



可读的EXPLAIN JSON

@大自在真人





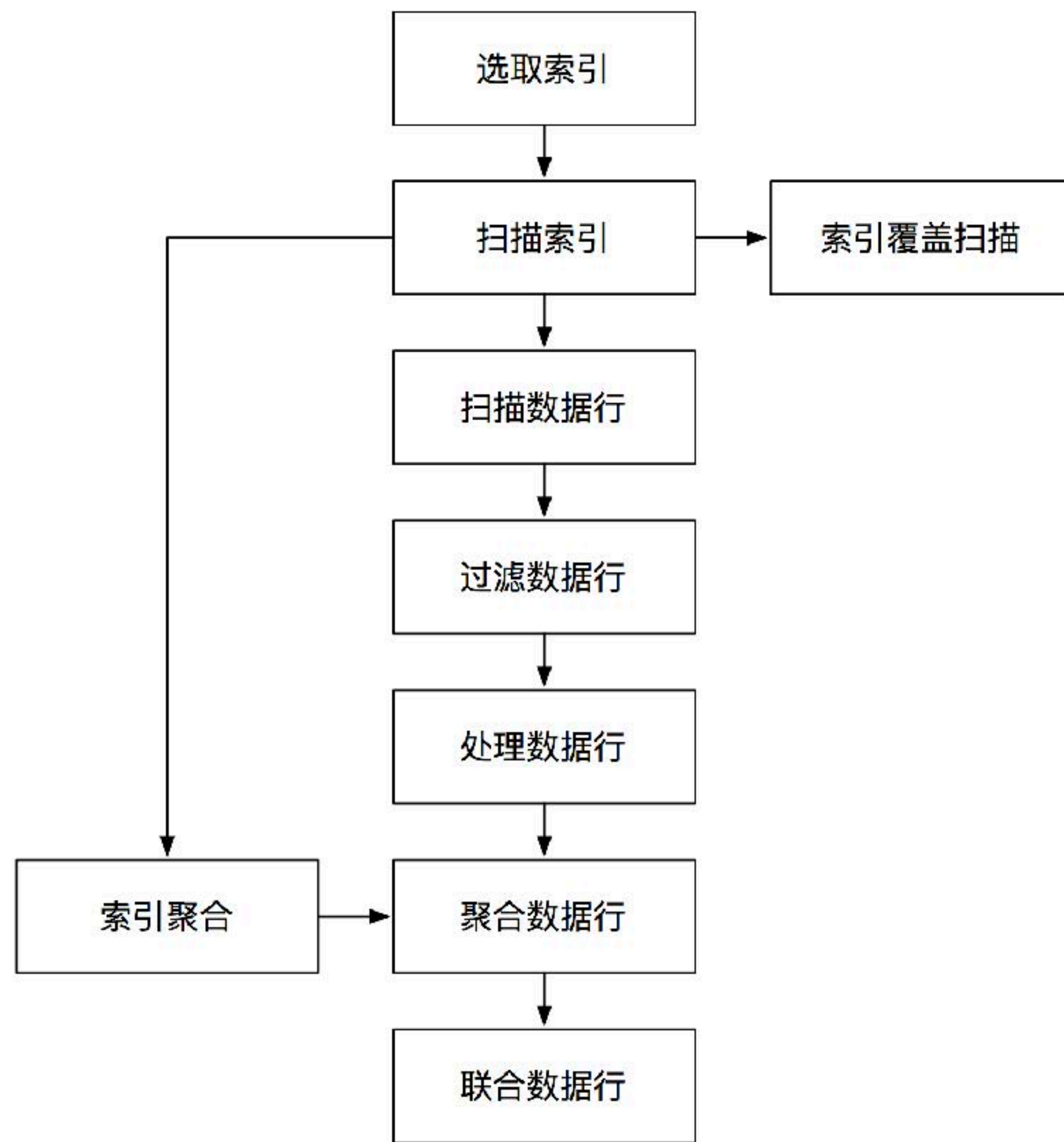
DBA&DEVELOPER
刘伟@云和开创



EXPLAIN

EXPLAIN是什么

- SQL解析
- SQL执行
- 单线程 (for MySQL)
- 通常顺序



经典
E
X
P
L
A
I
N

```
mysql> explain SELECT CONCAT(customer.last_name, ' ', customer.first_name) AS customer,
-> address.phone, film.title
-> FROM rental INNER JOIN customer ON rental.customer_id = customer.customer_id
-> INNER JOIN address ON customer.address_id = address.address_id
-> INNER JOIN inventory ON rental.inventory_id = inventory.inventory_id
-> INNER JOIN film ON inventory.film_id = film.film_id
-> WHERE rental.return_date IS NULL
-> AND rental_date + INTERVAL film.rental_duration DAY < CURRENT_DATE()
-> LIMIT 5;
```

id	select_type	table	partitions	type	possible_keys	key
1	SIMPLE	rental	NULL	ALL	idx_fk_inventory_id,idx_fk_customer_id	NULL
			16008	10.00	Using where	
1	SIMPLE	customer	NULL	eq_ref	PRIMARY,idx_fk_address_id	PRIMARY
2		sakila.rental.customer_id	1	100.00	NULL	
1	SIMPLE	address	NULL	eq_ref	PRIMARY	PRIMARY
2		sakila.customer.address_id	1	100.00	NULL	
1	SIMPLE	inventory	NULL	eq_ref	PRIMARY,idx_fk_film_id	PRIMARY
3		sakila.rental.inventory_id	1	100.00	NULL	
1	SIMPLE	film	NULL	eq_ref	PRIMARY	PRIMARY
2		sakila.inventory.film_id	1	100.00	Using where	

EXPLAIN

```
{
  "query_block": {
    "select_id": 1,
    "cost_info": {
      "query_cost": "10982.44"
    },
    "nested_loop": [
      {
        "table": {
          "table_name": "rental",
          "access_type": "ALL",
          "possible_keys": [
            "idx_fk_inventory_id",
            "idx_fk_customer_id"
          ],
          "rows_examined_per_scan": 16008,
          "rows_produced_per_join": 1600,
          "filtered": "10.00",
          "cost_info": {
            "read_cost": "2978.44",
            "eval_cost": "320.16",
            "prefix_cost": "3298.60",
            "data_read_per_join": "50K"
          },
          "used_columns": [
            "rental_date",
            "inventory_id",
            "customer_id",
            "return_date"
          ],
          "attached_condition": "isnull(`sakila`.`rental`.`return"
        }
      },
      {
        "table": {
          "table_name": "customer",
          "access_type": "eq_ref",
```

```
      "access_type": "eq_ref",
      "possible_keys": [
        "PRIMARY",
        "idx_fk_address_id"
      ],
      "key": "PRIMARY",
      "used_key_parts": [
        "customer_id"
      ],
      "key_length": "2",
      "ref": [
        "sakila.rental.customer_id"
      ],
      "rows_examined_per_scan": 1,
      "rows_produced_per_join": 1600,
      "filtered": "100.00",
      "cost_info": {
        "read_cost": "1600.80",
        "eval_cost": "320.16",
        "prefix_cost": "5219.56",
        "data_read_per_join": "687K"
      },
      "used_columns": [
        "customer_id",
        "first_name",
        "last_name",
        "address_id"
      ]
    }
  },
  {
    "table": {
      "table_name": "address",
      "access_type": "eq_ref",
      "possible_keys": [
        "PRIMARY"
      ],
```

Operation	Access Type	Read Cost	Eval Cost	Sort Cost	Prefix Cost	Query Cost	Data_read_per_join	Attached Condition
[-] query_block						10982.44		
[-] nested_loop								
table: [rental]	ALL	2978.44	320.16		3298.60		50K	isnull(`sakila`.`rental`.`return_...
[-] table: [customer] ref: [sakila.rental.customer_id]	eq_ref	1600.80	320.16		5219.56		687K	
key: [PRIMARY] used_key_parts: [customer_id]								
[-] table: [address] ref: [sakila.customer.address_id]	eq_ref	1600.80	320.16		7140.52		737K	
key: [PRIMARY] used_key_parts: [address_id]								
[-] table: [inventory] ref: [sakila.rental.inventory_id]	eq_ref	1600.80	320.16		9061.48		25K	
key: [PRIMARY] used_key_parts: [inventory_id]								
[-] table: [film] ref: [sakila.inventory.film_id]	eq_ref	1600.80	320.16		10982.44		1M	((`sakila`.`rental`.`rental_date`...
key: [PRIMARY] used_key_parts: [film_id]								

EXPLAIN JSON 可视化

@zCloud of MySQL

EXPLAIN的新发展

- **MySQL 5.6**
 - json format
 - 层级嵌套
 - Explain dml
- **MySQL 5.7**
 - explain connection
 - query cost

The image features a dark blue, textured background that resembles a fine mesh or fabric. A prominent white horizontal band runs across the center of the image. The text '预备的知识' is centered within this white band.

预备的知识

需要关注的字段

- **access type** 对表的访问方式
 - 直接决定查询效率
- **key** 所使用的索引
 - 决定数据分辨率
- **extra / message** 没地方放的东西
- **cost** 查询代价
 - only in json

ACCESS TYPE

- **good**

- system
- const
- Eq_ref
- Index
- Range
- ref
- unique_subquery

- **bad**

- index_merge
- ALL
- Fulltext

过滤计算

- **good**
 - Using index
 - Using index for group by
 - Using index condition
 - Using join buffer

- **bad**
 - index_merge
 - ALL
 - Fulltext

COST模型

- **disktemptable_create_cost 40**
- **disktemptable_row_cost 1**
- **key_compare_cost 0.1**
- **memorytemptable_create_cost 2**
- **memorytemptable_row_cost 0.2**
- **row_evaluate_cost 0.2**
- **io_block_read_cost 1**
- **memory_block_read_cost 1**
- **read cost 读取消耗**
- **eval cost 计算消耗**
- **prefix cost 累计消耗**
- **sort cost 排序消耗**
- **query cost 总查询消耗**



EXPLAIN 实例

SQL

```
CREATE TABLE `metric` (
```

```
  `id`  
  int(11) NOT NULL AUTO_INCREMENT,
```

```
  `target_id`  
  int(11) DEFAULT NULL,
```

```
  `target_type`  
  varchar(45) DEFAULT NULL,
```

```
  `name`  
  varchar(45) DEFAULT NULL,
```

```
  `ctime`  
  varchar(45) DEFAULT NULL,
```

```
  `description`  
  varchar(45) DEFAULT NULL,
```

```
  PRIMARY KEY(`id`),
```

```
  UNIQUE KEY `unq_target_type_id_name` (`  
target_type`, `target_id`, `name`)
```

```
) ENGINE = InnoDB
```

```
CREATE TABLE `metric_value` ( `metric`  
  int(11) NOT NULL,  
  `value` float DEFAULT NULL,  
  `ctime` datetime NOT NULL,  
  PRIMARY KEY (`metric`,  
  `ctime`) ) ENGINE=InnoDB DEF  
AULT CHARSET=utf8
```

S Q L

```
SELECT target_id,  
       name,  
       metric_value.ctime,  
       value  
FROM metric,metric_value  
WHERE metric.id=metric_value.metric  
      AND target_type='instance'  
      AND target_id IN (83,84,85)  
      AND name IN ('Com_select','Com_delete')  
      AND metric_value.ctime  
      BETWEEN '2017-11-13 00:00:00'  
      AND '2017-11-14 00:00:00'  
ORDER BY target_id,name,ctime
```



```
select target_id,name,metric_value.ctime,value from metric,metric_value where
metric.id=metric_value.metric and target_type='instance' and target_id in (83,84,85) and name in
('Co
'201
```

```
select target_id,name,metric_value.ctime,value from metric,metric_value where
metric.id=metric_value.metric and target_type='instance' and target_id in (83,84,85) and
name in ('Com_select','Com_delete') and metric_value.ctime between '2017-11-13 00:00:00' and
```



执行计划图

Operation	Access Type	Read Cost	Eval Cost	Sort Cost	Prefix Cost	Query Cost
query_block						654.82
ordering_operation: using_filesort: [true]						
nested_loop						
table: [metric]	range	2.38	1.20		3.58	
key: [uniq_target_type_id_name] used_key_parts: [target_type] used_key_parts: [target_id] used_key_parts: [name]						
table: [metric_value] ref: [mysql_paas.metric.id]	ref	9.56	71.29		654.82	
key: [PRIMARY] used_key_parts: [metric]						

EXPLAIN

@zCloud of MySQL

SQL

```
CREATE TABLE `dept_manager` (  
  `emp_no` int(11) NOT NULL,  
  `dept_no` char(4) NOT NULL,  
  `from_date` date NOT NULL,  
  `to_date` date NOT NULL,  
  PRIMARY KEY (`emp_no`, `dept_no`),  
  KEY `dept_no` (`dept_no`)  
) ENGINE = InnoDB
```

Operation	Access Type	Read Cost	Eval Cost	Sort Cost	Prefix Cost	Query Cost	Data_read_per_join
[-] query_block						5.80	
[-] table: [dept_manager]	Index	1.00	4.80		5.80		384
key: [dept_no] used_key_parts: [dept_no]							

**SELECT EMP_NO FROM
DEPT_MANAGER;**

@zCloud of MySQL

Operation	Access Type	Read Cost	Eval Cost	Sort Cost	Prefix Cost	Query Cost	Data_read_per_join	At
[-] query_block						29.80		
[-] table: [dept_manager]	Index	25.00	4.80		29.80		384	
key: [PRIMARY] used_key_parts: [emp_no] used_key_parts: [dept_no]								

```
SELECT EMP_NO FROM
DEPT_MANAGER FORCE INDEX(PRIMARY);
```

@zCloud of MySQL

SQL

```
SELECT dept_manager.emp_no, COUNT(1)
FROM employees, dept_emp, dept_manager
WHERE employees.emp_no = dept_emp.emp_no
      AND dept_emp.dept_no = dept_manager.dept_no
GROUP BY dept_manager.emp_no;
```

```
SELECT dept_manager.emp_no, COUNT(1)
FROM employees
      LEFT JOIN dept_emp ON employees.emp_no = dept_emp.emp_no
      LEFT JOIN dept_manager ON dept_emp.dept_no = dept_manager.dept_no
GROUP BY dept_manager.emp_no;
```

Operation	Access Type	Read Cost	Eval Cost	Sort Cost	Prefix Cost	Query Cost	Data_read_per_join	Attached Condition
[-] query_block						1240625.98		
[-] grouping_operation: using_filesort: [true]				590288.02				
[-] nested_loop								
[-] table: [dept_emp]	Index	481.00	44271.60		44752.60		3M	
key: [dept_no] used_key_parts: [dept_no]								
[-] table: [employees] ref: [employees.dept_emp.emp_no]	eq_ref	221358.00	44271.60		310382.20		10M	
key: [PRIMARY] used_key_parts: [emp_no]								
[-] table: [dept_manager] ref: [employees.dept_emp.dept_no]	ref	221898.16	118057.60		650337.96		9M	
key: [dept_no] used_key_parts: [dept_no]								

执行计划图

@zCloud for MySQL

Operation	Access Type	Read Cost	Eval Cost	Sort Cost	Prefix Cost	Query Cost	Data_read_per_join	Attached Condition
[-] query_block						1291291.77		
[-] grouping_operation: using_filesort: [true]				627090.94				
[-] nested_loop								
[-] table: [employees]	Index	673.00	42550.20		43223.20		9M	
key: [PRIMARY] used_key_parts: [emp_no]								
[-] table: [dept_emp] ref: [employees.employees.emp_no]	ref	212794.68	47031.82		303049.70		3M	
key: [PRIMARY] used_key_parts: [emp_no]								
[-] table: [dept_manager] ref: [employees.dept_emp.dept_no]	ref	235732.94	125418.19		664200.82		9M	
key: [dept_no] used_key_parts: [dept_no]								

执行计划图

@zCloud for MySQL



EXPLAIN IN ZCLOUD FOR MYSQL

SLOW LOG

➤ 定时分析

- flush slow logs
- pt query digest json

➤ 预切割

- 每五分钟slow log
- 按照sql执行时间切分

➤ 配图路径

- 复制slow log
- 分解slowlog
- pt分析
- 结果入库

- 排序规则 三
- Query_time_sum
- SQL文本
- ☆ 0x442D62978038B...
 - ★ 0xB08E1BB5E4AB0...
 - ☆ 0x5A70326E0CDB...
 - ★ 0xF2F246ABFFC51...
 - ★ 0x79DB25F9615F3...
 - ★ 0x5585C53B621FE...
 - ★ 0x78BF16FCC4F1A...
 - ☆ 0x68BF22EF3C790...
 - ☆ 0x3E396E02FB72E...
 - ☆ 0x285C261A17C56...
 - ☆ 0xA587088921068...
 - ☆ 0x567EBEF2B0F3D...
 - ☆ 0x1CBC03E301F38...
 - ☆ 131231
 - ☆ 0x9E67EA46E9C6F...
 - ☆ 0xC3D251A3F747D...
 - ☆ 0x2DA41889F0B01...
 - ☆ 0x813031B8BBC3B...
 - ☆ 0x9D0CB14565C99...
 - ☆ 0xAD392CB9F4E0...
 - ☆ 0xCD832198D8FDA...
 - ☆ 0x8B184BF485068...
 - ☆ 0x2C24703F79B29...

0xF2F246ABFFC51781 优化建议 备注设置

sql_id: 0xF2F246ABFFC51781

db: productdb

table: t_product_check_task_records,t_product

user: productdb

```

SELECT p.*,
       r.id record_id,
       r.task_id,
       r.apply_date,
       r.task_type,
       r.remark
FROM t_product_check_task_records r
LEFT JOIN t_product p
     ON r.product_id= p.product_id

```

更多



SLOW LOG 主页面

@zCloud for MySQL

SLOW LOG METRIC

- **rows send**
- **rows examind**
- **query time**
- **lock time**
- **sql count**

- **Sun**
- **Min**
- **Max**
- **pct 95**
- **Stddev**
- **Median**

- sql_count
- Query_time_sum
- Query_time_min
- Query_time_max
- Query_time_pct_95
- Query_time_stddev
- Query_time_median
- Lock_time_sum
- Lock_time_min
- Lock_time_max
- Lock_time_pct_95
- Lock_time_stddev
- Lock_time_median
- Rows_sent_sum
- Rows_sent_min
- Rows_sent_max
- Rows_sent_pct_95
- Rows_sent_stddev
- Rows_sent_median
- Rows_examined_...
- Rows_sent_median
- Rows_examined_...
- Rows_examined_...
- Rows_examined_...
- Rows_examined_...
- Rows_examined_...

Id	Host	Command	State	Time	User	db	Info	操作
93114	123.114.41.123:50058	Sleep	null	1909	root	sakila	null	kill killquery explain
94512	123.114.41.123:51259	Query	User sleep	4	root	sakila	select * from film where 1=sleep(1100)	kill killquery explain
94535	47.92.71.145:43038	Query	starting	null	root	Information_schema	show full processlist	kill killquery explain

执行计划图
✕

Operation	Access Type	Read Cost	Eval Cost	Sort Cost	Prefix Cost	Query Cost	Data_read_per_join	Attached Condition
query_block						212.00		
table: [film]	ALL	12.00	200.00		212.00		781K	

- Processlist
- Explain connection
- Kill query
- kill connection

PROCESSLIST

.....
@zCloud for MySQL



以后会做的事情

OPTIMIZER-TRACE

- **Optimizer-trace**
- **optimizer-trace-features**
 - greedy_search
 - range_optimizer
 - dynamic_range
 - repeated_subselect
- **优化器特性控制**

SQL优化建议

- **Hit建议**
- **索引建议**
 - 不可见索引
 - 虚拟索引
- **代价计算**
 - 更直观的代价模型
- **SQL改写建议**

MYSQL 8.0

- **新hit**
 - MERGE/INDEX_MERGE/
JOIN_ORDER/SET_VAR
- **列数据模型**
 - 直方图
- **不可见索引**
 - 优化调试
- **更优化的cost模型**



Q&A



谢谢观看

@大自在真人

