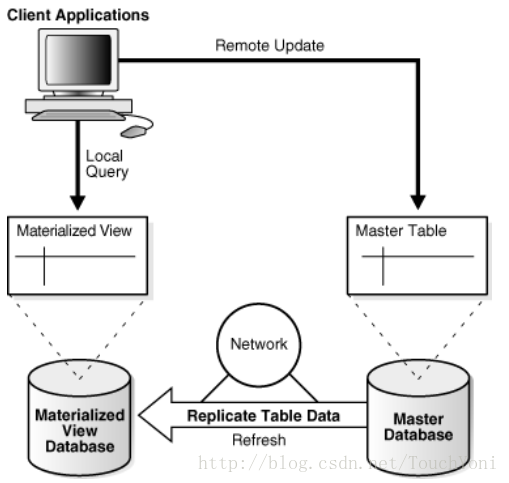
# 物化视图概述

**物化视图是查询结果被提前存储和或物化的模式对象。**在FROM字句后可以是表、视图或物化视图，通常称之为主表。**物化视图常用与数据同步、汇总、分发、计算等应用场景**。



基本语法：

CREATE MATERIALIZED VIEW [view\_name] REFRESH [force|fast|complete|never] on [demand|commit] AS SELECT…..

## 物化视图特征

物化视图与普通视图和索引有些类似的地方。物化视图主要在以下几点与索引类似：

- 它们包含实际数据，并占用存储空间

- 当其主表更新时，物化视图也能被更新

- 它用于查询重写操作，都可以提高SQL性能

- 他们的存在对SQL应用程序和用户是透明的

物化视图图也类似于非物化视图，因为他也呈现其他表和试图的数据。与索引不同的是用户可以直接SELECT语句查询物化视图并且物化视图不受主表丢失的影响。物化视图也可以用DML语句更新，取决于刷新类型。

物化视图可以进行分区，可以在一个分区表上定义一个物化视图，并在该视图上创建一个或多个索引。

## 物化视图刷新

在主表的数据更新后，数据库通过刷新物化视图来对其进行维护。

刷新方法可以是增量刷新（快速刷新）或者完全刷新。在创建时参数可以指定为fast（快速刷新）、complete（完全刷新）、force（默认值，判断能否进行快速刷新，不能则采用完全刷新）、never（不进行刷新）。

当物化视图定义时为BUILD IMMEDIATE,会发生完全刷新。除非物化视图引用了一个预建表。刷新涉及执行物化视图定义的查询。此过程可能很慢，尤其是在当数据库必须读取和处理大量的数据时。

快速刷新避免了重新生成物化视图，因此处理起来会非常快。快速刷新方法是由物化视图日志或直接加载器来记录主表的变更。物化视图日志基于ROWID来记录主表数据变更的模式对象以便增量刷新物化视图，每个物化视图日志与单个主表相关联，一般处于同一个数据库和Schema下。

CREATE MATERIALIZED VIEW LOG ON [table\_name] WITH ROWID

物化视图刷新类型分为两种ON DEMAND和ON COMMIT两种。ON DEMAND则表示在需要时刷新物化视图，ON COMMIT则表示在主表事务提交后自动刷新。当物化视图采用了dblink则无法使用ON COMMIT。

当采用ON DEMAND创建物化视图时，物化视图无法自动刷新。我们可以通过下面三种方式来指定刷新时间点：

1、在创建时指定下次的刷新时间

create meterialized view mv\_view refresh force on demand

start with sysdate next sysdate+1 as select ......

2、通过调用DBMS\_REFRESH

BEGIN DBMS\_REFRESH.MAKE(

NAME=>'AUTO\_REMV'

LIST=>'MV\_EMP'

NEXT\_DATE=>'SYSDATE'

END

/

3、创建JOB任务调用DBMS\_MVIEW.REFRESH

EXEC DBMS\_MVIEW.REFRESH('MV\_EMP')

采用ON COMMIT时物化视图会自动刷新，但需要注意的是主表需要有主键，没有主键也无法创建ON COMMIT类型的物化视图。

## 删除物化视图

DROP METERIALIZED VIEW MV\_EMP;

————————————————

版权声明：本文为CSDN博主「TouchYoni」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/TouchYoni/article/details/79151802