



معرفی **Materialized View** در اوراکل



MATERIALIZED VIEW

با استفاده از این شیء خروجی پرس و جو در یک جدول ذخیره می‌شود که می‌توان آن را پارتیشن بندی کرد و می‌توان بر آن ایندکس تعریف کرد.

Query Rewrite

با استفاده از این قابلیت اوراکل می‌تواند بدون تغییر در پرس و جوها قبلی از داده‌های ذخیره شده در Materialized View استفاده کند، برای استفاده از این امکان باید مقدار پارامتر `query_rewrite_enabled` برابر با `true` یا `force` باشد. در صورتی که پارامتر `optimizer_feature_enable >= 10` باشد مقدار پارامتر `query_rewrite_enabled` برابر `true` و در صورتی که مقدار پارامتر `optimizer_feature_enable <= 9` باشد مقدار پیش فرض آن برابر `false` خواهد بود.

از پارامتر فوق در تمامی MV ها استفاده می‌شود و در صورتی که قصد داشته باشیم برای یک MV خاص Query rewrite انجام شود می‌توان به هنگام ایجاد آن از عبارت `enable query rewrite` استفاده کرد.

Refresh Materialized view

یک MV باید بر اساس تغییرات داده‌ای سیستم‌های منبع بروز رسانی شود، زمان و نوع بروز رسانی MV به شرح ذیل است

1- زمان بروز رسانی

On Commit ✓

On Demand ✓

2- نوع بروز رسانی

Complete ✓

Fast Refresh ✓

Force ✓

Never ✓

(Partition Change Tracking) PCT ✓

Create Materialized View

برای ایجاد یک MV باید سه مرحله زیر انجام شود

1- اعطای مجوز

```
grant create materialized view to <user_name>;
```

```
grant query rewrite to <user_name>;
```

```
grant create any table to <user_name>;
grant select any table to <user_name>;
```

2- ایجاد جدول Log برای MV های Fast Refresh

```
create materialized view log on <table_name>;
```

3- ایجاد MV

```
create materialized view <mv_name>
build [immediate | deferred]
refresh [fast | complete | force ]
on [commit | demand ]
[[enable | disable] query rewrite]
as
select ...;
```

Data Dictionary

برای واکنشی اطلاعات مربوط به MV ها می توان از user_mvview ، user_mvview یا dba_mvview استفاده شود همچنین برای مدیریت MV ها نیز می توان از پکیج dbms_mvview استفاده کرد.