1. **Storage Engine là gì?**

Là thành phần chịu trách nhiệm tổ chức cách thức lưu trữ các dữ liệu, định dạng table,..MySQL hỗ trợ nhiều storage engine mà mỗi storage engine lại có cách xử lý dữ liệu và các loại tables khác nhau. Có thể tùy thuộc vào chức năng sử dụng dữ liệu của hệ thống mà sử dụng storage engine nào cho phù hợp. MySQL hỗ trợ cả 2 loại Storage Engine là :transaction và non-transaction.

* Transaction Storage Engine: có một số đặc điểm sau.
  + An toàn: Khi mà DB server bị lỗi hoặc hệ thống bị sự cố thì data không bị ảnh hưởng do DB tự động recovery hoặc back up lại dữ liệu từ transaction log.
  + Bạn có thể kết hợp các statement và accept chúng tại một thời điểm giống nhau thông qua câu lệnh COMMIT(không sử dụng cơ chế Auto\_Commit).
  + Bạn có thể thực hiện Roll Back để bỏ qua những thay đổi của dữ liệu về lại dữ liệu lúc trước đó.
  + Nếu một update bị lỗi, thì tất cả các thay đổi khác sẽ được revert lại.
  + Cung cấp cơ chế xử lý đồng thời tốt hơn đối với các bảng mà thường xuyên truy cập hay đọc bởi nhiều người sử dụng.
* Non-Transaction Storage Engine: mặc dù không có được các ưu điểm bên trên như Transaction Storage Engine nhưng nó cũng có một số đặc điểm sau:
  + Tốc độ nhanh
  + Yêu cầu không gian disk thấp.
  + Bộ nhớ yêu cầu để thực thi các updates là nhỏ.
* MySQL hỗ trợ một số loại Storage Engine như là: MyISAM, InnoDB, MEMORY, MERGE, ARCHIVE, FEDERATED,CSV,..

1. **MyISAM Storage Engine.**

* Là Storage Engine mặc định của phiên bản MySQL 5.1 trở về trước. Giới hạn lưu trữ của MyISAM là 256TB. MyISAM cung cấp cho người sử dụng một số lượng lớn các tính năng như: full text search indexing, cơ chế table level locking để xử lý truy cập đồng thời, hỗ trợ kiểu dữ liệu không gian(tọa độ địa lý, hoặc kiểu dữ liệu 3 chiều), B-Tree Index, clustered index, hỗ trợ cơ chế nén data, mã hóa data,…
* Mỗi một MyISAM table được lưu trên đĩa bởi 3 file có dạng là :
  + .frm: lưu định dạng của table
  + .myd: lưu data của table
  + .myi: lưu index của table
* Khi tạo table mà muốn sử dụng storage engine là MyISAM ta có thể đặt biến tùy chọn ENGINE = MYISAM.
* Cơ chế locking và concurrency của MyISAM: Khi xử lý vấn đề truy cập đồng thời của nhiều user, MyISAM cung cấp cơ chế table-locking. Những user mà thực hiện việc đọc dữ liệu thì được cung cấp các khóa chia sẻ(shared locks) cho việc đọc dữ liệu. Còn những user mà thực hiện việc ghi data thì được cung cấp các khóa riêng biệt(exclusive locks) cho việc ghi data.
* MyISAM cung cấp cơ chế check và sửa các MyISAM tables. Cơ chến này không phải là cơ chế xử lý transaction và recovery khi có sự cố. Ta có thể sử dụng các cách sau đây để kiểm tra và repair tables:
  + CHECK TABLE my\_table: check table
  + REPAIR TABLE my\_table: repair table.
  + Hoặc là sử dụng [mysqlcheck] client hoặc [myisamchk] ultility.
* Cơ chế nén MyISAM table bằng việc sử dụng myisampack sẽ giúp thu nhỏ không gian lưu trữ dữ liệu.
* MyISAM cung cấp các dạng Index là : B-Tree Index, R-Tree Index, và Hash.
* Nếu hệ thống sử dụng việc đọc dữ liệu nhiều hơn là việc update dữ liệu thì nên lựa chọn MyISAM.

1. **InnoDB Storage Engine.**

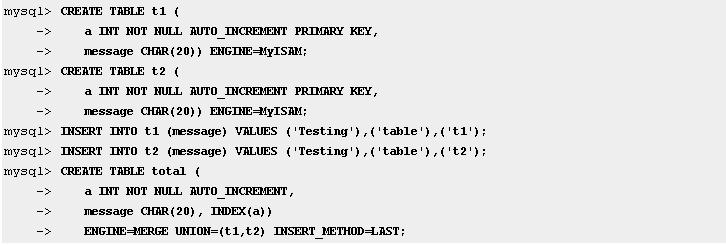
* InnoDB là storage engine có hỗ trợ transaction bao gồm các hành động như là : rollback, commit, recovery(khi có sự cố để bảo vệ dữ liệu người dùng).
* InnoDB hỗ trợ cơ chế row-level lock để xử lý truy cập đông thời từ nhiều người dùng.
* InnoDB hỗ trợ ràng buộc foreign key, giúp cho việc đảm bảo tính toàn vẹn của dữ liệu.
* Giới hạn lưu trữ của InnoDB là 64TB.
* InnoDB lưu trữ data trong một vùng gọi là tablespace.

1. **Archive Storage Engine.**

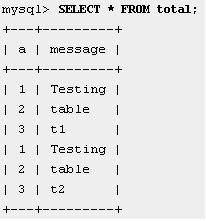
* Dùng để lưu trữ số lượng data lớn mà không cần phải có index.
* Khi tạo một Archive Table thì server sẽ tạo ra một file có dạng *ten\_table.frm* để lưu thông tin định dạng table. Các file data sẽ có định dạng là *ten\_table.arz*.
* Archive chỉ hỗ trợ 2 câu lệnh là Insert và SELECT. Không hỗ trợ UPDATE, DELETE, REPLACE. Không hỗ trợ ORDER BY.
* Archive Engine sử dụng cơ chế Row-level Locking.

1. **MERGE Storage Engine.**

* Cho phép MySQL DBA hoặc developer có thể kết hợp các MyISAM table giống nhau lại thành một nhóm logic mà có thể tham chiếu nó như với một đối tượng.
* Các MyISAM table có thển kết hợp được thì chúng phải giống nhau về thông tin column và index.
* Khi tạo một Merge Table, ta phải chỉ định UNION=(table1, table2,…).
* Ví dụ tạo một table merge:



Khi đó kết quả của talbe [total] là:



* Một số ưu điểm của Merge Table:
  + Có thể dễ dàng quản lý các bảng mà ghi lại log của hệ thống.
  + Tăng tốc độ đọc ghi dữ liệu. Thay vì bạn phải đọc dữ liệu của nhiều bảng, mà có thể được lưu ở nhiều vị trí khác nhau trên disk thì khi tạo Merge table, bạn chỉ cẩn đọc dữ liệu trên một table.
  + Thực hiện repair hiệu quả hơn: Sẽ dễ dàng để thực hiện repair một bảng hớn hơn là trên nhiều bảng nhỏ riêng biệt.
* Một số khuyết điểm của Merge Table:
  + Chỉ sử dụng đối với các MyISAM Table giống nhau.
  + Một số tính năng của MyISAM Table không có trong Merge Table.

1. **MEMORY Storage Engine (thường được gọi là HEAP):**

* Lưu trữ tất cả dữ liệu trong bộ nhớ RAM để có thể truy cập nhanh nhất cho các ứng dụng đòi hỏi tốc độ tra cứu cao mà rất ít có các thay đổi data, và không cần lưu data bền vững sau khi restart.
* Hỗ trợ cơ chế table-level loking.
* Hỗ trợ B-Tree Index và Hash Index.
* Mỗi MEMORY Table khi tạo sẽ có một file .frm để lưu thông tin định dạng table và được lưu trên đĩa. Khi server shutdown thì dữ liệu của Memory Table sẽ mất nhưng thông tin định dạng vẫn còn trên đĩa.

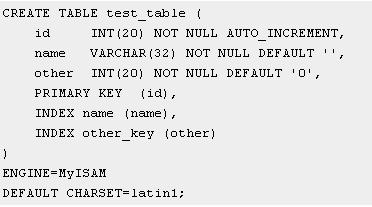
1. **CSV Storage Engine.**

* CSV Engine lưu data trong một file text theo định dạng “comma-separated values”(CSV).
* Bạn có thể sử dụng CSV Engine để dễ dàng trao đổi dữ liệu gữa các phần mềm, giữa các ứng dụng bằng việc import và export theo định dạng CSV.
* Với mỗi CSV Table thì server sẽ tạo ra 2 file. Một là file có dạng *table\_name.frm* lưu thông tin định dạng table, hai là dạng *table\_name.CSV* để lưu data của table. Data được lưu dưới dạng text file.

1. **FEDERATED Storage Engine.**

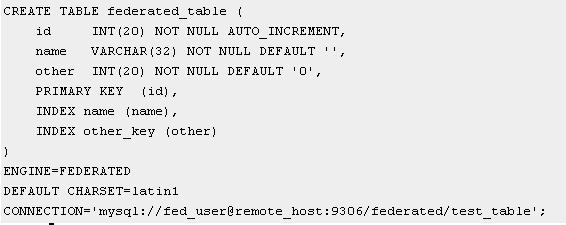
* Là Engine mà bạn có thể sử dụng truy cập dữ liệu từ một DB MySQL từ xa mà không phải là kĩ thuật replication hoặc cluster.
* Các bước để tạo FEDERATED Storage Engine:
  + B1: Tạo table trên remote server.

Vd: Tạo một table trên remote serve:

****

* + B2: Tạo một table trên local server giống với table trên remote server và thêm các thông tin connection để kết nối với table ở remote server.Ta có 2 cách để tạo là sử dụng lệnh CONNECTION để tạo mới connection, hoặc sử dụng connection đã có và tạo server với lệnh CREATE SERVER.

**VD1: Tạo FEDERATED TABLE với CONNECTION:**



Định dạng của chuỗi CONNECTION:

scheme://user\_name[:password]@host\_name[:port\_num]/db\_name/tbl\_name

+) scheme: là giao thức kết nối(mysql)

+) user\_name, password:tên và mật khẩu của tài khoản kết nối, phải được tạo trên remote server và cung cấp các quyền với remote table.

+) host\_name:host\_name hoặc địa chị IP của remote server.

+)port\_num: cổng của remote server.

+)db\_name và table\_name: tên db và table ở remote server mà muốn tạo rederated table.

**VD2: Dùng Create Server**

+) Tạo một conection server :

CREATE SERVER fedlink

FOREIGN DATA WRAPPER mysql

OPTIONS (USER 'fed\_user', HOST 'remote\_host', PORT 9306, DATABASE 'federated');

+) Tạo FEDERATED Table thông qua conection server trên:

CREATE TABLE test\_table (

id INT(20) NOT NULL AUTO\_INCREMENT,

name VARCHAR(32) NOT NULL DEFAULT '',

other INT(20) NOT NULL DEFAULT '0',

PRIMARY KEY (id),

INDEX name (name),

INDEX other\_key (other)

)

ENGINE=FEDERATED

DEFAULT CHARSET=latin1

CONNECTION='fedlink/test\_table';