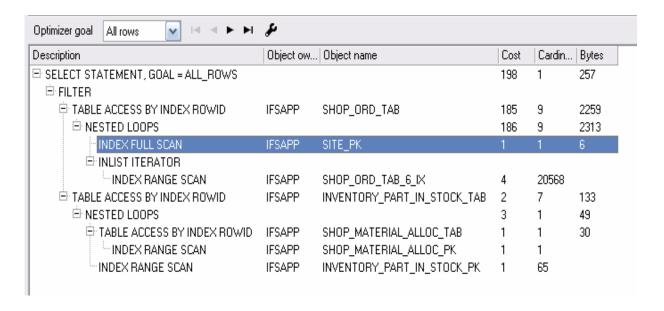
This is the query that takes more than 1 hour to execute.

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
	IFSAPP	SHOP_ORD_TAB	1	1	249
			9268	370	9805
			8202	5328	8524
- INDEX FULL SCAN	IFSAPP	SITE_PK	1	1	4
BUFFER SORT			8201	5328	6393
i VIEW	SYS	VW_SQ_1	8201	5328	6393
i⊕ HASH UNIQUE				5328	2664.
			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



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.