

Oracle数据库中物化视图的原理剖析 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/143/2021_2022_Oracle_E6_95_B0_E6_c102_143755.htm

物化视图 (MV) 在一个段中存储查询结果，并且能够在提交查询时将结果返回给用户，从而不再需要重新执行查询。在查询要执行几次时（这在数据仓库环境中非常常见），这是一个很大的好处。物化视图可以利用一个快速刷新机制从基础表中全部或增量刷新。假定您已经定义了一个物化视图，如下：

```
create materialized view  
mv_hotel_resv refresh fast enable query rewrite as  
select distinct
```

```
city, resv_id, cust_name from hotels h, reservations r where
```

```
r.hotel_id = h.hotel_id.
```

您如何才能知道已经为这个物化视图创建了其正常工作所必需的所有对象？在 Oracle 数据库 10g 之前，这是用 DBMS_MVIEW 程序包中的 EXPLAIN_MVIEW 和 EXPLAIN_REWRITE 过程来判断的。这些过程（在 10g 中仍然提供）非常简要地说明一种特定的功能，如快速刷新功能或查询重写功能，可能用于上述的物化视图，但不提供如何实现这些功能的建议。相反，需要对每一个物化视图的结构进行目视检查，这是非常不实际的。在 10g 中，新的 DBMS_ADVISOR 程序包中的一个名为 TUNE_MVIEW 的过程使得这项工作变得非常容易：您利用 IN 参数来调用程序包，这构造了物化视图创建脚本的全部内容。该过程创建一个顾问程序任务 (Advisor Task)，它拥有一个特定的名称，仅利用 OUT 参数就能够把这个名称传回给您。下面是一个例子。因为第一个参数是一个 OUT 参数，所以您需要在 SQL*Plus 中定义一个变量来保存它。

```
SQL> -- 首先定义一个变量来保
```

存 OUT 参数 SQL> var adv_name varchar2(20) SQL> begin 2
dbms_advisor.tune_mview 3 (4 :adv_name, 5 create materialized
view mv_hotel_resv refresh fast enable query rewrite as 0select
distinct city, resv_id, cust_name from hotels h, reservations r where
r.hotel_id = h.hotel_id). 6* end. 现在您可以在该变量中找出顾问
程序的名称。 SQL> print adv_name ADV_NAME

----- TASK_117 接下来，通过查询一个新的
DBA_TUNE_MVIEW 来获取由这个顾问程序提供的建议。务必在运行该命令之前执行 SET LONG 999999，因为该视图中的
列语句是一个 CLOB，默认情况下只显示 80 个字符。

0select script_type, statement from dba_tune_mview where
task_name = TASK_117 order by script_type, action_id. 下面是输出：
SCRIPT_TYPE STATEMENT -----

IMPLEMENTATION CREATE MATERIALIZED VIEW LOG
ON "ARUP"."HOTELS" WITH ROWID, SEQUENCE
("HOTEL_ID","CITY") INCLUDING NEW VALUES
IMPLEMENTATION ALTER MATERIALIZED VIEW LOG
FORCE ON "ARUP"."HOTELS" ADD ROWID, SEQUENCE
("HOTEL_ID","CITY") INCLUDING NEW VALUES
IMPLEMENTATION CREATE MATERIALIZED VIEW LOG
ON "ARUP"."RESERVATIONS" WITH ROWID, SEQUENCE
("RESV_ID","HOTEL_ID","CUST_NAME") INCLUDING NEW
VALUES IMPLEMENTATION ALTER MATERIALIZED VIEW
LOG FORCE ON "ARUP"."RESERVATIONS" ADD ROWID,
SEQUENCE ("RESV_ID","HOTEL_ID","CUST_NAME")

```
INCLUDING NEW VALUES IMPLEMENTATION CREATE
MATERIALIZED VIEW ARUP.MV_HOTEL_RESV REFRESH
FAST WITH ROWID ENABLE QUERY REWRITE AS SELECT
ARUP.RESERVATIONS.CUST_NAME C1,
ARUP.RESERVATIONS.RESV_ID C2, ARUP.HOTELS.CITY C3,
COUNT(*) M1 FROM ARUP.RESERVATIONS, ARUP.HOTELS
WHERE ARUP.HOTELS.HOTEL_ID =
ARUP.RESERVATIONS.HOTEL_ID GROUP BY
ARUP.RESERVATIONS.CUST_NAME,
ARUP.RESERVATIONS.RESV_ID, ARUP.HOTELS.CITY
UNDO DROP MATERIALIZED VIEW
ARUP.MV_HOTEL_RESV SCRIPT_TYPE 列显示建议的性质。
大多数行将要执行，因此名称为 IMPLEMENTATION。如果
接受，则需按照由 ACTION_ID 列指出的特定顺序执行建议
的操作。如果您仔细查看这些自动生成的建议，那么您将注
意到它们与您自己通过目视分析生成的建议是类似的。这些
建议合乎逻辑；快速刷新的存在需要在拥有适当子句（如那
些包含新值的子句）的基础表上有一个 MATERIALIZED
VIEW LOG。STATEMENT 列甚至提供了实施这些建议的确
切 SQL 语句。在实施的最后一个步骤中，顾问程序建议改变
创建物化视图的方式。注意我们的例子中的不同之处：将一
个 count(*) 添加到了物化视图中。因为我们将这个物化视图
定义为可快速刷新的，所以必须有 count(*)，以便顾问程序
纠正遗漏。 100Test 下载频道开通，各类考试题目直接下载。
详细请访问 www.100test.com
```