

01 Uvod



01.

Šta je optimizacija upita?

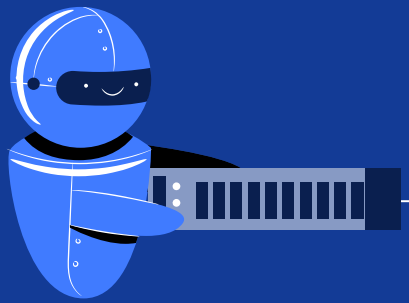
02.

Šta je plan izvršenja?

03.

Kako MySQL vrši optimizaciju upita?

02 MySQL optimizator



MySQL optimizator

- Skup rutina koje određuju optimalan plan izvršenja
- Trasa optimizatora i *EXPLAIN*
- Primarne i ostale optimizacije

```
handle_select()
  mysql_select()
    JOIN::prepare()
      setup_fields()
    JOIN::optimize()            /* optimizer is from here ... */
      optimize_cond()
      opt_sum_query()
      make_join_statistics()
        get_quick_record_count()
        choose_plan()
          /* Find the best way to access tables */
          /* as specified by the user.          */
          optimize_straight_join()
            best_access_path()
          /* Find a (sub-)optimal plan among all or subset */
          /* of all possible query plans where the user    */
          /* controls the exhaustiveness of the search.    */
          greedy_search()
            best_extension_by_limited_search()
              best_access_path()
          /* Perform an exhaustive search for an optimal plan */
          find_best()
      make_join_select()        /* ... to here */
    JOIN::exec()
```

Primarne optimizacije:

Konstantne relacije

01 Obrada tranzitivnosti

02 Uklanjanje „mrtvog “ koda

```
use sakila;
set optimizer_trace="enabled=on";
select * from payment where customer_id=staff_id and staff_id = 1;
select * from information_schema.optimizer_trace;
```

```
"steps": [
  {
    "condition_processing": {
      "condition": "WHERE",
      "original_condition": "((`payment`.`customer_id` = `payment`.`staff_id`) and (`payment`.`staff_id` = 1))",
```

```
use sakila;
set optimizer_trace="enabled=on";
select * from payment where 0=0 and customer_id=1;
select * from information_schema.optimizer_trace;
```

```
"condition_processing": {
  "condition": "WHERE",
  "original_condition": "((`payment`.`customer_id` = 1))",
```

Primarne optimizacije:

Konstantne relacije

03 Obrada IS NULL uslova

04 Konstantne tabele

```
use sakila;
create table test_null_cond (label char(1));
...
set optimizer_trace="enabled=on";
select * from test_null_cond where label="test";
select * from information_schema.optimizer_trace;
```

```
"condition_processing": {
  "condition": "WHERE",
  "original_condition": "(`test_null_cond`.`label` = 'test')",
```

```
use sakila;
set optimizer_trace="enabled=on";
explain select language.language_id, film.language_id from language, film
where language.language_id=film.language_id and language.language_id=1;
select * from information_schema.optimizer_trace;
```

```
"attaching_conditions_to_tables": {
  "original_condition": "(`film`.`language_id` = 1)",
  "attached_conditions_computation": [],
  "attached_conditions_summary": [
    {
      "table": "`film`",
      "attached": "(`film`.`language_id` = 1)"
    }
...
"rows_estimation": [
  {
    "table": "`language`",
    "rows": 1,
    "cost": 1,
    "table_type": "const",
    "empty": false
  },
```

```
explain select language.language_id, film.language_id from language, film
where language.language_id=film.language_id and language.language_id=100;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	no matching row in const table

Primarne optimizacije

Tipovi pristupa

01

Indeksi

02

Pretraga opsega (range)

```
use sakila;
set optimizer_trace="enabled=on";
select rental_duration from film where rental_duration>3;
```

```
"considered_access_paths": [
  {
    "rows_to_scan": 797,
    "access_type": "range",
    "range_details": {
      "used_index": "dur_index"
    },
    "resulting_rows": 797,
    "cost": 159.73,
    "chosen": true
```

```
use sakila;
set optimizer_trace="enabled=on";
select rental_duration from film where rental_duration>0;
```

```
"considered_access_paths": [
  {
    "rows_to_scan": 1000,
    "access_type": "scan",
    "resulting_rows": 1000,
    "cost": 103,
    "chosen": true
```

Primarne optimizacije:

Tipovi pristupa

03

Spajanje indeksa

```
1 • use sakila;
2 • set optimizer_trace="enabled=on";
3 • explain select * from film where language_id>1 or rental_duration>14;
4 • select * from information_schema.optimizer_trace optimizer_trace;
```


```
"chosen_range_access_summary": {
  "range_access_plan": {
    "type": "index_merge",
    "index_merge_of": [
      {
        "type": "range_scan",
        "index": "idx_fk_language_id",
        "rows": 1,
        "ranges": [
          "1 < language_id"
        ]
      },
      {
        "type": "range_scan",
        "index": "dur_index",
        "rows": 1,
        "ranges": [
          "14 < rental_duration"
        ]
      }
    ]
  },
```


```
1 • use sakila;
2 • set optimizer_trace="enabled=on";
3 • explain select * from film where language_id>0 and rental_duration>14;
4 • select * from information_schema.optimizer_trace optimizer_trace;
```


```
"range_scan_alternatives": [
  {
    "index": "idx_fk_language_id",
    "ranges": [
      "0 < language_id"
    ],
    "index_dives_for_eq_ranges": true,
    "rowid_ordered": false,
    "using_mrr": false,
    "index_only": false,
    "rows": 1000,
    "cost": 350.26,
    "chosen": false,
    "cause": "cost"
  },
  {
    "index": "dur_index",
    "ranges": [
      "14 < rental_duration"
    ],
    "index_dives_for_eq_ranges": true,
    "rowid_ordered": false,
    "using_mrr": false,
    "index_only": false,
    "rows": 1,
    "cost": 0.61,
    "chosen": true
  }
],
```

Primarne optimizacije

ORDER BY

Order by

- Indeksi
- filesort

```
1 • use sakila;
2 • set optimizer_trace="enabled=on";
3 • explain select rental_duration from film order by rental_duration;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	film	NULL	index	NULL	dur_index	1	NULL	1000	100.00	Using index

```
1 • use sakila;
2 • explain select rental_duration from film order by rental_duration+1;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	film	NULL	index	NULL	dur_index	1	NULL	1000	100.00	Using index; Using filesort

Primarne optimizacije:

GROUP BY

01 Privremene tabele i indeksi za grupisanje

02 DISTINCT u GROUP BY

```
1 • use sakila;
2 • set optimizer_trace="enabled=on";
3 • select distinct title from film;
4 • select * from information_schema.optimizer_trace;
```

```
{
  "optimizing_distinct_group_by_order_by": {
    "changed_distinct_to_group_by": true,
    "simplifying_group_by": {
      "original_clause": "`film`.`title`",
      "items": [
        {
          "item": "`film`.`title`"
        }
      ],
```

```
1 • use sakila;
2 • explain select rental_duration from film group by rental_duration;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	film	NULL	range	dur_index	dur_index	1	NULL	6	100.00	Using index for group-by

```
1 • use sakila;
2 • explain select rating from film group by rating;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	film	NULL	ALL	NULL	NULL	NULL	NULL	1000	100.00	Using temporary

Ostale optimizacije

- Rana i kasna obrada NULL vrednosti
- Particionisanje

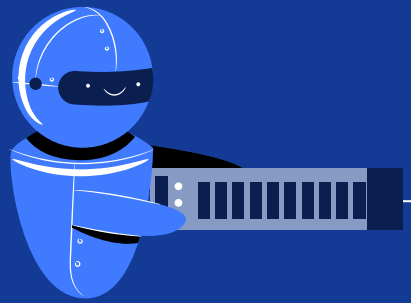
```
use sakila;
set optimizer_trace="enabled=on";
explain select staff_id, store.address_id from staff, store where staff_id=manager_staff_id;
select * from information_schema.optimizer_trace;
```

```
"ref_optimizer_key_uses": [
  {
    "table": "`staff`",
    "field": "staff_id",
    "equals": "`store`.`manager_staff_id`",
    "null_rejecting": true
  },
  {
    "table": "`store`",
    "field": "manager_staff_id",
    "equals": "`staff`.`staff_id`",
    "null_rejecting": true
  }
]
```

```
use sakila;
set optimizer_trace="enabled=on";
select * from film where film_id=5/0;
select * from information_schema.optimizer_trace;
```

```
"ref_optimizer_key_uses": [
  {
    "table": "`film`",
    "field": "film_id",
    "equals": "(5 / 0)",
    "null_rejecting": true
  }
]
```

03 Pisanje optimizovanih upita



SELECT naredbe

- Upotreba indeksa i potiskivanje indeksa
- Nametanje korišćenja indeksa optimizatoru

```
use sakila;
explain select address_id from store  where manager_staff_id<4 and address_id=4;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	store	NULL	ref	idx_unique_manager,idx_fk_address_id	idx_fk_address_id	2	const	1	100.00	Using where

```
use sakila;
explain select address_id from store use index (idx_unique_manager) where manager_staff_id<4 and address_id=4;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	store	NULL	range	idx_unique_manager	idx_unique_manager	1	NULL	2	50.00	Using index condition; Using where

```
use sakila;
explain select address_id from store force index (idx_unique_manager) where manager_staff_id<4 and address_id=4;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	store	NULL	range	idx_unique_manager	idx_unique_manager	1	NULL	2	50.00	Using index condition; Using where

```
use sakila;
explain select address_id from store ignore index (idx_unique_manager) where manager_staff_id<4 and address_id=4;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	store	NULL	ref	idx_fk_address_id	idx_fk_address_id	2	const	1	50.00	Using where

SELECT naredbe-range scan

- Džoker karakteri kod LIKE operatora
- Row constructor expressions

```
use sakila;
explain select title from film where title like "c%";
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	film	NULL	range	idx_title	idx_title	514	NULL	92	100.00	Using where; Using index

```
use sakila;
explain select title from film where title like "%c";
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	film	NULL	index	NULL	idx_title	514	NULL	1000	11.11	Using where; Using index

```
use sakila;
alter table customer add index row_const_ind (customer_id, store_id, address_id);
analyze table customer;
explain select customer_id, store_id from customer where (customer_id, store_id, address_id) in ((1, 1, 1) , (2, 2, 1));
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	customer	NULL	range	PRIMARY,idx_fk_store_id,idx_fk_address_id,ro...	row_const_ind	5	NULL	2	100.00	Using where; Using index

```
use sakila;
explain select customer_id, store_id from customer where (address_id, store_id) in ((1, 1) , (2, 2));
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	customer	NULL	range	idx_fk_store_id,idx_fk_address_id	idx_fk_address_id	2	NULL	2	100.00	Using where

SELECT naredbe-range scan

Skip scan index

```
1  use sakila;
2  CREATE TABLE test_skip_scan (col1 INT NOT NULL, col2 INT NOT NULL, PRIMARY KEY(col1, col2));
3  INSERT INTO test_skip_scan VALUES
4    (1,1), (1,2), (1,3), (1,4), (1,5),
5    (2,1), (2,2), (2,3), (2,4), (2,5);
6  INSERT INTO test_skip_scan SELECT col1, col2 + 5 FROM test_skip_scan;
7  INSERT INTO test_skip_scan SELECT col1, col2 + 10 FROM test_skip_scan;
8  INSERT INTO test_skip_scan SELECT col1, col2 + 20 FROM test_skip_scan;
9  INSERT INTO test_skip_scan SELECT col1, col2 + 40 FROM test_skip_scan;
10 ANALYZE TABLE test_skip_scan;
11 EXPLAIN SELECT col1, col2 FROM test_skip_scan WHERE col2 > 40;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	test_skip_scan	NULL	range	PRIMARY	PRIMARY	8	NULL	53	100.00	Using where; Using index for skip scan

SELECT naredbe – spoljašnji spoj

01 Nemogući (NULL rejected) uslovi

02 Transformacija u unutrašnji spoj

```
1  use sakila;
2  set optimizer_trace="enabled=on";
3  explain select film.film_id, category_id from film_category left join film on film_category.film_id=film.film_id
4  where language_id=1 or 0=1;
5  select * from information_schema.optimizer_trace;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	film	NULL	ref	PRIMARY,idx_fk_language_id	idx_fk_language_id	1	const	1000	100.00	Using where; Using index
1	SIMPLE	film_category	NULL	ref	PRIMARY	PRIMARY	2	sakila.film.film_id	1	100.00	Using index


```
{
  "transformations_to_nested_joins": {
    "transformations": [
      "outer_join_to_inner_join",
      "JOIN_condition_to_WHERE",
      "parenthesis_removal"
    ],
```

SELECT naredbe – ORDER BY

01 Indeksi i primarni ključ

```
use sakila;
set optimizer_trace="enabled=on";
alter table inventory add index ind_film_store (film_id, store_id);
explain select inventory_id, film_id, store_id from inventory order by film_id, store_id
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	inventory	NULL	index	NULL	ind_film_store	3	NULL	4581	100.00	Using index

02 Više kolona u klauzuli

```
use sakila;
set optimizer_trace="enabled=on";
CREATE TABLE test_order_dir (
  col1 INT, col2 INT,
  INDEX idx1 (col1 ASC, col2 DESC)
);

set optimizer_trace="enabled=on";
explain select * from test_order_dir order by col1 asc, col2 desc;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	test_order_dir	NULL	index	NULL	idx1	10	NULL	5	100.00	Using index

```
explain select * from test_order_dir order by col1 desc, col2 asc;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	test_order_dir	NULL	index	NULL	idx1	10	NULL	5	100.00	Backward index scan; Using index

```
explain select * from test_order_dir order by col1 desc, col2 desc;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	test_order_dir	NULL	index	NULL	idx1	10	NULL	5	100.00	Using index; Using filesort

SELECT naredbe – ORDER BY

03

Korišćenje alijasa

```
1 • use sakila;
2 • explain select country_id from country order by country_id;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	country	NULL	index	NULL	PRIMARY	2	NULL	109	100.00	Using index

```
1 • use sakila;
2 • explain select country_id+1 as country_id from country order by country_id;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	country	NULL	index	NULL	PRIMARY	2	NULL	109	100.00	Using index; Using filesort

SELECT naredbe – GROUP BY

01 Slobodan indeks (loose scan index)

02 Čvrst indeks (tight scan index)

```
1 • use sakila;
2 • set optimizer_trace="enabled=on";
3 • alter table test_loose add index all_ind(col1, col2, col3);
4 • explain select * from test_loose where col2=1 group by col1, col3;
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	test_loose	NULL	index	all_ind	all_ind	15	NULL	5	20.00	Using where; Using index

```
1 • use sakila;
2 • set optimizer_trace="enabled=on";
3 • alter table test_loose add index col1(col1, col2);
4 • explain select col1, max(col2) from test_loose group by col1;
5 • select * from information_schema.optimizer_trace;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	test_loose	NULL	range	col1	col1	5	NULL	2	100.00	Using index for group-by

```
1 • use sakila;
2 • set optimizer_trace="enabled=on";
3 • explain select min(col1), col2 from test_loose group by col2;
4 • select * from information_schema.optimizer_trace;
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	test_loose	NULL	index	col1	col1	10	NULL	5	100.00	Using index; Using temporary

SELECT naredbe – DISTINCT

01

Slično kao kog GROUP BY

02

Odbacivanje tabela koje se ne koriste kod upita s više tabela

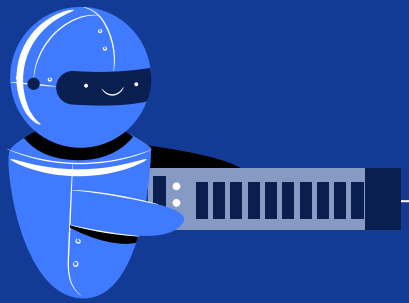
```
1 • use sakila;
2 • set optimizer_trace="enabled=on";
3 • explain select distinct title from film, film_category where film.film_id=film_category.film_id;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	film_category	NULL	index	PRIMARY	fk_film_category_category	1	NULL	1000	100.00	Using index; Using tempo
1	SIMPLE	film	NULL	eq_ref	PRIMARY,idx_title	PRIMARY	2	sakila.film_category.film_id	1	100.00	NULL

INSERT, UPDATE i DELETE naredbe

- Učitavanje iz fajla (LOAD DATA)
- INSERT s više vrsti vrednosti
- Postavljanje veličine bulk bafera
- Odlaganje ažuriranja
- TRUNCATE umesto DELETE FROM

03 Zaključak



Zaključak

01 Optimizacija upita poboljšava performanse aplikacije.

02 MySQL optimizator vrši transformacije na osnovu kojih automatski optimizuje upite.

03 Na osnovu trase optimizatora i EXPLAIN izlaza, korisnik može da vrši optimizaciju pisanjem upitea koji imaju optimalne planove izvršenja.

Hvala na pažnji!