SQL Query Tuning

Rules to live by

Patricia Mariani
Univera Healthcare

November 5, 2012

univera HEALTHCARE
an excellus company

About me….

- Employed with Univera/Excellus for over 6 years
- Oracle DBA for 15+ years
- OCA/OCP in 10g
- Finds SQL Tuning “fun”
- Worked with small (<10g) and large (>5T) databases
- Support both OLTP and DSS systems

Top 10 pieces of advice for sql tuning

10) Having too many indexes is sometimes worse than not enough indexes

9) The inner table in an execution plan on a Nested Loop operation should rarely be a full table scan

8) If your query is not partition pruning, make sure to verify the data type of the column you are evaluating with a literal

7) Just because there is an index on a column, and you are restricting the data in the predicate, does not mean the index will be used

6) Distributed queries using DB Links can cause unexpected performance problems

5) Setting a degree of parallel on a table or index can dramatically change the execution plan, sometimes not for the better

4) Just because you set a degree of parallel for your objects (or use a hint) doesn't mean it will run in parallel

3) Moving a table from the FROM clause to the SELECT clause (when only one column is selected) can improve performance dramatically

2) A Merge join Cartesian operation in your execution plan is not always caused by a missing table join

univera HEALTHCARE

And the number 1 piece of advice for SQL Tuning…

Don’t attempt to tune anything if statistics haven’t been run!

univera HEALTHCARE

Background

- Most examples are from our data warehouse environment
  - 5T in size
  - Oracle 11gr2 (11.2.0.3)
  - 64g allocated to memory_target
- Other examples taken from our ODS
  - 440G in size
  - Oracle 11gr1 (11.1.0.7)
  - 17g allocated to memory_target

univera HEALTHCARE

#10 Original Query – Uses an inappropriate index on a highly skewed column (RATING_RGN_DTL_DIM_SK)

```
SELECT T0.C0 Claim_Source_Application_Code, T0.C1 Rating_Region, T0.C2 Paid_YYYYMM_Text, T0.C3 Claim_Payable_To_Code,
   T0.C4 Claim_Payable_To, T0.C5 Claim_Type, T0.C6 Claim_Type_Category, T0.C7 Service_Claim_Type, T0.C8 Claim_Expense,
   T0.C9 ITS_Access_Fee, SUM(T0.C8) over() Claim_Expense11, SUM(T0.C9) over() ITS_Access_Fee12
FROM
  (SELECT Claim_Line_FACT.SRC_APPL_CODE C0, Rating_Region_Detail_Dim.RATING_RGN_NAME C1,
     Claim_Line_Paid_Date_Dim.YEAR_MNTH_TEXT C2, Claim_WDim.CLM_PAYBL_TO_CODE C3,
     Claim_WDim.CLM_PAYBL_TO_CODE_NAME C4,  Claim_WDim.CLM_TYPE_CODE_NAME C5,
     Claim_WDim.CLM_TYPE_CAT_CODE_NAME C6, Service_Dim.SRVC_CLM_TYPE_CODE_DESC C7,
     SUM(Claim_Line_FACT.CLM_LINE_EXPNS_AMT) C8, SUM(Claim_Line_FACT.ITS_ACS_FEE_AMT) C9
   FROM EDW.T_CLAIM_LINE_FACT Claim_Line_FACT, EDW.T_RATING_REGION_DETAIL_DIM Rating_Region_Detail_Dim,
        EDW.T_DATE_DIM Claim_Line_Paid_Date_Dim, EDW.T_CLAIM_WDIM Claim_WDim, EDW.T_SERVICE_DIM Service_Dim
   WHERE Claim_Line_FACT.SRC_APPL_CODE IN ('LRSP')
   AND Claim_Line_Paid_Date_Dim.YEAR_MNTH_TEXT IN ('201201', '201202', '201203')
   AND Rating_Region_Detail_Dim.RATING_RGN_NAME IN ('ROCHESTER')
   AND Claim_WDim.CLM_WDIM_SK = Claim_Line_FACT.CLM_WDIM_SK
   AND Claim_Line_Paid_Date_Dim.DATE_SK = Claim_Line_FACT.CLM_LINE_PAID_DATE_SK
   AND Rating_Region_Detail_Dim.RATING_RGN_DTL_DIM_SK = Claim_Line_FACT.RATING_RGN_DTL_DIM_SK
   AND Service_Dim.SRVC_DIM_SK = Claim_Line_FACT.SRVC_DIM_SK
   GROUP BY Claim_Line_FACT.SRC_APPL_CODE, Rating_Region_Detail_Dim.RATING_RGN_NAME,
     Claim_Line_Paid_Date_Dim.YEAR_MNTH_TEXT, Claim_WDim.CLM_PAYBL_TO_CODE,
     Claim_WDim.CLM_PAYBL_TO_CODE_NAME, Claim_WDim.CLM_TYPE_CODE_NAME,
     Claim_WDim.CLM_TYPE_CAT_CODE_NAME, Service_Dim.SRVC_CLM_TYPE_CODE_DESC) T0
```

univera HEALTHCARE

			Cardinality	Cost	Partition Start	Partition Stop	Temp space
SELECT STATEMENT, GOAL = ALL_ROWS			1528	5206229			
WINDOW BUFFER			1528	5206229			
HASH GROUP BY			1528	5206229			
HASH JOIN			676731	5205452			71468000
TABLE ACCESS FULL	EDWA	T_SERVICE_DIM	2646128	1478			
HASH JOIN			676731	5201270			6357000
TABLE ACCESS FULL	EDWA	T_CLAIM_WDIM	126544	157			
HASH JOIN			810967	5199985			
NESTED LOOPS			2267	104			
INLIST ITERATOR							
TABLE ACCESS BY INDEX ROWID	EDWA	T_DATE_DIM	91	21			
BITMAP CONVERSION TO ROWIDS							
BITMAP INDEX SINGLE VALUE	EDWA	XIE15_DATE_DIM					
TABLE ACCESS FULL	EDWA	T_RATING_REGION_DETAIL_DIM	25	1			
PARTITION RANGE SUBQUERY			179745174	5176245	KEY(SUBQUERY)	KEY(SUBQUERY)	
TABLE ACCESS BY LOCAL INDEX ROWID	EDW	T_CLAIM_LINE_FACT	179745174	5176245	KEY(SUBQUERY)	KEY(SUBQUERY)	
BITMAP CONVERSION TO ROWIDS							
BITMAP INDEX SINGLE VALUE	EDW	XIE53_CLAIM_LINE_FACT			KEY(SUBQUERY)	KEY(SUBQUERY)	

univera HEALTHCARE

Query executes 2 minutes

Performed alter index edw.xie53_claim_line_fact invisible;
(use of /*+ no_index (Claim_Line_FACT xie53_claim_line_fact) */ hint is also an option)

			Cardinality	Cost	Partition Start	Partition Stop	Temp space
SELECT STATEMENT, GOAL = ALL_ROWS			1528	3165971			
WINDOW BUFFER			1528	3165971			
HASH GROUP BY			1528	3165971			
HASH JOIN			676731	3165193			71468000
TABLE ACCESS FULL	EDWA	T_SERVICE_DIM	2646128	1478			
HASH JOIN			676731	3161012			6357000
TABLE ACCESS FULL	EDWA	T_CLAIM_WDIM	126544	157			
HASH JOIN			810967	3159727			
PART JOIN FILTER CREATE	SYS	:BF0000	2267	104			
NESTED LOOPS			2267	104			
INLIST ITERATOR							
TABLE ACCESS BY INDEX ROWID	EDWA	T_DATE_DIM	91	21			
BITMAP CONVERSION TO ROWIDS							
BITMAP INDEX SINGLE VALUE	EDWA	XIE15_DATE_DIM					
TABLE ACCESS FULL	EDWA	T_RATING_REGION_DETAIL_DIM	25	1			
PARTITION RANGE JOIN-FILTER			179745174	3135986	:BF0000	:BF0000	
TABLE ACCESS FULL	EDW	T_CLAIM_LINE_FACT	179745174	3135986	:BF0000	:BF0000	

Query executes 31 seconds

#9 Original Query – Nested loops and full table scans

```
SELECT W.CMS_MBR_ATRBT_WDIM_SK AS CMS_MBR_ATRBT_WDIM_SK, M.CMS_MBR_SK AS CMS_MBR_SK, S.SBMT_DATE_SK AS SBMT_DATE_SK,
  M.EFF_DATE_SK AS EFF_DATE_SK, M.EFF_THRU_DATE_SK AS EFF_THRU_DATE_SK
FROM IDL.T_CMS_MEMBER_ATTRIBUTE M, STAGING.T_CMS_RAPS_SBMT_DT S, MRA.T_CMS_MEMBER_ATTRIBUTE_WDIM W
WHERE S.SBMT_DATE_SK BETWEEN M.OBS_BGN_DATE_SK AND M.OBS_END_DATE_SK
AND M.PART_D_LOW_INCOME_CODE_SK = W.PART_D_LOW_INCOME_CODE_SK
AND NVL(M.PART_D_LOW_INCOME_MLTPLR_PCT, 0) = NVL(W.PART_D_LOW_INCOME_MLTPLR_PCT, 0)
AND NVL(M.PART_D_LTI_MLTPLR_PCT, 0) = NVL(W.PART_D_LTI_MLTPLR_PCT, 0)
AND M.MCAID_IND = W.MCAID_IND
AND M.DSBL_IND = W.DSBL_IND
AND M.AGE_DSBL_MSP_IND = W.AGE_DSBL_MSP_IND
AND M.HSPC_IND = W.HSPC_IND
AND M.EMPLYR_GRP_HLTH_PLAN_IND = W.EMPLYR_GRP_HLTH_PLAN_IND
AND M.MDCR_PART_A_ENTLMNT_IND = W.MDCR_PART_A_ENTLMNT_IND
AND M.MDCR_PART_B_ENTLMNT_IND = W.MDCR_PART_B_ENTLMNT_IND
AND M.LTI_IND = W.PART_C_LTI_IND
AND M.PART_D_LTI_CODE_SK = W.PART_D_LTI_CODE_SK
AND M.ESRD_MSP_IND = W.ESRD_MSP_IND
AND M.ENTLMNT_RSN_CODE_SK = W.ENTLMNT_RSN_CODE_SK
AND M.LOW_INCOME_CPY_CAT_CODE_SK = W.LOW_INCOME_CPY_CAT_CODE_SK
AND M.PART_D_LI_PREM_SUB_LVL_CODE_SK = W.PART_D_LI_PREM_SUB_LVL_CODE_SK
AND M.ESRD_IND = W.ESRD_IND
AND M.INST_IND = W.INST_IND
AND M.NRSNG_HOME_CRTFBL_IND = W.NRSNG_HOME_CRTFBL_IND
AND M.RISK_ADJ_FCTR_TYPE_CODE_SK = W.RISK_ADJ_FCTR_TYPE_CODE_SK
AND M.DFLT_RISK_ADJ_FCTR_CODE_SK = W.DFLT_RISK_ADJ_FCTR_CODE_SK
AND M.MCARE_BNFCRY_MCAID_STS_CODE_SK = W.MDCR_BNFCRY_MCAID_STS_CODE_SK
AND M.PART_D_ENRL_FLAG_CODE_SK = W.PART_D_ENRL_CODE_SK
AND NVL(M.DFLT_RISK_ADJ_FCTR_IND, 'N') = NVL(W.DFLT_RISK_ADJ_FCTR_IND, 'N')
AND NVL(M.MSP_FCTR, 0) = NVL(W.MSP_FCTR, 0)
AND M.MCAID_DUAL_STS_CODE_SK = W.MCAID_DUAL_STS_CODE_SK
AND M.PART_D_RISK_ADJ_FCTR_TYP_CD_SK = W.PART_D_RISK_ADJ_FCTR_TYP_CD_SK
AND m.dflt_part_d_risk_fctr_cd_sk = w.dflt_part_d_risk_fctr_cd_sk
ORDER BY CMS_MBR_SK, SBMT_DATE_SK, EFF_DATE_SK, EFF_THRU_DATE_SK, CMS_MBR_ATRBT_WDIM_SK
```

univera HEALTHCARE

SELECT STATEMENT, GOAL = ALL_ROWS			12051139	263432	
SORT ORDER BY			12051139	263432	3871540000
HASH JOIN			12051139	16995	15762000
NESTED LOOPS			125004	357	
TABLE ACCESS FULL	STAGING	T_CMS_RAPS_SBMT_DT	6	2	
TABLE ACCESS FULL	MRA	T_CMS_MEMBER_ATTRIBUTE_WDIM	20834	59	
TABLE ACCESS FULL	IDL	T_CMS_MEMBER_ATTRIBUTE	4097600	6083	

#8 Original Query – Data type conversion preventing partition pruning
Run time reduced from 5+ hours to 90 minutes

```
SELECT T1.SRC_APPL_CODE, FLOOR(T1.CLM_SRVC_EFF_DATE_SK/100) AS INCYYYYMM, FLOOR(T1.CLM_LINE_PAID_DATE_SK//100) AS PAIDYYYYMM,
  T5.RATING_RGN_NAME, T3.GRP_ID, T3.GRP_NAME, T3.SUBGRP_ID, T3.SUBGRP_NAME, T3.LGCY_GRP_ID, T4.RISK_CAT_CODE, T4.RISK_CAT_NAME,
  T4.MJR_RISK_CLS_CODE, T4.MJR_RISK_CLS_NAME, T2.PROD_NAME, T2.PROD_CLS, T6.MSTR_RATING_POOL_ID, T6.MSTR_RATING_POOL_NAME,
  T7.CLM_TYPE_CODE, T7.CLM_TYPE_CODE_NAME, T7.CLM_TYPE_CAT_CODE, T7.CLM_TYPE_CAT_CODE_NAME, T8.DNTL_BNFT_CLS_CODE,
  T8.DNTL_BNFT_CLS_CODE_NAME, T8.DNTL_TRMNLGY_TOS_CODE, T8.DNTL_TRMNLGY_TOS_CODE_NAME, T11.MI_CARRIER_CMS_PART_D_CODE,
  SUM(T1.CLM_LINE_PAID_AMT) AS CLM_LINE_PAID_AMT, SUM(T1.MLPRCTC_AMT) AS MLPRCTC_AMT,
  SUM(T1.RISK_WTHLDNG_AMT) AS RISK_WTHLDNG_AMT, SUM(T1.GME_AMT) AS GME_AMT, SUM(T1.ITS_ACS_FEE_AMT) AS ITS_ACS_FEE_AMT,
  SUM(T1.ITS_SRCHRG_AMT) AS ITS_SRCHRG_AMT, SUM(T1.BDCC_AMT) AS BDCC_AMT, SUM(T1.CLM_LINE_EXPNS_AMT) AS CLM_LINE_EXPNS_AMT,
  SUM(T1.ALW_AMT) AS ALW_AMT, SUM(T1.DDCTBL_AMT) AS DDCTBL_AMT, SUM(T1.CPY_AMT) AS CPY_AMT
FROM EDW.T_CLAIM_LINE_FACT T1, EDW.T_PRODUCT_DIM T2, EDW.T_GROUP_ACCOUNT_DIM T3,
    EDW.T_RISK_CATEGORY_DIM T4, EDW.T_RATING_REGION_DETAIL_DIM T5,
    EDW.T_MASTER_RATING_POOL_DIM T6, EDW.T_CLAIM_WDIM T7, EDW.T_DENTAL_WDIM T8,
    EDW.V_CLM_LINE_DISP_CODE T9, EDW.T_DATE_DIM T10, EDW.T_MEDIMPACT_WDIM T11
WHERE T10.YEAR_NBR_TEXT = 2012    --- Literal requires single quotes to prevent automatic type conversion
AND T9.TRGT_CLM_LINE_DISP = 'P'
AND (T4.MJR_RISK_CLS_CODE IN ('SF', 'ER', 'MP') OR T2.PROD_CLS = 'DENTAL')
AND T10.DATE_SK = T1.CLM_LINE_PAID_DATE_SK
AND T9.CLM_LINE_DISP_CODE_SK = T1.CLM_LINE_DISP_CODE_SK
AND T4.RISK_CAT_DIM_SK = T1.RISK_CAT_DIM_SK
AND T2.PROD_DIM_SK = T1.PROD_DIM_SK
AND T3.GRP_ACNT_DIM_SK = T1.GRP_ACNT_DIM_SK
AND T5.RATING_RGN_DTL_DIM_SK = T1.RATING_RGN_DTL_DIM_SK
AND T6.MSTR_RATING_POOL_DIM_SK = T1.MSTR_RATING_POOL_DIM_SK
AND T7.CLM_WDIM_SK = T1.CLM_WDIM_SK
AND T8.DNTL_WDIM_SK = T1.DNTL_WDIM_SK
AND T1.MI_WDIM_SK = T11.MI_WDIM_SK
GROUP BY T1.SRC_APPL_CODE, FLOOR(T1.CLM_SRVC_EFF_DATE_SK/100), FLOOR(T1.CLM_LINE_PAID_DATE_SK/100),
  T5.RATING_RGN_NAME, T3.GRP_ID, T3.GRP_NAME, T3.SUBGRP_ID, T3.SUBGRP_NAME, T3.LGCY_GRP_ID,
  T4.RISK_CAT_CODE, T4.RISK_CAT_NAME, T4.MJR_RISK_CLS_CODE, T4.MJR_RISK_CLS_NAME, T2.PROD_NAME, T2.PROD_CLS,
  T6.MSTR_RATING_POOL_ID, T6.MSTR_RATING_POOL_NAME, T7.CLM_TYPE_CODE, T7.CLM_TYPE_CODE_NAME,
  T7.CLM_TYPE_CAT_CODE, T7.CLM_TYPE_CAT_CODE_NAME, T8.DNTL_BNFT_CLS_CODE, T8.DNTL_BNFT_CLS_CODE_NAME,
  T8.DNTL_TRMNLGY_TOS_CODE, T8.DNTL_TRMNLGY_TOS_CODE_NAME, T11.MI_CARRIER_CMS_PART_D_CODE
```

univera HEALTHCARE

			Partition Start	Partition Stop
SELECT STATEMENT, GOAL = ALL_ROWS				
HASH GROUP BY				
NESTED LOOPS				
NESTED LOOPS				
HASH JOIN				
NESTED LOOPS				
NESTED LOOPS				
NESTED LOOPS				
NESTED LOOPS				
HASH JOIN				
HASH JOIN				
HASH JOIN				
HASH JOIN				
HASH JOIN				
TABLE ACCESS BY INDEX ROWID	IDL	T_CODE_MAPPING		
BITMAP CONVERSION TO ROWIDS				
BITMAP OR				
BITMAP CONVERSION FROM ROWIDS				
SORT ORDER BY				
INDEX RANGE SCAN	IDL	XAK1_CODE_MAPPING		
BITMAP CONVERSION FROM ROWIDS				
SORT ORDER BY				
INDEX RANGE SCAN	IDL	XPK_CODE_MAPPING		
PARTITION RANGE ALL			1	95
TABLE ACCESS FULL	EDW	T_CLAIM_LINE_FACT	1	95
TABLE ACCESS FULL	EDWA	T_DENTAL_WDIM		
TABLE ACCESS FULL	EDWA	T_PRODUCT_DIM		
TABLE ACCESS FULL	EDWA	T_RISK_CATEGORY_DIM		
TABLE ACCESS FULL	EDWA	T_MASTER_RATING_POOL_DIM		
TABLE ACCESS BY INDEX ROWID	EDWA	T_DATE_DIM		
INDEX UNIQUE SCAN	EDWA	XPK_DATE_DIM		
TABLE ACCESS BY INDEX ROWID	EDWA	T_CLAIM_WDIM		
INDEX UNIQUE SCAN	EDWA	XPK_CLAIM_WDIM		
INDEX UNIQUE SCAN	EDWA	XPK_GROUP_ACCOUNT_DIM		
TABLE ACCESS BY INDEX ROWID	EDWA	T_GROUP_ACCOUNT_DIM		
TABLE ACCESS FULL	EDWA	T_RATING_REGION_DETAIL_DIM		
INDEX UNIQUE SCAN	EDWA	XPK_MEDIMPACT_WDIM		
TABLE ACCESS BY INDEX ROWID	EDWA	T_MEDIMPACT_WDIM		

Since the year_nbr_text column is a character field, partition pruning on the partitioned table will not occur because of automatic type conversion

SELECT STATEMENT, GOAL = ALL_ROWS			Partition Start	Partition Stop
HASH GROUP BY				
NESTED LOOPS				
NESTED LOOPS				
HASH JOIN				
NESTED LOOPS				
NESTED LOOPS				
NESTED LOOPS				
NESTED LOOPS				
HASH JOIN				
HASH JOIN				
TABLE ACCESS FULL	EDWA	T_PRODUCT_DIM		
HASH JOIN				
HASH JOIN				
HASH JOIN				
TABLE ACCESS BY INDEX ROWID	IDL	T_CODE_MAPPING		
BITMAP CONVERSION TO ROWIDS				
BITMAP OR				
BITMAP CONVERSION FROM ROWIDS				
SORT ORDER BY				
INDEX RANGE SCAN	IDL	XAK1_CODE_MAPPING		
BITMAP CONVERSION FROM ROWIDS				
SORT ORDER BY				
INDEX RANGE SCAN	IDL	XPK_CODE_MAPPING		
PARTITION RANGE SUBQUERY			KEY(SUBQUERY)	KEY(SUBQUERY)
TABLE ACCESS FULL	EDW	T_CLAIM_LINE_FACT	KEY(SUBQUERY)	KEY(SUBQUERY)
TABLE ACCESS FULL	EDWA	T_DENTAL_WDIM		
TABLE ACCESS FULL	EDWA	T_RISK_CATEGORY_DIM		
TABLE ACCESS FULL	EDWA	T_MASTER_RATING_POOL_DIM		
TABLE ACCESS BY INDEX ROWID	EDWA	T_DATE_DIM		
INDEX UNIQUE SCAN	EDWA	XPK_DATE_DIM		
TABLE ACCESS BY INDEX ROWID	EDWA	T_CLAIM_WDIM		
INDEX UNIQUE SCAN	EDWA	XPK_CLAIM_WDIM		
INDEX UNIQUE SCAN	EDWA	XPK_GROUP_ACCOUNT_DIM		
TABLE ACCESS BY INDEX ROWID	EDWA	T_GROUP_ACCOUNT_DIM		
TABLE ACCESS FULL	EDWA	T_RATING_REGION_DETAIL_DIM		
INDEX UNIQUE SCAN	EDWA	XPK_MEDIMPACT_WDIM		
TABLE ACCESS BY INDEX ROWID	EDWA	T_MEDIMPACT_WDIM		

By enclosing the year_nbr_text field in single quotes, partition pruning will now occur.

#7 Original Query – Index on column not being used

column PART_D_LOW_INCOME_MLTPLR_PCT is indexed

```
SELECT W.CMS_MBR_ATRBT_WDIM_SK AS CMS_MBR_ATRBT_WDIM_SK, M.CMS_MBR_SK AS CMS_MBR_SK, S.SBMT_DATE_SK AS SBMT_DATE_SK,
  M.EFF_DATE_SK AS EFF_DATE_SK, M.EFF_THRU_DATE_SK AS EFF_THRU_DATE_SK
FROM IDL.T_CMS_MEMBER_ATTRIBUTE M, STAGING.T_CMS_RAPS_SBMT_DT S, MRA.T_CMS_MEMBER_ATTRIBUTE_WDIM W
WHERE S.SBMT_DATE_SK BETWEEN M.OBS_BGN_DATE_SK AND M.OBS_END_DATE_SK
AND M.PART_D_LOW_INCOME_CODE_SK = W.PART_D_LOW_INCOME_CODE_SK
AND NVL(W.PART_D_LOW_INCOME_MLTPLR_PCT, 0) = NVL(M.PART_D_LOW_INCOME_MLTPLR_PCT, 0)
AND NVL(M.PART_D_LTI_MLTPLR_PCT, 0) = NVL(W.PART_D_LTI_MLTPLR_PCT, 0)
AND M.MCAID_IND = W.MCAID_IND
AND M.DSBL_IND = W.DSBL_IND
AND M.AGE_DSBL_MSP_IND = W.AGE_DSBL_MSP_IND
AND M.HSPC_IND = W.HSPC_IND
AND M.EMPLYR_GRP_HLTH_PLAN_IND = W.EMPLYR_GRP_HLTH_PLAN_IND
AND M.MDCR_PART_A_ENTLMNT_IND = W.MDCR_PART_A_ENTLMNT_IND
AND M.MDCR_PART_B_ENTLMNT_IND = W.MDCR_PART_B_ENTLMNT_IND
AND M.LTI_IND = W.PART_C_LTI_IND
AND M.PART_D_LTI_CODE_SK = W.PART_D_LTI_CODE_SK
AND M.ESRD_MSP_IND = W.ESRD_MSP_IND
AND M.ENTLMNT_RSN_CODE_SK = W.ENTLMNT_RSN_CODE_SK
AND M.LOW_INCOME_CPY_CAT_CODE_SK = W.LOW_INCOME_CPY_CAT_CODE_SK
AND M.PART_D_LI_PREM_SUB_LVL_CODE_SK = W.PART_D_LI_PREM_SUB_LVL_CODE_SK
AND M.ESRD_IND = W.ESRD_IND
AND M.INST_IND = W.INST_IND
AND M.NRSNG_HOME_CRTFBL_IND = W.NRSNG_HOME_CRTFBL_IND
AND M.RISK_ADJ_FCTR_TYPE_CODE_SK = W.RISK_ADJ_FCTR_TYPE_CODE_SK
AND M.DFLT_RISK_ADJ_FCTR_CODE_SK = W.DFLT_RISK_ADJ_FCTR_CODE_SK
AND M.MCARE_BNFCRY_MCAID_STS_CODE_SK = W.MDCR_BNFCRY_MCAID_STS_CODE_SK
AND M.PART_D_ENRL_FLAG_CODE_SK = W.PART_D_ENRL_CODE_SK
AND NVL(M.DFLT_RISK_ADJ_FCTR_IND, 'N') = NVL(W.DFLT_RISK_ADJ_FCTR_IND, 'N')
AND NVL(M.MSP_FCTR, 0) = NVL(W.MSP_FCTR, 0)
AND M.MCAID_DUAL_STS_CODE_SK = W.MCAID_DUAL_STS_CODE_SK
AND M.PART_D_RISK_ADJ_FCTR_TYP_CD_SK = W.PART_D_RISK_ADJ_FCTR_TYP_CD_SK
AND m.dflt_part_d_risk_fctr_cd_sk = w.dflt_part_d_risk_fctr_cd_sk
ORDER BY CMS_MBR_SK, SBMT_DATE_SK, EFF_DATE_SK, EFF_THRU_DATE_SK, CMS_MBR_ATRBT_WDIM_SK
```

univera HEALTHCARE

SELECT STATEMENT, GOAL = ALL_ROWS			12051139	263432	
SORT ORDER BY			12051139	263432	3871540000
HASH JOIN			12051139	16995	15762000
NESTED LOOPS			125004	357	
TABLE ACCESS FULL	STAGING	T_CMS_RAPS_SBMT_DT	6	2	
TABLE ACCESS FULL	MRA	T_CMS_MEMBER_ATTRIBUTE_WDIM	20834	59	
TABLE ACCESS FULL	IDL	T_CMS_MEMBER_ATTRIBUTE	4097600	6083	

#6 Distributed transactions with DB Links

```
SELECT DISTINCT to_char(mpe.MEME_CK), mpe.CSPD_CAT, to_char(mpe.MEPE_EFF_DT, 'YYYYMMDD'),
  CASE mpe.PDPD_ID WHEN ' ' THEN 'UNKNOWN‘ ELSE mpe.PDPD_ID END, nvl(pln.cspi_id, ' '), ), to_char(mpe.SGSG_CK),
  to_char(mpe.MEPE_TERM_DT, 'YYYYMMDD'), to_char(cmm.SBSB_CK), to_char(mpe.MEPE_CREATE_DTM, 'YYYYMMDD'), nvl(cls.CSCS_ID, ' '),
  to_char(mpe.MEPE_PLAN_ENTRY_DT, 'YYYYMMDD'), nvl(ccp.GPAI_ID, ' '), nvl(cpp.LOBD_ID, ' '), nvl(merd.merd_units, 0)
FROM facetcor.cmc_mepe_prcs_elig@frepl mpe, facetcor.cmc_meme_member@frepl cmm,
    facetcor.cmc_pdpd_product@frepl cpp, facetcor.CMC_PLDS_PLAN_DESC@frepl pln,
    facetcor.CMC_CSCS_CLASS@frepl cls, facetcor.cmc_cspi_cs_plan@frepl ccp,
    facetcor.cmc_merd_rate_data@frepl merd
where mpe.MEME_CK = cmm.MEME_CK(+)
and mpe.PDPD_ID = cpp.PDPD_ID(+)
and mpe.CSPI_ID = pln.cspi_id(+)
and mpe.CSCS_ID = cls.CSCS_ID(+)
and mpe.GRGR_CK = ccp.GRGR_CK(+) AND mpe.CSCS_ID = ccp.CSCS_ID(+)
AND mpe.CSPD_CAT = ccp.CSPD_CAT(+) AND mpe.CSPI_ID = ccp.CSPI_ID(+)
and mpe.MEME_CK = merd.meme_ck(+) AND mpe.GRGR_CK = merd.GRGR_CK(+)
AND mpe.CSPI_ID = merd.CSPI_ID(+) AND merd.merd_type = 'EU' AND mpe.CSPD_CAT = 'R'
AND merd.merd_eff_dt(+) BETWEEN mpe.MEPE_EFF_DT AND mpe.MEPE_TERM_DT
MINUS
SELECT to_char(md.mem_srckey), pcd.prod_cat_cd, to_char(ef.elig_dt_key), pd.prod_id, pld.class_plan_id, to_char(gsd.grp_srckey),
  to_char(gsd.subgrp_srckey), to_char(ef.rec_cncl_dt_key), to_char(md.sub_srckey), to_char(ef.creatd_dt_key), cd.class_id,
  to_char(ef.plan_entry_dt_key), gad.group_admin_id, CASE lobd.lob_id WHEN '-1' THEN ' ‘ ELSE lobd.lob_id END, ef.mdcr_part_d_uncovrd_months
FROM hpxr.eligibility_fact ef
LEFT JOIN hpxr.member_dim md ON (ef.mem_key = md.mem_key)
LEFT JOIN hpxr.prod_category_dim pcd ON (ef.prod_cat_key = pcd.prod_cat_key)
LEFT JOIN hpxr.product_dim pd ON (ef.prod_key = pd.prod_key)
LEFT JOIN hpxr.plan_dim pld ON (ef.plan_key = pld.plan_key)
LEFT JOIN hpxr.group_subgrp_dim gsd ON (ef.grp_subgrp_key = gsd.grp_subgrp_key)
LEFT JOIN hpxr.class_dim cd ON (ef.class_key = cd.class_key)
LEFT JOIN hpxr.lineofbus_dim lobd ON (ef.lob_key = lobd.lob_key)
LEFT JOIN hpxr.group_admin_dim gad ON (ef.group_admin_key = gad.group_admin_key)
```

SELECT STATEMENT, GOAL = ALL_ROWS		
MINUS		
SORT UNIQUE		
HASH JOIN RIGHT OUTER		
REMOTE		CMC_CSCS_CLASS
NESTED LOOPS OUTER		
HASH JOIN RIGHT OUTER		
REMOTE		CMC_PDPD_PRODUCT
HASH JOIN RIGHT OUTER		
REMOTE		CMC_PLDS_PLAN_DESC
REMOTE		
REMOTE		CMC_MEME_MEMBER
SORT UNIQUE		
HASH JOIN RIGHT OUTER		
TABLE ACCESS FULL	HPXRA	CLASS_DIM
HASH JOIN RIGHT OUTER		
TABLE ACCESS FULL	HPXRA	PRODUCT_DIM
HASH JOIN RIGHT OUTER		
TABLE ACCESS FULL	HPXRA	PLAN_DIM
HASH JOIN RIGHT OUTER		
TABLE ACCESS FULL	HPXRA	GROUP_ADMIN_DIM
HASH JOIN RIGHT OUTER		
TABLE ACCESS FULL	HPXRA	MEMBER_DIM
HASH JOIN RIGHT OUTER		
TABLE ACCESS FULL	HPXRA	GROUP_SUBGRP_DIM
HASH JOIN RIGHT OUTER		
TABLE ACCESS FULL	HPXRA	LINEOFBUS_DIM
HASH JOIN RIGHT OUTER		
TABLE ACCESS FULL	HPXRA	PROD_CATEGORY_DIM
TABLE ACCESS FULL	HPXRA	ELIGIBILITY_FACT

#6 Distributed transactions with DB Links

```
SELECT /*+ driving_site (mpe) */ DISTINCT to_char(mpe.MEME_CK), mpe.CSPD_CAT, to_char(mpe.MEPE_EFF_DT, 'YYYYMMDD'),
  CASE mpe.PDPD_ID WHEN ' ' THEN 'UNKNOWN‘ ELSE mpe.PDPD_ID END, nvl(pln.cspi_id, ' '), ), to_char(mpe.SGSG_CK),
  to_char(mpe.MEPE_TERM_DT, 'YYYYMMDD'), to_char(cmm.SBSB_CK), to_char(mpe.MEPE_CREATE_DTM, 'YYYYMMDD'), nvl(cls.CSCS_ID, ' '),
  to_char(mpe.MEPE_PLAN_ENTRY_DT, 'YYYYMMDD'), nvl(ccp.GPAI_ID, ' '), nvl(cpp.LOBD_ID, ' '), nvl(merd.merd_units, 0)
FROM facetcor.cmc_mepe_prcs_elig@frepl mpe, facetcor.cmc_meme_member@frepl cmm,
    facetcor.cmc_pdpd_product@frepl cpp, facetcor.CMC_PLDS_PLAN_DESC@frepl pln,
    facetcor.CMC_CSCS_CLASS@frepl cls, facetcor.cmc_cspi_cs_plan@frepl ccp,
    facetcor.cmc_merd_rate_data@frepl merd
where mpe.MEME_CK = cmm.MEME_CK(+)
and mpe.PDPD_ID = cpp.PDPD_ID(+)
and mpe.CSPI_ID = pln.cspi_id(+)
and mpe.CSCS_ID = cls.CSCS_ID(+)
and mpe.GRGR_CK = ccp.GRGR_CK(+) AND mpe.CSCS_ID = ccp.CSCS_ID(+)
AND mpe.CSPD_CAT = ccp.CSPD_CAT(+) AND mpe.CSPI_ID = ccp.CSPI_ID(+)
and mpe.MEME_CK = merd.meme_ck(+) AND mpe.GRGR_CK = merd.GRGR_CK(+)
AND mpe.CSPI_ID = merd.CSPI_ID(+) AND merd.merd_type = 'EU' AND mpe.CSPD_CAT = 'R'
AND merd.merd_eff_dt(+) BETWEEN mpe.MEPE_EFF_DT AND mpe.MEPE_TERM_DT
MINUS
SELECT to_char(md.mem_srckey), pcd.prod_cat_cd, to_char(ef.elig_dt_key), pd.prod_id, pld.class_plan_id, to_char(gsd.grp_srckey),
  to_char(gsd.subgrp_srckey), to_char(ef.rec_cncl_dt_key), to_char(md.sub_srckey), to_char(ef.creatd_dt_key), cd.class_id,
  to_char(ef.plan_entry_dt_key), gad.group_admin_id, CASE lobd.lob_id WHEN '-1' THEN ' ‘ ELSE lobd.lob_id END, ef.mdcr_part_d_uncovrd_months
FROM hpxr.eligibility_fact ef
LEFT JOIN hpxr.member_dim md ON (ef.mem_key = md.mem_key)
LEFT JOIN hpxr.prod_category_dim pcd ON (ef.prod_cat_key = pcd.prod_cat_key)
LEFT JOIN hpxr.product_dim pd ON (ef.prod_key = pd.prod_key)
LEFT JOIN hpxr.plan_dim pld ON (ef.plan_key = pld.plan_key)
LEFT JOIN hpxr.group_subgrp_dim gsd ON (ef.grp_subgrp_key = gsd.grp_subgrp_key)
LEFT JOIN hpxr.class_dim cd ON (ef.class_key = cd.class_key)
LEFT JOIN hpxr.lineofbus_dim lobd ON (ef.lob_key = lobd.lob_key)
LEFT JOIN hpxr.group_admin_dim gad ON (ef.group_admin_key = gad.group_admin_key)
```

SELECT STATEMENT REMOTE, GOAL = ALL_ROWS		
MINUS		
SORT UNIQUE		
HASH JOIN OUTER		
NESTED LOOPS OUTER		
NESTED LOOPS OUTER		
HASH JOIN OUTER		
NESTED LOOPS OUTER		
HASH JOIN		
TABLE ACCESS FULL	FACETCOR	CMC_MERD_RATE_DATA
TABLE ACCESS FULL	FACETCOR	CMC_MEPE_PRCS_ELIG
INDEX UNIQUE SCAN	FACETCOR	CMCX_PLDS_PRIMARY
TABLE ACCESS FULL	FACETCOR	CMC_PDPD_PRODUCT
TABLE ACCESS BY INDEX ROWID	FACETCOR	CMC_CSPI_CS_PLAN
INDEX RANGE SCAN	FACETCOR	CMCX_CSPI_PRIMARY
TABLE ACCESS BY INDEX ROWID	FACETCOR	CMC_MEME_MEMBER
INDEX UNIQUE SCAN	FACETCOR	CMCX_MEME_PRIMARY
INDEX FAST FULL SCAN	FACETCOR	CMCX_CSCS_PRIMARY
SORT UNIQUE		
REMOTE		

#5 When parallel hurts instead of helps
Parallel query killed after 2 hours, serial query returns rows in 6 minutes

```
SELECT ….
FROM HPXR_prev.PROD_GRP_SUBGRP_RATE_FACT PGFACT_CURRENT
INNER JOIN HPXR_prev.RATE_DIM RATE_CURRENT
ON PGFACT_CURRENT.RATE_KEY = RATE_CURRENT.RATE_KEY
AND TO_NUMBER(TO_CHAR((TO_DATE(20120101, 'YYYYMMDD') - 1), 'YYYYMMDD')) BETWEEN
  PGFACT_CURRENT.PROD_GRPSBGRP_RATE_EFF_DT_KEY AND PGFACT_CURRENT.PROD_GRPSBGRP_RATE_THRU_DT_KEY
RIGHT OUTER JOIN (SELECT /*+ parallel (pgfact 4) parallel (elig 4)*/  …
FROM HPXR_prev.PROD_GRP_SUBGRP_RATE_FACT PGFACT, HPXR_prev.GROUP_SUBGRP_DIM GS,
HPXR_prev.SUBGRP_COUNTS SBC, HPXR_prev.GROUP_PACKAGE_DIM PKG, HPXR_prev.PROD_CATEGORY_DIM PRCD,
HPXR_prev.CLASS_DIM CLASS, HPXR_prev.PLAN_DIM PLN, HPXR_prev.BILLING_COMPONENT_DIM BC, HPXR_prev.PRODUCT_DIM
PRD, HPXR_prev.RATE_DIM RATE, HPXR_prev.ELIGIBILITY_FACT ELIG, HPXR_prev.ELIGIBILITY_DIM ELIGDIM,
HPXR_prev.ELIGIBILITY_FACT_EVENT_BRIDGE ELIG_EVT_BRI, HPXR_prev.ELIGIBILITY_EVENT_DIM ELIG_EVT_DIM,
HPXR_prev.MEMBER_PACKAGE_FACT MBRFACT, HPXR_prev.MEMBER_DIM MBR, HPXR_prev.LINEOFBUS_DIM LINEBUS
WHERE PGFACT.GRP_PACKAGE_KEY = PKG.GRP_PACKAGE_KEY
            AND PGFACT.SUBGRP_KEY = GS.GRP_SUBGRP_KEY
            AND GS.STS_CD = 'AC'
            AND PGFACT.CLASS_KEY = CLASS.CLASS_KEY
…
            AND LENGTH(MBRFACT.MEM_PKG_TERM_DT_KEY) = 8) MAIN_TBL
ON PGFACT_CURRENT.PROD_CAT_KEY = MAIN_TBL.PROD_CAT_KEY
AND PGFACT_CURRENT.CLASS_KEY = MAIN_TBL.CLASS_KEY
AND PGFACT_CURRENT.PLAN_KEY = MAIN_TBL.PLAN_KEY
AND PGFACT_CURRENT.PROD_KEY = MAIN_TBL.PROD_KEY
AND PGFACT_CURRENT.GRP_PACKAGE_KEY = MAIN_TBL.GRP_PACKAGE_KEY
AND PGFACT_CURRENT.SUBGRP_KEY = MAIN_TBL.GRP_SUBGRP_KEY
AND PGFACT_CURRENT.BILL_COMP_KEY = MAIN_TBL.BILL_COMP_KEY;
```

univera HEALTHCARE

SELECT STATEMENT, GOAL = ALL_ROWS			1	269	1440988574	62
RESULT CACHE		axudk38u2pbkff78s2s2v6aqt9				
PX COORDINATOR						
PX SEND QC (RANDOM)	SYS	:TQ10005	1	269	1440988574	62
HASH UNIQUE			1	269	1440988574	62
PX RECEIVE			1	268	1434011116	62
PX SEND HASH	SYS	:TQ10004	1	268	1434011116	62
NESTED LOOPS OUTER			1	268	1434011116	62
VIEW	SYSTEM		1	265	1434000050	59
NESTED LOOPS OUTER			1	265	1434000050	59
NESTED LOOPS OUTER			1	264	1433999811	59
NESTED LOOPS			1	264	1433999596	58
NESTED LOOPS			1	264	1433999346	58
NESTED LOOPS			1	263	1433999109	58
NESTED LOOPS			1	263	1433998858	58
NESTED LOOPS			1	263	1433998626	57
NESTED LOOPS			1	263	1433998382	57
NESTED LOOPS			1	262	1433998145	57
NESTED LOOPS			1	262	1433997859	56
NESTED LOOPS			1	262	1433997620	56
NESTED LOOPS			1	261	1433997388	56
NESTED LOOPS			4	257	1433936536	52
HASH JOIN			176153	58	320242489	12
BUFFER SORT						
PX RECEIVE			67246	4	16308974	2
PX SEND BROADCAST	SYS	:TQ10001	67246	4	16308974	2
TABLE ACCESS FULL	HPXRA	CLASS_DIM	67246	4	16308974	2
HASH JOIN			176153	52	293519086	10
PX RECEIVE			176154	17	82429191	5
PX SEND HASH	SYS	:TQ10003	176154	17	82429191	5
HASH JOIN			176154	17	82429191	5
BUFFER SORT						
PX RECEIVE			7	2	37367	2
PX SEND BROADCAST	SYS	:TQ10000	7	2	37367	2
TABLE ACCESS FULL	HPXRA	PROD_CATEGORY_DIM	7	2	37367	2
PX BLOCK ITERATOR			176154	14	74498894	3
TABLE ACCESS FULL	HPXRA	PROD_GRP_SUBGRP_RATE_FACT	176154	14	74498894	3
BUFFER SORT						
PX RECEIVE			92777	33	196359816	5
PX SEND HASH	SYS	:TQ10002	92777	33	196359816	5
TABLE ACCESS FULL	HPXRA	GROUP_SUBGRP_DIM	92777	33	196359816	5
TABLE ACCESS BY INDEX ROWID	HPXRA	MEMBER_PACKAGE_FACT	1	257	1433936536	52
BITMAP CONVERSION TO ROWIDS						
BITMAP AND						
BITMAP CONVERSION FROM ROWIDS						
INDEX RANGE SCAN	HPXRA	XIF4MEMBER_PACKAGE_FACT	62	1	1305	1
BITMAP CONVERSION FROM ROWIDS						
INDEX RANGE SCAN	HPXRA	XIF3MEMBER_PACKAGE_FACT	62	1	1345	1
TABLE ACCESS BY INDEX ROWID	HPXRA	ELIGIBILITY_FACT	1	4	54767	4
INDEX RANGE SCAN	HPXRA	XIF1ELIGIBILITY_FACT	281	1	5645	1
TABLE ACCESS BY INDEX ROWID	HPXRA	LINEOFBUS_DIM	1	1	834	1
INDEX UNIQUE SCAN	HPXRA	FACETS_LOB_DIM_PK	1	1	105	1
TABLE ACCESS BY INDEX ROWID	HPXRA	ELIGIBILITY_DIM	1	1	859	1
INDEX UNIQUE SCAN	HPXRA	ELIGIBILITY_DIM_PK	1	1	105	1
TABLE ACCESS BY INDEX ROWID	HPXRA	MEMBER_DIM	1	1	1029	1
INDEX UNIQUE SCAN	HPXRA	MEMBER_DIM_PK	1	1	105	1
TABLE ACCESS BY INDEX ROWID	HPXRA	GROUP_PACKAGE_DIM	1	1	854	1
INDEX UNIQUE SCAN	HPXRA	XPKGROUP_PACKAGE_DIM	1	1	105	1
TABLE ACCESS BY INDEX ROWID	HPXRA	PRODUCT_DIM	1	1	879	1
INDEX UNIQUE SCAN	HPXRA	PRODUCT_DIM_PK	1	1	105	1
TABLE ACCESS BY INDEX ROWID	HPXRA	PLAN_DIM	1	1	836	1
INDEX UNIQUE SCAN	HPXRA	PLAN_DIM_PK	1	1	105	1
TABLE ACCESS BY INDEX ROWID	HPXRA	SUBGRP_COUNTS	1	1	901	1
INDEX RANGE SCAN	HPXRA	XIF1SUBGRP_COUNTS	1	1	125	1
TABLE ACCESS BY INDEX ROWID	HPXRA	BILLING_COMPONENT_DIM	1	1	855	1
INDEX UNIQUE SCAN	HPXRA	XPKBILLING_COMPONENT_DIM	1	1	105	1
TABLE ACCESS BY INDEX ROWID	HPXRA	RATE_DIM	1	1	900	1
INDEX UNIQUE SCAN	HPXRA	RATE_DIM_PK	1	1	105	1
INDEX RANGE SCAN	HPXRA	ELIGIBILITY_FACT_EVENT_BRID_PK	2	1	772	1
TABLE ACCESS BY INDEX ROWID	HPXRA	ELIGIBILITY_EVENT_DIM	1	1	862	1
INDEX UNIQUE SCAN	HPXRA	ELIGIBILITYEVENT_DIM_PK	1	1	105	1
VIEW PUSHED PREDICATE	SYS		1			
NESTED LOOPS						
NESTED LOOPS			1	2	3815	2
TABLE ACCESS BY INDEX ROWID	HPXRA	PROD_GRP_SUBGRP_RATE_FACT	1	1	2915	1
INDEX RANGE SCAN	HPXRA	AK_PROD_GRP_SUBGRP_RATE_FACT	1	1	2161	1
INDEX UNIQUE SCAN	HPXRA	RATE_DIM_PK	1	1	105	1
TABLE ACCESS BY INDEX ROWID	HPXRA	RATE_DIM	1	1	900	1

SELECT STATEMENT, GOAL = ALL_ROWS			1	269	1440988574	62
RESULT CACHE		axudk38u2pbkff78s2s2v6aqt9				
PX COORDINATOR						
PX SEND QC (RANDOM)	SYS	:TQ10005	1	269	1440988574	62
HASH UNIQUE			1	269	1440988574	62
PX RECEIVE			1	268	1434011116	62
PX SEND HASH	SYS	:TQ10004	1	268	1434011116	62
NESTED LOOPS OUTER			1	268	1434011116	62
VIEW	SYSTEM		1	265	1434000050	59
NESTED LOOPS OUTER			1	265	1434000050	59
NESTED LOOPS OUTER			1	264	1433999811	59
NESTED LOOPS			1	264	1433999596	58
NESTED LOOPS			1	264	1433999346	58
NESTED LOOPS			1	263	1433999109	58
NESTED LOOPS			1	263	1433998858	58
NESTED LOOPS			1	263	1433998626	57
NESTED LOOPS			1	263	1433998382	57
NESTED LOOPS			1	262	1433998145	57
NESTED LOOPS			1	262	1433997859	56
NESTED LOOPS			1	262	1433997620	56
NESTED LOOPS			1	261	1433997388	56
NESTED LOOPS			4	257	1433936536	52
HASH JOIN			176153	58	320242489	12
BUFFER SORT						
PX RECEIVE			67246	4	16308974	2
PX SEND BROADCAST	SYS	:TQ10001	67246	4	16308974	2
TABLE ACCESS FULL	HPXRA	CLASS_DIM	67246	4	16308974	2
HASH JOIN			176153	52	293519086	10
PX RECEIVE			176154	17	82429191	5
PX SEND HASH	SYS	:TQ10003	176154	17	82429191	5
HASH JOIN			176154	17	82429191	5
BUFFER SORT						
PX RECEIVE			7	2	37367	2
PX SEND BROADCAST	SYS	:TQ10000	7	2	37367	2

Partial plan

SELECT STATEMENT, GOAL = ALL_ROWS			1	451	1919970725	176
RESULT CACHE		axudk38u2pbkff78s2s2v6aqt9				
HASH UNIQUE			1	451	1919970725	176
NESTED LOOPS OUTER			1	450	1912993267	176
VIEW	SYSTEM		1	448	1912989452	174
NESTED LOOPS OUTER			1	448	1912989452	174
NESTED LOOPS OUTER			1	447	1912988590	173
NESTED LOOPS			1	446	1912987818	172
NESTED LOOPS			1	445	1912986958	171
NESTED LOOPS			1	444	1912986124	170
NESTED LOOPS			1	443	1912984510	169
NESTED LOOPS			1	442	1912983481	168
NESTED LOOPS			1	441	1912982581	167
NESTED LOOPS			1	440	1912981680	166
HASH JOIN			1	439	1912980701	165
TABLE ACCESS FULL	HPXRA	GROUP_PACKAGE_DIM	72813	7	34062250	2
HASH JOIN			45167	297	1847269469	32
TABLE ACCESS FULL	HPXRA	MEMBER_PACKAGE_FACT	27642	233	1513543325	16
HASH JOIN			109878	61	315103315	16
TABLE ACCESS FULL	HPXRA	CLASS_DIM	67246	4	16308974	2
HASH JOIN			109878	53	274230911	14
TABLE ACCESS FULL	HPXRA	PRODUCT_DIM	11827	4	10896321	2
HASH JOIN			110415	47	247030311	12
TABLE ACCESS FULL	HPXRA	BILLING_COMPONENT_DIM	6036	3	8995528	2
HASH JOIN			159099	41	217730754	10
TABLE ACCESS FULL	HPXRA	PLAN_DIM	1813	2	486899	2
HASH JOIN			176154	36	195867777	8
TABLE ACCESS FULL	HPXRA	PROD_CATEGORY_DIM	7	2	37367	2
INDEX FAST FULL SCAN	HPXRA	AK_PROD_GRP_SUBGRP_RATE_FACT	176154	31	174725230	6
TABLE ACCESS BY INDEX ROWID	HPXRA	GROUP_SUBGRP_DIM	1	1	979	1
INDEX UNIQUE SCAN	HPXRA	GROUP_SUBGRP_DIM_PK	1	1	105	1
TABLE ACCESS BY INDEX ROWID	HPXRA	SUBGRP_COUNTS	1	1	901	1
INDEX RANGE SCAN	HPXRA	XIF1SUBGRP_COUNTS	1	1	125	1
TABLE ACCESS BY INDEX ROWID	HPXRA	RATE_DIM	1	1	900	1
INDEX UNIQUE SCAN	HPXRA	RATE_DIM_PK	1	1	105	1
TABLE ACCESS BY INDEX ROWID	HPXRA	MEMBER_DIM	1	1	1029	1
INDEX UNIQUE SCAN	HPXRA	MEMBER_DIM_PK	1	1	105	1
TABLE ACCESS BY INDEX ROWID	HPXRA	ELIGIBILITY_FACT	1	1	1614	1
INDEX RANGE SCAN	HPXRA	AK_ELIGIBILITY_FACT	2	1	752	1
TABLE ACCESS BY INDEX ROWID	HPXRA	LINEOFBUS_DIM	1	1	834	1
INDEX UNIQUE SCAN	HPXRA	FACETS_LOB_DIM_PK	1	1	105	1
TABLE ACCESS BY INDEX ROWID	HPXRA	ELIGIBILITY_DIM	1	1	859	1
INDEX UNIQUE SCAN	HPXRA	ELIGIBILITY_DIM_PK	1	1	105	1
INDEX RANGE SCAN	HPXRA	ELIGIBILITY_FACT_EVENT_BRID_PK	2	1	772	1
TABLE ACCESS BY INDEX ROWID	HPXRA	ELIGIBILITY_EVENT_DIM	1	1	862	1
INDEX UNIQUE SCAN	HPXRA	ELIGIBILITYEVENT_DIM_PK	1	1	105	1
VIEW PUSHED PREDICATE	SYS		1	2	3815	2
NESTED LOOPS						
NESTED LOOPS			1	2	3815	2
TABLE ACCESS BY INDEX ROWID	HPXRA	PROD_GRP_SUBGRP_RATE_FACT	1	1	2915	1
INDEX RANGE SCAN	HPXRA	AK_PROD_GRP_SUBGRP_RATE_FACT	1	1	2161	1
INDEX UNIQUE SCAN	HPXRA	RATE_DIM_PK	1	1	105	1
TABLE ACCESS BY INDEX ROWID	HPXRA	RATE_DIM	1	1	900	1

#4 – Parallel operations – Why don't I see parallel slaves?

- Parameters controlling parallelism are not set
- Hints are invalid or contradictory
- Alter session enable parallel dml/ddl
- Object DOP not set
- Some operations are not eligible for parallel processing
  - Anecdotal experience indicates nested loop operations are not eligible (even though the documentation says they are)
  - Materialized view refreshes have many rules around refreshing in parallel, and even when they are followed, it doesn't always happen
  - An excerpt from Oracle Support regarding a recently submitted SR on performance:
    - *Please note that parallel execution is useful only if the object size is huge and resources are available. If the object size is not big and parallel execution is used it wont be beneficial as this operation could be done by a single slave very well and there is no need to multiple slaves (resource consumption and finally performance degradation).*

      It is advised to use parallel execution if size is more than 3-4GB.

univera HEALTHCARE

#3 Original query – Single column returned from a table in select clause

```
SELECT T1.MBR_DIM_SK, T10.PRSN_UNIQ_ID AS UNIQ_ID, T2.SRC_APPL_CODE, T2.CLM_ID, T2.CLM_LINE_SQNC_NBR, T2.CLM_ADJ_SQNC_NBR,
    T2.SRC_APPL_ADJ_SQNC_NBR, T2.CLM_SRVC_EFF_DATE_SK, T2.CLM_LINE_PAID_DATE_SK, T2.ADMSN_DATE_SK,
    T2.DSCHRG_DATE_SK, T3.SRC_CLM_LINE_DISP, T4.PROV_ID, T4.FRST_NAME, T4.LAST_NAME, T4.FULL_NAME, T5.HOME_HOST_CODE,
    T6.CLM_TYPE_CAT_CODE_NAME, T6.POS_CODE, T6.POS_CODE_DESC, T7.RPT_CAT_CODE, T7.RPT_CAT_CODE_DESC,
    T8.FAC_TYPE_BILL_CODE, T8.FAC_TYPE_BILL_CODE_NAME, T2.SRVC_QTY_CNT, T11.RVNU_CODE, T2.ALW_AMT, T2.CLM_LINE_EXPNS_AMT,
    T2.CLM_LINE_PAID_AMT, T2.COINS_AMT, T2.CPY_AMT, T2.DDCTBL_AMT, T2.COB_PAID_AMT, T9.SRC_CLM_STS
FROM sasrpt.TEMP_MBR_DIM_SK_TABLE T1
INNER JOIN EDW.T_CLAIM_LINE_FACT T2
ON T1.MBR_DIM_SK = T2.MBR_DIM_SK
INNER JOIN EDW.V_CLM_LINE_DISP_CODE T3
ON T2.CLM_LINE_DISP_CODE_SK = T3.CLM_LINE_DISP_CODE_SK
INNER JOIN EDW.T_PROVIDER_DIM T4
ON T2.BLNG_PROV_SK = T4.PROV_DIM_SK
INNER JOIN EDW.T_ITS_CLAIM_WDIM T5
ON T2.ITS_CLM_WDIM_SK = T5.ITS_CLM_WDIM_SK
INNER JOIN EDW.T_CLAIM_WDIM T6
ON T2.CLM_WDIM_SK = T6.CLM_WDIM_SK
INNER JOIN EDW.T_REPORTING_CATEGORY_DIM T7
ON T2.RPT_CAT_DIM_SK = T7.RPT_CAT_DIM_SK
INNER JOIN EDW.T_CLAIM_FACILITY_WDIM T8
ON T2.CLM_FAC_WDIM_SK = T8.CLM_FAC_WDIM_SK
INNER JOIN EDW.V_CLM_STS_CODE T9
ON T2.CLM_STS_CODE_SK = T9.CLM_STS_CODE
INNER JOIN EDW.T_MEMBER_DIM T10
ON T1.MBR_DIM_SK = T10.MBR_DIM_SK
INNER JOIN EDW.T_REVENUE_CODE_DIM T11
ON T2.AGMNT_RVNU_CODE_DIM_SK = T11.RVNU_CODE_DIM_SK
WHERE (T2.CLM_LINE_PAID_DATE_SK >= 20100101)
AND (T2.CLM_SRVC_EFF_DATE_SK >= 20100101)
```

univera HEALTHCARE

SELECT STATEMENT, GOAL = ALL_ROWS			Partition Start	Partition Stop
NESTED LOOPS				
NESTED LOOPS				
NESTED LOOPS				
NESTED LOOPS				
NESTED LOOPS				
NESTED LOOPS				
NESTED LOOPS				
NESTED LOOPS				
NESTED LOOPS				
NESTED LOOPS				
NESTED LOOPS				
TABLE ACCESS FULL	SASRPT	TEMP_MBR_DIM_SK_TABLE		
PARTITION RANGE ITERATOR			61	95
TABLE ACCESS BY LOCAL INDEX ROWID	EDW	T_CLAIM_LINE_FACT	61	95
BITMAP CONVERSION TO ROWIDS				
BITMAP INDEX SINGLE VALUE	EDW	XIE56_CLAIM_LINE_FACT	61	95
TABLE ACCESS BY INDEX ROWID	EDWA	T_REVENUE_CODE_DIM		
INDEX UNIQUE SCAN	EDWA	XPK_REVENUE_CODE_DIM		
TABLE ACCESS BY INDEX ROWID	IDL	T_CODE_MAPPING		
INDEX UNIQUE SCAN	IDL	XPK_CODE_MAPPING		
TABLE ACCESS BY INDEX ROWID	EDWA	T_CLAIM_FACILITY_WDIM		
INDEX UNIQUE SCAN	EDWA	XPK_CLAIM_FACILITY_WDIM		
TABLE ACCESS BY INDEX ROWID	EDWA	T_REPORTING_CATEGORY_DIM		
INDEX UNIQUE SCAN	EDWA	XPK_REPORTING_CATEGORY_DIM		
TABLE ACCESS BY INDEX ROWID	EDWA	T_CLAIM_WDIM		
INDEX UNIQUE SCAN	EDWA	XPK_CLAIM_WDIM		
TABLE ACCESS BY INDEX ROWID	EDWA	T_ITS_CLAIM_WDIM		
INDEX UNIQUE SCAN	EDWA	XPK_ITS_CLAIM_WDIM		
TABLE ACCESS BY INDEX ROWID	EDWA	T_PROVIDER_DIM		
INDEX UNIQUE SCAN	EDWA	XPK_PROVIDER_DIM		
TABLE ACCESS BY INDEX ROWID	IDL	T_CODE_MAPPING		
INDEX UNIQUE SCAN	IDL	XPK_CODE_MAPPING		
INDEX UNIQUE SCAN	EDWA	XPK_MEMBER_DIM		
TABLE ACCESS BY INDEX ROWID	EDWA	T_MEMBER_DIM		

Original Explain Plan

#3 Rewritten query – Original was aborted after 3 hours, new query runs in 51 seconds.

```
SELECT T1.MBR_DIM_SK,
      (select T10.PRSN_UNIQ_ID
       from EDW.T_MEMBER_DIM T10
       where T1.MBR_DIM_SK = T10.MBR_DIM_SK) AS UNIQ_ID,
      T2.SRC_APPL_CODE, T2.CLM_ID, T2.CLM_LINE_SQNC_NBR, T2.CLM_ADJ_SQNC_NBR, T2.SRC_APPL_ADJ_SQNC_NBR,
      T2.CLM_SRVC_EFF_DATE_SK, T2.CLM_LINE_PAID_DATE_SK, T2.ADMSN_DATE_SK, T2.DSCHRG_DATE_SK,
      (select T3.SRC_CLM_LINE_DISP
       from EDW.V_CLM_LINE_DISP_CODE T3
       where T2.CLM_LINE_DISP_CODE_SK = T3.CLM_LINE_DISP_CODE_SK) SRC_CLM_LINE_DISP,
      T4.PROV_ID, T4.FRST_NAME, T4.LAST_NAME, T4.FULL_NAME,
      (select T5.HOME_HOST_CODE
       from EDW.T_ITS_CLAIM_WDIM T5
       where T2.ITS_CLM_WDIM_SK = T5.ITS_CLM_WDIM_SK) HOME_HOST_CODE,
      T6.CLM_TYPE_CAT_CODE_NAME, T6.POS_CODE, T6.POS_CODE_DESC, T7.RPT_CAT_CODE, T7.RPT_CAT_CODE_DESC,
      T8.FAC_TYPE_BILL_CODE, T8.FAC_TYPE_BILL_CODE_NAME, T2.SRVC_QTY_CNT,
      (select T11.RVNU_CODE
       from EDW.T_REVENUE_CODE_DIM T11
       where T2.AGMNT_RVNU_CODE_DIM_SK = T11.RVNU_CODE_DIM_SK) RVNU_CODE,
      T2.ALW_AMT, T2.CLM_LINE_EXPNS_AMT, T2.CLM_LINE_PAID_AMT, T2.COINS_AMT, T2.CPY_AMT, T2.DDCTBL_AMT, T2.COB_PAID_AMT,
      (select T9.SRC_CLM_STS
       from EDW.V_CLM_STS_CODE T9
       where T2.CLM_STS_CODE_SK = T9.CLM_STS_CODE) SRC_CLM_STS
FROM sasrpt.TEMP_MBR_DIM_SK_TABLE T1
INNER JOIN EDW.T_CLAIM_LINE_FACT T2
ON T1.MBR_DIM_SK = T2.MBR_DIM_SK
INNER JOIN EDW.T_PROVIDER_DIM T4
ON T2.BLNG_PROV_SK = T4.PROV_DIM_SK
INNER JOIN EDW.T_CLAIM_WDIM T6
ON T2.CLM_WDIM_SK = T6.CLM_WDIM_SK
INNER JOIN EDW.T_REPORTING_CATEGORY_DIM T7
ON T2.RPT_CAT_DIM_SK = T7.RPT_CAT_DIM_SK
INNER JOIN EDW.T_CLAIM_FACILITY_WDIM T8
ON T2.CLM_FAC_WDIM_SK = T8.CLM_FAC_WDIM_SK
WHERE (T2.CLM_LINE_PAID_DATE_SK >= 20100101)
AND (T2.CLM_SRVC_EFF_DATE_SK >= 20100101)
```

SELECT STATEMENT, GOAL = ALL_ROWS			partition start	partition stop
TABLE ACCESS BY INDEX ROWID	EDWA	T_MEMBER_DIM		
INDEX UNIQUE SCAN	EDWA	XPK_MEMBER_DIM		
TABLE ACCESS BY INDEX ROWID	IDL	T_CODE_MAPPING		
INDEX UNIQUE SCAN	IDL	XPK_CODE_MAPPING		
TABLE ACCESS BY INDEX ROWID	EDWA	T_ITS_CLAIM_WDIM		
INDEX UNIQUE SCAN	EDWA	XPK_ITS_CLAIM_WDIM		
TABLE ACCESS BY INDEX ROWID	EDWA	T_REVENUE_CODE_DIM		
INDEX UNIQUE SCAN	EDWA	XPK_REVENUE_CODE_DIM		
TABLE ACCESS BY INDEX ROWID	IDL	T_CODE_MAPPING		
INDEX UNIQUE SCAN	IDL	XPK_CODE_MAPPING		
NESTED LOOPS				
NESTED LOOPS				
NESTED LOOPS				
NESTED LOOPS				
NESTED LOOPS				
NESTED LOOPS				
TABLE ACCESS FULL	SASRPT	TEMP_MBR_DIM_SK_TABLE		
PARTITION RANGE ITERATOR			61	95
TABLE ACCESS BY LOCAL INDEX ROWID	EDW	T_CLAIM_LINE_FACT	61	95
BITMAP CONVERSION TO ROWIDS				
BITMAP INDEX SINGLE VALUE	EDW	XIE56_CLAIM_LINE_FACT	61	95
TABLE ACCESS BY INDEX ROWID	EDWA	T_CLAIM_WDIM		
INDEX UNIQUE SCAN	EDWA	XPK_CLAIM_WDIM		
TABLE ACCESS BY INDEX ROWID	EDWA	T_CLAIM_FACILITY_WDIM		
INDEX UNIQUE SCAN	EDWA	XPK_CLAIM_FACILITY_WDIM		
TABLE ACCESS BY INDEX ROWID	EDWA	T_PROVIDER_DIM		
INDEX UNIQUE SCAN	EDWA	XPK_PROVIDER_DIM		
INDEX UNIQUE SCAN	EDWA	XPK_REPORTING_CATEGORY_DIM		
TABLE ACCESS BY INDEX ROWID	EDWA	T_REPORTING_CATEGORY_DIM		

#2 – Cartesian product with all tables joined

```
SELECT DISTINCT T2.CLM_SRVC_EFF_DATE_SK, T2.PRMRY_DIAG_CODE_SK, T4.FULL_NAME AS SRVC_PROV_FULL_NAME,
  T4.FDRL_TIN AS SRVC_PROV_FED_ID, T4.PROV_ID AS SRVC_PROVIDER_ID, T4.PROV_DIM_SK AS SRVC_PROV_DIM_SK,
  T4.TXNMY_CODE_NAME AS SRVC_PROV_TYPE_NAME, T6.CITY_NAME AS SRVC_PROV_CITY, T6.ST_CODE AS SRVC_PROV_STATE,
  T6.PO_CODE AS SRVC_PROV_ZIP_CODE, T4.PROV_TYPE_CODE_DESC AS SRVC_PROV_TYPE,
  T4.PROV_SPCLTY_CODE_DESC AS SRVC_PROV_SPECIALTY, T7.FULL_NAME AS FACILITY_NAME, T7.FDRL_TIN AS BLNG_FED_TAX_ID,
  T7.PROV_ID AS BLNG_PROV_ID, T7.PROV_DIM_SK AS BLNG_PROV_DIM_SK, T7.TXNMY_CODE_NAME AS BLNG_PROV_TYPE_NAME,
  T9.CITY_NAME AS BLNG_PROV_CITY, T9.ST_CODE AS BLNG_PROV_STATE, T9.PO_CODE AS BLNG_PROV_ZIP_CODE,
  T10.DIAG_CODE AS DIAG_CODE_1, T11.DIAG_CODE_2, T11.DIAG_CODE_3, T11.DIAG_CODE_4, T11.DIAG_CODE_5, T11.DIAG_CODE_6,
  T11.DIAG_CODE_7, T11.DIAG_CODE_8, T11.DIAG_CODE_9, T11.DIAG_CODE_10, T11.DIAG_CODE_11, T11.DIAG_CODE_12, T11.DIAG_CODE_13,
  T11.DIAG_CODE_14, T11.DIAG_CODE_15, T11.DIAG_CODE_16, T11.DIAG_CODE_17, T11.DIAG_CODE_18, T11.DIAG_CODE_19, T11.DIAG_CODE_20,
  T11.DIAG_CODE_21, T11.DIAG_CODE_22, T11.DIAG_CODE_23, T11.DIAG_CODE_24, T11.DIAG_CODE_25, T3.PRSN_UNIQ_ID
FROM EDW.T_CLAIM_LINE_FACT T2, EDW.T_MEMBER_DIM T3, EDW.T_PROVIDER_DIM T4, EDW.T_PROVIDER_ADDRESS_FACT T5,
  EDW.T_PROVIDER_ADDRESS_DIM T6, EDW.T_PROVIDER_DIM T7, EDW.T_PROVIDER_ADDRESS_FACT T8,  EDW.T_PROVIDER_ADDRESS_DIM T9,
  EDW.T_DIAGNOSIS_CODE_DIM T10, EDW.T_CLAIM_LINE_DIAGNOSIS_WDIM T11
WHERE T2.CLM_SRVC_EFF_DATE_SK >= 20110101 AND T2.MBR_DIM_SK = T3.MBR_DIM_SK
AND T2.SRVCNG_PROV_SK = T4.PROV_DIM_SK AND T2.SRVCNG_PROV_SK = T5.PROV_DIM_SK
AND T5.PROV_PRMRY_ADRS_DIM_SK = T6.PROV_ADRS_DIM_SK AND T2.BLNG_PROV_SK = T7.PROV_DIM_SK
AND T2.BLNG_PROV_SK = T8.PROV_DIM_SK AND T8.PROV_PRMRY_ADRS_DIM_SK = T9.PROV_ADRS_DIM_SK
AND T2.PRMRY_DIAG_CODE_SK = T10.DIAG_CODE_DIM_SK AND T2.CLM_LINE_DIAG_WDIM_SK = T11.CLM_LINE_DIAG_WDIM_SK
AND T10.DIAG_CODE IN ('7140', '7141', '7142', '71481') OR T11.DIAG_CODE_2 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_3 IN ('7140', '7141', '7142', '71481')  OR T11.DIAG_CODE_4 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_5 IN ('7140', '7141', '7142', '71481')  OR T11.DIAG_CODE_6 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_7 IN ('7140', '7141', '7142', '71481')  OR T11.DIAG_CODE_8 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_9 IN ('7140', '7141', '7142', '71481') OR T11.DIAG_CODE_10 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_11 IN ('7140', '7141', '7142', '71481') OR T11.DIAG_CODE_12 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_13 IN ('7140', '7141', '7142', '71481') OR T11.DIAG_CODE_14 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_15 IN ('7140', '7141', '7142', '71481')  OR T11.DIAG_CODE_16 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_17 IN ('7140', '7141', '7142', '71481') OR T11.DIAG_CODE_18 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_19 IN ('7140', '7141', '7142', '71481') OR T11.DIAG_CODE_20 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_21 IN ('7140', '7141', '7142', '71481') OR T11.DIAG_CODE_22 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_23 IN ('7140', '7141', '7142', '71481') OR T11.DIAG_CODE_24 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_25 IN ('7140', '7141', '7142', '71481')
```

univera HEALTHCARE

Explain Plan Excerpt (original is 730 lines)

			Cost	partition start	partition stop
SELECT STATEMENT, GOAL = ALL_ROWS			1.84E+19		
HASH UNIQUE			1.84E+19		
CONCATENATION					
MERGE JOIN CARTESIAN			1.84E+19		
MERGE JOIN CARTESIAN			1.84E+19		
MERGE JOIN CARTESIAN			1.84E+19		
MERGE JOIN CARTESIAN			1.84E+19		
MERGE JOIN CARTESIAN			1.84E+19		
MERGE JOIN CARTESIAN			1.84E+19		
MERGE JOIN CARTESIAN			2.74E+18		
MERGE JOIN CARTESIAN			2.53429E+13		
MERGE JOIN CARTESIAN			369360049		
TABLE ACCESS FULL	EDWA	T_CLAIM_LINE_DIAGNOSIS_WDIM	5383		
BUFFER SORT			68613		
TABLE ACCESS FULL	EDWA	T_PROVIDER_ADDRESS_DIM	68613		
BUFFER SORT			68613		
TABLE ACCESS FULL	EDWA	T_PROVIDER_ADDRESS_DIM	68613		
BUFFER SORT			107958		
TABLE ACCESS FULL	EDWA	T_DIAGNOSIS_CODE_DIM	107958		
BUFFER SORT			4491680		
TABLE ACCESS FULL	EDWA	T_PROVIDER_DIM	4491680		
BUFFER SORT			4491680		
TABLE ACCESS FULL	EDWA	T_PROVIDER_DIM	4491680		
BUFFER SORT			720747034		
PARTITION RANGE ALL			720747034	1	95
TABLE ACCESS FULL	EDW	T_CLAIM_LINE_FACT	720747034	1	95
BUFFER SORT			21829563		

Fixed Query

```
SELECT DISTINCT T2.CLM_SRVC_EFF_DATE_SK, T2.PRMRY_DIAG_CODE_SK, T4.FULL_NAME AS SRVC_PROV_FULL_NAME,
  T4.FDRL_TIN AS SRVC_PROV_FED_ID, T4.PROV_ID AS SRVC_PROVIDER_ID, T4.PROV_DIM_SK AS SRVC_PROV_DIM_SK,
  T4.TXNMY_CODE_NAME AS SRVC_PROV_TYPE_NAME, T6.CITY_NAME AS SRVC_PROV_CITY, T6.ST_CODE AS SRVC_PROV_STATE,
  T6.PO_CODE AS SRVC_PROV_ZIP_CODE, T4.PROV_TYPE_CODE_DESC AS SRVC_PROV_TYPE,
  T4.PROV_SPCLTY_CODE_DESC AS SRVC_PROV_SPECIALTY, T7.FULL_NAME AS FACILITY_NAME, T7.FDRL_TIN AS BLNG_FED_TAX_ID,
  T7.PROV_ID AS BLNG_PROV_ID, T7.PROV_DIM_SK AS BLNG_PROV_DIM_SK, T7.TXNMY_CODE_NAME AS BLNG_PROV_TYPE_NAME,
  T9.CITY_NAME AS BLNG_PROV_CITY, T9.ST_CODE AS BLNG_PROV_STATE, T9.PO_CODE AS BLNG_PROV_ZIP_CODE,
  T10.DIAG_CODE AS DIAG_CODE_1, T11.DIAG_CODE_2, T11.DIAG_CODE_3, T11.DIAG_CODE_4, T11.DIAG_CODE_5, T11.DIAG_CODE_6,
  T11.DIAG_CODE_7, T11.DIAG_CODE_8, T11.DIAG_CODE_9, T11.DIAG_CODE_10, T11.DIAG_CODE_11, T11.DIAG_CODE_12, T11.DIAG_CODE_13,
  T11.DIAG_CODE_14, T11.DIAG_CODE_15, T11.DIAG_CODE_16, T11.DIAG_CODE_17, T11.DIAG_CODE_18, T11.DIAG_CODE_19, T11.DIAG_CODE_20,
  T11.DIAG_CODE_21, T11.DIAG_CODE_22, T11.DIAG_CODE_23, T11.DIAG_CODE_24, T11.DIAG_CODE_25, T3.PRSN_UNIQ_ID
FROM EDW.T_CLAIM_LINE_FACT T2, EDW.T_MEMBER_DIM T3, EDW.T_PROVIDER_DIM T4, EDW.T_PROVIDER_ADDRESS_FACT T5,
  EDW.T_PROVIDER_ADDRESS_DIM T6, EDW.T_PROVIDER_DIM T7, EDW.T_PROVIDER_ADDRESS_FACT T8,  EDW.T_PROVIDER_ADDRESS_DIM T9,
  EDW.T_DIAGNOSIS_CODE_DIM T10, EDW.T_CLAIM_LINE_DIAGNOSIS_WDIM T11
WHERE T2.CLM_SRVC_EFF_DATE_SK >= 20110101 AND T2.MBR_DIM_SK = T3.MBR_DIM_SK
AND T2.SRVCNG_PROV_SK = T4.PROV_DIM_SK AND T2.SRVCNG_PROV_SK = T5.PROV_DIM_SK
AND T5.PROV_PRMRY_ADRS_DIM_SK = T6.PROV_ADRS_DIM_SK AND T2.BLNG_PROV_SK = T7.PROV_DIM_SK
AND T2.BLNG_PROV_SK = T8.PROV_DIM_SK AND T8.PROV_PRMRY_ADRS_DIM_SK = T9.PROV_ADRS_DIM_SK
AND T2.PRMRY_DIAG_CODE_SK = T10.DIAG_CODE_DIM_SK AND T2.CLM_LINE_DIAG_WDIM_SK = T11.CLM_LINE_DIAG_WDIM_SK

AND (T10.DIAG_CODE IN ('7140', '7141', '7142', '71481') OR T11.DIAG_CODE_2 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_3 IN ('7140', '7141', '7142', '71481')  OR T11.DIAG_CODE_4 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_5 IN ('7140', '7141', '7142', '71481')  OR T11.DIAG_CODE_6 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_7 IN ('7140', '7141', '7142', '71481')  OR T11.DIAG_CODE_8 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_9 IN ('7140', '7141', '7142', '71481') OR T11.DIAG_CODE_10 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_11 IN ('7140', '7141', '7142', '71481') OR T11.DIAG_CODE_12 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_13 IN ('7140', '7141', '7142', '71481') OR T11.DIAG_CODE_14 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_15 IN ('7140', '7141', '7142', '71481')  OR T11.DIAG_CODE_16 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_17 IN ('7140', '7141', '7142', '71481') OR T11.DIAG_CODE_18 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_19 IN ('7140', '7141', '7142', '71481') OR T11.DIAG_CODE_20 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_21 IN ('7140', '7141', '7142', '71481') OR T11.DIAG_CODE_22 IN ('7140', '7141', '7142', '71481')
OR T11.DIAG_CODE_23 IN ('7140', '7141', '7142', '71481') OR T11.DIAG_CODE_24 IN ('7140', '7141', '7142', '71481')

OR T11.DIAG_CODE_25 IN ('7140', '7141', '7142', '71481'))
```

New Explain Plan

SELECT STATEMENT, GOAL = ALL_ROWS			143201		
HASH UNIQUE			1.43E+05		
HASH JOIN			1.43E+05		
HASH JOIN			143201		
HASH JOIN			1.55E+05		
HASH JOIN			1.55E+05		
TABLE ACCESS FULL	EDWA	T_PROVIDER_ADDRESS_DIM	6.86E+04		
HASH JOIN			1.67E+05		
HASH JOIN			1.63E+05		
TABLE ACCESS FULL	EDWA	T_CLAIM_LINE_DIAGNOSIS_WDIM	1.50E+07		
HASH JOIN			1.10E+08		
TABLE ACCESS FULL	EDWA	T_DIAGNOSIS_CODE_DIM	107958		
HASH JOIN			155522573		
HASH JOIN			4279734		
TABLE ACCESS FULL	EDWA	T_PROVIDER_ADDRESS_DIM	68613		
TABLE ACCESS FULL	EDWA	T_PROVIDER_ADDRESS_FACT	4631007		
PARTITION RANGE ALL			163376884	1	95
TABLE ACCESS FULL	EDW	T_CLAIM_LINE_FACT	163376884	1	95
TABLE ACCESS FULL	EDWA	T_PROVIDER_ADDRESS_FACT	4631007		
TABLE ACCESS FULL	EDWA	T_MEMBER_DIM	21829563		
TABLE ACCESS FULL	EDWA	T_PROVIDER_DIM	4491680		
TABLE ACCESS FULL	EDWA	T_PROVIDER_DIM	4491680		

Things to keep in mind regarding stats…

- Just because statistics are not stale, doesn't mean they are accurate
  - Check sample size
  - Run dbms_stats.flush_database_monitoring_info
  - Check all_tab_modifications
  - Look at your explain plan to see if dynamic sampling is being used
- Even if your table has only a few rows, you should still gather stats on it
- Lock statistics if you don't want the nightly job to do it for you
  - Especially important if your load runs at the same time as the SYS job
- Oracle 10g and 11g gathering schemes behave differently
  - In general, for 11g, allow Oracle to determine the sample size
  - Specify a estimate_percent on very large tables so statistics gathering does not run for days
  - Look into new 11g statistics gathering features if you are on that platform (incremental and extended statistics)

univera HEALTHCARE

Questions?

- Email address: pmariani@excellus.com

univera HEALTHCARE