MATERIALIZED VIEW

MATERIALIZED VIEW là 1 object đặc thù của Oracle. Trên MySQL bạn cũng có thể implement MATERIALIZED VIEW dưới dạng 1 table mới.

Needs của MATERIALZED VIEW phát sinh khi bạn có 1 complex computation hoặc 1 complex JOIN statement . Dĩ nhiên bạn không muốn mỗi lần query DB, DB engine lại bắt đầu select từ các table và thực hiện lại các thao tác tính toán phức tạp.

materialized\_view.sql

|  |  |
| --- | --- |
|  | CREATE MATERIALIZED VIEW demo\_materialized\_view AS  -- select ... (comlex JOIN or computation) |

* Khác với VIEW, MATERIALIZED VIEW thực sự chiếm storage của DB. Khi được tạo ra MATERIALZED VIEW sẽ đi tính toán theo công thức được chỉ định sẵn và lưu vào 1 object trong DB. Mỗi lần bạn SELECT FROM MATERIALZED\_VIEW thì sẽ nhận được kết quả tính toán của lần gần nhất.
* Kết quả tính toán sẽ được update trong mỗi lần REFRESH. Giữa 2 lần REFRESH thì kết quả tính toán là không đổi.
* REFRESH có thể được kích hoạt bẳng COMMIT, bằng TRIGGER hoặc được đặt SCHEDULE.
* Chiến lược REFRESH của MATERIALZED\_VIEW bao gồm COMPLETE (mới hoàn toàn), FAST (chỉ lấy thêm phần khác biệt so với lần trước). FAST REFRESH đòi hỏi phải có 1 object nữa là MATERIALZED VIEW LOG, sẽ được đề cập trong bài tiếp.
* Khi dữ liệu quá lớn và tính toán quá phức tạp, MATERIALZED VIEW sẽ đưa toàn bộ phần load của complex computation về thời điểm REFRESH và giúp câu query tại các thời điểm khác trả về kết quả tức thì. Nói hình tượng, bạn có thể schedule cho MATERIALZED VIEW được REFRESH vào lúc nửa đêm, khi user của bạn không mấy khi phát sinh request nào đến Application có thể động chạm đến DB, và trong 1 ngày tiếp theo bạn sẽ có kết quả tính toán được query ra trong 1s và đảm bảo là luôn dúng cho đến ngày hôm trước

MATERIALZED VIEW: là object chiếm storage trực tiếp của DB, thường tổng hợp 1 set các tính toán hoặc JOIN phức tạp và được REFRESH dựa theo chiến lược được định nghĩa sẵn. Với khả năng index chính các kết quả sau khi tính toán, MATERIALZED VIEW cho kết quả trả lại gần như ngay lập tức đối với những data up-to-date đến 1 thời điểm nhất định.

MATERIALIZED VIEW LOG

MATERIALIZED VIEW LOG là object bắt buộc phải có nếu bạn lựa chọn chiến lược FAST REFRESH của object MATERIALIZED VIEW. Chúng ta sẽ nhắc lại 1 chút về chiến lược REFRESH của object MATERIALZED VIEW.

MATERIALIZED VIEW có 3 kiểu REFRESH sau:

* COMPLETE: REFRESH mới hoàn toàn, Oracle sẽ query lại và tính toán lại, Nếu Table chứa lượng data lớn và việc tính toán mất nhiều thời gian thì mỗi lần COMPLETE REFRESH sẽ tốn nhiều thời gian,
* FAST: REFRESH những phần mới từ lần gần đây nhất. thời gian cho mỗi lần FAST REFRESH sẽ được rút ngắn tối thiểu.
* FORCE: là default của REFRESH. Oracle sẽ cố FAST REFRESH, và nếu không được thì sẽ COMPLETE REFRESH

Bạn có thể hình dung mỗi lần Oracle tính toán và ra kết quả cho MATERIALZED VIEW, bạn sẽ có 1 snapshot. Đến lần sau khi FAST REFRESH bạn sẽ update laị kết quả từ last snapshot lần trước. Tất nhiên cái giá phải trả cho việc có được thời gian REFRESH ngắn là sẽ mất thêm dung lượng đĩa cứng để lưu các snapshot ! Tuy nhiên để application chạy được smoothly hết mức có thể thì tốc độ luôn là ưu tiên hàng đầu :D

Vậy snapshot (hay là change log) của MATERIALZED VIEW là gì ? Chúng ta đang nói đến object đề cập ở bên trên: MATERIALIZED VIEW LOG

Cần lưu ý là COMPLETE hay FAST là phương pháp REFRESH (how). Còn thời điểm REFRESH (when) sẽ định nghĩa khi nào thì MATERIALZIED VIEW được REFRESH. Có 2 mode cơ bản là manually (ON DEMAND) và automatically (ON COMMIT, DBMS\_JOB). ON DEMAND là khi nào bạn (user) ra lệnh REFRESH, ON COMMIT là khi nào MATERIALZED bị thay đổi (COMMIT), còn DBMS\_JOB là cho REFRESH thành 1 job được đặt lịch sẵn (giống như cron của Unix system :D)

MATERIALIZED còn có nhiều điểm cần lưu ý khi áp dụng cụ thể. Bài viết chỉ trình bày những khái niệm cơ bản nhất. Bạn có thể xem thêm các restriction và cách create cụ thể tại [Oracle Doc](http://docs.oracle.com/cd/E11882_01/server.112/e10706/repmview.htm)

=> MATERIALIZED VIEW LOG : là snapshot, là object bắt buộc phải có khi dùng MATERIALZED VIEW với chiến lược FAST REFRESH .