#### Safe Harbor

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Statements in this presentation relating to Oracle's future plans, expectations, beliefs, intentions and prospects are "forward-looking statements" and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle's Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading "Risk Factors." These filings are available on the SEC's website or on Oracle's website at <a href="http://www.oracle.com/investor">http://www.oracle.com/investor</a>. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.

### ORACLE

## Real-World Performance Techniques for Extreme Data Warehousing

Robert Carlin Mihajlo Tekic

Real-World Performance Oracle Database Development



opyright © 2019 Oracle and/or its affiliates

# What is Real-World Performance?

- Getting the most out of Software and Hardware
- Achieving Performance Excellence



#### **Real-World Performance Team**

### Who We Are

- Part of Oracle Database Development
- Team members in USA, Europe, and Asia
- Over a hundred years of experience combined

## How We Work

- Use the product as designed
- Take a holistic view
- Aim for best performance
- Apply data-driven analysis
- Share what we learn



#### What we do

## Customer Engagements

- Design Review
- Escalations
- Performance
  Projects

Database Development

- Tools
- Applications

Customer Education

- In Person
- Online





#### **Root Causes of Suboptimal Database Performance**





The database is not being used as it was designed to be used The application architecture/ code design is suboptimal

There is a suboptimal algorithm in the database



#### **Data Warehousing Performance**

- Oracle has enhanced its data warehousing capabilities dramatically over the last decade
  - Exadata Database In-Memory Smart Scans Bloom Filtering Vector Processing Storage Indexes
  - Columnar Storage



**Data Warehousing Performance** 

- Many older Data Warehouse projects were built with an OLTP mindset
  - Over indexing Home grown parallelism
- Understanding of Data Warehousing concepts
  - Storage IO Bandwidth Parallel Processing Set-based processing techniques



Copyright © 2019 Oracle and/or its affiliates

• Ideal data model is Star Schema





• Ideal data model is Star Schema





- Ideal data model is Star Schema
- Or a derivative Snowflake Schema



- Ideal data model is Star Schema
- Or a derivative Snowflake Schema
- Well understood design pattern





- Ideal data model is Star Schema
- Or a derivative Snowflake Schema
- Well understood design pattern
- Proven to scale well



REAL-WORLD PERFORMANCE

- Ideal data model is Star Schema
- Or a derivative Snowflake Schema
- Well understood design pattern
- Proven to scale well
- Works well with query and ETL tools





**Schema and SQL Statement** SELECT CATEGORY ID , COUNTRY , SUM(CTRL) as CTRL , SUM(QUANTITY) AS S Q FROM (SELECT P.CATEGORY ID , SP.COUNTRY , CASE WHEN SP.CATEGORY='CAT '||P.CATEGORY ID THEN 1 ELSE 0 END as CTRL , QUANTITY FROM SALES SL INNER JOIN DEPARTMENTS D ON SL.DEPARTMENT ID=D.DEPARTMENT ID INNER JOIN STORES S ON D.STORE\_ID=S.STORE\_ID INNER JOIN LOCATIONS L ON S.LOCATION ID=L.LOCATION ID INNER JOIN PRODUCTS P ON SL.PRODUCT\_ID=P.PRODUCT\_ID INNER JOIN SUPPLIERS SP ON SL.SUPPLIER\_ID=SP.SUPPLIER\_ID WHERE (L.CITY) IN ('CITY\_1', 'CITY\_2', 'CITY\_3') Products AND S.STORE TYPE=0 GROUP BY CATEGORY\_ID, COUNTRY



#### **Dimension Model**

- The advantage of the dimensional data model is that the desired execution plan is predictable
- Knowing the data sizes, we would expect our query to return in 10s or less

We should at least be able to estimate the data acquisition phase





///X











• Update the statistics?



- Update the statistics?
- Histograms?



- Update the statistics?
- Histograms?
- Column Groups?



- Update the statistics?
- Histograms?
- Column Groups?
- Increase the degree of parallelism?



- Update the statistics?
- Histograms?
- Column Groups?
- Increase the degree of parallelism?
- Set optimizer\_index\_cost\_adj?



- Update the statistics?
- Histograms?
- Column Groups?
- Increase the degree of parallelism?
- Set optimizer\_index\_cost\_adj?
- Set cursor\_sharing = FORCE?



- Update the statistics?
- Histograms?
- Column Groups?
- Increase the degree of parallelism?
- Set optimizer\_index\_cost\_adj?
- Set cursor\_sharing = FORCE?
- Increase the block size?



- Update the statistics?
- Histograms?
- Column Groups?
- Increase the degree of parallelism?
- Set optimizer\_index\_cost\_adj?
- Set cursor\_sharing = FORCE?
- Increase the block size?
- Google for some magic hidden parameter?







### "Let's work the problem people; let's not make things worse by guessing"

Gene Kranz Flight Director Apollo 13





**Schema and SQL Statement** SELECT CATEGORY ID , COUNTRY , SUM(CTRL) as CTRL , SUM(QUANTITY) AS S Q FROM (SELECT P.CATEGORY ID , SP.COUNTRY , CASE WHEN SP.CATEGORY='CAT '||P.CATEGORY ID THEN 1 ELSE 0 END as CTRL , QUANTITY FROM SALES SL INNER JOIN DEPARTMENTS D ON SL.DEPARTMENT ID=D.DEPARTMENT ID INNER JOIN STORES S ON D.STORE\_ID=S.STORE\_ID INNER JOIN LOCATIONS L ON S.LOCATION ID=L.LOCATION ID INNER JOIN PRODUCTS P ON SL.PRODUCT\_ID=P.PRODUCT\_ID INNER JOIN SUPPLIERS SP ON SL.SUPPLIER\_ID=SP.SUPPLIER\_ID WHERE (L.CITY) IN ('CITY\_1', 'CITY\_2', 'CITY\_3') Products AND S.STORE TYPE=0 GROUP BY CATEGORY\_ID, COUNTRY



## Demo



Copyright © 2019 Oracle and/or its affiliates.

Monitored SQL Execution Details: azmv8ma1147ff 🥑 🏻 💣 Navigate to SQL Details

Save Page Refreshed 5:15:31 PM GMT-0700



🖂 Details

🥅 Plan Statistics 🛛 🗛 Plan 🖓 Parallel 🛛 📐 Activity 🛛 🔀 Metrics

peratio	ion I	Name	Line ID	Estimated Rows	Cost	Timeline(89s)	Executions	Actual Rows	Memory (Max)	Temp (Max)	Other	IO Requests	IO Bytes	Activity %
E-SE	SELECT STATEMENT		0				9	10						.29
	-PX COORDINATOR		1				9	10						
Ē	E-PX SEND QC (RANDOM) :	:TQ10006	2	71	871K		4	10						
	HASH GROUP BY		3	71	871K		4	10	6MB					
	- PX RECEIVE		4	71	871K		4	40						
	-PX SEND HASH :	:TQ10005	5	71	871K		4	40			<u> 66</u>			
	HASH GROUP BY		6	71	871K		4	40	17MB					3.2
			7	29K	871K		4	16M						.58
			8	29K	871K		4	16M						.58
	- HASH JOIN		9	29K	856K		4	16M	5GB	1GB	<u> 8</u>	4,176	2GB	4.65
	-PX RECEIVE		10	29K	850K		4	66M						.58
	PX SEND BROADCAST :	:TQ10004	11	29K	850K		4	66M						.58
			12	29K	850K		4	17M	17MB		<u> 8</u>			.58
		:BF0000	13	10	34	1	4	6,000			<u> 8</u>			
	□-JOIN FILTER USE :	:BF0000	37	147M	850K		4	19M						
			38	147M	850K		4	19M						
	TABLE ACCESS STORAGE S	SALES	39	147M	850K		91	19M	48MB		<u> </u>	265K	86GB	2.33
			40	1,000K	5,820		4	1,000K						
	TABLE ACCESS STORAGE FULL	SUPPLIERS	41	1,000K	5,820		54	1,000K	18MB		<u> 8</u>	1,844	601MB	
	INDEX UNIQUE SCAN	PRODUCTS_PK	42	1	1		19M	16M				5,246K	40GB	
	TABLE ACCESS BY INDEX ROWID	PRODUCTS	43	1	2		25M	16M				<b>905</b> K	7GB	37



Monitored SQL Execution Details: azmv8ma1147ff 🥑 🏻 💣 Navigate to SQL Details

Save Page Refreshed 5:15:31 PM GMT-0700



🖂 Details

🥅 Plan Statistics 🛛 🗛 Plan 🖓 Parallel 🛛 📐 Activity 🛛 🔀 Metrics

perati	tion	Name	Line ID	Estimated Rows	Cost	Timeline(89s)	Executions	Actual Rows	Memory (Max)	Temp (Max)	Other	IO Requests	IO Bytes	Activity %
⊟−S	SELECT STATEMENT		0				9	10						.29
È	- PX COORDINATOR		1				9	10						
1	PX SEND QC (RANDOM)	:TQ10006	2	71	871K		4	10						
			3	71	871K		4	10	6MB					
	- PX RECEIVE		4	71	871K		4	40						
	- PX SEND HASH	:TQ10005	5	71	871K		4	40			66			
	HASH GROUP BY		6	71	871K		4	40	17MB					3.2
	- NESTED LOOPS		7	29К	871K		4	16M						.58
	- NESTED LOOPS		8	29К	871K		4	16M						.58
	- HASH JOIN		9	29К	856K		4	16M	5GB	1GB	8	4,176	2GB	4.65
	PX RECEIVE		10	29К	850K		4	66M						.58
	PX SEND BROADCAST	:TQ10004	11	29К	850K		4	66M						.58
	- HASH JOIN		12	29К	850K		4	17M	17MB		<u> 8</u>			.58
	- JOIN FILTER CREATE	:BF0000	13	10	34	1	4	6,000			<u> 8</u>			
	- JOIN FILTER USE	:BF0000	37	147M	850K		4	19M						
	PX BLOCK ITERATOR		38	147M	850K		4	19M						
	TABLE ACCESS STORAGE	SALES	39	147M	850K		91	19M	48MB		<u> 6</u>	265K	86GB	2.33
	E-PX BLOCK ITERATOR		40	1,000K	5,820		4	1,000K						
	TABLE ACCESS STORAGE FULL	SUPPLIERS	41	1,000K	5,820		54	1,000K	18MB		<u> 8</u>	1,844	601MB	
	INDEX UNIQUE SCAN	PRODUCTS_PK	42	1	1		19M	16M				5,246K	40GB	
	TABLE ACCESS BY INDEX ROWID	PRODUCTS	43	1	2		25M	16M				<mark>_</mark> 905К	7GB	37



Monitored SQL Execution Details: azmv8ma1147ff 🥑 🎯 Navigate to SQL Details

Save Page Refreshed 5:15:31 PM GMT-0700

///x

General			Time &	Wait Statistics				IO Statistics							
SQL Text SEL Execution Plan cution Started Sun st Refresh Time Sun Execution ID 335 User MTE Fetch Calls 1	LECT /* q15 */ CATEGORY_ID , COUNTRY 4 1 Sep 15, 2019 5:13:48 PM 1 Sep 15, 2019 5:15:17 PM 554432 EKIC@PDB1	, SUM(C	E Databa PL/SQL Ac	Duration Li.5m Buffer Gets 10 Requests 10 Bytes 13 Get 13 Get 13 Get 14 Get 13 Get 13 Get 14											
etails															
Plan Statistics	हिंक Plan 🖓 Parallel 📐 Activity	/ Metrics							_						
Plan Hash Value 115	55856492 Plan Note									Almo	set all DR	Time sno			
peration		Name	Line ID	Estimated Rows	Cost	Timeline(89s) Executions	Actual Rows	Memory (Max)	Temp (Max)	AIIII		inne spe			
- SELECT STATEM	ENT		0			9	10			on a	accessing	data fror			
	TOR		1			9	10								
PX SEND QC	C (RANDOM)	:TQ10006	2	71	871K	4	10			ł	RODUC	IS table			
HASH GRO	DUP BY		3	71	871K	4	10	6MB							
	EIVE		4	71	871K	4	40					1			
E PX SE	ND HASH	:TQ10005	5	71	871K	4	40			<b>61</b> 0					
E HAS	SH GROUP BY		6	71	871K	4	40	17MB				3.2			
	ESTED LOOPS		7	29K	871K	4	16M					.58			
<b>-</b>	NESTED LOOPS		8	29K	871K	4	16M					.58			
Ē	- HASH JOIN		9	29K	856K	4	16M	5GB	1GB	4,176	2GB	4.65			
	PX RECEIVE		10	29K	850K	4	66M					.58			
	PX SEND BROADCAST	:TQ10004	11	29K	850K	4	66M					.58			
	HASH JOIN		12	29K	850K	4	17M	17MB		<b>66</b>		.58			
	JOIN FILTER CREATE	:BF0000	13	10	34	4	6,000			<u> </u>					
	D-JOIN FILTER USE	:BF0000	37	147M	850K	4	19M								
	PX BLOCK ITERATOR		38	147M	850K	4	19M								
	TABLE ACCESS STORAGE	SALES	39	147M	850K	91	19M	48MB		365K	860	iB 2.33			
	- PX BLOCK ITERATOR		40	1,000K	5,820	4	1,000K								
	TABLE ACCESS STORAGE FULL	SUPPLIERS	41	1,000K	5,820	54	1,000K	18MB		1,844	601MB				
	INDEX UNIQUE SCAN	PRODUCTS_PK	42	1	1	19M	16M				<b></b> 5,246K <b></b> 40GB				
Т	TABLE ACCESS BY INDEX ROWID	PRODUCTS	43	1	2	25M	16M			<b></b> 905K	7GB	37			



Monitored SQL Execution Details: azmv8ma1147ff 🥑 🛛 😵 Navigate to SQL Details

Save Page Refreshed 5:15:31 PM GMT-0700

///x

eneral		Time & W	ait Statistics					IO Statistics							
SQL Text SELECT /* q15 */ CATEGORY_ID , COUN Execution Plan \$4 soution Started Sun Sep 15, 2019 5:13:48 PM Exercise Sun Sep 15, 2019 5:15:17 PM	Dur Database PL/SQL & Activi	ation Time Time Taylor Os			Buffi IO Re IC	Buffer Gets 50 IO Requests 6,42 IO Bytes 136									
Execution ID 33554432						100	E>	kecutio	n tim	e 89s					
Fetch Calls 1		Neste	d Lo	ops join v	when										
etails			ioinin	g PR	ODUCTS	table									
Plan Statistics	ctivity 🔀 Metrics		Je	<b>o</b> · · ·											
lan Hash Value 1155856492 Plan Note				\	Why?					Aln	nost all DB	Time spe			
peration	Name	Line ID	Estimated Rows	Cost	Timeline(89s)	Executions	Actual Rows	Memory (Max)	Temp (Max)						
3-SELECT STATEMENT		0				9	10			or	i accessing	g data tro			
E-PX COORDINATOR		1				9	10					TC tabla			
È−PX SEND QC (RANDOM)	:TQ10006	2	71	871K		4	10				PRODUC	IS LADIE			
		3	71	871K		4	10	6MB							
	1T01000F	4	71	871K		4	40			<b>AA</b>		1			
	.1010005	5	71	871K		4	40	17MB		00		3.2			
		7	29K	871K		4	16M	17115				.58			
		8	29K	871K		4	16M					.58			
		9	29K	856K		4	16M	5GB	1GB	<b>80</b> 4,3	176 2GB	4.65			
PX RECEIVE		10	29K	850K	•	4	66M					.58			
- PX SEND BROADCAST	:TQ10004	11	29K	850K		4	66M					.58			
		12	29K	850K		4	17M	17MB		<b>60</b>		.58			
	:BF0000	13	10	34		4	6,000			<del>60</del>					
	:BF0000	37	147M	850K		4	19M								
		38	147M	850K		4	19M								
	E SALES	39	147M	850K		91	19M	48MB		30 26	55K 860	GB 2.33			
		40	1,000K	5,820		4	1,000K	1015				[			
	SUPPLIERS	41	1,000K	5,820		54	1,000K	18MB		30 1,8	844 601MB				
	PRODUCTS_PK	42	1	1		19M	16M				5,246K 40GB				
└─ TABLE ACCESS BY INDEX ROWID	PRODUCTS	43	1	2		25M	16M			<mark></mark> 9	905K 7GB	37			



Monitored SQL Execution Details: azmv8ma1147ff 🥑 🏻 💣 Navigate to SQL Details

Save Page Refreshed 5:15:31 PM GMT-0700

///x

eneral			buration  1.5m    tabase Time  7.3m    /SQL & Java  0s    Activity %  100						IO Statistics								
SQL Te Execution Pl ecution Start at Refresh Tin Execution	ext SELECT /* q15 */ CATEGORY_ID , COUNTRY lan 3 4 ted Sun Sep 15, 2019 5:13:48 PM Sun Sep 15, 2019 5:15:17 PM ID 33554432	Dur Database PL/SQL & Activi							Buffer Gets 50 IO Requests 6,42 IO Bytes 1360 Cell								
User MTEKIC@PDB1 Fetch Calls 1				Neste	d Loops	join when											
etails				joinin	g PRODL	JCTS table											
Plan Statis	stics 🖧 Plan 🎲 Parallel 📐 Activit	y Metrics			- M/by/	ר											
Plan Hash Value 1155856492 Plan Note							Δ	Imost		Time sne	en						
peration		Name	Line ID	Estimated Rows	· · · · ·				Memory (Max)	Temp (Max)	<i></i>				~'		
E-SELECT STATEMENT			0		Estimated cardinality for			or			(	on aco	cessing	data fro	m		
		1															
PX SEND QC (RANDOM) :TQ1000		:TQ10006	2	71	row source operation 9							PR	ODUCI	S table			
<u>—</u> н	ASH GROUP BY		3	71		4 0 17			6MB								
Ð.	PX RECEIVE		4	71		18K								1			
E	- PX SEND HASH	:TQ10005	5	71							60						
	HASH GROUP BY		6	71	871K		4	40	17MB					3.2			
			/	29K	871k		4	16M						.58			
			9	29K	856K		4	16M	5GB	1GB	<i>6</i> 82	4 176	2GB	4 65			
			10	29K	850K		4	66M	565	100	00	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	1200	.58			
	PX SEND BROADCAST	:TQ10004	11	29K	850K		4	66M						.58			
			12	29K	850K		4	17M	17MB		<b>6</b>			.58			
	- JOIN FILTER CREATE	:BF0000	13	10	34		4	6,000			<u> 8</u>						
	JOIN FILTER USE	:BF0000	37	147M	850K		4	19M									
			38	147M	850K		4	19M									
	TABLE ACCESS STORAGE	SALES	39	147M	850K		91	19M	48MB		<u> 8</u>	265K	86GB	2.33			
	PX BLOCK ITERATOR		40	1,000K	5,820		4	1,000K									
	TABLE ACCESS STORAGE FULL	SUPPLIERS	41	1,000K	5,820		54	1,000K	18MB		<u> 8</u>	1,844	601MB				
	INDEX UNIQUE SCAN	PRODUCTS_PK	42	1	1	1	9M	16M				5,	,246K <u>4</u> 40GB		-		
0	TABLE ACCESS BY INDEX ROWID	PRODUCTS	43	1	2	2	5M	16M				📕 905К	7GB	37	7		


Monitored SQL Execution Details: azmv8ma1147ff 🥑 🛛 😵 Navigate to SQL Details

Save Page Refreshed 5:15:31 PM GMT-0700

///x

eneral			Time & Wa	it Statistics				IO Statistics					
SQL Te Execution Pla ecution Starte it Refresh Tim Execution 1	xt SELECT /* q15 */ CATEGORY_ID , COUNTRY an 4 4 ad Sun Sep 15, 2019 5:13:48 PM e Sun Sep 15, 2019 5:15:17 PM (D) 33554432	7 , SUM(C	Durat Database Ti PL/SQL & J Activity	tion ime ava Os 7 %			1.5m Z.3m 100	Buff IO Re IC Cell	er Gets equests D Bytes	a tina a Q	0.0		6,4 13
Us Fetch Ca	er MTEKIC@PDB1 lls 1			Neste	d Loops jo	oin when			ecution	i une o	95		
etails				joinin	g PRODU(	CTS table							
Plan Statis	tics 🦉 Plan 👪 Parallel 📐 Activit	y 🔀 Metrics			N/h.J								
ขลก Hash Valu	ue 1155856492 🦲 Plan Note				vviiy :					Almos	t all DB	Time sr	pen
peration		Name	Line ID E	stimated Rows				Memory (Max)	Temp (Max)	,		· · · · · ·	
SELECT S	STATEMENT		0		Estimate	d cardinality	for			on ac	cessing	data fr	om
	ORDINATOR		1			, aa anaratian	0:0			DE		·C tahla	
		:1Q10006	2	71	row sour	ce operation	915	6MB		L 1		JLable	
	PX RECEIVE		4	71	29К	Actual 16M		UND					
	PX SEND HASH	:TQ10005	5	71	2317,					<b>61</b>			
			6	71	871K	4	40	17MB				3.2	
	- NESTED LOOPS		7	29K	871K	4	16M					.58	
			8	29K	87TK	4	16M	FCB	100	A 176	208	.58	
			10	29K	850K	4	66M	200	IGD	4,170	ZGD	- 4.05	
	PX SEND BROADCAST	:TQ10004	11	29K	850K	4	66M					.58	
			12	29K	850K	4	17M	17MB		<del>60</del>		.58	
	JOIN FILTER CREATE	:BF0000	13	10	34	4	6,000			8 <b>6</b>			
		:BF0000	37	147M	850K	4	19M						
	E-PX BLOCK ITERATOR		38	147M	850K	4	19M						
		SALES	39	147M	850K	91	19M	48MB		Збородов 265К	86GI	3 2.33	
			40	1,000K	5,820	4	1,000K	10MD		<b>49</b> 1 944	601MP		
		PRODUCTS PK	41	1,000K	3,820	10M	1,000K	TOWR		1,044			
			74	1	1	1911	10141						



Monitored SQL Execution Details: azmv8ma1147ff 🥥 🛛 😁 Navigate to SQL Details

Save Page Refreshed 5:15:31 PM GMT-0700

///X

Overview														
General			Time & W	ait Statistics				IO Statistics						
SQL Execution I Execution Sta Last Refresh T Execution Execution Execution	Text         SELECT /* q15 */ CATEGORY_ID , COUNTRY           Plan         4           rted         Sun Sep 15, 2019 5:13:48 PM           ime         Sun Sep 15, 2019 5:15:17 PM           i ID         33554432           Jser         MTEKIC@PDB1           alls         1	<sup>и</sup> , sum(с	Dur Database PL/SQL & Activi	ation Time Java 0s ty % Neste	ed Loops joi	n when	1.5m Z.3m 100		er Gets equests D Bytes	n tim	e 89s			6,42 136
Details				ininin		S tabla								
Plan Stat	istics 🚋 Plan 🍪 Parallel 📐 Activit alue 1155856492 📄 Plan Note	ty Metrics		Johnin	Why?					Aln	nost a	all DB	Time sp	ben
Operation		Name	Line ID	Estimated Rows	E at instant a d		<b>f</b>	Memory (Max)	Temp (Max)	<b>^</b>		ccina	data fre	h
			0		Estimated	cardinality	tor			U	Iacce	essing	uatant	וווע
	SEND OC (BANDOM)	:TO10006	1	71	row cource	onoration	Q ic				PRO	DUCT	S table	
	HASH GROUP BY		3	71		eoperation	515	6MB				0001		
6 – 6 –	PX RECEIVE		4	71	29K. A	ctual 16M								
ĥ	PX SEND HASH	:TQ10005	5	71	23107					<u> 8</u>				
<u>ĝ</u>			6	71	871K	4	40	17MB					3.2	
ĝ.			7	29K	871K	4	16M						.58	
å.			8	29K	871K	4	16M						.58	
ដ			9	29К	856K	4	16M	5GB	1GB	3 4,	176	2GB	4.65	
2	- PX RECEIVE		10	29K	850K -	4	66M						.58	
3	E-PX SEND BROADCAST	:TQ10004	11	29K	850K	4	66M						.58	
5	E-HASH JOIN		12	29K	850K	4	17M	17MB		<b>60</b>			.58	
		:BF0000	13	10	34	4	6,000			66				
2		:BF0000	37	14/M	850K	4	19M							
6 D		CAL 50	38	14/M	850K	4	19M	10145			- FIZ	0.00		
2		. SALES	39	1 000%		91	1 000K	48MB		<b>30</b> 2	лсо	86G	2.33	
d 2			40	1,000K	5,820	4	1,000K	19MD		AGA 1	944	601MP		
r S		PRODUCTS PK	42	1	1	19M	1,000K	TOMB		<b>00</b> 11,	5.246	40GB		
μ β	TABLE ACCESS BY INDEX ROWID	PRODUCTS	43	1	-	1014	1011				905K	7GB		37
					The und	derestimate								
					originates f	rom row so	urce							



Monitored SQL Execution Details: azmv8ma1147ff 🥑 🏻 💣 Navigate to SQL Details

Save Page Refreshed 5:15:31 PM GMT-0700

///>

I	Time & Wait Statistics			IO Statistics				
SQL Text SELECT /* q15 */ CATEGORY_ID , COUNTRY , SUM(C	Duration		1.5m	Buffer	r Gets			
ution Plan 🖓 4 on Started Sun Sen 15, 2019 5:13:48 PM	PL/SQL & Java Os		7.3m	IO Red	Bytes			6
fresh Time Sun Sep 15, 2019 5:15:17 PM	Activity %		100	Cell				
ecution ID 33554432				Fx	ecutio	n time 890	:	
User MTEKIC@PDB1	Nected	oons join when			cculio			
	Nesteu L	oops join when						
Broadcasting ~17M rows 4	joining P	RODUCTS table						
times, that is ~66M rows,		Why?				Almost	ם ח	Timo coo
causing 1G to shill to	Line ID Estimated Davis			amami (Mavi)	Tomp (Mau)	AIIIOSU		inne spe
		stimated cardinality	for	emory (Max)	Temp (Max)	on acce	essing	data fron
PX SEND QC (RANDOM) :TQ10006	<sup>2</sup> <sup>71</sup> ro	w source operation	9 is			PRC	DUCT	S table
E- HASH GROUP BY	3 71			6MB				
- PX RECEIVE	4 71	29K, ACTUAI 16M						1
E-PX SEND HASH :TQ10005	5 71					<b>6</b> 0		
	6 71 871	4	40	17MB				3.2
	29K 8/1	4	16M					.58
	9 29K 856	T 4	16M	5GB	1GB	<b>4</b> 176	2GB	4 65
	10 29K 850	K = 4	66M					.58
PX SEND BROADCAST :TO10004	11 29K 850	K 4	66M					.58
	12 29K 850	K 4	17M	17MB		66		.58
	13 10 3	4 4	6,000			<del>60</del>		
□ JOIN FILTER USE :BF0000	37 147M 850	K 4	19M					
E-PX BLOCK ITERATOR	38 147M 850	K 4	19M					
TABLE ACCESS STORAGE SALES	39 147M 850	91	19M	48MB		📸 <mark>і</mark> 265К	86G	3 2.33
	40 1,000K 5,82	4	1,000K					
TABLE ACCESS STORAGE FULL SUPPLIERS	41 1,000K 5,82	54	1,000K	18MB		1,844	601MB	
INDEX UNIQUE SCAN PRODUCTS_PK	42 1	1 19M	16M			5,24	5K 📥 40GB	
TABLE ACCESS BY INDEX ROWID PRODUCTS	43					<mark></mark> 905K	7GB	37
		The underestimate						



#### 🗐 Plan Statistics 🛛 🙀 Plan 🖓 Parallel 📐 Activity 🖳 Metrics

Plan Hash Value 1155856492 📄 Plan Note

Operation		Name	Line ID	Estimated Rows	Cost	Timeline(89s)	Executions	Actual Rows	Memory (Max)	Temp (Max)	Ot	IO Requests	IO Bytes	Activity %
8 <b>6</b> 6	D-JOIN FILTER CREATE	:BF0000	13	10	34	I	4	6,000			<u> 8</u>			
8 <b>6</b> 6	- PX RECEIVE		14	10	34	i	4	6,000						
8 <u>3</u> 6	-PX SEND BROADCAST	:TQ10003	15	10	34	i	4	6,000						
2 <u>5</u> 2	HASH JOIN BUFFERED		16	10	34	i .	4	1,500	33MB		<u> 66</u>			
2 <u>25</u>	- JOIN FILTER CREATE	:BF0001	17	1	20	i	4	600			<u>66</u>			
දියුර	PX RECEIVE		18	1	20	i .	4	600						
660 A	PX SEND HYBRID HASH	:TQ10001	19	1	20	i	4	600			<u>66</u>			
8 <b>6</b> 8	- STATISTICS COLLECTOR		20			i .	4	150						
<b>65</b>		VW_GBF_16	21	1	20	i -	4	150						
8 <u>8</u>	HASH GROUP BY		22	1	20	i .	4	150	19MB					
<b>65</b>			23	1	20	i	4	150						
2 <b>3</b> 2	PX SEND HASH	:TQ10000	24	1	20	i .	4	150			<u> 66</u>			
2 <u>5</u> 5	HASH GROUP BY		25	1	20	i	4	150	14MB					
දියුර	- HASH JOIN		26	1	20	i .	4	150	17MB		<u> 8</u>			
2 <u>2</u> 6	E-JOIN FILTER	:BF0002	27	3	17	i	4	240			<u>66</u>			
250 260	TABLE ACCE	LOCATIONS	28	3	17	i .	4	240						
2 <u>5</u> 6	E-JOIN FILTER	:BF0002	29	700	3	i -	4	150			<u> 8</u>			
226	E PX BLOCK I		30	700	3	i .	4	150						
25	TABLE AC	STORES	31	700	3	i -	22	150				22	176KB	
	- PX RECEIVE		32	50K	14		4	0						
2 <u>5</u> 6		:TQ10002	33	50K	14	i	4	0						
<b>25</b> 0	- JOIN FILTER USE	:BF0000	37	147M	850K		4	19M						
<b>65</b>			38	147M	850K		4	19M						
8 <b>6</b>	TABLE ACCESS STORAGE FULL	SALES	39	147M	850K		91	19M	48MB		<u>66</u>	265K	<b>86GB</b>	2.33
2 <u>2</u> 6			40	1,000K	5,820		4	1,000K						
2 <u>5</u> 6	TABLE ACCESS STORAGE FULL	SUPPLIERS	41	1,000K	5,820		54	1,000K	18MB		<u> </u>	1,844	601MB	
2 <u>5</u> 2	INDEX UNIQUE SCAN	PRODUCTS_PK	42	1	1		19M	16M				5,246K	40GB	50
26	TABLE ACCESS BY INDEX ROWID	PRODUCTS	43	1	2		25M	16M				<b>905</b> K	7GB	37



#### 🗐 Plan Statistics 🛛 🙀 Plan 🖓 Parallel 📐 Activity 🖳 Metrics

Plan Hash Value	1155856492		Plan Note
-----------------	------------	--	-----------

											ng unuere:	sumaters	
Operation		Name	Line ID	Estimated Rows	Cost	Timeline(89s)	Executions	Actual Rows	Memory (Max)	coming fr	om row co	1000000	
8 <b>3</b> 3	D-JOIN FILTER CREATE	:BF0000	13	10	34	I	4	6,000		connig n		Juice 20.	•
8 <b>66</b>	- PX RECEIVE		14	10	34	l	4	6,000					т
2 <u>2</u> 2	PX SEND BROADCAST	:TQ10003	15	10	34	1	4	6,000			~		
2 <u>3</u> 2			16	10	34	l	4	1,500	33MB	Estimated	:3		
2 <u>.5</u>		:BF0001	17	1	20	l.	4	600		Actual: 60			
220 C			18	1	20	I	4	600					_
<b>63</b> 0	PX SEND HYBRID HASH	:TQ10001	19	1	20	l.	4	600					
8 <b>6</b> 6	- STATISTICS COLLECTOR		20			1	4	150					
8 <b>3</b> 3		VW_GBF_16	21	1	20	l.	4	150		Due to sm	iall table re	plication,	
<b>65</b> 0			22	1	20	I.	4	150	19MB	actual car	dinality ic r	nultiplied	
8 <b>3</b> 3			23	1	20	1	4	150		actuartan	uniancy is i	nuitipiieu	
2 <u>2</u> 6	D-PX SEND HASH	:TQ10000	24	1	20		4	150		by the nur	mber of PX	servers	
8 <b>6</b> 6	E-HASH GROUP BY		25	1	20		4	150	14MB				
220 C	E-HASH JOIN		26	1	20		4	150	17MB	<u>(4), hence</u>	<u>240 repor</u>	ted	
220 C	D JOIN FILTER	:BF0002	27	3	17		4	240	-	æ			
8 <u>7</u> 8	TABLE ACCE	LOCATIONS	28	3	17		4	240					
2 <u>2</u> 2	⊟– JOIN FILTER	:BF0002	29	700	3		4	150		<u> </u>			
<u> </u>	E PX BLOCK I		30	700	3		4	150					
2 <u>2</u> 2	TABLE AC	STORES	31	700	3	1	22	150		2	22 176KB		≣
	D PX RECEIVE		32	50K	14		4	0					
8 <b>3</b> 8	PX SEND HYBRID HASH	:TQ10002	33	50K	14	1	4	0					
8 <b>6</b> 0	D-JOIN FILTER USE	:BF0000	37	147M	850K		4	19M					
8 <b>6</b> 0	- PX BLOCK ITERATOR		38	147M	850K		4	19M					
8 <b>6</b> 0	TABLE ACCESS STORAGE FULL	SALES	39	147M	850K		91	19M	48MB	3 1 2	265K 86GB	2.33	
220 C			40	1,000K	5,820		4	1,000K					
220 C	TABLE ACCESS STORAGE FULL	SUPPLIERS	41	1,000K	5,820		54	1,000K	18MB	<b>a</b> 1	L,844 601MB		
220 C	-INDEX UNIQUE SCAN	PRODUCTS_PK	42	1	1		19M	16M		_	5,246K <mark>4</mark> 0GB	50	)
220 C	TABLE ACCESS BY INDEX ROWID	PRODUCTS	43	1	2		25M	16M		-	905K 7GB	37	•

///%

The first his underestimate is



Plan Statistics	s 🚰 Plan 🆓 Parallel 📐 Activit	y 🎘 Metrics							
Display Tabula	ar Value 1155856492	Plan Note							
Operation			Object	Line ID	Predicate	Pruning	Operation Cost	Estimated Rows	Estimated Bytes
	HASH JOIN BUFFERED			16	•			10	390
	JOIN FILTER CREATE		:BF0001	17				1	30
	- PX RECEIVE			18				1	30
		ASH	:TQ10001	19				1	30
		ECTOR		20					
			VW_GBF_16	21				1	30
	HASH GROUP	BY		22				1	27
				23				1	27
	PX SEND I	HASH	:TQ10000	24				1	27
	HASH G	ROUP BY		25				1	27
	- HASH	JOIN		26	•			1	27
	E-JOI	N FILTER CREATE	:BF0002	27				3	48
	ТА	BLE ACCESS STORAGE FULL	LOCATIONS	28	🍸 <b>*</b> 🗖		17	3	48
		N FILTER USE	:BF0002	29	filter: ("L	"."CITY"='CITY_1	' OR "L"."CITY"='CITY_2' OR	700	7,700
	E P	K BLOCK ITERATOR		30	"L"."CITY	("='CITY_3')		700	7,700
		TABLE ACCESS STORAGE FULL	STORES	31	T • 🗖		3	700	7,700
	PX RECEIVE			32				50K	450K
	-PX SEND HYBRID HAS	н	:TQ10002	33				50K	450K
	D-JOIN FILTER USE		:BF0001	34				50K	450K
		OR		35				50K	450K
	TABLE ACCESS S	TORAGE FULL	DEPARTMENTS	36	T • 🗐		14	50K	450K
	JOIN FILTER USE		:BF0000	37				147M	2,786M
	PX BLOCK ITERATOR			38				147M	2,786M
	TABLE ACCESS STORAGE FUL	L	SALES	39	T • 🗐		85	50K 147M	2,786M
				40				1,000K	<mark>.</mark> 30M
	TABLE ACCESS STORAGE FULL		SUPPLIERS	41	•		5,820	1,000K	<mark>.</mark> 30M
	INDEX UNIQUE SCAN		PRODUCTS_PK	42	•		1	1	
	TABLE ACCESS BY INDEX ROWID		PRODUCTS	43			2	1	9



Plan Statistics	류 Plan 🆓 Paralle	el 📐 Activity 🚬 Metrics								
Display Tabular	▼ Plan Hash Value	1155856492 📄 Plan Note				At row source 28 t	the following f	filter		
Operation			Object	Line ID	Pr	prodicato is	waluatad	st	imated Bytes	
	IOL HASH JOIN	N BUFFERED		16		predicate is	evaluateu.	90	)	•
		TER CREATE	:BF0001	17				D		
	E-PX REC	CEIVE		18				D		
	⊟-PX S	END HYBRID HASH	:TQ10001	19	CI	ITY in ('CITY 1',	'CITY 2', 'CI	TY 3') 🖓		
	E ST	ATISTICS COLLECTOR		20		_	—			
		VIEW	VW_GBF_16	21			1	30		
		- HASH GROUP BY		22			1	27		
				23			1	27		
		E PX SEND HASH	:TQ10000	24			1	27		
				25			1	27		
				26	<u>•</u>		1	27		
		JOIN FILTER CREATE	:BF0002	27			3	48		
		TABLE ACCESS STORAGE FULL	LOCATIONS	28	T 1	17	3	48		
			:BF0002	29	filter: ("	L"."CITY"='CITY_1' OR "L"."CITY"='CITY_2' OR	700	7,7	00	
				30	"L"."CIT	Y"='CITY_3')	700	7,7	00	
		TABLE ACCESS STORAGE FULL	STORES	31	Ý • 🗖	3	700	7,7	00	
	E-PX RECE	IVE		32			50K	450	Ж	
		ND HYBRID HASH	:TQ10002	33			50K	450	Ж	
		I FILTER USE	:BF0001	34			50K	450	Ж	=
	⊟–PX	BLOCK ITERATOR		35			50K	450	Ж	
	L L	ABLE ACCESS STORAGE FULL	DEPARTMENTS	36	Ý • 🗖	14	50K	450	Ж	
			:BF0000	37				147M	2,	786M
		RATOR		38				147M	2,	786M
	TABLE ACCES	S STORAGE FULL	SALES	39	¥ •⊟		850K	147M	2,	786M
	- PX BLOCK ITERATOR			40			1,000K	30	м	
	TABLE ACCESS STORA	AGE FULL	SUPPLIERS	41	• 🗖	5,820	1,000K	30	м	
	LINDEX UNIQUE SCAN		PRODUCTS_PK	42	• 🗖	1	1			
	TABLE ACCESS BY INDEX RO	OWID	PRODUCTS	43		2	1	9		-



///X)

ORACLE

REAL-WORLD PERFORMANCE

Ο



```
GROUP BY CATEGORY ID, COUNTRY;
```

#### AND S.STORE\_TYPE=0

SALES SL INNER JOIN DEPARTMENTS D ON SL.DEPARTMENT\_ID=D.DEPARTMENT\_ID INNER JOIN STORES S ON D.STORE\_ID=S.STORE\_ID INNER JOIN LOCATIONS L ON S.LOCATION\_ID=L.LOCATION\_ID INNER JOIN PRODUCTS P ON SL.PRODUCT\_ID=P.PRODUCT\_ID INNER JOIN SUPPLIERS SP ON SL.SUPPLIER\_ID=SP.SUPPLIER\_ID WHERE (L.CITY) IN ('CITY\_1','CITY\_2','CITY\_3')

WHEN SP.CATEGORY='CAT\_'||P.CATEGORY\_ID

#### FROM

, QUANTITY

END as CTRL

Analysis

CATEGORY ID

, SUM(CTRL) as CTRL , SUM(QUANTITY) AS S\_Q

> P.CATEGORY ID , SP.COUNTRY

> > THEN 1 ELSE 0

SELECT

FROM (SELECT

, COUNTRY

, CASE



GROUP BY CATEGORY ID, COUNTRY;

AND S.STORE\_TYPE=0

#### WHERE (L.CITY) IN ('CITY\_1', 'CITY\_2', 'CITY\_3')

FROM SALES SL INNER JOIN DEPARTMENTS D ON SL.DEPARTMENT\_ID=D.DEPARTMENT\_ID INNER JOIN STORES S ON D.STORE\_ID=S.STORE\_ID INNER JOIN LOCATIONS L ON S.LOCATION\_ID=L.LOCATION\_ID INNER JOIN PRODUCTS P ON SL.PRODUCT\_ID=P.PRODUCT\_ID INNER JOIN SUPPLIERS SP ON SL.SUPPLIER\_ID=SP.SUPPLIER\_ID

, QUANTITY

END as CTRL

ELSE 0

THEN 1

WHEN SP.CATEGORY='CAT\_'||P.CATEGORY\_ID

, CASE

Analysis

CATEGORY ID

, SUM(CTRL) as CTRL , SUM(QUANTITY) AS S\_Q

SELECT

FROM (SELECT

, COUNTRY

, SP.COUNTRY

**P.CATEGORY ID** 



SELECT num rows

NUM\_ROWS

100000

\_\_\_\_\_

FROM user\_tab\_statistics

WHERE table name = 'LOCATIONS';

FROM user tab col statistics

AND column name = 'CITY';

NUM\_DISTINCT NUM\_NULLS

FROM user\_tab\_histograms

WHERE column name='CITY'

98672

0

SELECT COUNT(\*)

COUNT(\*)

WHERE table name = 'LOCATIONS'

SELECT num\_distinct, num\_nulls, histogram, num\_buckets

AND endpoint\_actual\_value in ('CITY\_1','CITY\_2','CITY\_3');

0

HISTOGRAM NUM\_BUCKETS

254

HYBRID

GROUP BY CATEGORY ID, COUNTRY;

AND S.STORE TYPE=0

#### WHERE (L.CITY) IN ('CITY\_1', 'CITY\_2', 'CITY\_3')

FROM SALES SL INNER JOIN DEPARTMENTS D ON SL.DEPARTMENT\_ID=D.DEPARTMENT\_ID INNER JOIN STORES S ON D.STORE ID=S.STORE ID INNER JOIN LOCATIONS L ON S.LOCATION\_ID=L.LOCATION\_ID INNER JOIN PRODUCTS P ON SL.PRODUCT\_ID=P.PRODUCT\_ID INNER JOIN SUPPLIERS SP ON SL.SUPPLIER ID=SP.SUPPLIER ID

, QUANTITY

END as CTRL

ELSE 0

THEN 1

WHEN SP.CATEGORY='CAT\_'||P.CATEGORY\_ID

, SUM(QUANTITY) AS S Q

Analysis

CATEGORY ID

SELECT

FROM

, COUNTRY

(SELECT

**P.CATEGORY ID** 

, SP.COUNTRY

, CASE

, SUM(CTRL) as CTRL



GROUP BY CATEGORY ID, COUNTRY;

AND S.STORE\_TYPE=0

#### WHERE (L.CITY) IN ('CITY\_1', 'CITY\_2', 'CITY\_3')

SALES SL INNER JOIN DEPARTMENTS D ON SL.DEPARTMENT\_ID=D.DEPARTMENT\_ID INNER JOIN STORES S ON D.STORE\_ID=S.STORE\_ID INNER JOIN LOCATIONS L ON S.LOCATION\_ID=L.LOCATION\_ID INNER JOIN PRODUCTS P ON SL.PRODUCT\_ID=P.PRODUCT\_ID INNER JOIN SUPPLIERS SP ON SL.SUPPLIER\_ID=SP.SUPPLIER\_ID

WHEN SP.CATEGORY='CAT\_'||P.CATEGORY\_ID

Analysis: Histogram with 2048 buckets?

FROM

SELECT

FROM (SELECT

, COUNTRY

CATEGORY ID

, SUM(CTRL) as CTRL , SUM(QUANTITY) AS S\_Q

THEN 1

ELSE 0

**P.CATEGORY ID** , SP.COUNTRY

END as CTRL

, QUANTITY

, CASE



SQL> exec dbms\_stats.gather\_table\_stats(user,'LOCATIONS',

SELECT num distinct, num nulls, histogram, num buckets

AND endpoint actual value in ('CITY\_1','CITY\_2','CITY\_3');

PL/SQL procedure successfully completed.

0

FROM user tab col statistics

AND column name ='CITY';

NUM DISTINCT NUM NULLS

FROM user tab histograms

WHERE column name='CITY'

98672

0

SELECT COUNT(\*)

COUNT(\*)

WHERE table name = 'LOCATIONS'

AUTO');

\_\_\_\_\_

METHOD OPT=>'FOR COLUMNS CITY SIZE 2048 FOR ALL COLUMNS SIZE

HISTOGRAM NUM BUCKETS

2048

HYBRID

GROUP BY CATEGORY ID, COUNTRY;

AND S.STORE TYPE=0

#### WHERE (L.CITY) IN ('CITY\_1', 'CITY\_2', 'CITY\_3')

FROM SALES SL INNER JOIN DEPARTMENTS D ON SL.DEPARTMENT ID=D.DEPARTMENT ID INNER JOIN STORES S ON D.STORE\_ID=S.STORE\_ID INNER JOIN LOCATIONS L ON S.LOCATION ID=L.LOCATION ID INNER JOIN PRODUCTS P ON SL.PRODUCT ID=P.PRODUCT ID INNER JOIN SUPPLIERS SP ON SL.SUPPLIER ID=SP.SUPPLIER ID

WHEN SP.CATEGORY='CAT '||P.CATEGORY ID

Analysis: Histogram with 2048 buckets?

SELECT

FROM (SELECT

, COUNTRY

CATEGORY ID

, SUM(CTRL) as CTRL , SUM(QUANTITY) AS S Q

, CASE

END as CTRL

, SP.COUNTRY

P.CATEGORY ID

THEN 1

ELSE 0

, QUANTITY

### 

CITY is a column with > 2048 popular values Histogram on column CITY may not always be helpful

WHERE column\_name='CITY'
AND endpoint\_actual\_value in ('CITY\_1','CITY\_2','CITY\_3');

COUNT(\*)

0

AUTO');

SELECT COUNT(\*) FROM user\_tab\_histograms WHERE column name='CITY'

SELECT num\_distinct, num\_nulls, histogram, num\_buckets
FROM user\_tab\_col\_statistics
WHERE table name = 'LOCATIONS'

PL/SQL procedure successfully completed.

SQL> exec dbms\_stats.gather\_table\_stats(user,'LOCATIONS', METHOD OPT=>'FOR COLUMNS CITY SIZE 2048 FOR ALL COLUMNS SIZE

Analysis: Histogram with 2048 buckets?

#### SELECT

AND S.STORE TYPE=0

GROUP BY CATEGORY ID, COUNTRY;

CATEGORY ID , COUNTRY , SUM(CTRL) as CTRL , SUM(QUANTITY) AS S Q FROM (SELECT P.CATEGORY ID , SP.COUNTRY , CASE WHEN SP.CATEGORY='CAT '||P.CATEGORY ID THEN 1 ELSE 0 END as CTRL , QUANTITY FROM SALES SL INNER JOIN DEPARTMENTS D ON SL.DEPARTMENT ID=D.DEPARTMENT ID INNER JOIN STORES S ON D.STORE ID=S.STORE ID INNER JOIN LOCATIONS L ON S.LOCATION ID=L.LOCATION ID INNER JOIN PRODUCTS P ON SL.PRODUCT ID=P.PRODUCT ID INNER JOIN SUPPLIERS SP ON SL.SUPPLIER ID=SP.SUPPLIER ID WHERE (L.CITY) IN ('CITY\_1', 'CITY\_2', 'CITY\_3')

n		Name	Line ID	Estimated Rows	Cost Timeline(45s)	Executions	Actual Rows	Memory (Max)	Temp (Max)	Other	IO Requests	IO Bytes	Activity %
LECT STATEMENT			0			9	10						
2X COORDINATOR			1			9	10						
- PX SEND QC (RANDOM)		:TQ10006	2	71	1,173K	4	10						
HASH GROUP BY			3	71	1,173K	- 4	10	6MB					
PX RECEIVE			4	71	1,173K	- 4	40						
PX SEND HASH		:TQ10005	5	71	1,173K	4	40			<b>8</b>			
HASH GROUP BY	•		6	71	1,173K	4	40	17MB					7.18
- NESTED LOOP	S		7	571K	1,173K	4	16M						.55
NESTED LOC	0PS		8	571K	1,173K	4	16M						2.76
E-HASH JOIN	4		9	571K	856K	4	16M	1GB		<del>60</del>			6.08
E-PX RECE	IVE		10	587K	850K	4	17M						1.1
E-PX SE	ND HYBRID HASH	:TQ10003	11	587K	850K	4	17M			80			1.1
E-STA	TISTICS COLLECTOR		12		-	4	17M						
E H	ASH JOIN		13	587K	850K	4	17M	17MB		<del>60</del>			.55
8	JOIN FILTER CREATE	:BF0000	14	200	35	4	6,000			<b>80</b>			
e	PX RECEIVE		15	200	35	4	6,000						
	PX SEND BROADCAST	:TQ10002	16	200	35	4	6,000						
	HASH JOIN		17	200	35	4	1,500	17MB		<b>80</b>			
	JOIN FILTER CREATE	:BF0001	18	20	21	4	600			<u>80</u>			
	- PX RECEIVE		19	20	21	4	600						
	- PX SEND BROADCAST	:TQ10001	20	20	21	4	600						
	VIEW	VW_GBF_16	21	20	21	4	150						
	- HASH GROUP BY		22	20	21	4	150	19MB					
	- PX RECEIVE		23	20	21	4	150						
	PX SEND HASH	:TQ10000	24	20	21	4	150			<b>8</b>			
	HASH GROUP BY		25	20	21	4	150	14MB					
	- HASH JOIN		26	20	20	4	150	17MB		<b>6</b>			
	D JOIN FILTER CREATE	:BF0002	27	60	17	4	240			<b>6</b>			
	TABLE ACCESS STORAGE FULL	LOCATIONS	28	60	17 =	4	240						
	- JOIN FILTER USE	:BF0002	29	700	3	4	150			<b>#</b>			
	D JOIN FILTER USE	:BF0001	32	50K	14 =	4	1,510			<b>6</b>			
Ð	JOIN FILTER USE	:BF0000	35	147M	850K	4	20M						
- PX RECF	IVE		38	1,000K	5,820	4	1,000K						
E PX SE	ND HYBRID HASH	:TQ10004	39	1,000K	5,820	4	1,000K			<b>#</b>			
INDEX UNI	DUE SCAN	PRODUCTS PK	42	1	1	17M	16M				11K	87MB	10
TABLE ACCES	S BY INDEX ROWID	PRODUCTS	43	- 1	2	19M	16M				842	6GB	
MULL ACCES			15	1	-	1.014	2011				042		



<table-container>          n</table-container>		
	() Other IO Requests	IO Bytes Activity %
in random reference in a second reference i		
	<b>6</b> 9	
		7.18
B         B		.55
		2.76
	<b>60</b>	6.08
		1.1
	80	1.1
Image: Normal sectorImage: Normal sector <th< td=""><td><b>8</b></td><td>.55</td></th<>	<b>8</b>	.55
$ \  \  \  \  \  \  \  \  \  \  \  \  \ $	<b>80</b>	
$ \left  $		
$ \  \  \  \  \  \  \  \  \  \  \  \  \ $		
Image: service ser	<b>80</b>	
$ \left  \begin{array}{c c c c c c c c c c c c c c c c c c c $	8	
I <td< td=""><td></td><td></td></td<>		
$ \left  \begin{array}{c c c c c c c c c c c c c c c c c c c $		
$ \left[ $		
$ \left[ \begin{array}{c c c c c c c c c c c c c c c c c c c $		
$ \left  \begin{array}{c c c c c c c c c c c c c c c c c c c $		
$ \left  \begin{array}{c c c c c c c c c c c c c c c c c c c $	<b>80</b>	
$ \left  \begin{array}{c c c c c c c c c c c c c c c c c c c $		
Image:	<b>6</b> 0	
TABLE ACCESS STORAGE FULL     LOCATIONS     28     G     17     C     G     240       Image: Construction of the storage full of the storage ful	68	
Image: Construction of the co		
B. JOIN FLITER USE     BF001     32     50K     14     A     1,50     A       B. JOIN FLITER USE     BF000     32     147M     880K     A     20M     A       B. PX RECEIVE     B     1,000K     5,82     A     A     1,000K     A       B. PX RECEIVE     C     A     1,000K     5,82     A     A     1,000K       B. PX RECEIVE     C     A     1,000K     A     A     1,000K     A       B. PX RECEIVE     C     A     1,000K     A     A     A     A       LINDEX UNIQUE SCAN     PRODUCTS_PK     A4     1     A     A     A	<b>F</b> 0	
B::001 FUTE USE         iB::001 FUTE USE </td <td><b>#</b></td> <td></td>	<b>#</b>	
IP KREEEVE       TQ10004       38       1,000K       5,820       A       1,000K       1,000K       A       A       A       A       1,000K       A       A       1,000K       A		
EP-PX SRID PMBRID PASH         TQ10004         39         1,000K         5,820         4         1,000K         4         1,000K <th< td=""><td></td><td></td></th<>		
LINDEX UNIQUE SCAN     PRODUCTS_PK     42     1     17M     16M       TABLE ACCESS BY INDEX ROWID     PRODUCTS     43     1     2     19M     16M	<b>60</b>	
TABLE ACCESS BY INDEX ROWID     PRODUCTS     43     1     2     19M     16M	11K	87MB 10
	84	42K 6GB



## Dynamic statistics with higher sampling level

Je 90	03553302 Plan Note	All Parallel Servers											
			Name	Line ID	Estimated Rows	Cost Timeline(45s)	Executions	Actual Rows	Memory (Max)	Temp (Max)	Other IO Requests	IO Bytes	Activity %
ECT STATE	MENT			0			9	10					
PX COORDIN	IATOR			1									
PX SEND Q	(RANDOM)		:TQ10006	2	71	1,173K							
E HASH GR	ROUP BY			3	71	1,173K	•	Execution	on tim	e 45s			
E PX REC	CEIVE			4	71	1,173K	· · · · · · · · · · · · · · · · · · ·						
E PX S	END HASH		:TQ10005	5	71	1,173K					<del>80</del>		
E HA	SH GROUP BY			6	71	1,173K	4	40	17MB				7.18
<u> </u>	NESTED LOOPS			7	571K	1,173K	4	16M					
Θ	NESTED LOOPS			8	571K	1,173K	4	16M					
	E-HASH JOIN			9	571K	856K	4	16M			• _ •		
	PX RECEIVE			10	587K	850K	4	17M		Still NL	IOIN W	nen ioi	ning_
	PX SEND HYBRID HASH		:TQ10003	11	587K	850K -	4	17M			<b>,</b>		
	E-STATISTICS COLLECTO	2		12		-	4	17M					
	E-HASH JOIN			13	587K	850K	4	17M	17		FRODU		
	JOIN FILTER CREAT	E	:BF0000	14	200	35	4	6,000					
	PX RECEIVE			15	200	35	4	6,000					
	E PX SEND BROA	DCAST	:TQ10002	16	200	35	4	6,000					
	E HASH JOIN			17	200	35	4	1,500	17MB		<i>8</i> 8	1	
	- JOIN FILTE	R CREATE	:BF0001	18	20	21	4	600			66		
	E PX RECEI	νE		19	20	21	4	600					
	E-PX SEN	D BROADCAST	:TQ10001	20	20	21	4	600				\	
	- VIEW		VW_GBF_16	21	20	21	4	150				1	
	E HA	5H GROUP BY		22	20	21	4	150	19MB				
	E-P	X RECEIVE		23	20	21	4	150					
	ė	PX SEND HASH	:TQ10000	24	20	21	4	150			ð		
		HASH GROUP BY		25	20	21	4	150	14MB				
		E HASH JOIN		26	20	20	4	150	17MB		<b>60</b>	1	
		D JOIN FILTER CREATE	:BF0002	27	60	17	4	240			60		
		L TABLE ACCESS STORAGE FULL	LOCATIONS	28	60	17 -	4	240					
		JOIN FILTER USE	:BF0002	29	700	3	4	150			<b>60</b>	\	
	DOIN FILTE	RUSE	:BF0001	32	50K	14	4	1,510			<b>60</b>		
	JOIN FILTER USE		:BF0000	35	147M	850K	4	20M					
	PX RECEIVE			38	1,000K	5,820	4	1,000K					1
	E PX SEND HYBRID HASH		:TQ10004	39	1,000K	5,820	4	1,000K			<b>60</b>		
	INDEX UNIQUE SCAN		PRODUCTS_PK	42	1	1	17M	16M			<b>11K</b>	87MB	10
	TABLE ACCESS BY INDEX ROWIE		PRODUCTS	43	1	2	19M	16M				842K 6GB	_
													L

ATEMENT RDINATOR ND QC (RANDOM H GROUP BY ( RECEIVE PX SEND HASH 3- HASH GROUP E	)	Name	Line ID 0	Estimated Rows	Cost Timeline(45s)	Executions	Actual Rows	Memory (Max) 1	emp (Max) Other	IO Requests	IO Bytes	Activity %
TATEMENT RDINATOR ND QC (RANDOM SH GROUP BY K RECEIVE PX SEND HASH HASH GROUP E FH-NESTED LOO	)	:TO10006	0									
IRDINATOR END QC (RANDOM SH GROUP BY X RECEIVE PX SEND HASH HASH GROUP E HASH GROUP E	)	:TO10006				· •	10					
ASH GROUP BY PX RECEIVE PX SEND HASH PASH GROUP E PASH GROUP E	)	:TO10006	1									
PX RECEIVE     PX SEND HASH     PX SEND HASH     HASH GROUP E     HASH GROUP E			2	71	1,173K		Evoqueti	an tima 10				
PX RECEIVE     PX SEND HASH     HASH GROUP E     HASH GROUP E			3	71	1,173K		Execution	on ume 4:	5			
PX SEND HASH     HASH GROUP E     HASH GROUP E			4	71	1,173K							
HASH GROUP E		:TQ10005	5	71	1,173K				66			
- NESTED LOO	IY		6	71	1,173K	4	40	17MB				7.18
	PS		7	571K	1,173K	4	16M					
- NESTED LC	0095		8	571K	1,173K	4	16M					
E HASH JO	IN		9	571K	856K	4	16M	CTI		ملير من	an ini	ain a
E PX REC	EIVE	2010000	10	587K	850K	4	17M	Stil		<u>m wn</u>	<u>en joir</u>	πng_
⊟-PX SI	END HYBRID HASH	:TQ10003	11	587K	850K	4	17M		-		-	-
E-ST)	ALISTICS COLLECTOR		12		-	4	17M		PF		$\gamma T \gamma$	
⊡-+			13	58/K		4	1/M	1,		0000	-15	
8		:BF0000	14	200	35	4	6,000					
		704000	15	200	35 -	4	6,000					
	E-PX SEND BROADCAST	: (Q10002	16	200	35 -	4	6,000		_			
		BERRAR	17	200	35 -	4	1,500	Now	<u>v w</u> e h	ave a	ccurate	2
		:RF0001	18	20	21 -	4	600					
		7010001	19	20	21 -	4	600	cipa	la tab	lo core	linality	,
	E-PA SEND BROADCAST	:/Q10001	20	20	21	4	600	Sing	ie tabi	e carc	manty	
		vW_GBF_16	21	20	21 -	4	150					
	HASH GROUP BY		22	20	21 -	4	150	estin	<u>nate fo</u>	or I OC	ATION	S
			23	20	21 -	4	150	Cotin				
	E PX SEND HASH	:1Q10000	24	20	21	4	150	1410				
			25	20	21 -	4	150	14MB				
		-BE0000	20	20	20	4	150	TIMB	<b>60</b>		\	
	TABLE ACCESS STORAGE SHILL	LOCATIONS	2/	60	17	4	240		69			
	TABLE ACCESS STORAGE FULL	LOCATIONS	28	60	1/	4	240					
		:DF0002	29	700	3	4	150		<b>00</b>		1	
		-BE0000	32	30K	1-1 850K	4	2014		66		1	
	EV/F	.510000	33	1 0004	5.820	4	1 000K					1
E PA REC		·TO10004	38	1,000K	5,020	4	1,000K		, <del>25</del> ,			
	TO LE SCAN	PRODUCTS PK	39	1,000K	1	17M	1,0004		66	11K	87MB	10
TADLE ACCO		PRODUCTS_PR	42	1	1	17M	10M			110		10
- TABLE ACCE	35 DT TINDEY KOMID	PRODUCTS	43	1	2	19M	16M				042N OGB	



Plan Stat	tistics 🔆 Plan 👸 Parallel 📐 Activity 🔀 Metrics alue 90355302 📄 Plan Note 🛛 All Parallel Servers	×	But join	cardinality is still			
Operation Name		off, hence NL join		Executions Actual Rows	Memory (Max) Temp (Max) Other IO Req	Jests IO Bytes Activity %	
🍦 🖻 SELECT	T STATEMENT		- /	J-	9 10		
🍦 🖻 PX O	COORDINATOR						
🚳 🖻 PX	(SEND QC (RANDOM)	:TQ10006			Execut	ion time 1Ec	
	HASH GROUP BY		3	71 1,173K	Execut	Ion lime 455	
<u>ଲ</u> ମ୍ବର ଅନ	- PX RECEIVE	.TO1000F	4	71 1,173K			
908 848		:1010005	5	1 1,173K	4 40	17MP	7.19
4.04 8:8	- NESTED LOOPS		7	5714 1 1734	4 16M	1740	7.10
88	NESTED LOOPS		8	571K 1.173K	4 16M		
355 255	- HASH JOIN		9	571K 856K	4 16M		
856 856	- PX RECEIVE		10	587K 850K	4 17M	Still NI ioin	when ining
<b>35</b>	PX SEND HYBRID HASH	:TQ10003	11	587K 850K -	4 17M	JUILINE	
85	- STATISTICS COLLECTOR		12		4 17M		
<b>65</b>	- HASH JOIN		13	587K 850K -	4 17M	I PROL	JUCIS
8 <b>6</b> 9	- JOIN FILTER CREATE	:BF0000	14	200 35	4 6,000		
8 <b>6</b> 8	- PX RECEIVE		15	200 35	4 6,000		
860 C	E-PX SEND BROADCAST	:TQ10002	16	200 35	4 6,000		
666 (	- HASH JOIN		17	200 35	4 1,500	Now we have	e accurate
859 1	JOIN FILTER CREATE	:BF0001	18	20 21	4 600		
839 1	-PX RECEIVE		19	20 21	4 600	cinale table a	ardinality
80 10	- PX SEND BROADCAST	:TQ10001	20	20 21	4 600	single table (	arumanty
200 200		VW_GBF_16	21	20 21 -	4 150		
400 818			22	20 21 -	4 150	estimate for I	OCATIONS
98 88		·TO10000	23	20 21	4 150		
4047 838		.101000	25	20 21	4 150	14MB	
404 856	ET-HASH JOIN		25	20 20	4 150	17MB #	
85	E-JOIN FILTER CREATE	:BF0002	27	60 17	4 240		
86	TABLE ACCESS STORAGE FULL	LOCATIONS	28	60 17 =	4 240		
8 <u>5</u> 6	B-JOIN FILTER USE	:BF0002	29	700 3 -	4 150	80	
836	E-JOIN FILTER USE	:BF0001	32	50K 14	4 1,510	æ	
8 <b>6</b> 9	E JOIN FILTER USE	:BF0000	35	147M 850K -	4 20M		
8 <u>5</u> 6	- PX RECEIVE		38	1,000K 5,820	4 1,000K		
8 <b>6</b> 0	E PX SEND HYBRID HASH	:TQ10004	39	1,000K 5,820	4 1,000K	<b>(h)</b>	
8 <b>6</b> 9	INDEX UNIQUE SCAN	PRODUCTS_PK	42	1 1	17M 16M	11K	87MB 10
TABLE ACCESS BY INDEX ROWID		PRODUCTS	43	1 2	19M 16M		842K 6GB
							L



Plan Hash Value 903553302 Plan Note All Parallel Servers													
Operation Name		off. hence NL ioin		Executions	Actual Rows	Memory (Max)	Temp (Max)	Other	IO Requests	IO Bytes	Activity %		
E-SELECT STATEMENT		,		<b>J</b>	9	10							
E PX COORDINATOR													
EPX SEND QC (RANDOM) :TQ10006						Evoquti	on time						
HASH GROUP BY		3	71	1,173K		Executio	JH LIIII	2 4 3 5					
	-TO10005	4	71	1,173K									
	.1010005	6	V1	1.173K	4	40	17MB		00			7.18	
E-NESTED LOOPS		7	571	1,173K	- 4	16M							
NESTED LOOPS		8	571K	1,173K	4	16M							
HASH JOIN		9	571K	856K	4	16M							
E-PX RECEIVE		10	587K	850K -	4	17M		Still NI	io	in whe	<u>en ioi</u>	ning	
PX SEND HYBRID HASH	:TQ10003	11	587K	850K	4	17M			- , 0				
STATISTICS COLLESTOR		12			4	17M			DD		ידכ		
HASH JOIN		13	587K	850K	4	17M	17		FN		13		
DI JOIN FILTER CREATE	:BF0000	14	200	35	4	6,000							
	-TO10003	15	200	35	4	6,000							
	:1010002	10	200	35	4	1,500			- I		<b>-</b>		
	:BF0001	18	20	21	4	600		NOM M6	e na	ave ac	curat	e	
PX RECEIVE		19	20	21	4	600							
	C PLOTED PROJECT		20 20 21 4 600					single table cardinality					
Now we end up with		21	20	21	4	150							
			20 21	21	4	150		stimato for IOCATIO			ΛΤΙΩΝ		
	HYBRID HASH		20	21	4	150		sumate				13	
			20	21	4	150							
DISTRIBUTION instead of		25	20	21	4	150	14MB						
		20	20	17	4	150	17MB		80 (A)				
		27	60	17	4	240			88		\		
BROA	ADCAST	29	700	3 -	4	150			<b>#</b>				
JOIN FILTER USE	:BF0001	32	50K	14	4	1,510			80 80				
	:BF0000	35	147M	850К —	4	20M							
- PX RECEIVE		38	1,000K	5,820	4	1,000K							
E PX SEND HYBRID HASH	:TQ10004	39	1,000K	5,820	4	1,000K			8				
INDEX UNIQUE SCAN	PRODUCTS_PK	42	1	1	17M	16M				11K	87MB	10	
TABLE ACCESS BY INDEX ROWID	PRODUCTS	43	1	2	19M	16M					42K 🔒 6GB		_







GROUP BY CATEGORY ID, COUNTRY;

INNER JOIN DEPARTMENTS D ON SL.DEPARTMENT\_ID=D.DEPARTMENT\_ID INNER JOIN STORES S ON D.STORE\_ID=S.STORE\_ID INNER JOIN LOCATIONS L ON S.LOCATION ID=L.LOCATION ID INNER JOIN PRODUCTS P ON SL.PRODUCT\_ID=P.PRODUCT\_ID INNER JOIN SUPPLIERS SP ON SL.SUPPLIER ID=SP.SUPPLIER ID WHERE (L.CITY) IN ('CITY 1', 'CITY 2', 'CITY 3') AND S.STORE TYPE=0

WHEN SP.CATEGORY='CAT '||P.CATEGORY ID

FROM (SELECT

P.CATEGORY ID

, SP.COUNTRY

, CASE

, SUM(CTRL) as CTRL , SUM(QUANTITY) AS S Q

THEN 1

ELSE 0

FROM

SELECT

, COUNTRY

CATEGORY ID

END as CTRL , QUANTITY SALES SL

Analysis – Reason for join cardinality misestimate between STORES and LOCATIONS



GROUP BY CATEGORY ID, COUNTRY;

#### AND S.STORE\_TYPE=0

#### WHERE (L.CITY) IN ('CITY\_1', 'CITY\_2', 'CITY\_3')

INNER JOIN PRODUCTS P ON SL.PRODUCT\_ID=P.PRODUCT\_ID INNER JOIN SUPPLIERS SP ON SL.SUPPLIER ID=SP.SUPPLIER ID

#### INNER JOIN LOCATIONS L ON S.LOCATION\_ID=L.LOCATION\_ID

SALES SL INNER JOIN DEPARTMENTS D ON SL.DEPARTMENT\_ID=D.DEPARTMENT\_ID INNER JOIN STORES S ON D.STORE ID=S.STORE ID

WHEN SP.CATEGORY='CAT '||P.CATEGORY ID

Analysis – Reason for join cardinality misestimate between STORES and LOCATIONS

FROM

SELECT

FROM (SELECT

, COUNTRY

CATEGORY ID

, SUM(CTRL) as CTRL , SUM(QUANTITY) AS S Q

, CASE

END as CTRL

THEN 1 ELSE 0

, QUANTITY

**P.CATEGORY ID** , SP.COUNTRY

SELECT CATEGORY ID , COUNTRY , SUM(CTRL) as CTRL , SUM(QUANTITY) AS S Q FROM (SELECT **P.CATEGORY ID** , SP.COUNTRY , CASE WHEN SP.CATEGORY='CAT '||P.CATEGORY ID THEN 1 ELSE 0 END as CTRL , QUANTITY FROM SALES SL INNER JOIN DEPARTMENTS D ON SL.DEPARTMENT ID=D.DEPARTMENT ID INNER JOIN STORES S ON D.STORE ID=S.STORE ID INNER JOIN LOCATIONS L ON S.LOCATION\_ID=L.LOCATION\_ID INNER JOIN PRODUCTS P ON SL.PRODUCT ID=P.PRODUCT ID INNER JOIN SUPPLIERS SP ON SL.SUPPLIER ID=SP.SUPPLIER ID WHERE (L.CITY) IN ('CITY\_1', 'CITY\_2', 'CITY\_3') AND S.STORE\_TYPE=0 GROUP BY CATEGORY ID, COUNTRY;

SELECT num rows





HISTOGRAM NUM BUCKETS

HYBRID

About than 2 % of the LOCATIONS

used in STORES

254

GROUP BY CATEGORY ID, COUNTRY;

#### AND S.STORE\_TYPE=0

SELECT

FROM

(SELECT

, COUNTRY

CATEGORY ID

, SUM(CTRL) as CTRL

, SUM(QUANTITY) AS S Q

WHERE (L.CITY) IN ('CITY\_1', 'CITY\_2', 'CITY\_3')

ELSE 0

INNER JOIN PRODUCTS P ON SL.PRODUCT ID=P.PRODUCT ID INNER JOIN SUPPLIERS SP ON SL.SUPPLIER ID=SP.SUPPLIER ID

INNER JOIN LOCATIONS L ON S.LOCATION\_ID=L.LOCATION\_ID

SALES SL INNER JOIN DEPARTMENTS D ON SL.DEPARTMENT ID=D.DEPARTMENT ID INNER JOIN STORES S ON D.STORE ID=S.STORE ID

END as CTRL , QUANTITY FROM

P.CATEGORY ID , SP.COUNTRY , CASE WHEN SP.CATEGORY='CAT '||P.CATEGORY ID THEN 1

\_\_\_\_\_ 5000 SELECT num distinct, num nulls, histogram, num buckets FROM user tab col statistics

WHERE table name = 'STORES'

NUM DISTINCT NUM NULLS

2080

AND column name = 'LOCATION ID';

0

NUM ROWS

SELECT num rows FROM user tab statistics WHERE table name = 'STORES';

Analysis – Reason for join cardinality misestimate between STORES and LOCATIONS

SELECT CATEGORY ID , COUNTRY , SUM(CTRL) as CTRL , SUM(QUANTITY) AS S Q FROM (SELECT P.CATEGORY ID , SP.COUNTRY , CASE WHEN SP.CATEGORY='CAT '||P.CATEGORY ID THEN 1 ELSE 0 END as CTRL , QUANTITY FROM SALES SL INNER JOIN DEPARTMENTS D ON SL.DEPARTMENT ID=D.DEPARTMENT ID INNER JOIN STORES S ON D.STORE ID=S.STORE ID INNER JOIN LOCATIONS L ON S.LOCATION\_ID=L.LOCATION\_ID INNER JOIN PRODUCTS P ON SL.PRODUCT ID=P.PRODUCT ID INNER JOIN SUPPLIERS SP ON SL.SUPPLIER ID=SP.SUPPLIER ID WHERE (L.CITY) IN ('CITY\_1', 'CITY\_2', 'CITY\_3') -AND S.STORE\_TYPE=0 GROUP BY CATEGORY ID, COUNTRY; Row filtering on STORES and LOCATIONS





SELECT num rows SELECT FROM user tab statistics CATEGORY ID WHERE table name = 'STORES'; , COUNTRY , SUM(CTRL) as CTRL NUM ROWS , SUM(QUANTITY) AS S Q \_\_\_\_\_ FROM 5000 (SELECT **P.CATEGORY ID** , SP.COUNTRY SELECT num distinct, num nulls, histogram, num buckets , CASE FROM user tab col statistics WHEN SP.CATEGORY='CAT\_'||P.CATEGORY ID WHERE table name = 'STORES' THEN 1 AND column name = 'LOCATION ID'; ELSE 0 END as CTRL NUM DISTINCT NUM NULLS HISTOGRAM NUM BUCKETS , QUANTITY FROM 2080 HYBRID 0 SALES SL INNER JOIN DEPARTMENTS D ON SL.DEPARTMENT ID=D.DEPARTMENT ID INNER JOIN STORES S ON D.STORE ID=S.STORE ID INNER JOIN LOCATIONS L ON S.LOCATION\_ID=L.LOCATION\_ID INNER JOIN PRODUCTS P ON SL.PRODUCT ID=P.PRODUCT ID About than 2 % of the LOCATIONS INNER JOIN SUPPLIERS SP ON SL.SUPPLIER ID=SP.SUPPLIER ID WHERE (L.CITY) IN ('CITY\_1','CITY\_2','CITY\_3' used in STORES AND S.STORE TYPE=0 GROUP BY CATEGORY ID COUNTRY; Row filtering on STORES and LOCATIONS Possible correlation between L.CITY AND S.STORE\_TYPE





# End Demo1



**Analysis: For the query in question** 

- Standard statistics are not sufficient
- Due to data distribution, histograms are not effective either
- Dynamic statistics (sampling)

Could fix single table cardinality estimates

Due to data model, data characteristics and data distribution, join cardinality is still inaccurate











LOCATIONS





















///







///
### **Baseline: Join order and join method**





///X

### **Baseline: Join order and join method**





































#### **Schema and SQL Statement**

#### SELECT

CATEGORY ID

, COUNTRY

, SUM(CTRL) as CTRL

, SUM(QUANTITY) AS S\_Q

#### FROM

#### (SELECT

P.CATEGORY\_ID

```
, SP.COUNTRY
```

, CASE

```
WHEN SP.CATEGORY='CAT_'||P.CATEGORY_ID
```

THEN 1

ELSE 0

END as CTRL

, QUANTITY

#### FROM

SALES SL

```
INNER JOIN DEPARTMENTS_DIM D ON SL.DEPARTMENT_ID=D.DEPARTMENT_ID
INNER JOIN PRODUCTS P ON SL.PRODUCT_ID=P.PRODUCT_ID
INNER JOIN SUPPLIERS SP ON SL.SUPPLIER_ID=SP.SUPPLIER_ID
WHERE (D.CITY) IN ('CITY_1','CITY_2','CITY_3')
AND D.STORE_TYPE=0
)
```

GROUP BY CATEGORY\_ID, COUNTRY





# Demo



Copyright © 2019 Oracle and/or its affiliates.

peration		Name	Line ID	Estimated Rows	Cost	Timeline(5s)	Executions	Actual Rows	Memory (Max)	Temp (Max)	Other	IO Requests	IO Bytes	Activity %
-SELECT ST	TEMENT		0				9	10						
PX COOR	DINATOR		1			_	9	10				2	16KB	
PX SEN	D QC (RANDOM)	:TQ10004	2	71	1,198K		4	10						
- HASH	GROUP BY		3	71	1,198K		4	10	6MB					
E PX	RECEIVE		4	71	1,198K		4	40						
E P	K SEND HASH	:TQ10003	5	71	1,198K		4	40			<u> 8</u>			
ė.	HASH GROUP BY		6	71	1,198K		4	40	17MB					16
ſ	- HASH JOIN		7	16M	1,198K		4	16M	964MB		<u> 66</u>			
	PX RECEIVE		8	20M	343K		4	20M						4
	- PX SEND HYBRID HASH	:TQ10001	9	20M	343K	—	4	20M			<u> 8</u>			
	- STATISTICS COLLECTOR		10				4	20M						
	- PX BLOCK ITERATOR		11	20M	343K	—	4	20M						
	- TABLE ACCESS STORAGE FULL	PRODUCTS	12	20M	343K		55	20M	48MB		<u> 8</u>	<b>—</b> 94K	<b>31GB</b>	16
			13	16M	856K		4	16M						
	- PX SEND HYBRID HASH	:TQ10002	14	16M	856K		4	16M			<u> 8</u>			8
	HASH JOIN		15	16M	856K		4	16M	277MB		<u> 66</u>			4
			16	1,000K	5,820	_	4	4,000K						
	- PX SEND BROADCAST	:TQ10000	17	1,000K	5,820	—	4	4,000K						
	- PX BLOCK ITERATOR		18	1,000K	5,820	_	4	1,000K						
	TABLE ACCESS STORAGE FULL	SUPPLIERS	19	1,000K	5,820	—	54	1,000K	17MB		<u> 66</u>	1,844	601MB	
			20	16M	850K		4	17M	17MB		<u> 8</u>			8
		:BF0000	21	1,500	8	_	4	6,000			<u> 8</u>			
	TABLE ACCESS STORAGE FULL	DEPARTMENTS_DIM	22	1,500	8		4	6,000						
		:BF0000	23	147M	850K		4	19M						
			24	147M	850K		4	19M						
	TABLE ACCESS STORAGE FULL	SALES	25	147M	850K		91	19M	48MB		<u> </u>	265K	<b>86GB</b>	12



eration	Name	Line ID	Estimated Rows	Cost	Timeline(5s)	Executions	Actual Rows	Memory (Max)	Temp (Max)	Other	IO Requests	IO Bytes	Activity %
- SELECT STATEMENT		0				9	10						
PX COORDINATOR		1			_	9	10				2	16KB	
- PX SEND QC (RANDOM)	:TQ10004	2	71	1,198K		4	10						
		3	71	1,198K		4	10	Evoc	ution	tim	10 5 c		
- PX RECEIVE		4	71	1,198K		4	40		ution	UIII	16 72		
- PX SEND HASH	:TQ10003	5	71	1,198K		4	40						
HASH GROUP BY		6	71	1,198K		4	40	17MB					16
- HASH JOIN		7	16M	1,198K		4	16M	964MB		<u> </u>			
		8	20M	343K		4	20M						4
- PX SEND HYBRID HASH	:TQ10001	9	20M	343K	_	4	20M			<u> </u>			
- STATISTICS COLLECTOR		10			_	4	20M						
- PX BLOCK ITERATOR		11	20M	343K	—	4	20M						
TABLE ACCESS STORAGE FULL	PRODUCTS	12	20M	343K		55	20M	48MB		<u> 8</u>	<b>—</b> 94K	<b>_</b> 31GB	16
		13	16M	856K		4	16M						
	:TQ10002	14	16M	856K		4	16M			<u> 8</u>			8
-HASH JOIN		15	16M	856K		4	16M	277MB		<u> 66</u>			4
		16	1,000K	5,820	—	4	4,000K						
- PX SEND BROADCAST	:TQ10000	17	1,000K	5,820	—	4	4,000K						
		18	1,000K	5,820		4	1,000K						
TABLE ACCESS STORAGE FULL	SUPPLIERS	19	1,000K	5,820	_	54	1,000K	17MB		<u> 8</u>	1,844	601MB	
		20	16M	850K		4	17M	17MB		<u> 8</u>			8
- JOIN FILTER CREATE	:BF0000	21	1,500	8	_	4	6,000			<u> 8</u>			
TABLE ACCESS STORAGE FULL	DEPARTMENTS_DIM	22	1,500	8		4	6,000						
	:BF0000	23	147M	850K		4	19M						
		24	147M	850K		4	19M						
TABLE ACCESS STORAGE FULL	SALES	25	147M	850K		91	19M	48MB		<b>60</b>	265K	<b>86GB</b>	12



eratio	n	Name	Line ID	Estimated Rows	Cost	Timeline(5s)	Executions	Actual Rows	Memory (Max)	Temp (Max)	Other	IO Requests	IO Bytes	Activity %
🖃 SE	LECT STATEMENT		0			_	9	10					_	_
	PX COORDINATOR		1				9	10				2	16KB	
È	- PX SEND QC (RANDOM)	:TQ10004	2	71	1,198K		4	10						
			3	71	1,198K		4	10	Evec	ution	tim	10 Sc		
			4	71	1,198K		4	40	LACC	ution	um	16 22		
	PX SEND HASH	:TQ10003	5	71	1,198K		4	40						
ł	HASH GROUP BY		6	71	1,198K		4	40	17MB					16
			7	16M	1,198K		4	16M	964MB		<u> </u>			_
			8	20M	343K		4	20M						4
	- PX SEND HYBRID HASH	:TQ10001	9	20M	343K	_	4	20M			<b>60</b>			
			10											
	- PX BLOCK ITERATOR		11	20M		Corroct	ingla tak							
	TABLE ACCESS STORAGE FULL	PRODUCTS	12	20M		CONECTS	ingle tai	JIE	48MB		<u> 66</u>	<u>94K</u>	<b>31GB</b>	16
			13	16M		cardinali	vactim	ato						
	PX SEND HYBRID HASH	:TQ10002	14	16M		Caruman	y estina	ale			<u> 8</u>			8
			15	16M					277MB		<u> 8</u>			4
	- PX RECEIVE		16	1,000K	5,820		4	4,000K						
	- PX SEND BROADCAST	:TQ10000	17	1,000K	5,820		4	4,000K						
			18	1,000K	5,820		4	1,000K						
	TABLE ACCESS STORAGE FULL	SUPPLIERS	19	1,000K	5,820	/-	54	1,000K	17MB		<u> 8</u>	1,844	601MB	
	- HASH JOIN		20	16M	850K		4	17M	17MB		<del>60</del>			8
	- JOIN FILTER CREATE	:BF0000	21	1,500	8		4	6,000	_		<del>60</del>			
	TABLE ACCESS STORAGE FULL	DEPARTMENTS_DIM	22	1,500	8		4	6,000						
		:BF0000	23	147M	850K		4	19M						
			24	147M	850K		4	19M						
1	TABLE ACCESS STORAGE FULL	SALES	25	147M	850K		91	19M	48MB		<u> 66</u>	265K	<b>86GB</b>	12



Opera	ation			D	Estimated Rows	Cost	Timeline(5s)	Executions	Actual Rows	Memory (Max)	Temp (Max)	Other	IO Requests	IO Bytes	Activity %
Θ	SELECT STATEMENT	Plan Note		× 0				9	10						
E	- PX COORDINATOR			1			_	9	10				2	16KB	
	PX SEND QC (RANDON			2	71	1,198K	_	4	10						
	- HASH GROUP BY	Dynamic Sampling Level : (	5	3	71	1,198K	_	4	10	Evoc	ution	tim			
	- PX RECEIVE			4	71	1,198K	_	4	40		ution	un	ie JS		
	- PX SEND HASH		A OK	5	71	1,198K	_	4	40						
	HASH GROUP		W UK	6	71	1,198K		4	40	17MB					16
	HASH JOIN			7	16M	1,198K		4	16M	964MB		<b>6</b>			
	E-PX RECEIV	E		8	20M	343K		4	20M						4
		HYBRID HASH	:TQ10001	9	20M	343K		4	20M			<u> 8</u>			
		STICS COLLECTOR		10											
	E PX B	LOCK ITERATOR		11	20M		Correct ci	nala ta	hla						
	TAE	BLE ACCESS STORAGE FULL	PRODUCTS	12	20M		COLLECT SI	ingle la	DIE	48MB		<b>6</b>	<b>—</b> 94K	<b>31GB</b>	16
	- PX RECEIV	E		13	16M		cardinality	, estim	ate						
		HYBRID HASH	:TQ10002	14	16M		caramant		all			66			8
	- HASH :	JOIN		15	16M					277MB		60			4
	⊟ PX R	ECEIVE		16	1,000K	5,820		4	4,000K						
	Ė−PX	SEND BROADCAST	:TQ10000	17	1,000K	5,820		4	4,000K						
	ė-I	PX BLOCK ITERATOR		18	1,000K	5,820		4	1,000K						
		TABLE ACCESS STORAGE FULL	SUPPLIERS	19	1,000K	5,820		54	1,000K	17MB		60	1,844	601MB	
	⊟- HASI	NIOL H		20	16M	850K		4	17M	17MB		<b>6</b> 0			8
	Or−⊟		:BF0000	21	1,500	8		4	6,000	1		66			
	-1,	ABLE ACCESS STORAGE FULL	DEPARTMENTS_DIM	22	1,500	8		4	6,000						
	01-⊡		:BF0000	23	147M	850K		4	19M						
			CALEC	24	14/M	850K		4	19M	401/2			2651	0000	12
	L	TABLE AUCESS STORAGE FULL	SALES	25	147M	850K		91	19M	48MB		66	265K	86GB	12



Plan S	Statistics 😤 Plan 🖓 Pa	arallel 📐 Activity 🖳 Met	rics												
Plan Has	h Value 2473962328	an Note All Parallel Servers	•												
Operation	Dian	Nata	1	D	Estimated Rows	Cost	Timeline(5s)	Executions	Actual Rows	Memory (Max)	Temp (Max)	Other	IO Requests	IO Bytes	Activity %
) 🖂 SEL	ECT STATEMENT	Note		0				9	10						
⊟-P	X COORDINATOR		_	1				9	10				2	16KB	
	PX SEND QC (RANDON	amic Sampling Level : 6		2	71	1,198K		4	10						
[	HASH GROUP BY	amic sampling Level . c		3	71	1,198K		4	10	Fyer	ution t	tim	10 5 s		
	PX RECEIVE		$\sim$	4	71	1 1094		4	40	LACC					
	- PX SEND HASH			5 K			Dvnamic s	tatisti	c used			00			
	- HASH GROUP			6					0.000.						16
	E- HASH JOIN			7	Wh	at if	<sup>-</sup> dvnamic st	atistic	s coulc	ln't hel	p?	<u> 6</u>			
	PX RECEIVE			8											4
	PX SEND HYBRID	HASH	:TQ10001	9	20M	343K		4	20M			<u> 8</u>			
	STATISTICS COL	LECTOR		10											
	D PX BLOCK ITE	RATOR		11	20M		Correct sin	iøle ta	ble						
		S STORAGE FULL	PRODUCTS	12	20M			910 10		48MB		66	<b>94</b> K	<b>31GB</b>	16
				13	16M		cardinality	estim	ate						
		HASH	:1Q10002	14	16M					077140		<b>00</b>			8
				15	16M					277MB		99			4
		OADCAST	·TO10000	10	1,000K	5,820	_/	4	4,000K						
		ITERATOR	.1010000	17	1,000K	5,620		4	4,000K						
		CESS STORAGE FULL	SUPPLIERS	19	1,000K	5,820	_	-+	1,000K	17MR		<b>#</b>	1.844	601MB	
			55.7 LILIO	20	16M	850		4	17M	17MB		60 68	1 2/3 11	301110	8
		CREATE	:BF0000	21	1,500	8	—	4	6,000	17110					
	TABLE ACC	ESS STORAGE FULL	DEPARTMENTS_DIM	22	1,500	8	_	4	6,000	1		00			
		USE	:BF0000	23	147M	850K		4	19M						
		ITERATOR		24	147M	850K	_	4	19M						
	TABLE AC	CESS STORAGE FULL	SALES	25	147M	850K		91	19M	48MB		<b>8</b>	265K	<b>86GB</b>	12



### Denormalize the join between LOCATIONS, STORES and DEPARTMENTS

#### 📄 Plan Statistics 💏 Plan 🎆 Parallel 📐 Activity 🖳 Metrics

Pla	n Hash Value 38	33059184 Plan Note													
Ор	eration		Name	Line ID	Estimated Rows	Cost	Timeline(81s)	Executions	Actual Rows	Memory (Max)	Temp (Max)	Other	IO Requests	IO Bytes	Activity %
<b>0</b>	SELECT STATEM	MENT		0				9	10						
0	PX COORDINA	ATOR		1				9	10						
6 <u>6</u> 6	PX SEND QC	C (RANDOM)	:TQ10002	2	71	861K		4	10						
<b>66</b> 6	HASH GRO	COUP BY		3	71	861K		4	10	6MB					
<b>7</b> 5	PX REC	CEIVE		4	71	861K		4	40						
2,6	E PX SE	END HASH	:TQ10001	5	71	861K		4	40			æ			
222	E HAS	SH GROUP BY		6	71	861K		4	40	17MB					3.14
2 <u>7</u> 6	ė N	NESTED LOOPS		7	9,091	861K		4	16M						.31
26	8	- NESTED LOOPS		8	9,091	861K		4	16M						
276 1	E	E-HASH JOIN		9	9,091	856K		4	16M	4GB	788MB	æ	3,152	2GB	4.72
256		E PX RECEIVE		10	9,337	850K	-	4	66M						.63
<b>3</b> 5		E-PX SEND BROADCAST	:TQ10000	11	9,337	850K		4	66M						.63
255		- HASH JOIN		12	9,337	850K		4	17M	17MB		æ			
<b>3</b> 5		- JOIN FILTER CREATE	:BF0000	13	1	8	1	4	6,000			æ			
250		- TABLE ACCESS STORAGE FULL	DEPARTMENTS_DIM	14	1	8		4	6,000						
6		- JOIN FILTER USE	:BF0000	15	147M	850K		4	19M						
8		- PX BLOCK ITERATOR		16	147M	850K		4	19M						
8		TABLE ACCESS STORAGE FULL	SALES	17	147M	850K		91	19M	48MB		æ	265K	<b>86GB</b>	1.89
260 260		- PX BLOCK ITERATOR		18	1,000K	5,820		4	1,000K						
35 C		TABLE ACCESS STORAGE FULL	SUPPLIERS	19	1,000K	5,820		54	1,000K	19MB		<del>60</del>	1,844	601MB	.63
255		-INDEX UNIQUE SCAN	PRODUCTS_PK	20	1	1		19M	16M				4,849	K 🗕 37GB	46
350 1	L	TABLE ACCESS BY INDEX ROWID	PRODUCTS	21	1	2		24M	16M				<b>921K</b>	7GB	42



	Plan St	atistics	-	Plan 👪 Parallel 📐 Activity 🗾	Metrics												
Pla	an Hash	Value 38	33059	184 📄 Plan Note													
Ор	eration				Name	Line ID	Estimated Rows	Cost	Timeline(81s)	Executions	Actual Rows	Memory (Max)	Temp (Max)	Other	IO Requests	IO Bytes	Activity %
ŵ		CT STATE	MENT			0				9	10						
0	E PX		ATOR			1				9	10						
<b>65</b>	<u> </u>	PX SEND Q	C (RA	NDOM)	:TQ10002	2	71	861K		4	10						
<b>77</b>	Ē	HASH GR	ROUP I	BY		3	71	861K		4	10						
<b>66</b>		PX REC	EIVE			4	71	861K		4	40	Exec	ution t	tim	le 81s		
26		E-PX S	END H	IASH	:TQ10001	5	71	861K		4	40						
<b>3</b> 6		E-HA	SH GF	ROUP BY		6	71	861K		4	40	17MB					3.14
26		ė-1	NESTE	D LOOPS		7	9,091	861K		4	16M						.31
26		Ē	NES	TED LOOPS		8	9,091	861K		4	16M						
220			⊟ HA	ASH JOIN		9	9,091	856K		4	16M	4GB	788MB	<u> </u>	3,152	2GB	4.72
256				PX RECEIVE		10	9,337	850K	-	4	66M						.63
26			Ē	PX SEND BROADCAST	:TQ10000	11	9,337	850K		4	66M						.63
<b>7</b> 5				- HASH JOIN		12	9,337	850K		4	17M	17MB		<u> </u>			
25				JOIN FILTER CREATE	:BF0000	13	1	8	1	4	6,000			66			
8				TABLE ACCESS STORAGE FULL	DEPARTMENTS_DIM	14	1	8	1	4	6,000						
<b>7</b> 5					:BF0000	15	147M	850K		4	19M						
35						16	147M	850K		4	19M						
26				TABLE ACCESS STORAGE FULL	SALES	17	147M	850K		91	19M	48MB		<u> 66</u>	265K	86GB	1.89
<b>7</b> 5				PX BLOCK ITERATOR		18	1,000K	5,820		4	1,000K						
22				TABLE ACCESS STORAGE FULL	SUPPLIERS	19	1,000K	5,820		54	1,000K	19MB		8	1,844	601MB	.63
26				DEX UNIQUE SCAN	PRODUCTS_PK	20	1	1		19M	16M				4,849K	<b>37GB</b>	46
220			TABL	E ACCESS BY INDEX ROWID	PRODUCTS	21	1	2		24M	16M				<mark></mark> 921K	7GB	42



SELECT STATEMENT  SELECT STATEMENT  SELECT STATEMENT  PX COORDINATOR  PX SEND QC (RANDOM)  PX SEND QC (RANDOM)  PX RECEIVE  PX RECEIVE  PX SEND HASH  PX SEND HASH  PX SEND HASH  PX SEND LOOPS  PX SEND LOOPS  PX RECEIVE  PX RECEIVE PX RECEIVE PX RECEIVE PX RECEIVE PX RECEIVE PX REC	E BROADCAST	:TQ10002 :TQ10001	00 11 23 44 55 66 77 88 9	71 71 71 71 71 71 9,091 9,091	861K	Single table car underestim	° rdina	10 10 10	Execu	ution ti	me	e 81s		
PX COORDINATOR      PX SEND QC (RANDOM)      HASH GROUP BY      PX RECEIVE      PX SEND HASH      PX SEND	E BROADCAST	:TQ10002 :TQ10001	1 2 3 4 5 6 7 7 8 8 9	71 71 71 71 71 9,091 9,091	861K	Single table car underestim	rdina	10 10	Execu	ution ti	me	e 81s		
PX SEND QC (RANDOM)      HASH GROUP BY      PX RECEIVE      PX SEND HASH     PX SEND HASH     PX SEND HASH     PX SEND LASH     PX SEND LOOPS      PX SEND LOOPS      PX RECEIVE	E BROADCAST	:TQ10002 :TQ10001	2 3 4 5 6 7 8 8 9	71 71 71 71 71 9,091 9,091	861K	Single table car underestim	rdina	lity	Execu	ution ti	me	e 81s		
HASH GROUP BY     PX RECEIVE     PX SEND HASH     PX SEND HASH     PX SEND HASH     PX SEND LOOPS     PX RECEIVE     PX RECEIVE     PX RECEIVE     PX SEND     PX RECEIVE     PX SEND     PX SEND     PX TABI	BROADCAST	:TQ10001	3 4 5 6 7 8 9	71 71 71 9,091 9,091		Single table car underestim	rdina	lity	Execu	ution ti	me	e 81s		
PX RECEIVE	E BROADCAST	:TQ10001	4 5 6 7 8 9	71 71 71 9,091 9,091		Single table car underestim	rdina	lity	Execu	ution ti	me	e 81s	-	
PX SEND HASH     PX SEND HASH     PHASH GROUP BY     PNESTED LOOPS     PX RECEIVE     PX RECEIVE     PX RECEIVE     PX SEND     PASH J     PX SEND     PX SEN	E BROADCAST	:TQ10001	5 6 7 8 9	71 71 9,091 9,091		underestim	und ata	iiity						
HASH GROUP BY     HASH GROUP BY     HASH JOOPS     HASH JOIN     PX RECEIVE     HASH JOIN     H	BROADCAST		6 7 8 9	71 9,091 9,091		underestim	t.							
NESTED LOOPS     NESTED     NESTED LOOPS     NESTED LOOPS     NESTED LOOPS     NESTED LOOPS     NESTED	E BROADCAST		7 8 9	9,091 9,091		underestim			17MB					3.14
NESTED LOOPS     HASH JOIN     PX RECEIVE     PX SEND     PX SEND     HASH JOIN     PX SEND     PX SEND     PX SIND     PX SEND     P	BROADCAST		8	9,091			Idle							.31
HASH JOIN PX RECEIVE PX SEND PX SEND	BROADCAST		9											
	BROADCAST			9,091		-			4GB	788MB	<b>80</b>	3,152	2GB	4.72
- PX SEND	BROADCAST		10	9,337	850K	-	4	66M						.63
		:TQ10000	11	9,337	850K		4	66M						.63
	OIN		12	9,337	850K		4	17M	17MB		<u> </u>			
TAB	FILTER CREATE	:BF0000	13	1	8		4	6,000			<u> </u>			
	LE ACCESS STORAGE FULL	DEPARTMENTS_DIM	14	1	8	1	4	6,000						
E JOIN	FILTER USE	:BF0000	15	147M	850K		4	19M						
⊟– PX	BLOCK ITERATOR		16	147M	850K		4	19M						
TA	ABLE ACCESS STORAGE FULL	SALES	17	147M	850K		91	19M	48MB		<u> 8</u>	265K	86GB	1.89
	TERATOR		18	1,000K	5,820		4	1,000K						
TABLE ACC	CESS STORAGE FULL	SUPPLIERS	19	1,000K	5,820		54	1,000K	19MB		<u> 8</u>	1,844	601MB	.63
-INDEX UNIQU	E SCAN	PRODUCTS_PK	20	1	1		19M	16M			-	4,849K	<b>37GB</b>	
TABLE ACCESS E	BY INDEX ROWID	PRODUCTS	21	1	2		24M	16M			-	921K	7GB	



Plan Sta	ttistics 🖉 Plan 🚳 Parallel 📐 Activity Value 383059184 📄 Plan Note	Metrics						C	ould	a colu	umr	۱		
Operation		Name	Line ID	Estimated Rows	Cost	Timeline(81s) Execut		gro	o auc	n CIT	Y ar	nd		y %
じ 🖻 SELEC	CT STATEMENT		0				>	0.						
🍦 🖻 PX (	COORDINATOR		1							TVDF	hol	n?		
🏭 🖻 P	X SEND QC (RANDOM)	:TQ10002	2	71	861K	(		<u> </u>			IICI	<b>P</b> :		
🖓 🗖	HASH GROUP BY		3	71					_			~ .		
аран С.	PX RECEIVE		4	71		Single table care	linali	+.,	໌ ,,ឧດເ	ition t	ime	81s		
<u>8</u>	PX SEND HASH	:TQ10001	5	71		Single table card	IIIaII	ιy						
<u>ک</u>			6	71		underection	+~		17MB					3.14
356 1	- NESTED LOOPS		7	9,091		underestima	ile							.31
86 6			8	9,091										
266	E-HASH JOIN		9	9,091					4GB	788MB	3,	152	2GB	4.72
350 100	PX RECEIVE		10	9,337	850K	- /	4	66M						.63
<b>3</b> 2	PX SEND BROADCAST	:TQ10000	11	9,337	850K		4	66M						.63
<b>6</b>	E-HASH JOIN		12	9,337	850K		4	17M	17MB		<u> </u>			
<b>3</b>	JOIN FILTER CREATE	:BF0000	13	1	8		4	6,000			<b>66</b>			
<b>6</b>	TABLE ACCESS STORAGE FULL	DEPARTMENTS_DIM	14	1	8		4	6,000						
<b>6</b>	- JOIN FILTER USE	:BF0000	15	147M	850K		4	19M						
<b>6</b>	E PX BLOCK ITERATOR		16	147M	850K		4	19M						
<b>6</b> 2	TABLE ACCESS STORAGE FULL	SALES	17	147M	850K	g	1	19M	48MB		3 26	5K	<b>86GB</b>	1.89
56 C	E PX BLOCK ITERATOR		18	1,000K	5,820		4	1,000K						
55 C	TABLE ACCESS STORAGE FULL	SUPPLIERS	19	1,000K	5,820	5	4	1,000K	19MB		3 1,1	844	601MB	.63
<b>3</b> 5	INDEX UNIQUE SCAN	PRODUCTS_PK	20	1	1	19	1	16M				4,849K	<b></b> 37GB	46
<b>3</b>	TABLE ACCESS BY INDEX ROWID	PRODUCTS	21	1	2	24	1	16M			<mark>_</mark> 9	921K	7GB	42



### Analysis: Column group on DEPARTMENS\_DIM(CITY, STORE\_TYPE)

SELECT dbms\_stats.create\_extended\_stats(USER,'DEPARTMENTS\_DIM','(CITY,STORE\_TYPE)') from dual;

EXEC dbms\_stats.gather\_table\_stats(user, 'DEPARTMENTS\_DIM', METHOD\_OPT=>'FOR COLUMNS (CITY,STORE\_TYPE) SIZE 2048 FOR ALL COLUMNS SIZE AUTO');

PL/SQL procedure successfully completed.

SELECT column\_name, num\_distinct, num\_nulls, histogram, num\_buckets
FROM user\_tab\_col\_statistics
WHERE table\_name = 'DEPARTMENTS\_DIM';

COLUMN_NAME	NUM_DISTINCT	NUM_NULLS	HISTOGRAM	NUM_BUCKETS
DEPARTMENT_ID	50000	0	NONE	1
DEPARTMENT_NAME	50000	0	NONE	1
STORE_ID	5000	0	NONE	1
* * * * * * * * * *				
CITY	2080	0	HYBRID	254
STORE_TYPE	5	0	FREQUENCY	5
SYS STUMZODT3BT6MDTDUS 10YH3M#	2080	0	HYBRID	1878



Plan Hash Value	2473962328 Plan Note All Parallel Serve	ers 🔻												
Deration														
- SELECT STAT		Name	Line ID	Estimated Rows	Cost	Timeline(5s)	Executions	Actual Rows	Memory (Max)	Temp (Max)	Other	IO Requests	IO Bytes	Activity %
	EMENT		0				9	10						
	NATOR	7010001	1		4 4 9 9 14		9	10						
		:1Q10004	2	71	1,198K		4	10	CMD					
			3	71	1,198K		4	10	OMB					
		.TO10002	4	71	1,1986		4	40			<b>/20</b>			
		:1010003	5	71	1,196K		4	40	1700		90			10
			0	/1	1,196K		4	40			(89),			10
			/	20M	1,190K		4	20M	904MD		99			
		·TO10001	0	2014	2426		4	2014						10
		:1010001	9	2014	242K		4	2014			99			10
			10	20M	2424		4	2014						
		PRODUCTS	11	2014	2424		-4	2014	49MD			Odk	2100	10
ſ		FRODUCTS	12	18M	856K		4	16M			00	5410	<b>_</b> 516b	5
Ľ		·TO10002	14	18M	856K		4	16M			3 <b>8</b> 2			5
		.1010002	19	19M	OFER		7	16M	277MP		(C) (29)			15
			15	1 000K	6 920	_	T A	4 000K	277140		00			15
		·TO10000	17	1,000K	5,820	_	т Д	4,000K						
		.1010000	19	1,000K	5,820	_	т 4	1,000K						
			10	1,000K	5,820	_	54	1,000K	19MB		<i>3</i> 9	1 844	601MB	
		Soft Eleto	20	18M	850K		4	17M	17MB		60 ·	1,011	001110	
r .		:BE0000	20	1.650	8	_	4	6.000	17110		30 68			
		DEPARTMENTS DIM	22	1,650	8		4	6,000			90			
r -	E-JOIN FILTER USE	:BE0000	23	1,350 147M	850K		4	19M						
			24	147M	850K		4	19M						
r		SALES	25	147M	850K		91	19M	48MB		68	265K	86GB	10
1		JALLO	25	14714	USOK		91	1914	UND		00	ZUJK	3000	10



lan Hash	Value 2473962328 Plan Note All Parallel Serv	vers 🗸 🔻												
peration		Name	Line ID	Estimated Rows	Cost	Timeline(5s)	Executions	Actual Rows	Memory (Max)	Temp (Max)	Other	IO Requests	IO Bytes	Activity %
- SELE	CT STATEMENT		0				9	10						
E PX	COORDINATOR		1				9	10						
⊡ P	X SEND QC (RANDOM)	:TQ10004	2	71	1,198K	-	4	10						
	HASH GROUP BY		3	71	1,198K		4	10						
	- PX RECEIVE		4	71	1,198K		4	40	l Exec	cution	tim	ne 5s		
	PX SEND HASH	:TQ10003	5	71	1,198K		4	40						
			6	71	1,198K		4	40	17MB					10
	HASH JOIN		7	18M	1,198K		4	16M	964MB		<u> 66</u>			
	- PX RECEIVE		8	20M	343K		4	20M						
	PX SEND HYBRID HASH	:TQ10001	9	20M	343K		4	20M			<b>66</b>			10
	- STATISTICS COLLECTOR		10				4	20M						
			11	20M	343K		4	20M						
	TABLE ACCESS STORAGE FULL	PRODUCTS	12	20M	343K		55	20M	48MB		<u> 66</u>	94K	<b>31GB</b>	10
			13	18M	856K		4	16M						5
	PX SEND HYBRID HASH	:TQ10002	14	18M	856K		4	16M			<u> 66</u>			15
	- HASH JOIN		15	18M	856K		4	16M	277MB		<u> 66</u>			15
			16	1,000K	5,820		4	4,000K						
	- PX SEND BROADCAST	:TQ10000	17	1,000K	5,820	_	4	4,000K						
			18	1,000K	5,820	_	4	1,000K						
	TABLE ACCESS STORAGE FULL	SUPPLIERS	19	1,000K	5,820	_	54	1,000K	18MB		<u> 8</u>	1,844	601MB	
			20	18M	850K		4	17M	17MB		<u> 66</u>			
	- JOIN FILTER CREATE	:BF0000	21	1,650	8	_	4	6,000			<u> 66</u>			
	TABLE ACCESS STORAGE FULL	DEPARTMENTS_DIM	22	1,650	8		4	6,000						
		:BF0000	23	147M	850K		4	19M						
			24	147M	850K		4	19M						
	TABLE ACCESS STORAGE FULL	SALES	25	147M	850K		91	19M	48MB		<u> 8</u>	265K	86GB	10



an Hash Value 2473962328 Plan Note All Parallel Servers														
eratio	n	Name	Line ID	Estimated Rows	Cost	Timeline(5s)	Executions	Actual Rows	Memory (Max)	Temp (Max)	Other	IO Requests	IO Bytes	Activity %
SE	LECT STATEMENT		0				9	10						
	PX COORDINATOR		1				9	10						
Ē	PX SEND QC (RANDOM)	:TQ10004	2	71	1,198K		4	10						
			3	71	1,198K		4	10						
			4	71	1,198K		4	40	l Exec	cution	tim	ne 5s		
	- PX SEND HASH	:TQ10003	5	71	1,198K		4	40						
			6	71	1,198K		4	40	17MB					10
	- HASH JOIN		7	18M	1,198K		4	16M	964MB		<b>60</b>			
	PX RECEIVE		8	20M	343K		4	20M						
	PX SEND HYBRID HASH	:TQ10001	9	20M	343K		4	20M			<u> 8</u>			10
	STATISTICS COLLECTOR		10				4	20M						
			11	20M										
	TABLE ACCESS STORAGE FULL	PRODUCTS	12	20M		Vary good o	inala t	abla	48MB		<u> 66</u>	<b>—</b> 94K	<b>31GB</b>	10
	PX RECEIVE		13	18M		very good s	r estimate							5
	PX SEND HYBRID HASH	:TQ10002	14	18M							<u> </u>			15
	- HASH JOIN		15	18M		cardinality			277MB		<u> 8</u>			15
			16	1,000K										
	- PX SEND BROADCAST	:TQ10000	17	1,000K										
			18	1,000K	5,820	_/	4	1,000K						
	TABLE ACCESS STORAGE FULL	SUPPLIERS	19	1,000K	5,820	-	54	1,000K	18MB		<u> 8</u>	1,844	601MB	
			20	18M	850K		4	17M	17MB		<u> 8</u>			
		:BF0000	21	1,650	8	_	4	6,000	-		<b>60</b>			
	TABLE ACCESS STORAGE FULL	DEPARTMENTS_DIM	22	1,650	8		4	6,000						
		:BF0000	23	147M	850K		4	19M						
	PX BLOCK ITERATOR		24	147M	850K		4	19M						
	TABLE ACCESS STORAGE FULL	SALES	25	147M	850K		91	19M	48MB		8	265K	86GB	10





• "This approach requires schema changes..."



- "This approach requires schema changes..."
- "This approach requires code changes (ETL, queries, etc) ..."



- "This approach requires schema changes..."
- "This approach requires code changes (ETL, queries, etc) ..."
- "If we changed the code, we would have to test it ..."



- "This approach requires schema changes..."
- "This approach requires code changes (ETL, queries, etc) ..."
- "If we changed the code, we would have to test it ..."
- "The person who developed this code is no longer with the company hence we cannot implement such a solution ..."



- "This approach requires schema changes..."
- "This approach requires code changes (ETL, queries, etc) ..."
- "If we changed the code, we would have to test it ..."
- "The person who developed this code is no longer with the company hence we cannot implement such a solution ..."
- "It's a 3<sup>rd</sup> party application ..."



- "This approach requires schema changes..."
- "This approach requires code changes (ETL, queries, etc) ..."
- "If we changed the code, we would have to test it ..."
- "The person who developed this code is no longer with the company hence we cannot implement such a solution ..."
- "It's a 3<sup>rd</sup> party application ..."
- "After much consideration the management decided to stick with the current solution ..."



### **Enter the Materialized View**

• Traditionally the Materialized View (MV) is to

Materialize expensive joins Summarize or rollup frequently aggregated data

Some Key Properties

A query can automatically be re-written by the optimizer to use an MV

Transparent to user queries

No data model changes required

MV's can automatically be kept up to date by the database



#### **Create an MV for the join between LOCATIONS, STORES and DEPARTMENTS**

```
CREATE MATERIALIZED VIEW DEP_STORE_LOC_MV

PARALLEL 4

ENABLE QUERY REWRITE

AS

SELECT D.*

, L.CITY

, S.STORE_TYPE

from DEPARTMENTS D

, STORES S

, LOCATIONS L

WHERE D.STORE_ID=S.STORE_ID

AND S.LOCATION ID=L.LOCATION ID;
```



### **Benefit of Materialized View**

• In this example

The MV is not used to rollup aggregates The Joins in this case are not expensive

• The MV is used to improve cardinality estimates


# Query Rewrite for Materialized View

#### MV creation is fast

General	Tim	e & Wait Stat	istics					IO Statistics					
SQL Text         CREATE TABLE "MTEKIC"."DEP_STORE_LOC_MV" (           Execution Plan	 Dat PL/	Duration abase Time SQL & Java 0 Activity %	S	_		-	0.2s 0.4s 100	Buffer Gets IO Requests IO Bytes					2,096 77 2MB
Fetch Calls 0													
<ul> <li>✓ Details</li> </ul>													
Plan Statistics 🚋 Plan 🎲 Parallel 📐 Activity													
Plan Hash Value 114397671 Plan Note													
Operation	Name	Line ID	Estimated Rows	Cost	Timeline(0.155791s)	Executions	Actual Rows	Memory (Max)	Temp (Max)	Other	IO Requests	IO Bytes	Activity %
		0				9	8						
🝦 🖻 - PX COORDINATOR		1				9	8						
👪 📥 PX SEND QC (RANDOM)	:TQ10001	2	27К	34		4	8						
LOAD AS SELECT (HYBRID TSM/HWMB)	DEP_STORE_LOC_MV	3				4	8				20	1MB	
👪 🖨 HASH JOIN		4	27К	34		4	27К	19MB		<u> 66</u>			
3 E- PX RECEIVE		5	2,700	20		- 4	11K						
	:TQ10000	6	2,700	20		- 4	11K						
ASH JOIN		7	2,700	20		- 4	2,700	17MB		<u> 66</u>			
3 OIN FILTER CREATE	:BF0000	8	2,700	3		4	11K			<u> 66</u>			
TABLE ACCESS STORAGE FULL	STORES	9	2,700	3		4	11K				1	8KB	100
3 JOIN FILTER USE	:BF0000	10	100K	17		4	3,744			<u> </u>			
B-PX BLOCK ITERATOR		11	100K	17		4	3,744						
TABLE ACCESS STORAGE FULL	LOCATIONS	12	100K	17		43	3,744						
👪 📃 – PX BLOCK ITERATOR		13	50K	14		4	50K						
TABLE ACCESS STORAGE FULL	DEPARTMENTS	14	50K	14		43	50K				5	5 🔜 1MB	



# Query Rewrite for Materialized View

#### MV creation is fast

General	Tin	ne & Wait Stat	tistics					IO Statistics					
SQL Text CREATE TABLE "MTEKIC"."DEP_STORE_LOC_MV" (		Duration		_		_	0.2s	Buffer Gets					2,096
Execution Fiant war 4	PI		s				0.45	IO Requests					2MB
Last Refresh Time Sun Sep 15, 2019 6:46:50 PM		Activity %	5	_			100	10 5/65					LIND
Execution ID 33554433		,											
User MTEKIC@PDB1													
Fetch Calls 0							~ ~						
					IVIV Cre	eated in	1 U.2S						
Details													
Plan Statistics Plan 🎲 Parallel 📐 Activity								_					
Operation	Name	Line ID	Estimated Rows	Cost	Timeline(0.155791s)	Executions	Actual Rows	Memory (Max)	Temp (Max)	Other	IO Requests	IO Bytes	Activity %
		0				9	8						
		1				9	8						
PX SEND QC (RANDOM)	:TQ10001	2	27K	34		- 4	8						
LOAD AS SELECT (HYBRID TSM/HWMB)	DEP_STORE_LOC_MV	/ 3				4	8				20	1MB	
🖓 🖨 HASH JOIN		4	27К	34		4	27K	19MB		<b>6</b>			
A RECEIVE		5	2,700	20		4	11K						
A PX SEND BROADCAST	:TQ10000	6	2,700	20		4	11K						
📸 🖻 HASH JOIN		7	2,700	20		4	2,700	17MB		<u>60</u>			
3 JOIN FILTER CREATE	:BF0000	8	2,700	3		4	11K			60			
TABLE ACCESS STORAGE FULL	STORES	9	2,700	3		4	11K				1	8KB	100
3 D-JOIN FILTER USE	:BF0000	10	100K	17		4	3,744			<u> 66</u>			
B-PX BLOCK ITERATOR		11	100K	17		4	3,744						
TABLE ACCESS STORAGE FULL	LOCATIONS	12	100K	17		43	3,744						
📸 📴 - PX BLOCK ITERATOR		13	50K	14		4	50K						
TABLE ACCESS STORAGE FULL	DEPARTMENTS	14	50K	14		43	50K				5	5 1MB	



	eee Plan 🖓 Parallel 📉 Activity 🔀 Me	uncs												
lan Hash Value	2230777441 Plan Note													
peration		Name	Line ID	Estimated Rows	Cost	Timeline(82s) Exec	cutions	Actual Rows	Memory (Max)	Temp (Max)	Other	IO Requests	IO Bytes	Activity %
E-SELECT STAT	FEMENT		0			1	9	10						
	INATOR		1			1	9	10						
PX SEND	QC (RANDOM)	:TQ10002	2	71	902K	1	4	10						
- HASH (	GROUP BY		3	71	902K	1	4	10	6MB					
E PX R	ECEIVE		4	71	902K	1	4	40						
E-PX	SEND HASH	:TQ10001	5	71	902K	1	4	40			8			
<u> </u>	HASH GROUP BY		6	71	902K		4	40	17MB					.63
	- NESTED LOOPS		7	83K	902K		4	16M						.63
	- NESTED LOOPS		8	83K	902K		4	16M						
÷			9	83K	856K		4	16M	4GB	788MB	<b>60</b>	3,152	2GB	5.02
	PX RECEIVE		10	86K	850K	-	4	66M						1.25
	PX SEND BROADCAST	:TQ10000	11	86K	850K		4	66M						.63
	- HASH JOIN		12	86K	850K		4	17M	17MB		66			.63
	- JOIN FILTER CREATE	:BF0000	13	8	13	1	4	6,000			66			
	MAT_VIEW REWRITE ACCESS STORAGE	DEP_STORE_LOC_MV	14	8	13	1	4	6,000						
	- JOIN FILTER USE	:BF0000	15	147M	850K		4	19M						
	- PX BLOCK ITERATOR		16	147M	850K		4	19M						
	TABLE ACCESS STORAGE FULL	SALES	17	147M	850K		91	19M	48MB		<u>66</u>	265K	<b>86GB</b>	1.25
	PX BLOCK ITERATOR		18	1,000K	5,820		4	1,000K						
	TABLE ACCESS STORAGE FULL	SUPPLIERS	19	1,000K	5,820		54	1,000K	18MB		66	1,844	601MB	
÷	INDEX UNIQUE SCAN	PRODUCTS_PK	20	1	1		19M	16M				4,845K	<b></b> 37GB	
	TABLE ACCESS BY INDEX ROWID	PRODUCTS	21	1	2		24M	16M				<mark></mark> 923K	7GB	38



etails				origi	nal									
Plan Statistics	🚋 Plan 🛛 🖓 Parallel 📐 Activity 🎘 Me	trics		origi	lldl	query								
-														
Plan Hash Value 2														
Operation		Name	Line ID	Estimated Rows	Cost	Timeline(82s)	Executions	Actual Rows	Memory (Max)	Temp (Max)	Other	IO Requests	IO Bytes	Activity %
SELECT STATE	MENT		0				9	10						
PX COORDIN	NATOR		1				9	10						
D PX SEND C	QC (RANDOM)	:TQ10002	2	71	902K		4	10						
🖞 📋 HASH GI	ROUP BY		3	71	902K		4	10	6MB					
🖇 🖻 PX REG	CEIVE		4	71	902K		4	40						
🖇 🖻 PX S	SEND HASH	:TQ10001	5	71	902K		4	40			66			
<u>\$</u> — н/	ASH GROUP BY		6	71	902K		4	40	17MB					.63
\$ <u> </u>	NESTED LOOPS		7	83K	902K		4	16M						.63
já E	-NESTED LOOPS		8	83K	902K		4	16M						
jû (ji karal na serie de la se			9	83K	856K		4	16M	4GB	788MB	<b>60</b>	3,152	2GB	5.02
<u>\$</u>	E-PX RECEIVE		10	86K	850K	-	4	66M						1.25
<b>3</b>	PX SEND BROADCAST	:TQ10000	11	86K	850K		4	66M						.63
<b>3</b>	HASH JOIN		12	86K	850K		4	17M	17MB		<b>6</b>			.63
<b>\$</b>	- JOIN FILTER CREATE	:BF0000	13	8	13	i.	4	6,000			<b>6</b>			
<b>å</b>	MAT_VIEW REWRITE ACCESS STORAGE	DEP_STORE_LOC_MV	14	8	13	1	4	6,000						
<u>\$</u>	-JOIN FILTER USE	:BF0000	15	147M	850K		4	19M						
<u>\$</u>	- PX BLOCK ITERATOR		16	147M	850K		4	19M						
<u>\$</u>	TABLE ACCESS STORAGE FULL	SALES	17	147M	850K		91	19M	48MB		<u>60</u>	265K	<b>86GB</b>	1.25
<u>\$</u>	- PX BLOCK ITERATOR		18	1,000K	5,820		4	1,000K						
<u>\$</u>	TABLE ACCESS STORAGE FULL	SUPPLIERS	19	1,000K	5,820		54	1,000K	18MB		<u> 66</u>	1,844	601MB	
<u>۾</u>	INDEX UNIQUE SCAN	PRODUCTS_PK	20	1	1		19M	16M				4,845K	<b>37GB</b>	
-9	TABLE ACCESS BY INDEX ROWID	PRODUCTS	21	1	2		24M	16M				<mark></mark> 923K	7GB	38



Plan Statistics Plan Hash Value 2	2230777441 📄 Plan Note	trics		- 0										
Operation		Name	Line ID	Estimated Rows	Cost	Timeline(82 <u>s)</u>	Executions	Actual Rows	Memory (Max)	Temn (Max)	Other	IO Requests	IO Bytes	Activity %
🧯 🖻 – SELECT STATE	MENT		0				9	10						
🍦 🖕 PX COORDIN	NATOR		1				9	10	EX	ecutio	nτ	ime 87	2S	
👸 📄 PX SEND (	QC (RANDOM)	:TQ10002	2	71	902K		4	10						
🐉 🖻 HASH GI	ROUP BY		3	71	902K		4	10	6MB					
🐉 🖨 PX RE	CEIVE		4	71	902K		4	40						
🐉 🖻 PX S	SEND HASH	:TQ10001	5	71	902K		4	40			<b>60</b>			
🐉 😐 н/	ASH GROUP BY		6	71	902K		4	40	17MB					.63
<b>%</b> 🖻	NESTED LOOPS		7	83K	902K		4	16M						.63
16 E	-NESTED LOOPS		8	83K	902K		4	16M						
<u>j</u> û	E-HASH JOIN		9	83K	856K		4	16M	4GB	788MB	<u> 8</u>	3,152	2GB	5.02
<u>å</u>	PX RECEIVE		10	86K	850K	-	4	66M						1.25
<b>3</b> 2	E-PX SEND BROADCAST	:TQ10000	11	86K	850K		4	66M						.63
<b>5</b> 2	- HASH JOIN		12	86K	850K		4	17M	17MB		đ			.63
5 <b>5</b>	JOIN FILTER CREATE	:BF0000	13	8	13	1	4	6,000			<del>60</del>			
<b>ä</b> i	MAT_VIEW REWRITE ACCESS STORAGE	DEP_STORE_LOC_MV	14	8	13	1	4	6,000						
<b>5</b> 2	JOIN FILTER USE	:BF0000	15	147M	850K		4	19M						
<b>6</b> 0			16	147M	850K		4	19M						
8	TABLE ACCESS STORAGE FULL	SALES	17	147M	850K		91	19M	48MB		<u>8</u>	265K	<b>86GB</b>	1.25
<b>5</b> 2	PX BLOCK ITERATOR		18	1,000K	5,820		4	1,000K						
<b>5</b> 2	TABLE ACCESS STORAGE FULL	SUPPLIERS	19	1,000K	5,820		54	1,000K	18MB		<b>60</b>	1,844	601MB	
\$	INDEX UNIQUE SCAN	PRODUCTS_PK	20	1	1		19M	16M				4,845	37GB	
<b>S</b>	TABLE ACCESS BY INDEX ROWID	PRODUCTS	21	1	2		24M	16M				<mark></mark> 923K	7GB	38



#### Cardinality underestimate still a problem

#### Monitored SQL Execution Details: azmv8ma1147ff 🥪

									1					
peration		Name	Line ID	Estimated Rows	Cost	Timeline(82 <u>s)</u> E	ecutions A	ctual Rows M	lemony (Max)	Temp (Max) 0	ther	IO Requests	IO Bytes	Activity %
SELECT STAT	EMENT		0				9	10	Fva	ocution	h ti	ma 87	c	
E-PX COORDI	NATOR		1				9	10		cution	I U		.3	
PX SEND	QC (RANDOM)	:TQ10002	2	71	902K		4	10			_			
E HASH G	ROUP BY		3	71	902K		4	10	6MB					
⊟-PX RE	ECEIVE		4	71	902K		4	40						
Ė−PX	SEND HASH	:TQ10001	5	71	_						66			
⊡-H	ASH GROUP BY		6	71	R	ewrite took	place,	but t	here I	S				.63
Ξ	NESTED LOOPS		7	83K		1. 1.	• •							.63
[			8	83K	big	cardinality i	nis-es	timat	e at ro	)W	~			
	E-HASH JOIN		9	83K	C	,				88MB	<b>60</b>	3,152	2GB	5.02
			10	86K		source	e 14. V	Vhy?						1.25
	- PX SEND BROADCAST	:TQ10000	11	86K										.63
	⊡-HASH JOIN		12	86K	850K		4	17M	17MB		<b>60</b>			.63
	□-JOIN FILTER CREATE	:BF0000	13	8	13		4	6,000			66			
	MAT_VIEW REWRITE ACCESS STORAGE	DEP_STORE_LOC_MV	14	8	13		4	6,000						
		:BF0000	15	147M	850K		4	19M						
			16	147M	850K		4	19M						_
	TABLE ACCESS STORAGE FULL	SALES	17	147M	850K		91	19M	48MB		66	265K	<b>86GB</b>	1.25
			18	1,000K	5,820		4	1,000K						
	TABLE ACCESS STORAGE FULL	SUPPLIERS	19	1,000K	5,820		54	1,000K	18MB		66	1,844	601MB	
		PRODUCTS_PK	20	1	1		19M	16M				4,845K	37GB	
	TABLE ACCESS BY INDEX ROWID	PRODUCTS	21	1	2		24M	16M			-	923K	7GB	38

Cardinality underestimate still a problem

#### Monitored SQL Execution Details: azmv8ma1147ff 🥪

Overview			- 5	Same S	QL	ID as the								
Details				origi	nal	auerv								
📄 Plan Statis	tics 🖧 Plan 🎆 Parallel 📐 Activity 📜 Me	trics		UIBI	Tur	query								
Plan Hash Valu	ue 2230777441 Plan Note													
Operation		Name	Line ID	Estimated Rows	Cost	Timeline(82 <u>s)</u>	Executions	Actual Rows	Memory (Max)	Temp (Max)	Other	TO Requests	IO Bytes	Activity %
E-SELECT S	STATEMENT		0			1	9	10	Бу		+			
	ORDINATOR		1			1	9	10	EX	eculic	י חכ	.ime ŏ₄	25	
🗯 🖻 PX S	END QC (RANDOM)	:TQ10002	2	71	902K	1	4	10						
а 👝 на	SH GROUP BY		3	71	902K	1	4	10	6MB					
à 🗄 P	PX RECEIVE		4	71	902K	l. I	4	40						
	PX SEND HASH	:TQ10001	Nies			:					60			
}	E-HASH GROUP BY		Nes	sted Lo	ops	Join when	lac	e. but	there	is 🔄				.63
à	- NESTED LOOPS													.63
	- NESTED LOOPS		Join	ing PR	יטט	JCIS table	is-	estima	te at r	'OW				
i i	- HASH JOIN									8MB	<b>60</b>	3,152	2GB	5.02
1	E-PX RECEIVE		10	86K		sour	ce 14.	. Whv?						1.25
<b>}</b>	E-PX SEND BROADCAST	:TQ10000	11	86K				· · · · · , ·						.63
3	E-HASH JOIN		12	86K	850K		- 4	17M	17MB		60			.63
3	- JOIN FILTER CREATE	:BF0000	13	8	13		4	6,000			60			
}	MAT_VIEW REWRITE ACCESS STORAGE	DEP_STORE_LOC_MV	14	8	13		4	6,000						
3	- JOIN FILTER USE	:BF0000	15	147M	850K		4	19M						
	E-PX BLOCK ITERATOR		16	147M	850K		4	19M						
	TABLE ACCESS STORAGE FULL	SALES	17	147M	850K		91	19M	48MB		õ	265K	<b>86GB</b>	1.25
	E- PX BLOCK ITERATOR		18	1,000K	5,820		4	1,000K						
à	TABLE ACCESS STORAGE FULL	SUPPLIERS	19	1,000K	5,820		54	1,000K	18MB		đ	1,844	601MB	
ð		PRODUCTS_PK	20	1	1		19M	16M				4,845	37GB	
	TABLE ACCESS BY INDEX ROWID	PRODUCTS	21	1	2		24M	16M				<mark></mark> 923K	7GB	38









SELECT num\_rows
FROM user\_tab\_statistics
WHERE table\_name = 'DEP\_STORE\_LOC\_MV';

NUM\_ROWS

-----

27000

SELECT column\_name, num\_distinct, num\_nulls, num\_buckets
FROM user\_tab\_col\_statistics
WHERE table\_name = 'DEP\_STORE\_LOC\_MV'
AND column\_name IN ('CITY','STORE\_TYPE');

COLUMN_NAME	NUM_DISTINCT	NUM_NULLS	NUM_BUCKETS
CITY	2080	0	254
STORE_TYPE	5	0	5

SELECT count(DISTINCT CITY||' '||store\_type) c\_check
FROM DEP\_STORE\_LOC\_MV;

C\_CHECK

2080



SELECT num\_rows
FROM user\_tab\_statistics
WHERE table\_name = 'DEP\_STORE\_LOC\_MV';

NUM\_ROWS

-----

27000

SELECT column\_name, num\_distinct, num\_nulls, num\_buckets
FROM user\_tab\_col\_statistics
WHERE table\_name = 'DEP\_STORE\_LOC\_MV'
AND column\_name IN ('CITY','STORE\_TYPE');

COLUMN_NAME	NUM_DISTINCT	NUM_NULLS	NUM_BUCKETS
CITY	2080	0	254
STORE_TYPE	5	0	5

SELECT count(DISTINCT CITY||' '||store\_type) c\_check FROM DEP\_STORE\_LOC\_MV;

C\_CHECK 2080\_\_\_\_\_\_ Indicates strong correlation between CITY and STORE\_TYPE



### SELECT num\_rows FROM user\_tab\_statistics WHERE table name = 'DEP STORE LOC MV';

#### NUM\_ROWS

\_\_\_\_\_

27000

C CHECK

2080-

SELECT column\_name, num\_distinct, num\_nulls, num\_buckets
FROM user\_tab\_col\_statistics
WHERE table\_name = 'DEP\_STORE\_LOC\_MV'
AND column name IN ('CITY','STORE TYPE');

COLUMN_NAME	NUM_DISTINCT	NUM_NULLS	NUM_BUCKETS
CITY	2080	0	254
STORE_TYPE	5	0	5

SELECT count(DISTINCT CITY||' '||store type) c check FROM DEP\_STORE\_LOC\_MV;

Indicates strong correlation between CITY and STORE\_TYPE SELECT

dbms\_stats.create\_extended\_stats(USER, 'DEP\_STORE\_LOC\_MV', '(CITY,STORE\_TYPE)')
from dual;

EXEC dbms\_stats.gather\_table\_stats(user, 'DEP\_STORE\_LOC\_MV', METHOD\_OPT=>'FOR COLUMNS (CITY,STORE\_TYPE) SIZE 2048 FOR ALL COLUMNS SIZE AUTO');

PL/SQL procedure successfully completed.

SELECT column\_name, num\_distinct, num\_nulls, histogram, num\_buckets
FROM user\_tab\_col\_statistics
WHERE table name = 'DEP STORE LOC MV';

COLUMN_NAME	NUM_DISTINCT	NUM_NULLS	HISTOGRAM	NUM_BUCKETS
DEPARTMENT_ID	27000	0	NONE	1
DEPARTMENT_NAME	27000	0	NONE	1
STORE_ID	2700	0	NONE	1
CITY	2080	0	HYBRID	254
STORE_TYPE	5	0	FREQUENCY	5
SYS_STUMZODT3BT6MDTDUS_10YH3M#	2080	0	HYBRID	1878



#### With extended statistics

etails														
Dian Stati	stics 😓 Plan 88 Parallel 🕨 Activity 📜 Mei	rics												
- Han Stat														
Plan Hash Va	lue 3112689538 Plan Note All Parallel Servers	•												
Operation		Name	Line ID	Estimated Rows	Cost	Timeline(4s)	Executions	Actual Rows	Memory (Max)	Temp (Max)	Other	IO Requests	IO Bytes	Activity %
	STATEMENT		0				9	10						
🍦 🖻 PX C	DORDINATOR		1				9	10						
🐉 🖻 PX	SEND QC (RANDOM)	:TQ10004	2	71	1,198K		4	10						
🍇 🗄 H	ASH GROUP BY		3	71	1,198K		4	10	6MB					
👸 🗖	PX RECEIVE		4	71	1,198K		4	40						
<u>ل</u>	PX SEND HASH	:TQ10003	5	71	1,198K		4	40			6			
<u>1</u>			6	71	1,198K		4	40	17MB					18
<u>8</u>	HASH JOIN		7	15M	1,198K		4	16M	964MB		<u> 66</u>			
<u>86</u>	PX RECEIVE		8	20M	343K		4	20M						
<b>8</b>	PX SEND HYBRID HASH	:TQ10001	9	20M	343K		4	20M			<u> </u>			3.03
<b>8</b>	- STATISTICS COLLECTOR		10			_	4	20M						
<b>8</b>	- PX BLOCK ITERATOR		11	20M	343K	_	4	20M						
<b>3</b>	TABLE ACCESS STORAGE FULL	PRODUCTS	12	20M	343K		55	20M	48MB		<u> </u>	94K	<b>31GB</b>	9.09
<b>3</b>			13	15M	856K		4	16M						
<b>3</b>	- PX SEND HYBRID HASH	:TQ10002	14	15M	856K		4	16M			<b>60</b>			9.09
<b>3</b>			15	15M	856K		4	16M	277MB		<b>60</b>			6.06
<b>3</b> 3			16	1,000K	5,820		4	4,000K						
<u>36</u>	- PX SEND BROADCAST	:TQ10000	17	1,000K	5,820	_	4	4,000K						
<u>ک</u>			18	1,000K	5,820		4	1,000K						
<b>3</b>	TABLE ACCESS STORAGE FULL	SUPPLIERS	19	1,000K	5,820	_	54	1,000K	18MB		<b>60</b>	1,844	601MB	
<b>6</b>	- HASH JOIN		20	16M	850K		4	17M	17MB		<b>60</b>			
<b>3</b> 5	- JOIN FILTER CREATE	:BF0000	21	1,435	13	_	4	6,000			<b>6</b>			
<b>5</b>	MAT_VIEW REWRITE ACCESS STORAGE FU	DEP_STORE_LOC_MV	22	1,435	13		4	6,000						
5		:BF0000	23	147M	850K		4	19M						
35	- PX BLOCK ITERATOR		24	147M	850K		4	19M						
<b>a</b>	TABLE ACCESS STORAGE FULL	SALES	25	147M	850K		91	19M	48MB		<u> </u>	265K	86GB	



#### With extended statistics

#### Monitored SQL Execution Details: azmv8ma1147ff 🥪

				~~_										
			_	original query										
Plan St	atistics 👰 Plan 🖓 Parallel 📐 Activity 🎘 Me	trics		0.10	inter	query								
Plan Hash	Value 3112689538 Plan Note All Parallel Servers	•												
Operation		Name	Line ID	Estimated Rows	Cost	Timeline(4s)	Executions	Actual Rows	Memory (Max)	Temp (Max)	Other	IO Requests	IO Bytes	Activity %
- SELE	ECT STATEMENT		0				9	10						-
E-PX	COORDINATOR		1				9	10						
i اط	PX SEND QC (RANDOM)	:TQ10004	2	71	1,198K		4	10						
È	HASH GROUP BY		3	71	1,198K		4	10	6MB					
3			4	71	1,198K		4	40						
3	-PX SEND HASH	:TQ10003	5	71	1,198K	_	4	40			<b>60</b>			
1			6	71	1,198K		4	40	17MB					18
1			7	15M	1,198K		4	16M	964MB		<u> 6</u>			
}			8	20M	343K	_	4	20M						
	- PX SEND HYBRID HASH	:TQ10001	9	20M	343K		4	20M			<u> </u>			3.03
	- STATISTICS COLLECTOR		10			_	4	20M						
			11	20M	343K	—	4	20M						
	TABLE ACCESS STORAGE FULL	PRODUCTS	12	20M	343K		55	20M	48MB		<u> </u>	<b>94</b> K	<b>31GB</b>	9.09
			13	15M	856K		4	16M						
}	- PX SEND HYBRID HASH	:TQ10002	14	15M	856K		4	16M			<u> 66</u>			9.09
}	HASH JOIN		15	15M	856K		4	16M	277MB		<u> </u>			6.06
1	-PX RECEIVE		16	1,000K	5,820	_	4	4,000K						
	- PX SEND BROADCAST	:TQ10000	17	1,000K	5,820	—	4	4,000K						
}			18	1,000K	5,820		4	1,000K						
5	TABLE ACCESS STORAGE FULL	SUPPLIERS	19	1,000K	5,820		54	1,000K	18MB		<b>60</b>	1,844	601MB	
}	E-HASH JOIN		20	16M	850K		4	17M	17MB		<u> </u>			
}	- JOIN FILTER CREATE	:BF0000	21	1,435	13		4	6,000			<u> </u>			
	MAT_VIEW REWRITE ACCESS STORAGE FU	DEP_STORE_LOC_MV	22	1,435	13		4	6,000						
		:BF0000	23	147M	850K		4	19M						
3			24	147M	850K		4	19M						
5	TABLE ACCESS STORAGE FULL	SALES	25	147M	850K		91	19M	48MB		<b>60</b>	265K	86GB	24



#### With extended statistics

#### Monitored SQL Execution Details: azmv8ma1147ff 🥪

		ounie (	- <u> </u>										
Details			_	orig	inal	auerv							
Plan Statistic	s कुङ्गु Plan 👹 Parallel 📐 Activity 🖳 Me	trics											
Plan Hash Value	3112689538 Plan Note All Parallel Servers	▼											
Operation		Name	Line ID	Estimated Rows	Cost	Timeline(4s)	Executions	Actual Rows				ytes	Activity %
SELECT STA	TEMENT		0				9	10	Evocut	ion ti	mode		
	DINATOR		1			_	9	10	EXECU		1116 45		
PX SENI	D QC (RANDOM)	:TQ10004	2	71	1,198K		4	10					
HASH	GROUP BY		3	71	1,198K	_	4	10	6MB				
PX I	RECEIVE		4	71	1,198K		4	40					
E P	K SEND HASH	:TQ10003	5	71	1,198K	_	4	40		<u>66</u>			
	HASH GROUP BY		6	71	1,198K		4	40	17MB				18
) E	-HASH JOIN		7	15M	1,198K		4	16M	964MB	<b>60</b>			
3	PX RECEIVE		8	20M	343K		4	20M					
}	- PX SEND HYBRID HASH	:TQ10001	9	20M	343K		4	20M		<b>60</b>			3.03
6	- STATISTICS COLLECTOR		10				4	20M					
<b>6</b>	PX BLOCK ITERATOR		11	20M	343K	_	4	20M					
۶.	TABLE ACCESS STORAGE FULL	PRODUCTS	12	20M	343K		55	20M	48MB	<u> 66</u>	<u>—</u> 94К	<b>31GB</b>	9.09
ត្តិ	- PX RECEIVE		13	15M	856K		4	16M					
<b>\$</b>	- PX SEND HYBRID HASH	:TQ10002	14	15M	856K		4	16M		<u> 66</u>			9.09
<b>β</b>	HASH JOIN		15	15M	856K		4	16M	277MB	<b>66</b>			6.06
6	- PX RECEIVE		16	1,000K	5,820		4	4,000K					
<u>ß</u>	- PX SEND BROADCAST	:TQ10000	17	1,000K	5,820		4	4,000K					
<u>β</u>			18	1,000K	5,820		4	1,000K					
ĥ	TABLE ACCESS STORAGE FULL	SUPPLIERS	19	1,000K	5,820		54	1,000K	18MB	<del>66</del>	1,844	601MB	
<b>6</b>	- HASH JOIN		20	16M	850K		4	17M	17MB	<u> 66</u>			
<b>β</b>	JOIN FILTER CREATE	:BF0000	21	1,435	13		4	6,000		<u> 66</u>			
3	MAT_VIEW REWRITE ACCESS STORAGE FU	DEP_STORE_LOC_MV	22	1,435	13		4	6,000					
۵ ۵		:BF0000	23	147M	850K		4	19M					
<b>\$</b>			24	147M	850K		4	19M					
<b>\$</b>	TABLE ACCESS STORAGE FULL	SALES	25	147M	850K		91	19M	48MB	<b>6</b>	265K	86GB	24



#### With extended statistics

#### Monitored SQL Execution Details: azmv8ma1147ff 🥪

etails				•	•								
Plan Statistics	a 🐻 Plan 👸 Parallel 📐 Activity 📜 Me	rics		orig	Inal	query							
							_						
Plan Hash Value	3112689538   Plan Note   All Parallel Servers	•											
peration		Name	Line ID	Estimated Rows	Cost	Timeline(4s)	Executions	Actual Rows				ytes	Activity %
E-SELECT STAT	TEMENT		0				9	10	Fxecut	ion ti	me 4s		
	DINATOR		1				9	10	LACCUL				
PX SEND	QC (RANDOM)	:TQ10004	2	71	1,198K		4	10					
🖻 HASH	GROUP BY		3	71	1,198K		4	10	6MB				
	RECEIVE		4	71	1,198K		4	40					
E PX	SEND HASH	:TQ10003	5	71	1,198K		4	40		<b>6</b>			
<b>—</b>	HASH GROUP BY		6	71	1,198K		4	40	17MB				18
Ē	- HASH JOIN		7	15M	1,198K		4	16M	964MB	<u> </u>			
	- PX RECEIVE		8	20M	343K	—	4	20M					
	- PX SEND HYBRID HASH	:TQ10001	9	20M	343K		4	20M		<u> </u>			3.03
	- STATISTICS COLLECTOR		10				4	20M					
	- PX BLOCK ITERATOR		11	20M	343K	_	4	20M					
	TABLE ACCESS STORAGE FULL	PRODUCTS	12	20M	343K		55	20M	48MB	<b>66</b>	🗕 94К	<b>31GB</b>	9.09
	PX RECEIVE		13	15M	856K		4	16M					
	- PX SEND HYBRID HASH	:TQ10002	14	15M					i	<u> </u>			9.09
	HASH JOIN		15	15M						66			6.06
	- PX RECEIVE		16	1,000K		Rewrite to	ook nla	ice and	l the				
	- PX SEND BROADCAST	:TQ10000	17	1,000K									
			18	1,000K	C	ardinality is	estim	ated co	nrectly				
	TABLE ACCESS STORAGE FULL	SUPPLIERS	19	1,000K		aramancy is	- country		rectiy	<b>8</b>	1,844	601MB	
	- HASH JOIN		20	16M						<u> </u>			
	JOIN FILTER CREATE	:BF0000	21	1,435	13		4	6,000		<u> </u>			
	MAT_VIEW REWRITE ACCESS STORAGE FU	DEP_STORE_LOC_MV	22	1,435	13		4	6,000					
		:BF0000	23	147M	850K		4	19M					
			24	147M	850K		4	19M					
	TABLE ACCESS STORAGE FULL	SALES	25	147M	850K		91	19M	48MB	<b>1</b>	265K	86GB	24



### End Demo2



#### Data Model







///X



///



### **One more thing...**



**Materialized View; Additional Consideration** 

- Since the original join cardinality was wrong, its cost is also wrong
- The cost could well be lower than that of the MV-based plan
- May well have to set the following parameter QUERY\_REWRITE\_ENABLED=FORCE



### To Summarize...



Copyright © 2019 Oracle and/or its affiliates.

**Summary: Avoid Guesswork** 

- When things go wrong, work to find the root cause.
- When you have root cause, you then have a platform to recommend or deliver a robust solution



#### Summary: Data Model

Data model is important

The benefits of a good data model cannot be overstated

### You might not always have a perfect dimensional model

Not designed that way

Coercing some other system/model for data warehousing

### A poor data model can lead to sub-optimal execution plans

Wrong distribution methods

Large amounts of TEMP used

etc

Often these plans cannot take advantage of the performance features



Copyright © 2019 Oracle and/or its affiliates

#### Summary: Data Model

 In certain cases, data characteristics mean that statistics (including advanced statistic types) are not sufficient to estimate good join cardinality

Dynamic statistics may or may not help.

• A materialized view provides additional information to the optimizer, resulting in a good execution plan

The performance benefit in this example comes from the better execution plan, not the materializing of the joins



**Summary: Benefits of the Materialized View** 

- No change to the underlying data model
- No change to the code
- The MV's that address this type of problem are typically built on DIMENSIONS

Generally small compared to the FACT table Slowly changing

Opens up the possibility of additional optimizations

Eg. Column Groups

• Leverage

Can benefit many different queries



Copyright © 2019 Oracle and/or its affiliates

**Summary: Materialized View** 

- Another technique in your Performance Engineering arsenal
- Learn when/where/how to use it



### Thank You

### Robert Carlin Mihajlo Tekic

Real-World Performance Team Oracle Database Development

#### Safe Harbor

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Statements in this presentation relating to Oracle's future plans, expectations, beliefs, intentions and prospects are "forward-looking statements" and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle's Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading "Risk Factors." These filings are available on the SEC's website or on Oracle's website at <a href="http://www.oracle.com/investor">http://www.oracle.com/investor</a>. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.