实践练习六（必选）：查看 OceanBase 执行计划

# 1. 安装benchmark

下载地址：<https://github.com/obpilot/benchmarksql-5.0.git>

[root@observer175 run]# tar -zvxf benchmarksql-5.0-master.zip

# 2. 初始化配置文件

[root@observer175 run]# pwd  
/root/benchmarksql-5.0-master/run  
[root@observer175 run]# cat props.ob  
[root@observer175 run]# cat runBenchmark.sh   
#!/usr/bin/env bash  
  
if [ $# -ne 1 ] ; then  
 echo "usage: $(basename $0) PROPS\_FILE" >&2  
 exit 2  
fi  
  
SEQ\_FILE="./.jTPCC\_run\_seq.dat"  
if [ ! -f "${SEQ\_FILE}" ] ; then  
 echo "0" > "${SEQ\_FILE}"  
fi  
SEQ=$(expr $(cat "${SEQ\_FILE}") + 1) || exit 1  
echo "${SEQ}" > "${SEQ\_FILE}"  
  
source /root/benchmarksql-5.0-master/run/funcs.sh $1  
  
setCP || exit 1  
  
myOPTS="-Dprop=$1 -DrunID=${SEQ}"  
  
java -cp "$myCP" $myOPTS jTPCC  
[root@observer175 run]# cat props.ob  
db=oracle  
driver=com.alipay.oceanbase.obproxy.mysql.jdbc.Driver  
# 修改连接串  
conn=jdbc:oceanbase://192.168.11.175:2883/tpcc?useUnicode=true&characterEncoding=utf-8  
user=root@ob\_tent\_lhq#lhq\_dev  
password=19961203  
  
warehouses=2  
loadWorkers=2  
  
terminals=5  
//To run specified transactions per terminal- runMins must equal zero  
runTxnsPerTerminal=0  
//To run for specified minutes- runTxnsPerTerminal must equal zero  
runMins=10  
//Number of total transactions per minute  
limitTxnsPerMin=0  
  
//Set to true to run in 4.x compatible mode. Set to false to use the  
//entire configured database evenly.  
terminalWarehouseFixed=true  
  
//The following five values must add up to 100  
newOrderWeight=45  
paymentWeight=43  
orderStatusWeight=4  
deliveryWeight=4  
stockLevelWeight=4  
  
// Directory name to create for collecting detailed result data.  
// Comment this out to suppress.  
resultDirectory=my\_result\_%tY-%tm-%td\_%tH%tM%tS  
osCollectorScript=./misc/os\_collector\_linux.py

# 3. 初始化数据

* 导入表结构

# 初始化表解构  
root@observer175 run]# ./runSQL.sh props.ob sql.oceanbase/tableCreates.sql  
# 导入报错  
You have an error in your SQL syntax; check the manual that corresponds to your OceanBase version for the right syntax to use near 'varchar2(10),  
  
# 替换建表语句中varchar2为varchar  
[root@observer175 run]# sed -i 's/varchar2/varchar/g' sql.oceanbase/tableCreates.sql  
  
# 再次执行，创建tablegroup报错，sequence报错  
  
create tablegroup "tpcc\_group" ;  
You have an error in your SQL syntax; check the manual that corresponds to your OceanBase version for the right syntax to use near '"tpcc\_group"' at line 1  
create sequence bmsql\_hist\_id\_seq;  
You have an error in your SQL syntax; check the manual that corresponds to your OceanBase version for the right syntax to use near 'sequence bmsql\_hist\_id\_seq' at line 1  
  
# 删除建表语句中表组的引号，删除sequence  
create tablegroup "tpcc\_group" ;替换为 create tablegroup tpcc\_group ;  
sed -i '/sequence/d' sql.oceanbase/tableCreates.sql  
  
[root@observer175 run]# obclient -h 192.168.11.175 -u root@ob\_tent\_lhq#lhq\_dev -P 2883 -p19961203 -e "show tables from tpcc"  
# 确认表结构导入成功  
+------------------+  
| Tables\_in\_tpcc |  
+------------------+  
| bmsql\_config |  
| bmsql\_customer |  
| bmsql\_district |  
| bmsql\_history |  
| bmsql\_item |  
| bmsql\_new\_order |  
| bmsql\_oorder |  
| bmsql\_order\_line |  
| bmsql\_stock |  
| bmsql\_warehouse |  
+------------------+

* 导入数据

[root@observer175 run]# ./runLoader.sh props.ob  
Starting BenchmarkSQL LoadData  
  
driver=com.alipay.oceanbase.obproxy.mysql.jdbc.Driver  
conn=jdbc:oceanbase://192.168.11.175:2883/tpcc?useUnicode=true&characterEncoding=utf-8  
user=root@ob\_tent\_lhq#lhq\_dev  
password=\*\*\*\*\*\*\*\*\*\*\*  
warehouses=2  
loadWorkers=2  
fileLocation (not defined)  
csvNullValue (not defined - using default 'NULL')  
  
Worker 000: Loading ITEM  
Worker 001: Loading Warehouse 1  
Worker 000: ERROR: Transaction is timeout  
Worker 001: ERROR: Transaction is timeout  
  
# 事务超时报错，修改相关参数：  
  
[root@observer175 run]# obclient -h 192.168.11.175 -u root@ob\_tent\_lhq#lhq\_dev -P 2883 -p19961203 -e "set global ob\_query\_timeout=36000000000;set global ob\_trx\_timeout=36000000000;"  
  
[root@observer175 run]# ./runLoader.sh props.ob  
Starting BenchmarkSQL LoadData  
  
driver=com.alipay.oceanbase.obproxy.mysql.jdbc.Driver  
conn=jdbc:oceanbase://192.168.11.175:2883/tpcc?useUnicode=true&characterEncoding=utf-8  
user=root@ob\_tent\_lhq#lhq\_dev  
password=\*\*\*\*\*\*\*\*\*\*\*  
warehouses=2  
loadWorkers=2  
fileLocation (not defined)  
csvNullValue (not defined - using default 'NULL')  
  
Worker 000: Loading ITEM  
Worker 001: Loading Warehouse 1  
Worker 000: Loading ITEM done  
Worker 000: Loading Warehouse 2  
Worker 001: Loading Warehouse 1 done  
Worker 000: Loading Warehouse 2 done  
# 数据初始化完成

# 4.执行压力测试

[root@observer175 run]# ./runSQL.sh props.ob sql.oceanbase/indexCreates.sql   
# ------------------------------------------------------------  
# Loading SQL file sql.oceanbase/indexCreates.sql  
# ------------------------------------------------------------  
create index bmsql\_customer\_idx1  
on bmsql\_customer (c\_w\_id, c\_d\_id, c\_last, c\_first) local;  
create index bmsql\_oorder\_idx1  
on bmsql\_oorder (o\_w\_id, o\_d\_id, o\_carrier\_id, o\_id) local;  
  
[root@observer175 run]# sh runBenchmark.sh props.ob  
runBenchmark.sh: 第 15 行:source: funcs.sh: 没有找到文件  
  
# 报找不到文件，修改配置文件相关配置为绝对路径  
  
[root@observer175 run]# find /root/benchmarksql-5.0-master -name funcs.sh  
/root/benchmarksql-5.0-master/run/funcs.sh  
[root@observer175 run]# cat runBenchmark.sh   
#!/usr/bin/env bash  
  
if [ $# -ne 1 ] ; then  
 echo "usage: $(basename $0) PROPS\_FILE" >&2  
 exit 2  
fi  
  
SEQ\_FILE="./.jTPCC\_run\_seq.dat"  
if [ ! -f "${SEQ\_FILE}" ] ; then  
 echo "0" > "${SEQ\_FILE}"  
fi  
SEQ=$(expr $(cat "${SEQ\_FILE}") + 1) || exit 1  
echo "${SEQ}" > "${SEQ\_FILE}"  
  
source /root/benchmarksql-5.0-master/run/funcs.sh $1  
  
setCP || exit 1  
  
myOPTS="-Dprop=$1 -DrunID=${SEQ}"  
  
java -cp "$myCP" $myOPTS jTPCC  
[root@observer175 run]#

# 5. 查看sql执行计划

* 查找执行次数最多的前十条sql

MySQL [oceanbase]> select query\_sql,executions,elapsed\_time/executions avg\_elapsed\_time\_ms from oceanbase.gv$plan\_cache\_plan\_stat order by executions desc limit 10\G  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 query\_sql: SELECT i\_price, i\_name, i\_data FROM bmsql\_item WHERE i\_id = 2663  
 executions: 134560  
avg\_elapsed\_time\_ms: 645.1494  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 2. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 query\_sql: INSERT INTO bmsql\_order\_line ( ol\_o\_id, ol\_d\_id, ol\_w\_id, ol\_number, ol\_i\_id, ol\_supply\_w\_id, ol\_quantity, ol\_amount, ol\_dist\_info) VALUES (3001, 10, 2, 1, 2663, 2, 6, 534.96, 'yzgkLwBHIbZH1ZXVo1CfKdYN')  
 executions: 134433  
avg\_elapsed\_time\_ms: 515.4432  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 3. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 query\_sql: UPDATE bmsql\_stock SET s\_quantity = 12, s\_ytd = s\_ytd + 6, s\_order\_cnt = s\_order\_cnt + 1, s\_remote\_cnt = s\_remote\_cnt + 0 WHERE s\_w\_id = 2 AND s\_i\_id = 2663  
 executions: 134433  
avg\_elapsed\_time\_ms: 466.5327  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 4. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 query\_sql: SELECT s\_quantity, s\_data, s\_dist\_01, s\_dist\_02, s\_dist\_03, s\_dist\_04, s\_dist\_05, s\_dist\_06, s\_dist\_07, s\_dist\_08, s\_dist\_09, s\_dist\_10 FROM bmsql\_stock WHERE s\_w\_id = 2 AND s\_i\_id = 2663 FOR UPDATE  
 executions: 134433  
avg\_elapsed\_time\_ms: 1835.5766  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 5. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 query\_sql: SELECT c\_discount, c\_last, c\_credit, w\_tax FROM bmsql\_customer JOIN bmsql\_warehouse ON (w\_id = c\_w\_id) WHERE c\_w\_id = 2 AND c\_d\_id = 10 AND c\_id = 413  
 executions: 13388  
avg\_elapsed\_time\_ms: 4588.4065  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 6. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 query\_sql: SELECT d\_tax, d\_next\_o\_id FROM bmsql\_district WHERE d\_w\_id = 2 AND d\_id = 10 FOR UPDATE  
 executions: 13388  
avg\_elapsed\_time\_ms: 10564.8227  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 7. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 query\_sql: INSERT INTO bmsql\_new\_order ( no\_o\_id, no\_d\_id, no\_w\_id) VALUES (3001, 10, 2)  
 executions: 13388  
avg\_elapsed\_time\_ms: 637.6746  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 8. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 query\_sql: INSERT INTO bmsql\_oorder ( o\_id, o\_d\_id, o\_w\_id, o\_c\_id, o\_entry\_d, o\_ol\_cnt, o\_all\_local) VALUES (3001, 10, 2, 413, '2022-07-24 14:30:06.705', 14, 1)  
 executions: 13388  
avg\_elapsed\_time\_ms: 736.9303  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 9. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 query\_sql: UPDATE bmsql\_district SET d\_next\_o\_id = d\_next\_o\_id + 1 WHERE d\_w\_id = 2 AND d\_id = 10  
 executions: 13388  
avg\_elapsed\_time\_ms: 522.5645  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 10. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 query\_sql: INSERT INTO bmsql\_history ( h\_c\_id, h\_c\_d\_id, h\_c\_w\_id, h\_d\_id, h\_w\_id, h\_date, h\_amount, h\_data) VALUES (562, 4, 2, 4, 2, '2022-07-24 14:30:03.716', 3783.75, 'bd16S1Ij1 jBYYdh7')  
 executions: 12523  
avg\_elapsed\_time\_ms: 843.3396  
10 rows in set (0.016 sec)  
  
MySQL [oceanbase]>

* 查看前三条sql的详细执行计划

MySQL [tpcc]> explain extended SELECT i\_price, i\_name, i\_data FROM bmsql\_item WHERE i\_id = 2663\G  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
Query Plan: ========================================  
|ID|OPERATOR |NAME |EST. ROWS|COST|  
----------------------------------------  
|0 |TABLE GET|bmsql\_item|1 |53 |  
========================================  
  
Outputs & filters:   
-------------------------------------  
 0 - output([bmsql\_item.i\_price(0x2acc6ce1af80)], [bmsql\_item.i\_name(0x2acc6ccee300)], [bmsql\_item.i\_data(0x2acc6ccee820)]), filter(nil),   
 access([bmsql\_item.i\_price(0x2acc6ce1af80)], [bmsql\_item.i\_name(0x2acc6ccee300)], [bmsql\_item.i\_data(0x2acc6ccee820)]), partitions(p0),   
 is\_index\_back=false,   
 range\_key([bmsql\_item.i\_id(0x2acc6ce1aa60)]), range[2663 ; 2663],   
 range\_cond([bmsql\_item.i\_id(0x2acc6ce1aa60) = 2663(0x2acc6ce1a3e0)])  
  
Used Hint:  
-------------------------------------  
 /\*+  
 \*/  
  
Outline Data:  
-------------------------------------  
 /\*+  
 BEGIN\_OUTLINE\_DATA  
 FULL(@"SEL$1" "tpcc.bmsql\_item"@"SEL$1")  
 END\_OUTLINE\_DATA  
 \*/  
  
Plan Type:  
-------------------------------------  
LOCAL  
  
Optimization Info:  
-------------------------------------  
  
bmsql\_item:table\_rows:100000, physical\_range\_rows:1, logical\_range\_rows:1, index\_back\_rows:0, output\_rows:1, est\_method:local\_storage, optimization\_method=rule\_based, heuristic\_rule=unique\_index\_without\_indexback  
  
Parameters  
-------------------------------------  
  
1 row in set (0.007 sec)  
  
MySQL [tpcc]> explain extended INSERT INTO bmsql\_order\_line ( ol\_o\_id, ol\_d\_id, ol\_w\_id, ol\_number, ol\_i\_id, ol\_supply\_w\_id, ol\_quantity, ol\_amount, ol\_dist\_info) VALUES (3001, 10, 2, 1, 2663, 2, 6, 534.96, 'yzgkLwBHIbZH1ZXVo1CfKdYN')\G  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
Query Plan: ====================================  
|ID|OPERATOR |NAME|EST. ROWS|COST|  
------------------------------------  
|0 |INSERT | |1 |1 |  
|1 | EXPRESSION| |1 |1 |  
====================================  
  
Outputs & filters:   
-------------------------------------  
 0 - output([column\_conv(INT,PS:(11,0),NOT NULL,\_\_values.ol\_w\_id(0x2acc6ce1abf0))(0x2acc6cca2a60)], [column\_conv(INT,PS:(11,0),NOT NULL,\_\_values.ol\_d\_id(0x2acc6ce1a6d0))(0x2acc6cca3e30)], [column\_conv(INT,PS:(11,0),NOT NULL,\_\_values.ol\_o\_id(0x2acc6ce1a1b0))(0x2acc6cd80cd0)], [column\_conv(INT,PS:(11,0),NOT NULL,\_\_values.ol\_number(0x2acc6cc9f550))(0x2acc6cd82060)], [column\_conv(INT,PS:(11,0),NOT NULL,\_\_values.ol\_i\_id(0x2acc6cc9fa70))(0x2acc6cd83570)], [remove\_const(?)(0x2acc6cd6a760)], [column\_conv(DECIMAL,PS:(6,2),NULL,\_\_values.ol\_amount(0x2acc6cca09d0))(0x2acc6cd86d90)], [column\_conv(INT,PS:(11,0),NULL,\_\_values.ol\_supply\_w\_id(0x2acc6cc9ff90))(0x2acc6cd88120)], [column\_conv(INT,PS:(11,0),NULL,\_\_values.ol\_quantity(0x2acc6cca04b0))(0x2acc6cd89640)], [column\_conv(CHAR,utf8mb4\_general\_ci,length:24,NULL,\_\_values.ol\_dist\_info(0x2acc6cca0ef0))(0x2acc6cd8ae10)]), filter(nil),   
 columns([{bmsql\_order\_line: ({bmsql\_order\_line: (bmsql\_order\_line.ol\_w\_id(0x2acc6ce1a960), bmsql\_order\_line.ol\_d\_id(0x2acc6ce1a440), bmsql\_order\_line.ol\_o\_id(0x2acc6ce19f20), bmsql\_order\_line.ol\_number(0x2acc6ce1ae80), bmsql\_order\_line.ol\_i\_id(0x2acc6cc9f7e0), bmsql\_order\_line.ol\_delivery\_d(0x2acc6cca27a0), bmsql\_order\_line.ol\_amount(0x2acc6cca0740), bmsql\_order\_line.ol\_supply\_w\_id(0x2acc6cc9fd00), bmsql\_order\_line.ol\_quantity(0x2acc6cca0220), bmsql\_order\_line.ol\_dist\_info(0x2acc6cca0c60))})}]), partitions(p0)  
 1 - output([\_\_values.ol\_o\_id(0x2acc6ce1a1b0)], [\_\_values.ol\_d\_id(0x2acc6ce1a6d0)], [\_\_values.ol\_w\_id(0x2acc6ce1abf0)], [\_\_values.ol\_number(0x2acc6cc9f550)], [\_\_values.ol\_i\_id(0x2acc6cc9fa70)], [\_\_values.ol\_supply\_w\_id(0x2acc6cc9ff90)], [\_\_values.ol\_quantity(0x2acc6cca04b0)], [\_\_values.ol\_amount(0x2acc6cca09d0)], [\_\_values.ol\_dist\_info(0x2acc6cca0ef0)]), filter(nil)  
 values({3001, 10, 2, 1, 2663, 2, 6, 534.96, 'yzgkLwBHIbZH1ZXVo1CfKdYN'})  
  
Used Hint:  
-------------------------------------  
 /\*+  
 \*/  
  
Outline Data:  
-------------------------------------  
 /\*+  
 BEGIN\_OUTLINE\_DATA  
 END\_OUTLINE\_DATA  
 \*/  
  
Plan Type:  
-------------------------------------  
LOCAL  
  
Optimization Info:  
-------------------------------------  
  
  
Parameters  
-------------------------------------  
{obj:{"NULL":"NULL"}, accuracy:{length:-1, precision:-1, scale:-1}, flag:0, raw\_text\_pos:-1, raw\_text\_len:-1, param\_meta:{type:"NULL", collation:"binary", coercibility:"IGNORABLE"}}  
1 row in set (0.049 sec)  
  
MySQL [tpcc]> explain extended UPDATE bmsql\_stock SET s\_quantity = 12, s\_ytd = s\_ytd + 6, s\_order\_cnt = s\_order\_cnt + 1, s\_remote\_cnt = s\_remote\_cnt + 0 WHERE s\_w\_id = 2 AND s\_i\_id = 2663\G  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
Query Plan: ==========================================  
|ID|OPERATOR |NAME |EST. ROWS|COST|  
------------------------------------------  
|0 |UPDATE | |1 |55 |  
|1 | TABLE GET|bmsql\_stock|1 |54 |  
==========================================  
  
Outputs & filters:   
-------------------------------------  
 0 - output(nil), filter(nil), table\_columns([{bmsql\_stock: ({bmsql\_stock: (bmsql\_stock.s\_w\_id(0x2acc6cd3dc50), bmsql\_stock.s\_i\_id(0x2acc6cd3dee0), bmsql\_stock.s\_quantity(0x2acc6ce19f10), bmsql\_stock.s\_ytd(0x2acc6ce1a3e0), bmsql\_stock.s\_order\_cnt(0x2acc6cd8e750), bmsql\_stock.s\_remote\_cnt(0x2acc6cd8f530), bmsql\_stock.s\_data(0x2acc6cd3e170), bmsql\_stock.s\_dist\_01(0x2acc6cd3e400), bmsql\_stock.s\_dist\_02(0x2acc6cd3e690), bmsql\_stock.s\_dist\_03(0x2acc6cd3e920), bmsql\_stock.s\_dist\_04(0x2acc6cd3ebb0), bmsql\_stock.s\_dist\_05(0x2acc6cd3ee40), bmsql\_stock.s\_dist\_06(0x2acc6cd3f0d0), bmsql\_stock.s\_dist\_07(0x2acc6cd3f360), bmsql\_stock.s\_dist\_08(0x2acc6cd3f5f0), bmsql\_stock.s\_dist\_09(0x2acc6cd3f880), bmsql\_stock.s\_dist\_10(0x2acc6cd3fc90))})}]),  
 update([bmsql\_stock.s\_quantity(0x2acc6ce19f10)=?], [bmsql\_stock.s\_ytd(0x2acc6ce1a3e0)=column\_conv(INT,PS:(11,0),NULL,cast(bmsql\_stock.s\_ytd(0x2acc6ce1a3e0) + 6(0x2acc6ce1ab40), INT(-1, 0))(0x2acc6ce24110))(0x2acc6ce22c30)], [bmsql\_stock.s\_order\_cnt(0x2acc6cd8e750)=column\_conv(INT,PS:(11,0),NULL,cast(bmsql\_stock.s\_order\_cnt(0x2acc6cd8e750) + 1(0x2acc6cd8eeb0), INT(-1, 0))(0x2acc6ce26150))(0x2acc6ce24c70)], [bmsql\_stock.s\_remote\_cnt(0x2acc6cd8f530)=column\_conv(INT,PS:(11,0),NULL,cast(bmsql\_stock.s\_remote\_cnt(0x2acc6cd8f530) + 0(0x2acc6cd8fc90), INT(-1, 0))(0x2acc6ce28190))(0x2acc6ce26cb0)])  
 1 - output([bmsql\_stock.s\_w\_id(0x2acc6cd3dc50)], [bmsql\_stock.s\_i\_id(0x2acc6cd3dee0)], [bmsql\_stock.s\_quantity(0x2acc6ce19f10)], [bmsql\_stock.s\_ytd(0x2acc6ce1a3e0)], [bmsql\_stock.s\_order\_cnt(0x2acc6cd8e750)], [bmsql\_stock.s\_remote\_cnt(0x2acc6cd8f530)], [bmsql\_stock.s\_data(0x2acc6cd3e170)], [bmsql\_stock.s\_dist\_01(0x2acc6cd3e400)], [bmsql\_stock.s\_dist\_02(0x2acc6cd3e690)], [bmsql\_stock.s\_dist\_03(0x2acc6cd3e920)], [bmsql\_stock.s\_dist\_04(0x2acc6cd3ebb0)], [bmsql\_stock.s\_dist\_05(0x2acc6cd3ee40)], [bmsql\_stock.s\_dist\_06(0x2acc6cd3f0d0)], [bmsql\_stock.s\_dist\_07(0x2acc6cd3f360)], [bmsql\_stock.s\_dist\_08(0x2acc6cd3f5f0)], [bmsql\_stock.s\_dist\_09(0x2acc6cd3f880)], [bmsql\_stock.s\_dist\_10(0x2acc6cd3fc90)], [remove\_const(?)(0x2acc6cd4adf0)], [column\_conv(INT,PS:(11,0),NULL,cast(bmsql\_stock.s\_ytd(0x2acc6ce1a3e0) + 6(0x2acc6ce1ab40), INT(-1, 0))(0x2acc6ce24110))(0x2acc6ce22c30)], [column\_conv(INT,PS:(11,0),NULL,cast(bmsql\_stock.s\_order\_cnt(0x2acc6cd8e750) + 1(0x2acc6cd8eeb0), INT(-1, 0))(0x2acc6ce26150))(0x2acc6ce24c70)], [column\_conv(INT,PS:(11,0),NULL,cast(bmsql\_stock.s\_remote\_cnt(0x2acc6cd8f530) + 0(0x2acc6cd8fc90), INT(-1, 0))(0x2acc6ce28190))(0x2acc6ce26cb0)]), filter(nil),   
 access([bmsql\_stock.s\_quantity(0x2acc6ce19f10)], [bmsql\_stock.s\_ytd(0x2acc6ce1a3e0)], [bmsql\_stock.s\_order\_cnt(0x2acc6cd8e750)], [bmsql\_stock.s\_remote\_cnt(0x2acc6cd8f530)], [bmsql\_stock.s\_w\_id(0x2acc6cd3dc50)], [bmsql\_stock.s\_i\_id(0x2acc6cd3dee0)], [bmsql\_stock.s\_data(0x2acc6cd3e170)], [bmsql\_stock.s\_dist\_01(0x2acc6cd3e400)], [bmsql\_stock.s\_dist\_02(0x2acc6cd3e690)], [bmsql\_stock.s\_dist\_03(0x2acc6cd3e920)], [bmsql\_stock.s\_dist\_04(0x2acc6cd3ebb0)], [bmsql\_stock.s\_dist\_05(0x2acc6cd3ee40)], [bmsql\_stock.s\_dist\_06(0x2acc6cd3f0d0)], [bmsql\_stock.s\_dist\_07(0x2acc6cd3f360)], [bmsql\_stock.s\_dist\_08(0x2acc6cd3f5f0)], [bmsql\_stock.s\_dist\_09(0x2acc6cd3f880)], [bmsql\_stock.s\_dist\_10(0x2acc6cd3fc90)]), partitions(p0),   
 is\_index\_back=false,   
 range\_key([bmsql\_stock.s\_w\_id(0x2acc6cd3dc50)], [bmsql\_stock.s\_i\_id(0x2acc6cd3dee0)]), range[2,2663 ; 2,2663],   
 range\_cond([bmsql\_stock.s\_w\_id(0x2acc6cd3dc50) = 2(0x2acc6cd403f0)], [bmsql\_stock.s\_i\_id(0x2acc6cd3dee0) = 2663(0x2acc6cd40f40)])  
  
Used Hint:  
-------------------------------------  
 /\*+  
 \*/  
  
Outline Data:  
-------------------------------------  
 /\*+  
 BEGIN\_OUTLINE\_DATA  
 FULL(@"UPD$1" "tpcc.bmsql\_stock"@"UPD$1")  
 END\_OUTLINE\_DATA  
 \*/  
  
Plan Type:  
-------------------------------------  
LOCAL  
  
Optimization Info:  
-------------------------------------  
  
  
bmsql\_stock:table\_rows:200000, physical\_range\_rows:1, logical\_range\_rows:1, index\_back\_rows:0, output\_rows:1, est\_method:local\_storage, optimization\_method=rule\_based, heuristic\_rule=unique\_index\_without\_indexback  
  
Parameters  
-------------------------------------  
{obj:{"INT":12}, accuracy:{length:-1, precision:-1, scale:-1}, flag:0, raw\_text\_pos:-1, raw\_text\_len:-1, param\_meta:{type:"INT", collation:"binary", coercibility:"NUMERIC"}}  
1 row in set (0.089 sec)