知数堂公开课

优质、可靠的在线培训品牌

QQ群：579036588

官网：http://zhishutang.com

微信公众号：知数堂（izhishuedu）

概要

1. group by 不全，随着执行计划不同，导致结果不同。

2.left join 不满足，多对一 导致 结果出错。

3.根据MySQL 执行计划，判断为myisam 数据库，并添加索引优化。

1. group by 不全 随着执行计划不同导致结果不同

```
root@mysql3306.sock>[employees]>desc t_order2;
+-----------+---------+------+-----+---------+-------+
| Field     | Type    | Null | Key | Default | Extra |
+-----------+---------+------+-----+---------+-------+
| emp_no    | int(11) | YES  |     | NULL    |       |
| dept_no   | char(4) | YES  | MUL | NULL    |       |
| from_date | date    | YES  |     | NULL    |       |
| to_date   | date    | YES  |     | NULL    |       |
+-----------+---------+------+-----+---------+-------+
4 rows in set (0.00 sec)

root@mysql3306.sock>[employees]>show index from t_order2;
+---------+------------+-------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| Table   | Non_unique | Key_name    | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment |
+---------+------------+-------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| t_order2 |          1 | ix_torder_2 |            1 | dept_no     | A         |           6 |     NULL | NULL   | YES  | BTREE      |         |               |
| t_order2 |          1 | ix_torder_3 |            1 | dept_no     | A         |           6 |     NULL | NULL   | YES  | BTREE      |         |               |
| t_order2 |          1 | ix_torder_3 |            2 | emp_no      | A         |          10 |     NULL | NULL   | YES  | BTREE      |         |               |
+---------+------------+-------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+

root@mysql3306.sock>[employees]>select * from t_order2;
+--------+---------+------------+------------+
| emp_no | dept_no | from_date  | to_date    |
+--------+---------+------------+------------+
|  22744 | d006    | 1986-12-01 | 9999-01-01 |
|  54007 | d005    | 1986-12-01 | 9999-01-01 |
|  30970 | d005    | 1986-12-01 | 2017-03-29 |
|  31112 | d002    | 1986-12-01 | 1993-12-10 |
|  40983 | d005    | 1986-12-01 | 9999-01-01 |
|   NULL | d008    | 1986-12-01 | 1992-05-27 |
|  48317 | d008    | 1986-12-01 | 1989-01-11 |
|  49667 | d007    | 1986-12-01 | 9999-01-01 |
|  50449 | d005    | 1986-12-01 | 9999-01-01 |
|  10004 | d004    | 1986-12-01 | 9999-01-01 |
+--------+---------+------------+------------+
```

1. group by 不全 随着执行计划不同导致结果不同

如下所示 下面的SQL 是不完全group by 使用了索引ix_torder_2索引

```
root@mysql3306.sock>[employees]>desc select t.* ,count(1) from   t_order2 t group by t.dept_no ;
+----+-------------+-------+------------+-------+-------------------------+-------------+---------+------+------+----------+-------+
| id | select_type | table | partitions | type  | possible_keys           | key         | key_len | ref  | rows | filtered | Extra |
+----+-------------+-------+------------+-------+-------------------------+-------------+---------+------+------+----------+-------+
|  1 | SIMPLE      | t     | NULL       | index | ix_torder_2,ix_torder_3 | ix_torder_2 | 13      | NULL |   10 |   100.00 | NULL  |
+----+-------------+-------+------------+-------+-------------------------+-------------+---------+------+------+----------+-------+
1 row in set, 1 warning (0.00 sec)

root@mysql3306.sock>[employees]>select t.* ,count(1) from   t_order2 t group by t.dept_no ;
+--------+---------+------------+------------+----------+
| emp_no | dept_no | from_date  | to_date    | count(1) |
+--------+---------+------------+------------+----------+
|  31112 | d002    | 1986-12-01 | 1993-12-10 |        1 |
|  10004 | d004    | 1986-12-01 | 9999-01-01 |        1 |
|  54007 | d005    | 1986-12-01 | 9999-01-01 |        4 |
|  22744 | d006    | 1986-12-01 | 9999-01-01 |        1 |
|  49667 | d007    | 1986-12-01 | 9999-01-01 |        1 |
|   NULL | d008    | 1986-12-01 | 1992-05-27 |        2 |
+--------+---------+------------+------------+----------+
```

1. group by 不全 随着执行计划不同导致结果不同

如下所示 下面的SQL 是不完全group by 使用了索引ix_torder_3索引

```
root@mysql3306.sock>[employees]>desc select t.* ,count(1) from   t_order2 t force index(ix_torder_3) group by t.dept_no ;
+----+-------------+-------+------------+-------+---------------------------+-------------+---------+------+------+----------+-------+
| id | select_type | table | partitions | type  | possible_keys             | key         | key_len | ref  | rows | filtered | Extra |
+----+-------------+-------+------------+-------+---------------------------+-------------+---------+------+------+----------+-------+
|  1 | SIMPLE      | t     | NULL       | index | ix_torder_2,ix_torder_3 | ix_torder_3 | 18      | NULL |   10 |   100.00 | NULL  |
+----+-------------+-------+------------+-------+---------------------------+-------------+---------+------+------+----------+-------+
1 row in set, 1 warning (0.00 sec)

root@mysql3306.sock>[employees]>select t.* ,count(1) from   t_order2 t force index(ix_torder_3) group by t.dept_no ;
+--------+---------+------------+------------+----------+
| emp_no | dept_no | from_date  | to_date    | count(1) |
+--------+---------+------------+------------+----------+
|  31112 | d002    | 1986-12-01 | 1993-12-10 |        1 |
|  10004 | d004    | 1986-12-01 | 9999-01-01 |        1 |
|  30970 | d005    | 1986-12-01 | 2017-03-29 |        4 |
|  22744 | d006    | 1986-12-01 | 9999-01-01 |        1 |
|  49667 | d007    | 1986-12-01 | 9999-01-01 |        1 |
|   NULL | d008    | 1986-12-01 | 1992-05-27 |        2 |
+--------+---------+------------+------------+----------+
```

延伸 group by

我们用ignore index 变成全表扫描
从 extra 部分可以看到 using filesort 说明group by 默认含有排序

```
root@mysql3306.sock>[employees]>desc select t.* ,count(1) from   t_order2 t ignore index(ix_torder_3) ignore index(ix_torder_2) group by t.dept_no ;
+----+-------------+-------+------------+------+-------------------------+------+---------+------+------+----------+---------------------------------+
| id | select_type | table | partitions | type | possible_keys           | key  | key_len | ref  | rows | filtered | Extra                           |
+----+-------------+-------+------------+------+-------------------------+------+---------+------+------+----------+---------------------------------+
|  1 | SIMPLE      | t     | NULL       | ALL  | ix_torder_2,ix_torder_3 | NULL | NULL    | NULL |   10 |   100.00 | Using temporary; Using filesort |
+----+-------------+-------+------------+------+-------------------------+------+---------+------+------+----------+---------------------------------+
1 row in set, 1 warning (0.00 sec)

root@mysql3306.sock>[employees]>select t.* ,count(1) from   t_order2 t ignore index(ix_torder_3) ignore index(ix_torder_2) group by t.dept_no ;
+--------+---------+------------+------------+----------+
| emp_no | dept_no | from_date  | to_date    | count(1) |
+--------+---------+------------+------------+----------+
|  31112 | d002    | 1986-12-01 | 1993-12-10 |        1 |
|  10004 | d004    | 1986-12-01 | 9999-01-01 |        1 |
|  54007 | d005    | 1986-12-01 | 9999-01-01 |        4 |
|  22744 | d006    | 1986-12-01 | 9999-01-01 |        1 |
|  49667 | d007    | 1986-12-01 | 9999-01-01 |        1 |
|   NULL | d008    | 1986-12-01 | 1992-05-27 |        2 |
+--------+---------+------------+------------+----------+
```

延伸 group by

如果添加order by null 就可以发现 extra 部分就没有 using filesort
从执行结果中可以看出 dept_no 没有排序

```
root@mysql3306.sock>[employees]>desc select t.* ,count(1) from   t_order2 t ignore index(ix_torder_3) ignore index(ix_torder_2) group by t.dept_no order by null ;
+----+-------------+-------+------------+------+-------------------------+------+---------+------+------+----------+-----------------+
| id | select_type | table | partitions | type | possible_keys           | key  | key_len | ref  | rows | filtered | Extra           |
+----+-------------+-------+------------+------+-------------------------+------+---------+------+------+----------+-----------------+
|  1 | SIMPLE      | t     | NULL       | ALL  | ix_torder_2,ix_torder_3 | NULL | NULL    | NULL |   10 |   100.00 | Using temporary |
+----+-------------+-------+------------+------+-------------------------+------+---------+------+------+----------+-----------------+
1 row in set, 1 warning (0.00 sec)

root@mysql3306.sock>[employees]>select t.* ,count(1) from   t_order2 t ignore index(ix_torder_3) ignore index(ix_torder_2) group by t.dept_no order by null ;
+--------+---------+------------+------------+----------+
| emp_no | dept_no | from_date  | to_date    | count(1) |
+--------+---------+------------+------------+----------+
|  22744 | d006    | 1986-12-01 | 9999-01-01 |        1 |
|  54007 | d005    | 1986-12-01 | 9999-01-01 |        4 |
|  31112 | d002    | 1986-12-01 | 1993-12-10 |        1 |
|   NULL | d008    | 1986-12-01 | 1992-05-27 |        2 |
|  49667 | d007    | 1986-12-01 | 9999-01-01 |        1 |
|  10004 | d004    | 1986-12-01 | 9999-01-01 |        1 |
+--------+---------+------------+------------+----------+
```

2.left join 不满足多对一 导致 结果出错

Left join 的时候 一般情况下要满足 n:1原则 即 left join 右边表中跟左边的表的连接条件是保证唯一性。

```
root@mysql3306.sock>[employees]>select * from departments force index(PRI);
+---------+--------------------+
| dept_no | dept_name          |
+---------+--------------------+
| d001    | Marketing          |
| d002    | Finance            |
| d003    | Human Resources    |
| d004    | Production         |
| d005    | Development        |
| d006    | Quality Management |
| d007    | Sales              |
| d008    | Research           |
| d009    | Customer Service   |
+---------+--------------------+
9 rows in set (0.00 sec)

root@mysql3306.sock>[employees]>select * from t_order2 order by 2;
+--------+---------+------------+------------+
| emp_no | dept_no | from_date  | to_date    |
+--------+---------+------------+------------+
|  31112 | d002    | 1986-12-01 | 1993-12-10 |
|  10004 | d004    | 1986-12-01 | 9999-01-01 |
|  54007 | d005    | 1986-12-01 | 9999-01-01 |
|  30970 | d005    | 1986-12-01 | 2017-03-29 |
|  40983 | d005    | 1986-12-01 | 9999-01-01 |
|  50449 | d005    | 1986-12-01 | 9999-01-01 |
|  22744 | d006    | 1986-12-01 | 9999-01-01 |
|  49667 | d007    | 1986-12-01 | 9999-01-01 |
|   NULL | d008    | 1986-12-01 | 1992-05-27 |
|  48317 | d008    | 1986-12-01 | 1989-01-11 |
+--------+---------+------------+------------+
10 rows in set (0.00 sec)

root@mysql3306.sock>[employees]>select * from t_order2 t left join departments d on t.dept_no = d.dept_no order by 2;
+--------+---------+------------+------------+---------+--------------------+
| emp_no | dept_no | from_date  | to_date    | dept_no | dept_name          |
+--------+---------+------------+------------+---------+--------------------+
|  31112 | d002    | 1986-12-01 | 1993-12-10 | d002    | Finance            |
|  10004 | d004    | 1986-12-01 | 9999-01-01 | d004    | Production         |
|  54007 | d005    | 1986-12-01 | 9999-01-01 | d005    | Development        |
|  30970 | d005    | 1986-12-01 | 2017-03-29 | d005    | Development        |
|  40983 | d005    | 1986-12-01 | 9999-01-01 | d005    | Development        |
|  50449 | d005    | 1986-12-01 | 9999-01-01 | d005    | Development        |
|  22744 | d006    | 1986-12-01 | 9999-01-01 | d006    | Quality Management |
|  49667 | d007    | 1986-12-01 | 9999-01-01 | d007    | Sales              |
|   NULL | d008    | 1986-12-01 | 1992-05-27 | d008    | Research           |
|  48317 | d008    | 1986-12-01 | 1989-01-11 | d008    | Research           |
+--------+---------+------------+------------+---------+--------------------+
 10 rows in set (0.00 sec)
```

2.left join 不满足多对一 导致 结果出错

如下图所示 不满足n:1 的条件 会发现有重复值！

```
root@mysql3306.sock>[employees]>select * from departments force index(PRI);
+---------+--------------------+
| dept_no | dept_name          |
+---------+--------------------+
| d001    | Marketing          |
| d002    | Finance            |
| d003    | Human Resources    |
| d004    | Production         |
| d005    | Development        |
| d006    | Quality Management |
| d007    | Sales              |
| d008    | Research           |
| d009    | Customer Service   |
+---------+--------------------+
9 rows in set (0.00 sec)

root@mysql3306.sock>[employees]>select * from t_order2 order by 2;
+--------+---------+------------+------------+
| emp_no | dept_no | from_date  | to_date    |
+--------+---------+------------+------------+
|  31112 | d002    | 1986-12-01 | 1993-12-10 |
|  10004 | d004    | 1986-12-01 | 9999-01-01 |
|  54007 | d005    | 1986-12-01 | 9999-01-01 |
|  30970 | d005    | 1986-12-01 | 2017-03-29 |
|  40983 | d005    | 1986-12-01 | 9999-01-01 |
|  50449 | d005    | 1986-12-01 | 9999-01-01 |
|  22744 | d006    | 1986-12-01 | 9999-01-01 |
|  49667 | d007    | 1986-12-01 | 9999-01-01 |
|   NULL | d008    | 1986-12-01 | 1992-05-27 |
|  48317 | d008    | 1986-12-01 | 1989-01-11 |
+--------+---------+------------+------------+

root@mysql3306.sock>[employees]>select * from departments t left join t_order2 d on t.dept_no = d.dept_no order by 1;
+---------+--------------------+--------+---------+------------+------------+
| dept_no | dept_name          | emp_no | dept_no | from_date  | to_date    |
+---------+--------------------+--------+---------+------------+------------+
| d001    | Marketing          |   NULL | NULL    | NULL       | NULL       |
| d002    | Finance            |  31112 | d002    | 1986-12-01 | 1993-12-10 |
| d003    | Human Resources    |   NULL | NULL    | NULL       | NULL       |
| d004    | Production         |  10004 | d004    | 1986-12-01 | 9999-01-01 |
| d005    | Development        |  54007 | d005    | 1986-12-01 | 9999-01-01 |
| d005    | Development        |  30970 | d005    | 1986-12-01 | 2017-03-29 |
| d005    | Development        |  40983 | d005    | 1986-12-01 | 9999-01-01 |
| d005    | Development        |  50449 | d005    | 1986-12-01 | 9999-01-01 |
| d006    | Quality Management |  22744 | d006    | 1986-12-01 | 9999-01-01 |
| d007    | Sales              |  49667 | d007    | 1986-12-01 | 9999-01-01 |
| d008    | Research           |   NULL | d008    | 1986-12-01 | 1992-05-27 |
| d008    | Research           |  48317 | d008    | 1986-12-01 | 1989-01-11 |
| d009    | Customer Service   |   NULL | NULL    | NULL       | NULL       |
+---------+--------------------+--------+---------+------------+------------+
13 rows in set (0.00 sec)
```

2.left join 不满足多对一 导致 结果出错

为了解决重复值 往往开发人员就会如下图所示
添加了group by 但是这样 就会出现之前的问题 会根据执行计划结果就不同！

```
root@mysql3306.sock>[employees]>select * from departments t left join t_order2 d on t.dept_no = d.dept_no  group by t.dept_no  order by 1;
+---------+--------------------+--------+---------+------------+------------+
| dept_no | dept_name          | emp_no | dept_no | from_date  | to_date    |
+---------+--------------------+--------+---------+------------+------------+
| d001    | Marketing          |   NULL | NULL    | NULL       | NULL       |
| d002    | Finance            |  31112 | d002    | 1986-12-01 | 1993-12-10 |
| d003    | Human Resources    |   NULL | NULL    | NULL       | NULL       |
| d004    | Production         |  10004 | d004    | 1986-12-01 | 9999-01-01 |
| d005    | Development        |  54007 | d005    | 1986-12-01 | 9999-01-01 |
| d006    | Quality Management |  22744 | d006    | 1986-12-01 | 9999-01-01 |
| d007    | Sales              |  49667 | d007    | 1986-12-01 | 9999-01-01 |
| d008    | Research           |   NULL | d008    | 1986-12-01 | 1992-05-27 |
| d009    | Customer Service   |   NULL | NULL    | NULL       | NULL       |
+---------+--------------------+--------+---------+------------+------------+
```

2.left join 不满足多对一 导致 结果出错

解决这种问题的方法 就是保证left join 右边的连接条件保证唯一性！
而且需要使用全group by 就是group by 之后select 上使用min,max等聚合函数

```
root@mysql3306.sock>[employees]>desc select * from departments t left join
 (select dept_no,min(emp_no) emp_no, min(from_date) from_date , min(to_date) to_date  from t_order2 group by dept_no ) d on t.dept_no = d.dept_no     order by 1;
+----+-------------+------------+------------+-------+---------------------------+-------------+---------+---------------------+------+----------+-------+
| id | select_type | table      | partitions | type  | possible_keys             | key         | key_len | ref                 | rows | filtered | Extra |
+----+-------------+------------+------------+-------+---------------------------+-------------+---------+---------------------+------+----------+-------+
|  1 | PRIMARY     | t          | NULL       | index | NULL                      | PRIMARY     | 12      | NULL                |    9 |   100.00 | NULL  |
|  1 | PRIMARY     | <derived2> | NULL       | ref   | <auto_key0>               | <auto_key0> | 13      | employees.t.dept_no |    2 |   100.00 | NULL  |
|  2 | DERIVED     | t_order2   | NULL       | index | ix_torder_2,ix_torder_3 | ix_torder_2 | 13      | NULL                |   10 |   100.00 | NULL  |
+----+-------------+------------+------------+-------+---------------------------+-------------+---------+---------------------+------+----------+-------+

root@mysql3306.sock>[employees]>select * from departments t
 left join (select dept_no,min(emp_no) emp_no, min(from_date) from_date , min(to_date) to_date  from t_order2 group by dept_no ) d
 on t.dept_no = d.dept_no     order by 1;
+---------+--------------------+---------+--------+------------+------------+
| dept_no | dept_name          | dept_no | emp_no | from_date  | to_date    |
+---------+--------------------+---------+--------+------------+------------+
| d001    | Marketing          | NULL    |   NULL | NULL       | NULL       |
| d002    | Finance            | d002    |  31112 | 1986-12-01 | 1993-12-10 |
| d003    | Human Resources    | NULL    |   NULL | NULL       | NULL       |
| d004    | Production         | d004    |  10004 | 1986-12-01 | 9999-01-01 |
| d005    | Development        | d005    |  30970 | 1986-12-01 | 2017-03-29 |
| d006    | Quality Management | d006    |  22744 | 1986-12-01 | 9999-01-01 |
| d007    | Sales              | d007    |  49667 | 1986-12-01 | 9999-01-01 |
| d008    | Research           | d008    |  48317 | 1986-12-01 | 1989-01-11 |
| d009    | Customer Service   | NULL    |   NULL | NULL       | NULL       |
+---------+--------------------+---------+--------+------------+------------+
9 rows in set (0.00 sec)
```

3.根据MySQL执行计划 判断为myisam 数据库并添加索引优化

下面是 群里问的实际案例
原sql： 执行1s+
SELECT program_id FROM t_record WHERE id=(SELECT MAX(id) FROM t_record WHERE uid=UNHEX('323634343039'))

修改的sql，执行 0.01
SELECT program_id FROM t_record WHERE id=(SELECT id FROM t_record WHERE uid=UNHEX('323634343039') ORDER BY id DESC LIMIT 1)

3.根据MySQL 执行计划 判断为myisam 数据库并添加索引优化

下面是 群里问的实际案例

```
1  DESC
2  SELECT program_id FROM t_record WHERE id=(SELECT MAX(id) FROM t_record WHERE uid=UNHEX('323634343039'))
3
```

1 结果 | 2 信息 | 3 表数据 | 4 信息

（只读） 限制行 第一行: 0

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	t_record	(NULL)	OK	const	PRIMARY	PRIMARY	4	const	1	100.00	(NULL)
2	SUBQUERY	t_record	(NULL)	OK	ref	t_u_index	t_u_index	5	const	1594014	100.00	(NULL)

```
4  DESC
5  SELECT program_id FROM t_record WHERE id=(SELECT id FROM t_record WHERE uid=UNHEX('323634343039') ORDER BY id DESC LIMIT 1)
```

1 结果 | 2 信息 | 3 表数据 | 4 信息

（只读） 限制行 第一行: 0

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	t_record	(NULL)	OK	const	PRIMARY	PRIMARY	4	const	1	100.00	(NULL)
2	SUBQUERY	t_record	(NULL)	OK	index	t_u_index	PRIMARY	4	(NULL)	13	7.54	Using where

Indexes	Columns	Index Type
PRIMARY	id	Unique
t_u_index	uid	
t_di_index	device_id	

3.根据MySQL 执行计划 判断为myisam 数据库并添加索引优化

下面是我仿照 刚才的案例 在自己的数据中运行的

这是salaries 表

```
root@mysql3306.sock>[employees]>desc select * from employees e where e.emp_no=(select emp_no from salaries where from_date='1988-06-25' order by emp_no desc limit 1);
+----+-------------+----------+------------+-------+---------------+---------+---------+-------+------+----------+-------------------------+
| id | select_type | table    | partitions | type  | possible_keys | key     | key_len | ref   | rows | filtered | Extra                   |
+----+-------------+----------+------------+-------+---------------+---------+---------+-------+------+----------+-------------------------+
|  1 | PRIMARY     | e        | NULL       | const | PRIMARY       | PRIMARY | 4       | const |    1 |   100.00 | NULL                    |
|  2 | SUBQUERY    | salaries | NULL       | index | NULL          | emp_no  | 4       | NULL  |    1 |    10.00 | Using where; Using index |
+----+-------------+----------+------------+-------+---------------+---------+---------+-------+------+----------+-------------------------+
root@mysql3306.sock>[employees]>select * from employees e where e.emp_no=(select emp_no from salaries where from_date='1988-06-25' order by emp_no desc limit 1);
+--------+------------+------------+-----------+--------+------------+
| emp_no | birth_date | first_name | last_name | gender | hire_date  |
+--------+------------+------------+-----------+--------+------------+
| 498565 | 1959-03-29 | Dietrich   | Lung      | F      | 1985-06-26 |
+--------+------------+------------+-----------+--------+------------+
1 row in set (0.01 sec)

root@mysql3306.sock>[employees]>desc select * from employees e where e.emp_no=(select max(emp_no) from salaries where from_date='1988-06-25'  );
+----+-------------+----------+------------+-------+---------------+---------+---------+-------+---------+----------+-------------------------+
| id | select_type | table    | partitions | type  | possible_keys | key     | key_len | ref   | rows    | filtered | Extra                   |
+----+-------------+----------+------------+-------+---------------+---------+---------+-------+---------+----------+-------------------------+
|  1 | PRIMARY     | e        | NULL       | const | PRIMARY       | PRIMARY | 4       | const |       1 |   100.00 | NULL                    |
|  2 | SUBQUERY    | salaries | NULL       | index | NULL          | emp_no  | 4       | NULL  | 2844047 |    10.00 | Using where; Using index |
+----+-------------+----------+------------+-------+---------------+---------+---------+-------+---------+----------+-------------------------+
2 rows in set, 1 warning (0.90 sec)

root@mysql3306.sock>[employees]>select * from employees e where e.emp_no=(select max(emp_no) from salaries where from_date='1988-06-25'  );
+--------+------------+------------+-----------+--------+------------+
| emp_no | birth_date | first_name | last_name | gender | hire_date  |
+--------+------------+------------+-----------+--------+------------+
| 498565 | 1959-03-29 | Dietrich   | Lung      | F      | 1985-06-26 |
+--------+------------+------------+-----------+--------+------------+
1 row in set (0.90 sec)
```

3.根据MySQL 执行计划 判断为myisam 数据库并添加索引优化

下面是我仿照 刚才的案例 在自己的数据中运行的

这是salaries3 表

```
root@mysql3306.sock>[employees]>desc select * from employees e where e.emp_no=(select emp_no from salaries3 where from_date='1988-06-25' order by emp_no desc limit 1);
+----+-------------+-----------+------------+-------+---------------+---------+---------+-------+------+----------+-------------------------+
| id | select_type | table     | partitions | type  | possible_keys | key     | key_len | ref   | rows | filtered | Extra                   |
+----+-------------+-----------+------------+-------+---------------+---------+---------+-------+------+----------+-------------------------+
|  1 | PRIMARY     | e         | NULL       | const | PRIMARY       | PRIMARY | 4       | const |    1 |   100.00 | NULL                    |
|  2 | SUBQUERY    | salaries3 | NULL       | ref   | PRIMARY       | PRIMARY | 3       | const |  183 |   100.00 | Using where; Using index |
+----+-------------+-----------+------------+-------+---------------+---------+---------+-------+------+----------+-------------------------+
2 rows in set, 1 warning (0.00 sec)

root@mysql3306.sock>[employees]>select * from employees e where e.emp_no=(select emp_no from salaries3 where from_date='1988-06-25' order by emp_no desc limit 1);
+--------+------------+------------+-----------+--------+------------+
| emp_no | birth_date | first_name | last_name | gender | hire_date  |
+--------+------------+------------+-----------+--------+------------+
| 498565 | 1959-03-29 | Dietrich   | Lung      | F      | 1985-06-26 |
+--------+------------+------------+-----------+--------+------------+



root@mysql3306.sock>[employees]>desc select * from employees e where e.emp_no=(select max(emp_no) from salaries3 where from_date='1988-06-25'  );
+----+-------------+-------+------------+-------+---------------+---------+---------+-------+------+----------+-----------------------------+
| id | select_type | table | partitions | type  | possible_keys | key     | key_len | ref   | rows | filtered | Extra                       |
+----+-------------+-------+------------+-------+---------------+---------+---------+-------+------+----------+-----------------------------+
|  1 | PRIMARY     | e     | NULL       | const | PRIMARY       | PRIMARY | 4       | const |    1 |   100.00 | NULL                        |
|  2 | SUBQUERY    | NULL  | NULL       | NULL  | NULL          | NULL    | NULL    | NULL  | NULL |     NULL | Select tables optimized away |
+----+-------------+-------+------------+-------+---------------+---------+---------+-------+------+----------+-----------------------------+
2 rows in set, 1 warning (0.01 sec)

root@mysql3306.sock>[employees]>select * from employees e where e.emp_no=(select max(emp_no) from salaries3 where from_date='1988-06-25'  );
+--------+------------+------------+-----------+--------+------------+
| emp_no | birth_date | first_name | last_name | gender | hire_date  |
+--------+------------+------------+-----------+--------+------------+
| 498565 | 1959-03-29 | Dietrich   | Lung      | F      | 1985-06-26 |
+--------+------------+------------+-----------+--------+------------+
```

3.根据MySQL 执行计划 判断为myisam 数据库并添加索引优化

两个表的索引状况 从这几个执行计划的extra 部分可以看出
有什么不一样的吗 ？

```
root@mysql3306.sock>[employees]>show index from salaries;
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| Table    | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| salaries |          0 | PRIMARY  |            1 | emp_no      | A         |      300024 |     NULL | NULL   |      | BTREE      |         |               |
| salaries |          0 | PRIMARY  |            2 | from_date   | A         |     2844047 |     NULL | NULL   |      | BTREE      |         |               |
| salaries |          1 | emp_no   |            1 | emp_no      | A         |      300024 |     NULL | NULL   |      | BTREE      |         |               |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
3 rows in set (0.00 sec)

root@mysql3306.sock>[employees]>show index from salaries3;
+-----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| Table     | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment |
+-----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| salaries3 |          0 | PRIMARY  |            1 | from_date   | A         |        7425 |     NULL | NULL   |      | BTREE      |         |               |
| salaries3 |          0 | PRIMARY  |            2 | emp_no      | A         |     2838426 |     NULL | NULL   |      | BTREE      |         |               |
+-----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
2 rows in set (0.01 sec)
```

3.根据MySQL 执行计划， 判断为myisam 数据库并添加索引优化

我们可以看出， 我按照他给的条件， 做出的相似的sql
跟之前的相比多出了 using index ！
而原来的SQL 没有！ 联系着innodb 二级索引是pk必须包含的。
从这可以得出 不是innodb!

经过核实 果然不是innodb 是 myisam ！
结合myisam 的特点 创建了包含id 的联合索引
最后 速度非常快。

“知数堂培训”是由资深MySQL专家叶金荣、吴炳锡联合创办的优质在线培训品牌，主推MySQL DBA实战/优化课、Python运维开发课,大数据课程以及SQL优化等多门优质课程，课程品质和口碑享誉业界

现有课程
MySQL DBA 实战班
MySQL DBA 优化班
Python运维开发班
SQL开发优化班
大数据实战就业班