GIT

**Git là gì?**

Git: Hệ thống quản lý phiên bản phân tán ( là một mô hình hệ thống)

Clone: nhân bản

Responsitory: kho chứa mã nguồn

Lệnh **touch**: Tạo 1 file rỗng

Commit: Đưa mã nguồn lên kho chứa chính: Hoạt động để git lưu lại một bản chụp ( snapshot ) của các sự thay đổi

* Lưu lịch sử chỉnh sửa
* Tên, email của ng commit
* git commit -m “lời nhắn”
* Đưa tập tin lên kho chứa mã nguồn Responsitory:
  + Commit: **git commit -m “ lời nhắn ”**
  + Push: **git push origin master**

Workinh tree: Thư mục làm việc trên máy tính cá nhân

**Tại sao phải dùng git.**

* Miễn phí, dễ sử dụng.
* Lưu trữ được nhiều phiên bản của sourcecode
* Dễ dàng trao đổi source pcode với các thành viên trong teamwork.

**Câu lệnh cơ bản thường gặp**

* Clone 1 kho chứa có sẵn về máy: $ git clone git://github.com/schacon/grit.git
* Atracked file: **git add**
* Commit: lưu lại bản sao của sự thay đổi: **git commit -m “ lời nhắn ”**
* Pull: **git pull** : Lấy sourcode về máy local
* Push: **git push origin master**: Đẩy sourcecode lên kho chứa
* File status: **git status** : Kiểm tra trạng thái của working tree
* Lifecycle:

|  |  |  |  |
| --- | --- | --- | --- |
| Untracked  Tập tin không làm việc trong Git | Traked  Tập tin cần làm việc trong Git | | |
|  | Unmodified  Chưa chỉnh sửa | Modified  Đã chỉnh sửa | Staged  Đã sẵn sàng commit |
|  | | | |

**Phân nhánh: Branching**

Tạo branch mới: **git branch Ten\_Branch**

Truy cập branch: **git checkout Ten\_Branch**

Hiển thị các file trong branch: **ls**

Chuyển dữ liệu giữa các branch:

EX: từ branch A -> branch B

**git checkout B**

***git merge A***

**ls**

Xóa branch: **git branch -d Ten\_Branch**

**Change Author for commit: git commit --amend --author=”Tên\_Author<email Author>”**

Github: dịch vụ của máy chủ

<https://github.com/TranThiNhung>

**MySQL**

* **Structures query language:** ngôn ngữ truy vấn theo hướng cấu trúc
* **MySQL:** Hệ thống quản lý cơ sở dữ liệu quan hệ mã nguồn mở, thân thiện với PHP( nguôn ngữ dùng để phát triển web)
* CREAT
  + Tạo CSDL : **CREAT DATABASE Ten\_CSDL**
  + Tạo bảng: **CREAT TABLE Ten\_bang (**

**Ten\_cot\_1 Kieu\_du\_lieu**

**…**

**Ten\_cot\_n Kieu\_du\_lieu**

**)**

* Tạo view: **CREAT VIEW Ten\_view AS**

SELECT

FROM

WHERE

GROUP BY

HAVING

ORDER BY

* SELECT – FROM – WHERE
* Thêm cột vào bảng: **ALTER TABLE Ten\_bang ADD Ten\_cot Kieu\_du\_lieu**.
* Xóa
  + Xóa CSDL: **DROP DATABASE Ten\_CSDL**
  + Xóa Bảng: **DROP TABLE Ten\_bang**
  + Xóa cột trong bảng: **ALTER TABLE Ten\_bang DROP COLUM Ten\_cot**
* Các lệnh JOIN:
  + **INNER JOIN** : Trả về dữ liệu có chung giữa 2 bảng
  + **FULL JOIN**: Trả về dữ liệu có cả ở 2 bảng
  + **LEFT JOIN**: Trả về mọi bản ghi của bảng bên trái
  + **RIGHT JOIN**: Trả về mọi bản ghi của bảng bên phải
  + **OUTER JOIN**:
* Having
  + Lệnh Having đưa ra điều kiện truy vấn thay cho where khi cần truy vấn trên một nhóm.
  + Cấu trúc :
    - Select
    - From
    - Groub by (nhóm các đối tượng cần truy vấn)
    - Having
* Order by: Sắp xếp kết quả trả về của câu truy vấn( chỉ sử dụng trong truy vấn select)
  + ASC: Sắp xếp tăng dần
  + DESC: Sắp xếp giảm dần
* Like: so sánh kết quả với các giá trị tương tự.

|  |  |
| --- | --- |
| Ký tự | Mô tả |
| \_ | Thay cho một ký tự đơn |
| % | Thay cho một chuỗi |
| [] | Thay cho một ký tự đơn trong khoảng được bao bởi cặp dấu ngoặc vuông |
| [^] | Thay cho một ký tự đơn bất kỳ không nằm trong khoảng được bao bởi cặp dấu ngoặc vuông |

* Hàm:
  + Max: lấy giá trị lớn nhất
  + Min: lấy giá trị nhỏ nhất
  + Avg: lấy giá trị trung bình
  + Sum: lấy tổng
  + Count: đếm số lượng
* So sánh:
  + And
  + Or
  + Between: các phần tử đứng giữa
  + Not: phủ định lại
  + All: tất cả

**BÀI TẬP CÂU LEENJK SQL**

1. Cho biết kết quả cuối khóa (điểm, kết quả, xếp loại) của các học viên do giáo viên “Tran Thanh” dạy trong khóa 1

SELECT DIEM, KETQUA, XEPLOAI

FROM ketqua

INNER JOIN lophoc ON ketqua.MAKH=lophoc.MAKH

INNER JOIN giaovien ON lophoc.MAGV=giaovien.MAGV

(INNER JOIN hocvien ON ketqua.MAHV=hocvien.MAHV)-> Nếu lấy thêm họ tên học viên

INNER JOIN khoahoc ON lophoc.MAKH=khoahoc.MAKH

WHERE HOTENGV='Dai' AND MAKH=1

2. Cho biết danh sách lớp học và số lượng học viên thực sự của lớp đó.

SELECT TENLOP, COUNT(bienlai.MAHV)

FROM lophoc

INNER JOIN bienlai ON lophoc.MALOP=bienlai.MALOP

GROUP BY bienlai.MALOP

3. Cho biết họ tên, năm sinh , địa chỉ của học viên có điểm cao nhất:

SELECT HO, TEN, NTNS, DIACHI

FROM hocvien INNER JOIN ketqua ON hocvien.MAHV= ketqua.MAHV

WHERE DIEM = (

SELECT MAX(ketqua.DIEM)

FROM ketqua

)

4. Cho biết họ tên, năm sinh , địa chỉ của học viên có điểm cao nhất của khóa 2.

SELECT HO, TEN, NTNS, DIACHI, MAX(DIEM)

FROM hocvien INNER JOIN ketqua ON hocvien.MAHV=ketqua.MAHV

WHERE MAKH=2

5. Cho biết tên các lớp học và điểm trung bình,, điểm cao nhất của tất cả các học viên

SELECT MAHV, TENLOP, AVG(DIEM), MAX(DIEM)

FROM ketqua

INNER JOIN lophoc ON ketqua.MAKH=lophoc.MAKH

GROUP BY DIEM

6. Cho biết họ tên học viên , tên lớp học mà học viên đó theo học và số biên lai tương ứng.

SELECT HO, TEN, TENLOP ,MASOBL

FROM hocvien

INNER JOIN bienlai ON hocvien.MAHV = bienlai.MAHV

INNER JOIN lophoc ON lophoc.MALOP = bienlai.MALOP

7. Cho biết tên những lớp học có sĩ số thực sự vượt sĩ số dự kiến.

//TẠO VIEW SISOTHUCTE

CREATE VIEW SISOTHUCTE AS

SELECT lophoc.MALOP, lophoc.TENLOP, lophoc.SISODK, COUNT(bienlai.MAHV) AS SISOBL

FROM lophoc INNER JOIN bienlai ON lophoc.MALOP=bienlai.MALOP

GROUP BY bienlai.MALOP

//TRUY VẤN THEO YÊU CẦU

SELECT TENLOP

FROM sisothucte

WHERE SISODK < SISOBL

8. Cho biết tên và mã số các lớp có sĩ số thực sự ít nhất.

//TẠO VIEW NHƯ CÂU 7

//THỰC HIỆN TRUY VẤN

SELECT TENLOP

FROM sisothucte

WHERE SISOBL = (SELECT MIN(SISOBL) FROM sisothucte)

9. Cho biết họ tên giáo viên và số lượng lớp mà giáo viên đó đã dạy

SELECT HO, TEN, DIEM, XEPLOAI

FROM hocvien INNER JOIN bienlai ON hocvien.MAHV-bienlai.MAHV INNER JOIN ketqua ON hocvien.MAHV=ketqua.MAHV

10. Cho biết họ tên và kết quả học tập (điểm, xếp loại, kết quả) của những học viên được miễn phí

SELECT TEN, DIEM, XEPLOAI, KETQUA

FROM hocvien INNER JOIN ketqua ON hocvien.MAHV=ketqua.MAHV

INNER JOIN bienlai ON ketqua.MAHV=bienlai.MAHV

WHERE bienlai.HOCPHI=0

//TẠO VIEW HVMP=> bao gồm các học viên không có biên lai

CREATE VIEW HVMP AS

SELECT MAHV, TEN

FROM hocvien

WHERE MAHV NOT IN (

SELECT MAHV

FROM bienlai

)

//TRUY VẤN

SELECT TEN, DIEM, KETQUA, XEPLOAI

FROM hvmp INNER JOIN ketqua ON hvmp.MAHV=ketqua.MAHV

11. Cho biết tên khóa học và số lượng lớp học trong khóa đó

SELECT TENKHOA, COUNT(TENLOP) AS ‘SOLUONGLOP’

FROM lophoc INNER JOIN khoahoc ON lophoc.MAKH=khoahoc.MAKH

GROUP BY lophoc.MAKH

<https://viblo.asia/p/tap-hop-nhung-cau-lenh-git-huu-dung-dWrvwWr2vw38>

<https://vietjack.com/mysql/mysql_la_gi.jsp>

<https://vietjack.com/sql/lenh_create_table_trong_sql.jsp>

<https://mariadb.com/kb/vi/basic-sql-statements/>

<https://xuanthulab.net/git-va-github/>