
SQL Advisor Oracle

Module : DataBase Administration Oracle 12c
Name : Sid el mrabet Malek Aya

Table des matières

1	Introduction	2
1.1	Prérequis	2
2	Installation et Configuration de SQL Access Advisor	3
3	Utilisation de SQL Access Advisor via SQL Developer	4
3.1	Étape 1	4
3.1.1	Connexion à la base de données	4
3.1.2	Importing the data	4
3.1.3	Execute DB queries multiple times	4
3.2	Étape 2 : Création de SQL workload et de STS	5
3.2.1	Créer un SQL Tuning Set (STS)	5
3.2.2	Create a Snapshot :	5
3.3	Étape 3 : Création et exécution de la tâche SQL Access Advisor	6
3.3.1	Création d'une tâche SQL Access Advisor	6
3.3.2	Associez le SQL Workload à la tâche.	6
3.3.3	Exécutez la tâche.	6
3.4	Étape 4 : Afficher et Analyser les résultats	7
3.5	Étape 5 : Appliquer les recommandations	8
3.5.1	Créer les journeaux des vues matérialisés	8
3.5.2	Créer la Vues Materialisée proposé par advisor	8
3.6	benefit	9
4	Conclusion	9

1 Introduction

Le SQL Access Advisor est un outil fourni par Oracle pour optimiser les requêtes SQL en recommandant des index, des partitions et des regroupements matériels. Ce tutoriel vous guide sur la manière d'installer, configurer et utiliser SQL Access Advisor en utilisant SQL*Plus ou SQL Developer, sans avoir besoin d'Oracle Enterprise Manager (OEM).

1.1 Prérequis

- Une instance Oracle Database installée.
- Des privilèges d'administrateur sur la base de données.
- SQL*Plus ou SQL Developer installé.

2 Installation et Configuration de SQL Access Advisor

SQL Access Advisor est inclus dans Oracle Database, il n'y a donc pas d'installation et de configuration séparée nécessaire, puisqu'on va l'utiliser via des commandes SQL*Plus ou SQL Developer. Cependant, assurez d'avoir les privilèges nécessaires pour utiliser cet outil comme suit :

```
GRANT EXECUTE ON DBMS_ADVISOR TO TP_USER ;  
GRANT ADVISOR TO TP_USER;
```

3 Utilisation de SQL Access Advisor via SQL Developer

3.1 Étape 1

3.1.1 Connexion à la base de données

On Commence par connecter à notre base de données Oracle en utilisant SQL*Plus ou SQL Developer avec compte sysdba et accorder le rôle ADVISOR user.

```
C:\Users\ayama>sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Mar. Juin 4 03:15:30 2024
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connecté à :
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> GRANT ADVISOR TO TP_USER ;

Autorisation de privilèges (GRANT) acceptée.
```

3.1.2 Importing the data

3.1.3 Execute DB queries multiple times

3.2 Étape 2 : Création de SQL workload et de STS

3.2.1 Créer un SQL Tuning Set (STS)

STS to store SQL statements along with their execution context.

```
BEGIN
  DBMS_SQLSET.CREATE_SQLSET (
    sqlset_name => 'FINAL_STS'
  ,   description => 'STS to store SQL from the private SQL area'
  );
END;

/* vérifier si sts est bien créer */

SELECT NAME, STATEMENT_COUNT AS "SQLCNT", DESCRIPTION
FROM   USER_SQLSET;
```

3.2.2 Create a Snapshot :

- Exécutez des requêtes plusieurs fois.
- créer un instantané (snapshot).
- Identifiez les ID d'instantané corrects correspondant au moment d'exécution des requêtes.
- Charger le STS depuis AWR
- Vérifier que STS contient les requêtes de l'AWR

```
--1
EXEC dbms_workload_repository.create_snapshot;

--2
ALTER SESSION SET NLS_TIMESTAMP_FORMAT = 'YYYY-MM-DD HH:MI';

col begin_snap format a20
col end_snap format a20

SELECT snap_id, begin_interval_time begin_snap, end_interval_time end_snap
FROM dba_hist_snapshot
ORDER BY snap_id;

--3
DECLARE
  cur dbms_sqltune.sqlset_cursor;
BEGIN
  OPEN cur FOR
    SELECT VALUE(P)...
    FROM DBMS_SQLTUNE.SELECT_WORKLOAD_REPOSITORY(&begin_snap_id, &end_snap_id,NULL,NULL,NULL,NULL,NULL,NULL,NULL,'ALL') p;
  dbms_sqltune.load_sqlset (sqlset_name => 'MY_STS', populate_cursor => cur);
END;

--4
COLUMN SQL_TEXT FORMAT a30
COLUMN SCH FORMAT a3
COLUMN ELAPSED FORMAT 999999999

SELECT SQL_ID, PARSING_SCHEMA_NAME AS "SCHEMA", SQL_TEXT,
       ELAPSED_TIME AS "ELAPSED", BUFFER_GETS
FROM   TABLE( DBMS_SQLTUNE.SELECT_SQLSET( 'MY_STS' ) )
WHERE  PARSING_SCHEMA_NAME = 'TP_USER';
```

3.3 Étape 3 : Création et exécution de la tâche SQL Access Advisor

3.3.1 Création d'une tâche SQL Access Advisor

Dans cette étape, nous définissons ce que le SQL Access Advisor doit analyser et l'emplacement des résultats de l'analyse. Nous commençons par créer une tâche à l'aide de la procédure DBMS_ADVISOR.CREATE_TASK, puis nous définissons les paramètres de la tâche à l'aide de la procédure SET_TASK_PARAMETER, et enfin nous lions la tâche à un STS à l'aide de la procédure DBMS_ADVISOR.ADD_STS_REF.

```
--1
VARIABLE task_id NUMBER;
VARIABLE task_name VARCHAR2(255);

EXEC :task_name := 'FINAL_TASK';
EXEC DBMS_ADVISOR.CREATE_TASK('SQL Access Advisor', :task_id, :task_name);

EXEC DBMS_ADVISOR.SET_TASK_PARAMETER(:task_name, 'ANALYSIS_SCOPE', 'ALL');
```

3.3.2 Associez le SQL Workload à la tâche.

```
EXECUTE DBMS_ADVISOR.ADD_STS_REF(:task_name, 'TP_USER', 'MY_STS');
```

3.3.3 Exécutez la tâche.

```
EXECUTE DBMS_ADVISOR.EXECUTE_TASK(:task_name);

-- vérifier la tâche
COL TASK_ID FORMAT 999
COL TASK_NAME FORMAT a25
COL STATUS_MESSAGE FORMAT a25

SELECT * FROM USER_SQLSET;
```

3.4 Étape 4 : Afficher et Analyser les résultats

Vous pouvez afficher les recommandations générées par SQL Access Advisor à l'aide de vues telles que :

- DBA_ADVISOR_TASKS.
- DBA_ADVISOR_RECOMMENDATIONS.
- DBA_ADVISOR_ACTIONS.
- DBA_ADVISOR_RATIONALE.
- DBA_ADVISOR_SQLA_WK_STMTS

De plus, utilisez la procédure DBMS_ADVISOR.GET_TASK_SCRIPT pour un script complet de recommandations. Voici un exemple pour obtenir les détails de la tâche :

```
COL TASK_ID FORMAT 999
COL TASK_NAME FORMAT a25
COL STATUS_MESSAGE FORMAT a25

SELECT * FROM USER_ADVISOR_TASKS;
SELECT * FROM USER_ADVISOR_RECOMMENDATIONS;
SELECT * FROM USER_ADVISOR_ACTIONS;
```

Les results (table de recommandations) :

SQL Advisor fournit des recommandations pour optimiser les performances des bases de données. Voici les principales recommandations :

- Partitionnement
- Index
- Les vues matérialisées
- Optimisation des requêtes
- Statistiques ...ect

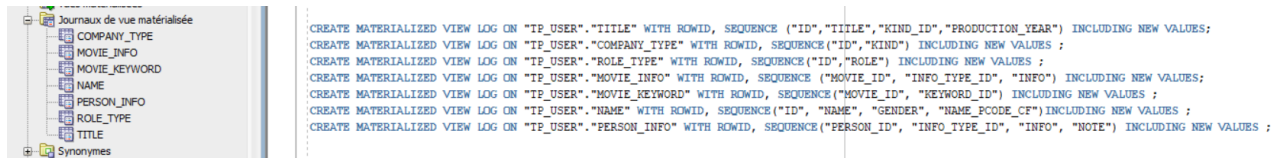
Ces recommandations visent à améliorer l'efficacité et les performances globales de bases de données SQL Server.

TASK_ID	TASK_NAME	EXECUTION_NAME	REC_ID	ACTION_ID	COMMAND	COMMAND_ID	FLAGS	ATTR1	ATTR2	ATTR3	ATTR4	ATTR5	ATTR6
1	65 FINAL_TASK	(null)	22	18	0 CREATE MATERIALIZED V...	4	0 "TP_USER"."COMPANY...	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "KIND")		
2	65 FINAL_TASK	(null)	23	18	0 CREATE MATERIALIZED V...	4	0 "TP_USER"."COMPANY...	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "KIND")		
3	65 FINAL_TASK	(null)	24	18	0 CREATE MATERIALIZED V...	4	0 "TP_USER"."COMPANY...	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "KIND")		
4	65 FINAL_TASK	(null)	26	18	0 CREATE MATERIALIZED V...	4	0 "TP_USER"."COMPANY...	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "KIND")		
5	65 FINAL_TASK	(null)	30	18	0 CREATE MATERIALIZED V...	4	0 "TP_USER"."COMPANY...	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "KIND")		
6	65 FINAL_TASK	(null)	8	18	0 CREATE MATERIALIZED V...	4	0 "TP_USER"."COMPANY...	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "KIND")		
7	65 FINAL_TASK	(null)	11	18	0 CREATE MATERIALIZED V...	4	0 "TP_USER"."COMPANY...	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "KIND")		
8	65 FINAL_TASK	(null)	17	18	0 CREATE MATERIALIZED V...	4	0 "TP_USER"."COMPANY...	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "KIND")		
9	65 FINAL_TASK	(null)	22	20	0 CREATE MATERIALIZED V...	4	5242888 "TP_USER"."TITLE"	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "TITLE")...		
10	65 FINAL_TASK	(null)	23	20	0 CREATE MATERIALIZED V...	4	5242888 "TP_USER"."TITLE"	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "TITLE")...		
11	65 FINAL_TASK	(null)	24	20	0 CREATE MATERIALIZED V...	4	5242888 "TP_USER"."TITLE"	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "TITLE")...		
12	65 FINAL_TASK	(null)	25	20	0 CREATE MATERIALIZED V...	4	5242888 "TP_USER"."TITLE"	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "TITLE")...		
13	65 FINAL_TASK	(null)	26	20	0 CREATE MATERIALIZED V...	4	5242888 "TP_USER"."TITLE"	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "TITLE")...		
14	65 FINAL_TASK	(null)	27	20	0 CREATE MATERIALIZED V...	4	5242888 "TP_USER"."TITLE"	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "TITLE")...		
15	65 FINAL_TASK	(null)	30	20	0 CREATE MATERIALIZED V...	4	5242888 "TP_USER"."TITLE"	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "TITLE")...		
16	65 FINAL_TASK	(null)	5	20	0 CREATE MATERIALIZED V...	4	5242888 "TP_USER"."TITLE"	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "TITLE")...		
17	65 FINAL_TASK	(null)	7	20	0 CREATE MATERIALIZED V...	4	5242888 "TP_USER"."TITLE"	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "TITLE")...		
18	65 FINAL_TASK	(null)	8	20	0 CREATE MATERIALIZED V...	4	5242888 "TP_USER"."TITLE"	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "TITLE")...		
19	65 FINAL_TASK	(null)	11	20	0 CREATE MATERIALIZED V...	4	5242888 "TP_USER"."TITLE"	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "TITLE")...		
20	65 FINAL_TASK	(null)	12	20	0 CREATE MATERIALIZED V...	4	5242888 "TP_USER"."TITLE"	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "TITLE")...		
21	65 FINAL_TASK	(null)	13	20	0 CREATE MATERIALIZED V...	4	5242888 "TP_USER"."TITLE"	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "TITLE")...		
22	65 FINAL_TASK	(null)	16	20	0 CREATE MATERIALIZED V...	4	5242888 "TP_USER"."TITLE"	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "TITLE")...		
23	65 FINAL_TASK	(null)	17	20	0 CREATE MATERIALIZED V...	4	5242888 "TP_USER"."TITLE"	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "TITLE")...		
24	65 FINAL_TASK	(null)	30	22	0 CREATE MATERIALIZED V...	4	0 "TP_USER"."ROLE_TYPE"	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "ROLE")		
25	65 FINAL_TASK	(null)	22	22	0 CREATE MATERIALIZED V...	4	0 "TP_USER"."ROLE_TYPE"	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "ROLE")		
26	65 FINAL_TASK	(null)	13	22	0 CREATE MATERIALIZED V...	4	0 "TP_USER"."ROLE_TYPE"	(null)	ROWID, SEQUENCE	INCLUDING NEW...	("ID", "ROLE")		

FIGURE 1 – Les recommandations proposer par SQL Advisor

3.5 Étape 5 : Appliquer les recommandations

3.5.1 Créer les journeaux des vues matérialisés



3.5.2 Créer la Vues Materialisée proposé par advisor

```
CREATE MATERIALIZED VIEW "TP_USER"."MV$$_0041000A"
REFRESH FAST WITH ROWID
ENABLE QUERY REWRITE
AS
SELECT
  "TP_USER"."MOVIE_COMPANIES"."NOTE" C1,
  "TP_USER"."INFO_TYPE"."INFO" C2,
  "TP_USER"."COMPANY_TYPE"."KIND" C3,
  MIN("TP_USER"."MOVIE_COMPANIES"."NOTE") M1,
  MIN("TP_USER"."TITLE"."PRODUCTION_YEAR") M2,
  MIN("TP_USER"."TITLE"."TITLE") M3,
  COUNT(*) M4
FROM
  "TP_USER"."TITLE",
  "TP_USER"."MOVIE_INFO_IDX",
  "TP_USER"."MOVIE_COMPANIES",
  "TP_USER"."INFO_TYPE",
  "TP_USER"."COMPANY_TYPE"
WHERE
  "TP_USER"."MOVIE_COMPANIES"."MOVIE_ID" = "TP_USER"."TITLE"."ID"
  AND "TP_USER"."MOVIE_COMPANIES"."MOVIE_ID" = "TP_USER"."MOVIE_INFO_IDX"."MOVIE_ID"
  AND "TP_USER"."INFO_TYPE"."ID" = "TP_USER"."MOVIE_INFO_IDX"."INFO_TYPE_ID"
  AND "TP_USER"."COMPANY_TYPE"."ID" = "TP_USER"."MOVIE_COMPANIES"."COMPANY_TYPE_ID"
  AND ("TP_USER"."COMPANY_TYPE"."KIND" = 'production companies')
  AND ("TP_USER"."INFO_TYPE"."INFO" = 'top 250 rank')
GROUP BY
  "TP_USER"."MOVIE_COMPANIES"."NOTE",
  "TP_USER"."INFO_TYPE"."INFO",
  "TP_USER"."COMPANY_TYPE"."KIND";
```

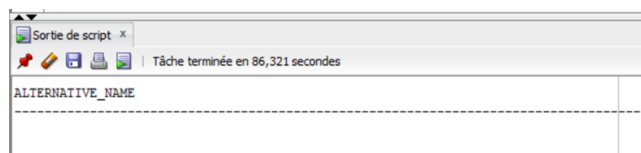
Materialized view "TP_USER"."MV\$\$_0041000A" créé(e).

3.6 bénéfice

L'avantage de cette vue matérialisée est évident dans la réduction significative du temps d'exécution des requêtes. Sans la vue matérialisée, les requêtes prennent 262 secondes pour s'exécuter,



tandis qu'avec la vue matérialisée, le temps d'exécution chute à seulement 86 secondes.



Cette amélioration montre le rôle de la vue matérialisée dans l'optimisation des performances des requêtes en pré-calculant et en stockant les résultats des requêtes complexes, réduisant ainsi la nécessité de recalculer à chaque fois et accélérant la récupération des données.

4 Conclusion