GIT commands

Git push origin HEAD

<https://abc.com/gitserver>

D:\MYPROJECT > git init

git status

**git add** a.txt b.txt

**git add** a.\*

**git add** .

**git commit -m “my first commit”**

git push ( one time pre requisite command is **git add remote origin** [**https://github.com/aw1app/my-first-repo.git**](https://github.com/aw1app/my-first-repo.git) )

Bring Files from the remote repo

**git pull**

**Just to know if remote repo has any new commits, use**

**git fetch**

**BRANCHING**

**git branch <branch\_name>**

**git branch release1**

SWITCHING TO A PARTICULAR BRANCH

**git switch release1**

VERIFY you are on which branch

**git branch**

**COMMITING to the above branch, after switching**

**touch g.txt**

**git add g.txt**

**git commit -m “some message”**

**git push --set-upstream origin release1**

**MERGING BRANCH**

**git merge release1 (this will merge release1 branch into your current branch)**

**git push**

**Git switch main**

**git merge release1**

**git log**

**git pull**

**git status**

**Git add .**

**Git commit**

**Git revert HEAD~2**

**git reset 454etwt41253535dydyh**

does the newly created branch will get the existing files of different branches of the same repo?

can we do partial pull... i dont want to pull c.txt... but i want to pull d.txt... is that possible.?

YES

git checkout cc26e45da53e9d1fe52889eaccfd2c59fd66d7d8

Is it possible that a team memeber can hide, protect or restrict acces to a file added no the local repository YES but not for remote repo? ... in a manner that the git fatch or pull command can not acces it.

DATABASES

A relational database stores data in **rows and columns that form a table**.

A distributed database is spread over different sites, computers, or networks of computers. Cassandra, Amazon DynamoDB

In a cloud database Amazon DynamoDB

SQL commands

sudo mysql -h <IPADDRESS> -u root -p

sudo mysql -u root -p

CREATE A NEW DATABASE

**create database estore;**

**LIST All Databases;**

**show databases;**

**CREATING TABLES**

**Use estore;**

**CREATE TABLE Users(username varchar(80), password varchar(12), age int );**

INFO about table structure:

**describe Users;**

**Store some data**

**INSERT INTO Users values('abc', '56##', 45);**

RETRIEVE / FETCH / SEARCH the data in the table

**SELECT \* FROM Users;**

**Adding more data**

**mysql> INSERT INTO Users values('Salim', '5e66ve%6##', 28);**

**Query OK, 1 row affected (0.01 sec)**

**INSERT INTO Users values('Sujata', '8888RR6##', 68);**

**INSERT INTO Users values('Sharon', 'uud8RR6##', 68);**

**INSERT INTO Users(username,age) values('Jane', 58);**

**INSERT INTO Users values('Z12', '888dff8RR6##', 68);**

**INSERT INTO Users values('Z13', 'uudggg8RR6##', 68);**

**INSERT INTO Users(username,age) values('Z14', 58);**

SELECTING ONLY particular column data

**Select username, age from Users;**

**CONDITION BASED Selection**

**SELECT \* from Users WHERE age>50;**

**WHERE and LIKE**

**PROBLEM - Fetch all usernames starts with S and end with a and age greater than 50**

**select \* from user where username like S%a and age>50**

**WHERE and IN**

**SELECT \* FROM Users WHERE username IN ('Z14', 'Sujata', 'Jane');**

**ORDERing**

**SELECT \* FROM Users Order By username;**

**SELECT \* FROM Users Order By username DESC;**

**PROBLEM Fetch all users such that lesser age users are listed first**

**select \* from users order by age;**

**ALIASES for Column name or Expressions**

**SELECT COUNT(\*) AS NO\_OF\_USERS FROM Users;**

**WHERE and GROUP BY**

**select username, SUM(age) FROM Users GROUP BY username;**

+----------+----------+

| username | SUM(age) |

+----------+----------+

| abc | 45 |

| Mohan | 25 |

| Salim | 28 |

| Sujata | 168 |

| Sharon | 68 |

| Jane | 58 |

| Z14 | 116 |

| Z12 | 68 |

| Z13 | 68 |

| Mehul | 24 |

+----------+----------+

10 rows in set (0.00 sec)

ADDING NEW COULMNS to an Existing Table

**ALTER table Users ADD COLUMN CITY varchar(80);**

**INSERT INTO Users values('Z22', '888dff8RR6##', 68,'LONDON' );**

**INSERT INTO Users values('Z23', 'uudggg8RR6##', 68,'LONDON');**

**INSERT INTO Users(username,age,city) values('Z15', 58, 'MUMBAI');**

**INSERT INTO Users(username,age,city) values('Z17', 58, 'MUMBAI');**

**INSERT INTO Users(username,age,city) values('Z16', 58, 'DUBAI');**

**mysql> select CITY, SUM(age) FROM Users GROUP BY city;**

**+--------+----------+**

**| CITY | SUM(age) |**

**+--------+----------+**

**| NULL | 668 |**

**| LONDON | 136 |**

**| MUMBAI | 116 |**

**| DUBAI | 58 |**

**+--------+----------+**

**4 rows in set (0.00 sec)**

**select CITY, SUM(age) FROM Users GROUP BY city HAVING SUM(Age)>300;**

**UPDATING / MODIFYING TABLE DATA**

**mysql> UPDATE Users SET age=50 WHERE username like 'Z%';**

**Query OK, 9 rows affected (0.00 sec)**

**Rows matched: 9 Changed: 9 Warnings: 0**

**mysql> SELECT \* FROM Users;**

**+----------+--------------+------+--------+**

**| username | password | age | CITY |**

**+----------+--------------+------+--------+**

**| abc | 56## | 45 | NULL |**

**| Mohan | 5etve%6## | 25 | NULL |**

**| Salim | 5e66ve%6## | 28 | NULL |**

**| Sujata | 8888RR6## | 68 | NULL |**

**| Sharon | uud8RR6## | 68 | NULL |**

**| Jane | NULL | 58 | NULL |**

**| Z14 | NULL | 50 | NULL |**

**| Z12 | 888dff8RR6## | 50 | NULL |**

**| Z13 | uudggg8RR6## | 50 | NULL |**

**| Z14 | NULL | 50 | NULL |**

**| Mehul | CASE23$ | 24 | NULL |**

**| Sujata | 8888R66666## | 100 | NULL |**

**| Z22 | 888dff8RR6## | 50 | LONDON |**

**| Z23 | uudggg8RR6## | 50 | LONDON |**

**| Z15 | NULL | 50 | MUMBAI |**

**| Z17 | NULL | 50 | MUMBAI |**

**| Z16 | NULL | 50 | DUBAI |**

**+----------+--------------+------+--------+**

**17 rows in set (0.00 sec)**

**DELETING TABLE DATA**

**DELETE FROM Users WHERE username like 'Z%';**

**PROBLEM: How will your delete an existing column from a table**

**ALTER TABLE Users DROP COLUMN CITY;**

**DROP TABLE Users; // BE CAUTIOUS, all data will be lost!**

**CREATE TABLE Product (PRODUCT\_ID int AUTO\_INCREMENT PRIMARY KEY, PRODUCT\_NAME varchar(80));**

**INSERT INTO Product(PRODUCT\_NAME) VALUES('Samsung Mobile');**

**INSERT INTO Product(PRODUCT\_NAME) VALUES('IPhone Mobile');**

**mysql> SELECT \* FROM Product;**

**+------------+----------------+**

**| PRODUCT\_ID | PRODUCT\_NAME |**

**+------------+----------------+**

**| 1 | Samsung Mobile |**

**| 2 | IPhone Mobile |**

**+------------+----------------+**

**2 rows in set (0.00 sec)**

**CREATING USERS**

**CREATE USER 'user1'@'localhost' IDENTIFIED BY <password>**

**OR**

**CREATE USER user1 IDENTIFIED BY <password>**

**CREATE USER 'user1' IDENTIFIED BY 'Pass1\*\*\*^^%';**

**GRANT ALL PRIVILEGES ON \*.\* TO 'user1';**

**OR**

**GRANT SELECT, INSERT ON \*.\* TO 'user1';**

**NOTE: The first start in \*.\* above refers to the database and second refers to the objects(tables, SP, triggers, views..) in the database.**

**GRANT ALL PRIVILEGES ON estore.Product TO 'user1';**

**GRANT ALL PRIVILEGES ON estore.\* TO 'user1';**

**CREATE TABLE UsersDB(username varchar(80) UNIQUE, password varchar(12), age int );**

**INSERT INTO UsersDB values('Z22', '888dff8RR6##', 68);**

**COMPOSITE KEYS**

**No 2 manc can having same product name registred in a given city/country**

**CREATE TABLE PRODUCT(**

**Product\_name varchar(80) ,**

**City\_of\_manc varchar(80) ,**

**Quantity INT,**

**PRIMARY KEY (Product\_name , City\_of\_manc)**

**);**

**FORIEGN KEY**

**CREATE TABLE Customer( CID int PRIMARY Key AUTO\_INCREMENT, name varchar(256),phone varchar(20),email varchar(256));**

**INSERT INTO Customer values(null, 'John', '90900 11111','john@example.com');**

**INSERT INTO Customer values(null, 'Jagat', '9668781111','jagga@example.com');**

**INSERT INTO Customer values(null, 'Jane', '9356365331','jane@example.com');**

**SELECT \* FROM Customer;**

**CREATE TABLE Address (**

**AddressID INT PRIMARY KEY AUTO\_INCREMENT,**

**Lane VARCHAR(255) NOT NULL,**

**City VARCHAR(255) NOT NULL,**

**Pincode VARCHAR(10) NOT NULL,**

**CustomerID INT,**

**FOREIGN KEY (CustomerID) REFERENCES Customer(CID)**

**);**

**INSERT INTO Address VALUES(null, 'Z144 Redwood Shores', 'Bangalore', '520001', 1);**

**INSERT INTO Address VALUES(null, 'PAL544 Bevery Tank', 'Kolkota', '320001', 2);**

**INSERT INTO Address VALUES(null, 'SOR2 Shalimar Regal', 'Noida', '145045', 3);**

**mysql> Select \* from Address;**

**+-----------+---------------------+-----------+---------+------------+**

**| AddressID | Lane | City | Pincode | CustomerID |**

**+-----------+---------------------+-----------+---------+------------+**

**| 1 | Z144 Redwood Shores | Bangalore | 520001 | 1 |**

**| 2 | PAL544 Bevery Tank | Kolkota | 320001 | 2 |**

**| 3 | SOR2 Shalimar Regal | Noida | 145045 | 3 |**

**+-----------+---------------------+-----------+---------+------------+**

**3 rows in set (0.00 sec)**

**mysql> SELECT \* FROM Customer;**

**+-----+-------+-------------+-------------------+**

**| CID | name | phone | email |**

**+-----+-------+-------------+-------------------+**

**| 1 | John | 90900 11111 | john@example.com |**

**| 2 | Jagat | 9668781111 | jagga@example.com |**

**| 3 | Jane | 9356365331 | jane@example.com |**

**+-----+-------+-------------+-------------------+**

**3 rows in set (0.00 sec)**

**mysql> INSERT INTO Address VALUES(null, 'Prestige gardens', 'Berlin', '145045', 30);**

**ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`estore`.`Address`, CONSTRAINT `Address\_ibfk\_1` FOREIGN KEY (`CustomerID`) REFERENCES `Customer` (`CID`))**

**mysql> DELETE FROM Customer where CID=3;**

**ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`estore`.`Address`, CONSTRAINT `Address\_ibfk\_1` FOREIGN KEY**

**Above errors ARE EXPECTED as per the Foriegn Key policy!**

**CHECKING CONSTRAINT/DEPENDENCIES DEFINITION**

**SELECT**

**CONSTRAINT\_NAME,**

**TABLE\_NAME,**

**COLUMN\_NAME,**

**REFERENCED\_TABLE\_NAME,**

**REFERENCED\_COLUMN\_NAME**

**FROM**

**INFORMATION\_SCHEMA.KEY\_COLUMN\_USAGE**

**WHERE**

**REFERENCED\_TABLE\_NAME = 'Customer'**

**AND TABLE\_NAME = 'Address';**

**JOINS**

**-- CREATE DATABASE JOINS\_DEMODB;**

**-- USE JOINS\_DEMODB;**

**-- CREATE TABLE Customer( CID int PRIMARY Key AUTO\_INCREMENT, name varchar(256),phone varchar(20),email varchar(256));**

**-- INSERT INTO Customer values(null, 'John', '90900 11111','john@example.com');**

**-- INSERT INTO Customer values(null, 'Jagat', '9668781111','jagga@example.com');**

**-- INSERT INTO Customer values(null, 'Jane', '9356365331','jane@example.com');**

**-- SELECT \* FROM Customer;**

**USE JOINS\_DEMODB;**

**CREATE TABLE Orders(**

**OrderID INT PRIMARY KEY AUTO\_INCREMENT,**

**Amount int NOT NULL,**

**CustomerID INT,**

**FOREIGN KEY (CustomerID) REFERENCES Customer(CID)**

**);**

**USE JOINS\_DEMODB;**

**INSERT into Orders (Amount, CustomerID) Values (120, 1);**

**INSERT into Orders (Amount, CustomerID) Values (150, 1);**

**INSERT into Orders (Amount, CustomerID) Values (160, 2);**

**INSERT into Orders (Amount, CustomerID) Values (90, 3);**

**Select CID, Customer.name, Orders.AMOUNT from Customer INNER JOIN Orders on Customer.CID=Orders.CustomerID;**

**LEFT JOIN**

**Select CID, Customer.name, Orders.AMOUNT from Customer LEFT JOIN Orders on Customer.CID=Orders.CustomerID;**

**Select CID, Customer.name, Orders.AMOUNT from Customer LEFT JOIN Orders on Customer.CID=Orders.CustomerID;**

**+-----+-------+--------+**

**| CID | name | AMOUNT |**

**+-----+-------+--------+**

**| 1 | John | 120 |**

**| 1 | John | 150 |**

**| 1 | John | 120 |**

**| 1 | John | 150 |**

**| 2 | Jagat | 160 |**

**| 2 | Jagat | 160 |**

**| 3 | Jane | 90 |**

**| 3 | Jane | 90 |**

**| 4 | John | NULL |**

**| 5 | Jagat | NULL |**

**| 6 | Jane | NULL |**

**+-----+-------+--------+**

**11 rows in set (0.00 sec)**

**RIGHT JOIN**

**Select Orders.OrderID, Customer.name, Orders.AMOUNT from Customer FULL JOIN Orders on Customer.CID=Orders.CustomerID;**

**SELF JOIN**

**Select E1.EmpID, E1.name, E2.name from Employees E1, Employees E2 where E1. ManagerID = E2. EmpID;**

**DELETE FROM TABLE WHERE age>50;**

**DROP**