**Người soạn: Lê Huy**

**Ngày soạn: 04/01/2024**

**Topic: Export có điều kiện 10 triệu record ra file CSV**

**BÁO CÁO**

1. **Yêu cầu:**

Export toàn bộ dữ liệu trên theo định dạng CSV theo điều kiện Address, Birthday và sắp xếp tăng giảm theo Birthday.

Viết ra 1 phương án thơ ngây nhất, tìm hiểu có vấn đề gì ko, sau đó cải thiện dần, đưa ra thời gian chạy của từng phương án, sau mỗi lần cải thiện thì tăng lên được bao nhiêu?

1. **Tổng hợp kết quả (xếp theo thời gian thực hiện từ chậm đến nhanh):**

*Ghi chú: Kết quả thời gian là lần có kết quả nhanh nhất (trong đó một lần test với keyword ‘VT’ và một lần keyword null trả ra 10tr record) đối với một phương án.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Phương án** | **Thời gian**  **(giây)** | **Toàn vẹn dữ liệu** | **Bảo mật** | **Ưu điểm** | **Nhược điểm** | **Đề xuất** |
| Batched Export Method | 65.162 (‘VT’)  1291.742  (NULL) | Bình thường | Bình thường | Thời gian nhanh, đơn giản | Nguy cơ SQL injection | Sử dụng khi yêu cầu bảo mật được ưu tiên thấp |
| Select Into Outfile | 44.002  (‘VT’)  137.729  (NULL) | Thấp | Bình thường  (raw data) | Nhanh nhất | Nguy cơ SQL injection, toàn vẹn dữ liệu thấp | Sử dụng khi dữ liệu lớn và đã được chuẩn hóa, ưu tiên tốc độ |

1. **Chi tiết:**
2. **Nhận định một số yếu tố ảnh hưởng đến hiệu suất:**

Tương tự đối với insert, tuy nhiên việc xuất file theo điều kiện có liên quan đến quá trình truy vấn, xử lý dữ liệu. Vì vậy, trong phạm vi yêu cầu này, chỉ tập trung tối ưu phần export có liên quan đến truy vấn.

1. **Quá trình thử nghiệm và tối ưu insert:**

* **Tóm tắt:**

File giao diện chạy: **index.php**

* Tạo giao diện import, export với các điều kiện export sau:
  + Export theo điều kiện tìm kiếm address
  + Export có lựa chọn sắp xếp theo ASC hoặc DESC của trường birthday.
* Triển khai import 10 triệu record từ máy local lên server và import database.
* Truy vấn dữ liệu từ database theo các điều kiện lựa chọn và export ra file csv.
* Với mỗi phương án, sử dụng hàm microtime() để tính toán thời gian chạy từ bắt đầu cho đến kết thúc quá trình.
* Test từ phương án đơn giản nhất và xoay quanh các yếu tố tác động. Từ đó tìm các cách để tối ưu.
* **Quá trình thực hiện như sau:**
  1. **Import vào MySQL:**
* **Phương pháp:**
  + Tạo form input file.
  + Với file dữ liệu lớn (~950MB), để thực thi input và upload file lên server, thực hiện điều chỉnh (setting php.ini):
    - upload\_max\_filesize: 300MB 🡪 1000MB
    - post\_max\_size: 300MB 🡪 1000MB
  + Sau khi script chạy input và upload, từ những cơ sở:
    - File dữ liệu đã được chuẩn hóa.
    - Mục đích của yêu cầu là tập trung tối ưu truy vấn có điều kiện.
* Chọn phương pháp LOAD DATA INFILE để insert dữ liệu vào MySQL để tối ưu tốc độ.
  + Ghi chú: Kết hợp chạy TRUNCATE để empty table dữ liệu trước khi insert.
* **File:** upload.php
  1. **Export theo từng batch (Batched Export Method) – Có điều kiện:**
* **Phương pháp:**
  + Tương tự export từng batch không có điều kiện (ở yêu cầu 2).
  + Trong đó với yêu cầu sort theo birthday, sử dụng hàm STR\_TO\_DATE(birthday, '%b-%d-%Y') để xử lý.
  + Tuy nhiên, thực hiện test lần lượt:
    1. Test case: search adress input = “VT” trong 2 trường hợp: gán index và không gán index (cho trường ‘address’ và ‘birthday’) để xem optimize của gán index.
    2. Gán index và thực hiện test case: case search address input = NULL (số lượng record nhiều nhất) để tìm mức batch-size tối ưu.
* **Kết quả:**

1. **Index – Non index with search input ‘VT’:**

|  |  |
| --- | --- |
| **Index** | **Non-index** |
| 65.162 giây | 358.608 giây |

1. **Batch-size:**

|  |  |
| --- | --- |
| **Batch-size** | **Kết quả** |
| 500,000 records | 1358.318 giây |
| 510,000 records | 1291.742 giây |
| >520.000 records | Out of memory |

* **Hiệu suất:** 
  + Việc gán index đã làm tốc độ truy vấn dữ liệu tăng đáng kể, gấp hơn 5 lần việc không gán. Index đã giúp database locate đến chính xác các cột cần thao tác, so với việc scan cả bảng và từng record để xử lý.
  + Trong trường hợp, xuất ra 10 triệu record có sắp xếp, mức tối ưu batch-size khoảng 510,000 tuy nhiên thời gian xử lý vẫn còn lâu.
  + Vì vậy, xem xét sử dụng phương thức SELECT INTO OUTFILE để load file về server sau đó tải về máy local. Nếu nhanh hơn, sẽ tiếp tục xem xét tối ưu đến những yếu tố khác. (Chi tiết ở phần tiếp theo)
* **File:** export.php
  1. **Export sử dụng SELECT INTO OUTFILE:**
* **Phương pháp:** 
  + Sử dụng SELECT INTO OUTFILE và query theo điều kiện để fetch data từ MySQL và lưu tại server.
  + Sau khi test thử, dowload file dung lượng lớn cần sử dụng fopen() và fread() để đọc file theo thông số length number of bytes read và lưu về local.
    - Test ở các mức chunk-size (length number of bytes read) của hàm fread() khác nhau để tối ưu.
  + Unlink() file trên server sau khi hoàn thành để làm sạch.
  + Test và tiếp tục tìm cách tối ưu.
* **Kết quả:**
  + 1. **Test với các mức chunk-size khác nhau:**

|  |  |
| --- | --- |
| **Chunk-size** | **Kết quả** |
| 1MB | 140.901 giây |
| 2MB | 140.846 giây |
| 4MB | 138.775 giây |
| **8MB** | **137.729 giây** |
| 9MB | 138.682 giây |
| 10MB | 145.589 giây |

Test với ‘VT’ search keyword ở thông số tối ưu: **44.002 giây**

* + 1. **Test tối ưu câu query:**

(Tiếp tục nghiên cứu với một số option như sort Date tại kiểu dữ liệu birthday)

* **Hiệu suất:** 
  + Hiệu suất về tốc độ tối ưu nhất do đã rút ngắn tối thiểu quá trình tương tác giữa server và database.
  + Mức chunk-size 8MB tạm thời là mức tối ưu ở phương án này.
* **File:** export\_select\_into\_outfile.php