

This is the query that takes more than 1 hour to execute.

```

select *
from SHOP_ORD where contract IN
(SELECT contract FROM site_public WHERE contract=
User_Allowed_Site_API.Authorized(contract))
and order_code_db in( 'M', 'F', 'P')
and (EXISTS (SELECT 1 FROM SHOP_MATERIAL_ALLOC SMA
WHERE (QTY_ISSUED+ SMA.QTY_ASSIGNED) < QTY_REQUIRED
AND SHOP_ORD.RELEASE_NO =SMA.RELEASE_NO
AND SHOP_ORD.ORDER_NO = SMA.ORDER_NO
and (sma.PART_NO IN (SELECT INVENTORY_PART_LOCATION.PART_NO
FROM INVENTORY_PART_LOCATION
WHERE INVENTORY_PART_LOCATION.QTY_ONHAND -
INVENTORY_PART_LOCATION.QTY_RESERVED >0)))
AND (OBJSTATE != 'Closed') AND (REVISED_DUE_DATE < SYSDATE +30))

```

When execute the explain plan for this in customer database I got following.

Description	Object o...	Object name	Cost	Cardi...	Bytes
SELECT STATEMENT GOAL = ALL_ROWS			9268	370	98050
TABLE ACCESS BY INDEX ROWID	IFSAPP	SHOP_ORD_TAB	1	1	249
NESTED LOOPS			9268	370	98050
MERGE JOIN CARTESIAN			8202	5328	85248
INDEX FULL SCAN	IFSAPP	SITE_PK	1	1	4
BUFFER SORT			8201	5328	63936
VIEW	SYS	VW_SQ_1	8201	5328	63936
HASH UNIQUE				5328	2664...
NESTED LOOPS			8201	5328	2664...
TABLE ACCESS FULL	IFSAPP	INVENTORY_PART_IN_STOCK_TAB	487	3084	61680
TABLE ACCESS BY INDEX ROWID	IFSAPP	SHOP_MATERIAL_ALLOC_TAB	3	2	60
INDEX RANGE SCAN	IFSAPP	C_SHOP_MATERIAL_ALLOC_IX1	1	25	
INDEX RANGE SCAN	IFSAPP	SHOP_ORD_TAB_PK	1	1	

Here you can see there is a buffer sort and I guess performance issue is raised due that.

When you run the same query in the in-house QA database you will get below explain plan

Description	Object ow...	Object name	Cost	Cardin...	Bytes
SELECT STATEMENT, GOAL = ALL_ROWS			198	1	257
FILTER					
TABLE ACCESS BY INDEX ROWID	IFSAPP	SHOP_ORD_TAB	185	9	2259
NESTED LOOPS			186	9	2313
INDEX FULL SCAN	IFSAPP	SITE_PK	1	1	6
INLIST ITERATOR					
INDEX RANGE SCAN	IFSAPP	SHOP_ORD_TAB_6_IX	4	20568	
TABLE ACCESS BY INDEX ROWID	IFSAPP	INVENTORY_PART_IN_STOCK_TAB	2	7	133
NESTED LOOPS			3	1	49
TABLE ACCESS BY INDEX ROWID	IFSAPP	SHOP_MATERIAL_ALLOC_TAB	1	1	30
INDEX RANGE SCAN	IFSAPP	SHOP_MATERIAL_ALLOC_PK	1	1	
INDEX RANGE SCAN	IFSAPP	INVENTORY_PART_IN_STOCK_PK	1	65	

When I update database statistics for three key tables in the query, shop_ord_tab, shop_material_alloc_tab and inventory_part_in_stock_tab, query populated with in few seconds. But again the performance dropped to it original state on the following day. I rebuild all the indexes and create few indexes for order_code in shop_ord_tab and part_no in inventory_part_in_stock_tab, but there was no any significant improvement. When modified the query as follows performance improved bit but still takes more than 45 minutes. In our QA database this populated with in few seconds.

```

select *
from SHOP_ORD where contract IN
(SELECT contract FROM site_public WHERE
contract=User_Allowed_Site_API.Authorized(contract))
and order_code_db in( 'M', 'F', 'P')
and (EXISTS (SELECT 1 FROM SHOP_MATERIAL_ALLOC SMA
WHERE (QTY_ISSUED+ SMA.QTY_ASSIGNED) < QTY_REQUIRED
AND SHOP_ORD.RELEASE_NO =SMA.RELEASE_NO
AND SHOP_ORD.ORDER_NO = SMA.ORDER_NO
and (EXISTS (SELECT 1
FROM INVENTORY_PART_LOCATION
WHERE INVENTORY_PART_LOCATION.QTY_ONHAND -
INVENTORY_PART_LOCATION.QTY_RESERVED >0
AND INVENTORY_PART_LOCATION.PART_NO = SMA.PART_NO)))
AND (OBJSTATE != 'Closed') AND (REVISED_DUE_DATE < SYSDATE +30))

```