同步服务数据库访问规范

sys_info 数据库中主要包括了订单表，商品表，退款表，库存表，现有表中的索引是按照同步服务程序访问 sys_info 数据库的 sql 语句统一建立，用户没有权限对 sys_info 中的表进行索引的操作，所以用户在访问 sys_info 中的表的时候，需要按照表中现有的索引进行访问，如果用户没有按照索引规则进行访问，则会出现问题：

一．查询条件应按照表中索引查询：

查询传入的条件由于没有适当的索引可以使用，则会进行全表扫描，过多的消耗数据库 io，超出实例允许的最大阀值；频繁的从磁盘读取数据，内存中的热点数据被冲刷出去，命中率下降，进而导致整个数据库的响应时间上升；

例如:

jdp_tb_trade 的索引为:

```
jdp_tb_trade:
PRIMARY KEY (`tid`),
  KEY `ind_jdp_tb_trade_seller_nick_jdp_modified` (`seller_nick`,`jdp_modified`),
  KEY `ind_jdp_tb_trade_jdp_modified` (`jdp_modified`),
  KEY `ind_jdp_tb_trade_seller_nick_modified` (`seller_nick`,`modified`),
  KEY `ind_jdp_tb_trade_modified` (`modified`);
```

用户在对 jdp_tb_trade 表进行查询的时候，只能按照上面的索引进行查询,例如：

```
select count(*) from jdp_tb_item o where o.seller_nick ='xuancan' and (o.jdp_modified> '2013-08-05 12:21:56'  and o.jdp_modified<'2013-08-05 22:21:56' ;
Delete  from  jdp_tb_trade o where  jdp_modified <'2013-11-01';
select  *  from  jdp_tb_trade  o  where  o.seller_nick ='xuancan'  and  (o.modified> '2013-08-05 12:21:56'  and o.modified<'2013-08-05 22:21:56' ;
select  count(*) from jdp_tb_trade o where o.modified>'2013-08-05 12:21:56' and o.modified<'2013-08-05 22:21:56';
```

二．谨慎使用子查询的使用(mysql)：

mysql 的查询优化器在优化子查询的时候很弱，需要对子查询进行改写成关联的写法，例如下面的查询；

普通写法：

```
select * from my_db.user_test where seller_id in (select seller_id from sys_info.orders where oid=31722463696732);
```

优化的写法为：

```
select t1.* from my_db.user_test t1 ， (select seller_id from sys_info.orders where oid=31722463696732) t2 where t1.seller_id=t2.seller_id;
```

三．分页的优化：

用户通常会对订单或者商品进行分页的拉取，普通的 limit 的分页写法有一个问题就是，越往后翻页，性能就会越慢，需要对原来普通的翻页写法进行优化：

普通写法：

```
SELECT * FROM sys_info.jdp_tb_trade       WHERE
seller_nick=’xuancan’ and jdp_modified>='2012-09-18 16:00:01'
AND jdp_modified<='2012-09-18 16:30:00'
ORDER BY jdp_modified DESC LIMIT 29000,50    ;
```

优化写法：

```
select t1.* from sys_info.jdp_tb_trade t1 ,
   ( select tid from sys_info.jdp_tb_trade WHERE
     seller_nick=’xuancan’ AND gmt_modified>='2012-09-18 16:00:01'
     AND jdp_modified<='2012-09-18 16:30:00'
     ORDER BY jdp_modified DESC LIMIT 29000,50
   )   t2
where t1.tid=t2.tid;
```

注意：在优化写法中，获取主键的过程必须保证完全是走索引取得，如果查询条件不包含在索引内，则需要回表过滤不符合查询条件中的数据,该范例中 tid 为主键，seller_nick 和 jdp_modified 在 jdp_tb_trade 表中已建有索引，所以完全符合优化翻页写法的条件；

四．注意点

a.同步服务数据库中支持的时间粒度为秒：

同步服务现在使用的数据库主要包括 mysql 和 mssql，由于 mysql 中的 datetime 和 timestamp 数据类型的最小单位是秒，没有存储毫秒，所以保持 mssql 和 mysql 一致，支持时间粒度为秒；

b.增量拉取订单请使用 jdp_modified 字段，而不要使用 modified 字段；

五．附属信息

附 sys_info 中各表的索引结构：

```
jdp_tb_item:
  PRIMARY KEY (`num_iid`),
  KEY `ind_jdp_tb_item_nick_jdp_modified` (`nick`,`jdp_modified`),
  KEY `ind_jdp_tb_item_jdp_modified` (`jdp_modified`),
  KEY `ind_jdp_tb_item_nick_modified` (`nick`,`modified`),
  KEY `ind_jdp_tb_item_modified` (`modified`)
```

```
jdp_tb_refund：
  PRIMARY KEY (`refund_id`),
  KEY `ind_jdp_tb_refund_seller_nick_jdp_modified` (`seller_nick`,`jdp_modified`),
  KEY `ind_jdp_tb_refund_jdp_modified` (`jdp_modified`),
  KEY `ind_jdp_tb_refund_seller_nick_modified` (`seller_nick`,`modified`),
  KEY `ind_jdp_tb_refund_modified` (`modified`)；
```

```
jdp_tb_trade：
  PRIMARY KEY (`tid`),
  KEY `ind_jdp_tb_trade_seller_nick_jdp_modified` (`seller_nick`,`jdp_modified`),
  KEY `ind_jdp_tb_trade_jdp_modified` (`jdp_modified`),
  KEY `ind_jdp_tb_trade_seller_nick_modified` (`seller_nick`,`modified`),
  KEY `ind_jdp_tb_trade_modified` (`modified`)；
```

```
jdp_tm_refund：
  PRIMARY KEY (`refund_id`),
  KEY `ind_jdp_tm_refund_seller_nick_jdp_modified` (`seller_nick`,`jdp_modified`),
  KEY `ind_jdp_tm_refund_jdp_modified` (`jdp_modified`),
  KEY `ind_jdp_tm_refund_seller_nick_modified` (`seller_nick`,`modified`),
  KEY `ind_jdp_tm_refund_modified` (`modified`)；
```

```
jdp_tm_return：
  PRIMARY KEY (`refund_id`),
  KEY `ind_jdp_tm_return_status_jdp_modified` (`status`,`jdp_modified`),
  KEY `ind_jdp_tm_return_jdp_modified` (`jdp_modified`)
```

六.案例分析：

select tid , jdp_response from jdp_tb_trade where seller_nick = :1 and jdp_modified >= :2 and jdp_modified <= :3 and type in (:4 , :5 , :6 , :7 , :8 , :9 , :10) order by tid limit 42050 , 50

该 sql 存在的问题：

1.没有按照优化的翻页写法进行查询，这样会导致在往后翻页的过程中性能越来越慢；

应该写为：

```
select tid , jdp_response from jdp_tb_trade t1,
    (select tid from jdp_tb_trad
      where seller_nick = :1 and jdp_modified >= :2
      and    jdp_modified <= :3 and type in    ( :4 , :5 , :6 , :7 , :8 , :9 , :10 )
      order by tid limit 42050 , 50
    ) t2
 Where t1.tid=t2.id;
```

2.jdp_tb_trade 的索引为：

```
KEY `ind_jdp_tb_refund_seller_nick_jdp_modified` (`seller_nick`,`jdp_modified`),
```

由于查询中含有 type 字段，所以会导致上述的子查询翻页中回表，导致性能下降，所以用户在应用设计中，需要去掉对 type 字段的查询，数据查询完我们做二次筛选即可。

最后的最佳 sql 优化写法为：

```
select tid , jdp_response from jdp_tb_trade t1,
    (select tid from jdp_tb_trad
      where seller_nick = :1 and jdp_modified >= :2
      and    jdp_modified <= :3    order by tid limit 42050 , 50
    ) t2
  Where t1.tid=t2.id;
```