SQL优化01

心得 ： explain的每一行记录代表查询语句里的一个表

SQL :

```
select  distinct
    t.LIST_ID, t.POLICY_ID, t.ITEM_ID, t.PRODUCT_ID, t.FEE_TYPE, t.POLICY_YEAR, t.AGENT_ID, t.CHANNEL_ID,
    t.PARTNER_ID, t.FACTOR_TYPE, t.FACTOR_ID, t.FACTOR_VALUE, t.STATUS, t.BUSINESS_TYPE, t.BUSINESS_DATE,
    t.CREATE_BY, t.CREATE_TIME, t.UPDATE_BY, t.UPDATE_TIME
         from t_chl_policy_factor t,
                t_chl_policy a
                where t.STATUS = '01'
                  and t.POLICY_ID = a.POLICY_ID
                  and t.FACTOR_TYPE = '101004'

        AND NOT exists ( select 1 from t_chl_comm_policy_detail e,t_chl_comm_settle f
        where e.POLICY_ID = t.POLICY_ID and e.SETTLE_ID = f.SETTLE_ID and f.STATUS = '03'
           AND f.settle_type='01' )
                and DATE_FORMAT(a.ISSUE_DATE,'%Y-%m') <= '2017-08'
                        AND a.ISSUE_DATE >= date_add('2017-08-31',interval - '180' day)
                        and a.ISSUE_DATE >= '2017-07-01'
                        and a.DELIVERY_DATE <= '2017-09-05'
                        and a.DELIVERY_CLOSE_DATE <= '2017-09-05'
                        and a.CALLBACK_DATE <= '2017-09-20'
                AND t.AGENT_ID IN (SELECT DISTINCT d.agent_id FROM
                                v_chl_org_ins_leader b,
                                v_chl_org_insurance c,
                                v_agent_channel_change d
                                        WHERE c.`PARENT_PATH` LIKE CONCAT('%_',b.`CHANNEL_ID`,'_%')
                                          AND c.CHANNEL_ID = d.`CHANNEL_ID`
                                          AND DATE_FORMAT(b.END_DATE,'%Y-%m')   >=  '2017-08'
                                          AND DATE_FORMAT(b.START_DATE,'%Y-%m')  <=  '2017-08'
                                          AND DATE_FORMAT(d.END_DATE,'%Y-%m')  >=  '2017-08'
                                          AND DATE_FORMAT(d.START_DATE,'%Y-%m')  <=  '2017-08'
                                          AND b.`LEADER_ID`= 1214072
                                          AND d.agent_id  <>  1214072
                );
```

b t_chl_org_ins_leader
a t_chl_policy
t t_chl_policy_factor
d t_agent_channel_change
c t_chl_org_insurance
f t_chl_comm_settle
e t_chl_comm_policy_detail

```
*************************** 1. row ***************************
           id: 1
  select_type: PRIMARY
        table: b t_chl_org_ins_leader  v_chl_org_ins_leader
   partitions: NULL
         type: ref
possible_keys: idx_chl_org_ins_leader__lead_id
          key: idx_chl_org_ins_leader__lead_id
      key_len: 9
          ref: const
         rows: 1
```

```
     filtered: 100.00
        Extra: Using where; Using temporary; Start temporary
*************************** 1. row ***************************
```

查询where条件用到了CHANNEL_ID（无索引），END_DATE（无索引），START_DATE（无索引），LEADER_ID（有索引）

视图，基表 ： `wift_iiws`.`t_chl_org_ins_leader`.

```
mysql> select count(*) from t_chl_org_ins_leader;
+----------+
| count(*) |
+----------+
|      476 |
+----------+
1 row in set (0.00 sec)
mysql>
```

基表只有476条数据，暂时无必要创建索引。

如果急需优化也可以现在创建索引看看效果。

可以考虑给CHANNEL_ID（无索引），END_DATE（无索引），START_DATE（无索引）列创建索引。

```
*************************** 2. row ***************************
           id: 1
  select_type: PRIMARY
        table: a t_chl_policy
   partitions: NULL
         type: ALL
possible_keys: idx_chl_policy__policy_id
          key: NULL
      key_len: NULL
          ref: NULL
         rows: 36742
     filtered: 0.41
        Extra: Using where; Using join buffer (Block Nested Loop)
*************************** 2. row ***************************
```

查询where条件用到了POLICY_ID（有索引），ISSUE_DATE（无索引），DELIVERY_DATE（无索引），DELIVERY_CLOSE_DATE（无索引），CALLBACK_DATE（无索引），

t_chl_policy表的policy_id有索引，别的select where条件用到的列都是date类型，可以考虑创建索引。

但是若建了索引的字段使用了函数如DATE_FORMAT()等则任然用不到索引。

```
*************************** 3. row ***************************
           id: 1
  select_type: PRIMARY
        table: t t_chl_policy_factor
   partitions: NULL
         type: ref
possible_keys: idx_chl_policy_factor__policy_id
          key: idx_chl_policy_factor__policy_id
      key_len: 122
          ref: wift_ccms.a.POLICY_ID
         rows: 9
     filtered: 1.00
        Extra: Using where
*************************** 3. row ***************************
```

t_chl_policy_factor t,

status列只有两个值01，02。但是定义却是varchar ： `STATUS` varchar(10) DEFAULT NULL COMMENT '状态',

```
mysql> SELECT COUNT(*) FROM t_chl_policy_factor;
+----------+
| COUNT(*) |
+----------+
|   289877 |
+----------+
```

1 row in set (0.29 sec)
mysql>
优化建议 ：
可以考虑将status列转换成tinyint类型以提高查询效率。
factor_type列只有8个不同值，定义却是varchar : `FACTOR_TYPE` varchar(40) NOT NULL COMMENT '指标类型',
mysql> select distinct FACTOR_TYPE from t_chl_policy_factor;
+-------------+
| FACTOR_TYPE |
+-------------+
| 101006 |
| 101003 |
| 101007 |
| 101008 |
| 101002 |
| 101005 |
| 101001 |
| 101004 |
+-------------+
8 rows in set (0.28 sec)
mysql>
鉴于该表总共有二十多万条记录，而status列和factor_type列基数小，不适宜创建索引
查询where条件用到了STATUS（无索引，基数小不适宜建索引），POLICY_ID（有索引），FACTOR_TYPE（无索引），AGENT_ID（无索引）
优化建议 ：
可以考虑将factor_type列转换成bigint类型再创建索引，或直接创建索引。
agent_id上没有索引，可以考虑创建索引。

*************************** 4. row ***************************
           id: 1
  select_type: PRIMARY
        table: d t_agent_channel_change
   partitions: NULL
         type: ALL
possible_keys: idx_agt_chl_chg__chl_id,idx_agt_chl_chg__agt_id
          key: NULL
      key_len: NULL
          ref: NULL
         rows: 1925
     filtered: 9.00
        Extra: Using where; Distinct; Using join buffer (Block Nested Loop)
*************************** 4. row ***************************
v_agent_channel_change d
视图，基表 ：`wift_iiws`.`t_agent_channel_change`.
mysql> select count(*) from t_agent_channel_change;
+----------+
| count(*) |
+----------+
| 1966 |
+----------+
1 row in set (0.00 sec)
mysql>
查询where条件用到了agent_id（有索引），CHANNEL_ID（有索引），END_DATE（无索引），START_DATE（无索引）
该表记录较少，暂时没有必要创建索引
可以考虑给END_DATE（无索引），START_DATE（无索引）两列创建索引。

*************************** 5. row ***************************
           id: 1
  select_type: PRIMARY

```
        table: c t_chl_org_insurance v_chl_org_insurance
   partitions: NULL
         type: eq_ref
possible_keys: PRIMARY
          key: PRIMARY
      key_len: 8
          ref: wift_iiws.t_agent_channel_change.CHANNEL_ID
         rows: 1
     filtered: 11.11
        Extra: Using where; Distinct; End temporary
*************************** 5. row ***************************
```

v_chl_org_insurance c,

视图，基表 ： `wift_iiws`.`t_chl_org_insurance`.

```
mysql> select count(*) from t_chl_org_insurance;
+----------+
| count(*) |
+----------+
|      925 |
+----------+
1 row in set (0.00 sec)
mysql>
```

该表只有925条数据，暂时无必要创建索引

查询where条件用到了PARENT_PATH（无索引），CHANNEL_ID（有索引）

可以考虑给PARENT_PATH（无索引）创建索引。

```
*************************** 6. row ***************************
           id: 2
  select_type: DEPENDENT SUBQUERY
        table: f t_chl_comm_settle
   partitions: NULL
         type: ALL
possible_keys: PRIMARY
          key: NULL
      key_len: NULL
          ref: NULL
         rows: 18
     filtered: 5.56
        Extra: Using where
*************************** 6. row ***************************
```

查询where条件用到了SETTLE_ID（有索引），STATUS（无索引，基数小不适合建索引），settle_type（无索引，基数小不适合建索引），

status列只有三个值，定义是 ： `STATUS` varchar(10) NOT NULL COMMENT '核算状态01-核算中 02-核算完成 04-审核 04-取消',

```
mysql> select distinct status from t_chl_comm_settle;
+--------+
| status |
+--------+
| 02     |
| 01     |
| 03     |
+--------+
3 rows in set (0.00 sec)
mysql> select distinct settle_type from t_chl_comm_settle;
+-------------+
| settle_type |
+-------------+
| 03          |
| 01          |
+-------------+
2 rows in set (0.00 sec)
```

```
mysql> select count(*) from t_chl_comm_settle;
+----------+
| count(*) |
+----------+
|       18 |
+----------+
1 row in set (0.00 sec)
mysql>
```

鉴于该表总共只有18条记录，暂时无必要创建索引。

```
*************************** 7. row ***************************
           id: 2
  select_type: DEPENDENT SUBQUERY
        table: e t_chl_comm_policy_detail
   partitions: NULL
         type: ref
possible_keys: I_CHL_COMM_PO_DETAIL__POL_SET_ID
          key: I_CHL_COMM_PO_DETAIL__POL_SET_ID
      key_len: 245
          ref: wift_ccms.f.SETTLE_ID,wift_ccms.t.POLICY_ID
         rows: 3
     filtered: 100.00
        Extra: Using index
7 rows in set, 6 warnings (0.01 sec)
*************************** 7. row ***************************
```

查询where条件用到了POLICY_ID，SETTLE_ID，

POLICY_ID和SETTLE__ID列上没有索引，只有(settle_id,policy_id)符合索引，可以考虑单独创建单列索引。