Oracle Enterprise Manager

Oracle SQL Explain Plan

November 1, 2011 8:44:15 AM CST

Target:

FP32

Version: Oracle 10.2.0.4.0
Database: FP32
Schema: DMDOC1
Date: Nov 1, 2011 12:00:00 AM

SQL Statement:

```
SELECT ALL aeb_cam_claim_dental.r_object_id, aeb_cam_claim_dental.object_name,
       aeb_cam_claim_dental.title, aeb_cam_claim_dental.subject,
       aeb_cam_claim_dental.resolution_label, aeb_cam_claim_dental.owner_name,
       aeb_cam_claim_dental.owner_permit, aeb_cam_claim_dental.group_name,
       aeb_cam_claim_dental.group_permit, aeb_cam_claim_dental.world_permit,
       aeb_cam_claim_dental.log_entry, aeb_cam_claim_dental.acl_domain,
       aeb_cam_claim_dental.acl_name, aeb_cam_claim_dental.language_code,
       aeb_cam_claim_dental.aiz_bus_unit, aeb_cam_claim_dental.aiz_division,
       aeb_cam_claim_dental.aiz_record_class,
       aeb_cam_claim_dental.aiz_sox_matter,
       aeb_cam_claim_dental.aiz_hipaa_matter,
       aeb_cam_claim_dental.aiz_reg_matter,
       aeb_cam_claim_dental.aeb_document_id,
       aeb_cam_claim_dental.aeb_template_type_code,
       aeb_cam_claim_dental.aeb_source_system_name,
       aeb_cam_claim_dental.aeb_doc_cat_name,
       aeb_cam_claim_dental.aeb_rim_record_series_code,
       aeb_cam_claim_dental.aeb_ldap_proxy_id,
       aeb_cam_claim_dental.aeb_delete_flag,
       aeb_cam_claim_dental.aeb_ldap_create_id,
       aeb_cam_claim_dental.aeb_cam_group_case_id,
       aeb_cam_claim_dental.aeb_cam_part_no,
       aeb_cam_claim_dental.aeb_cam_pol_no, aeb_cam_claim_dental.aeb_claim_no,
       aeb_cam_claim_dental.aeb_control_no,
       aeb_cam_claim_dental.aeb_pnr_claim_no,
       aeb_cam_claim_dental.aeb_file_claim_no,
       aeb_cam_claim_dental.aeb_file_type, aeb_cam_claim_dental.aeb_file_path,
       aeb_cam_claim_dental.aeb_client_id_code,
       aeb_cam_claim_dental.aeb_patient_name_first,
       aeb_cam_claim_dental.aeb_patient_name_last,
       aeb_cam_claim_dental.aeb_member_name_first,
       aeb_cam_claim_dental.aeb_member_name_last,
       aeb_cam_claim_dental.aeb_member_no,
       aeb_cam_claim_dental.aeb_provider_name,
       aeb_cam_claim_dental.r_object_type, aeb_cam_claim_dental.r_creation_date,
       aeb_cam_claim_dental.r_modify_date, aeb_cam_claim_dental.a_content_type,
       dm_repeating.aeb_procedure_code, aeb_cam_claim_dental.r_content_size,
       dm_repeating.aeb_date_of_service
  FROM aeb_cam_claim_dental_sp aeb_cam_claim_dental,
       aeb_cam_claim_dental_rp dm_repeating
```

```
WHERE aeb_cam_claim_dental.aeb_member_no = :sys_b_00
  AND aeb_cam_claim_dental.r_creation_date > to_date(:sys_b_01, :sys_b_02)
  AND aeb_cam_claim_dental.i_has_folder = :sys_b_03
  AND aeb_cam_claim_dental.i_is_deleted = :sys_b_04
  AND (aeb_cam_claim_dental.owner_name IN (:sys_b_05, :sys_b_06, :sys_b_07,
       :sys_b_08, :sys_b_09, :sys_b_10)
  OR  EXISTS (SELECT :sys_b_11
                FROM dm_acl_s acl_s0, dm_acl_r acl_r
                WHERE acl_s0.r_object_id = acl_r.r_object_id
                  AND aeb_cam_claim_dental.acl_domain = acl_s0.owner_name
                  AND aeb_cam_claim_dental.acl_name = acl_s0.object_name
                  AND (acl_r.r_accessor_name IN (:sys_b_12, :sys_b_13)
                  OR  acl_r.r_is_group = :sys_b_14
                  AND acl_r.r_accessor_name IN (:sys_b_15, :sys_b_16,
                       :sys_b_17, :sys_b_18, :sys_b_19, :sys_b_20))
                  AND (acl_r.r_permit_type = :sys_b_21
                  OR  acl_r.r_permit_type IS NULL)
                  AND acl_r.r_accessor_permit >= :sys_b_22))
  AND dm_repeating.r_object_id = aeb_cam_claim_dental.r_object_id
```

Optimizer Mode Used:

ALL_ROWS

Total Cost:

8

Execution Steps:

Step #	Step Name
27	SELECT STATEMENT
26	FILTER
20	NESTED LOOPS
17	NESTED LOOPS
15	NESTED LOOPS
12	NESTED LOOPS
9	NESTED LOOPS
6	NESTED LOOPS
3	DMDOC1.AEB_CAM_CLAIM_DENTAL_S TABLE ACCESS [BY INDEX ROWID]
2	BITMAP CONVERSION [TO ROWIDS]
1	DMDOC1.AEB_CAM_CLAIM_DENTAL_S_IDX04 BITMAP INDEX [SINGLE VALUE]
5	DMDOC1.AEB_CAM_S TABLE ACCESS [BY INDEX ROWID]
4	DMDOC1.D_1F01A5AE80002900 INDEX [UNIQUE SCAN]
8	DMDOC1.AEB_DOCUMENT_S TABLE ACCESS [BY INDEX ROWID]
7	DMDOC1.D_1F01A5AE80000501 INDEX [UNIQUE SCAN]
11	DMDOC1.AIZ_DOCUMENT_S TABLE ACCESS [BY INDEX ROWID]
10	DMDOC1.D_1F01A5AE80000500 INDEX [UNIQUE SCAN]
14	DMDOC1.DM_SYSOBJECT_S TABLE ACCESS [BY INDEX ROWID]
13	DMDOC1.D_1F01A5AE80000109 INDEX [UNIQUE SCAN]
16	DMDOC1.D_1F01A5AE8000010A INDEX [RANGE SCAN]
19	DMDOC1.AEB_CAM_CLAIM_DENTAL_R TABLE ACCESS [BY INDEX ROWID]

```
18      └─DMDOC1.D_1F01A5AE80003104 INDEX [UNIQUE SCAN]
25    └─NESTED LOOPS
22      ├─DMDOC1.DM_ACL_S TABLE ACCESS [BY INDEX ROWID]
21      │ └─DMDOC1.D_1F01A5AE80000103 INDEX [UNIQUE SCAN]
24      ├─DMDOC1.DM_ACL_R TABLE ACCESS [BY INDEX ROWID]
23      │ └─DMDOC1.D_1F01A5AE80000102 INDEX [RANGE SCAN]
```

Step #	Description	Est. Cost	Est. Rows Returned	Est. KBytes Returned
1	This plan step retrieves a single ROWID by checking the bits in the bitmap index AEB_CAM_CLAIM_DENTAL_S_IDX04 to find the row which satisfies a condition specified in the querys WHERE clause.	--	--	--
2	This plan step accepts a bitmap representation of an index from its child node, and converts it to a ROWID that can be used to access the table.			
3	This plan step retrieves rows from table AEB_CAM_CLAIM_DENTAL_S through ROWID(s) returned by an index.	1	24	2.391
4	This plan step retrieves a single ROWID from the B*-tree index D_1F01A5AE80002900.	1	1	--
5	This plan step retrieves rows from table AEB_CAM_S through ROWID(s) returned by an index.	1	1	0.026
6	This plan step joins two sets of rows by iterating over the driving, or outer, row set (the first child of the join) and, for each row, carrying out the steps of the inner row set (the second child). Corresponding pairs of rows are tested against the join condition specified in the query's WHERE clause.	2	24	3.023
7	This plan step retrieves a single ROWID from the B*-tree index D_1F01A5AE80000501.	1	1	--
8	This plan step retrieves rows from table AEB_DOCUMENT_S through ROWID(s) returned by an index.	1	1	0.052
9	This plan step joins two sets of rows by iterating over the driving, or outer, row set (the first child of the join) and, for each row, carrying out the steps of the inner row set (the second child). Corresponding pairs of rows are tested against the join condition specified in the query's WHERE clause.	4	1	0.178
10	This plan step retrieves a single ROWID from the B*-tree index D_1F01A5AE80000500.	1	1	--
11	This plan step retrieves rows from table AIZ_DOCUMENT_S through ROWID(s) returned by an index.	1	1	0.033
12	This plan step joins two sets of rows by iterating over the driving, or outer, row set (the first child of the join) and, for each row, carrying out the steps of the inner row set (the second child). Corresponding pairs of rows are tested	5	1	0.211

	against the join condition specified in the query's WHERE clause.			
13	This plan step retrieves a single ROWID from the B*-tree index D_1F01A5AE80000109.	1	1	--
14	This plan step retrieves rows from table DM_SYSOBJECT_S through ROWID(s) returned by an index.	1	1	0.161
15	This plan step joins two sets of rows by iterating over the driving, or outer, row set (the first child of the join) and, for each row, carrying out the steps of the inner row set (the second child). Corresponding pairs of rows are tested against the join condition specified in the query's WHERE clause.	6	1	0.372
16	This plan step retrieves one or more ROWIDs in ascending order by scanning the B*-tree index D_1F01A5AE8000010A.	1	1	0.021
17	This plan step joins two sets of rows by iterating over the driving, or outer, row set (the first child of the join) and, for each row, carrying out the steps of the inner row set (the second child). Corresponding pairs of rows are tested against the join condition specified in the query's WHERE clause.	7	1	0.393
18	This plan step retrieves a single ROWID from the B*-tree index D_1F01A5AE80003104.	1	1	--
19	This plan step retrieves rows from table AEB_CAM_CLAIM_DENTAL_R through ROWID(s) returned by an index.	1	1	0.033
20	This plan step joins two sets of rows by iterating over the driving, or outer, row set (the first child of the join) and, for each row, carrying out the steps of the inner row set (the second child). Corresponding pairs of rows are tested against the join condition specified in the query's WHERE clause.	8	1	0.426
21	This plan step retrieves a single ROWID from the B*-tree index D_1F01A5AE80000103.	1	1	--
22	This plan step retrieves rows from table DM_ACL_S through ROWID(s) returned by an index.	1	1	0.05
23	This plan step retrieves one or more ROWIDs in ascending order by scanning the B*-tree index D_1F01A5AE80000102.	1	4	--
24	This plan step retrieves rows from table DM_ACL_R through ROWID(s) returned by an index.	1	1	0.038
25	This plan step joins two sets of rows by iterating over the driving, or outer, row set (the first child of the join) and, for each row, carrying out the steps of the inner row set (the second child). Corresponding pairs of rows are tested against the join condition specified in the query's WHERE clause.	2	1	0.088
26	This plan step accepts multiple sets of rows. Rows from the			

	first set are eliminated using the data found in the second through n sets.			
27	This plan step designates this statement as a SELECT statement.	8	1	0.426