PostgreSQL

PostgreSQL is an open-source database management system. It supports both SQL and JSON for relational and non-relational queries.

KEY DIFFERENCE:

PostgreSQL is an Object Relational Database Management System (ORDBMS) whereas MySQL is a community driven DBMS system.

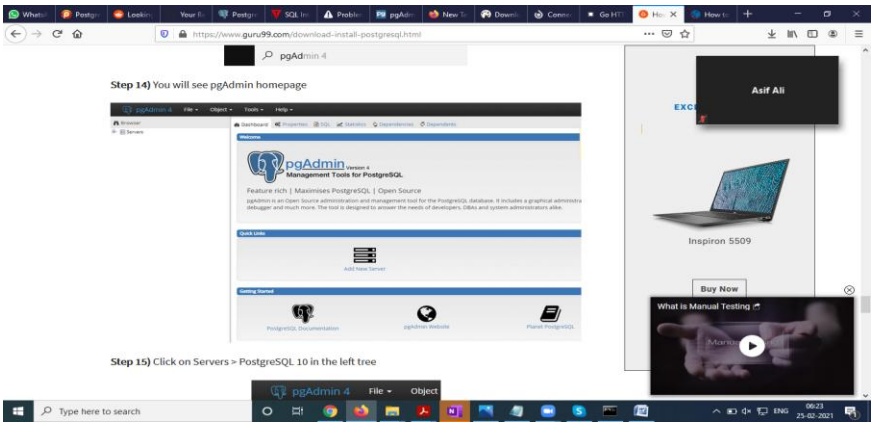
PostgreSQL support modern applications feature like JSON, XML etc. while MySQL only supports JSON.

Install PostgreSQL

**Link:-https://www.postgresql.org/download/

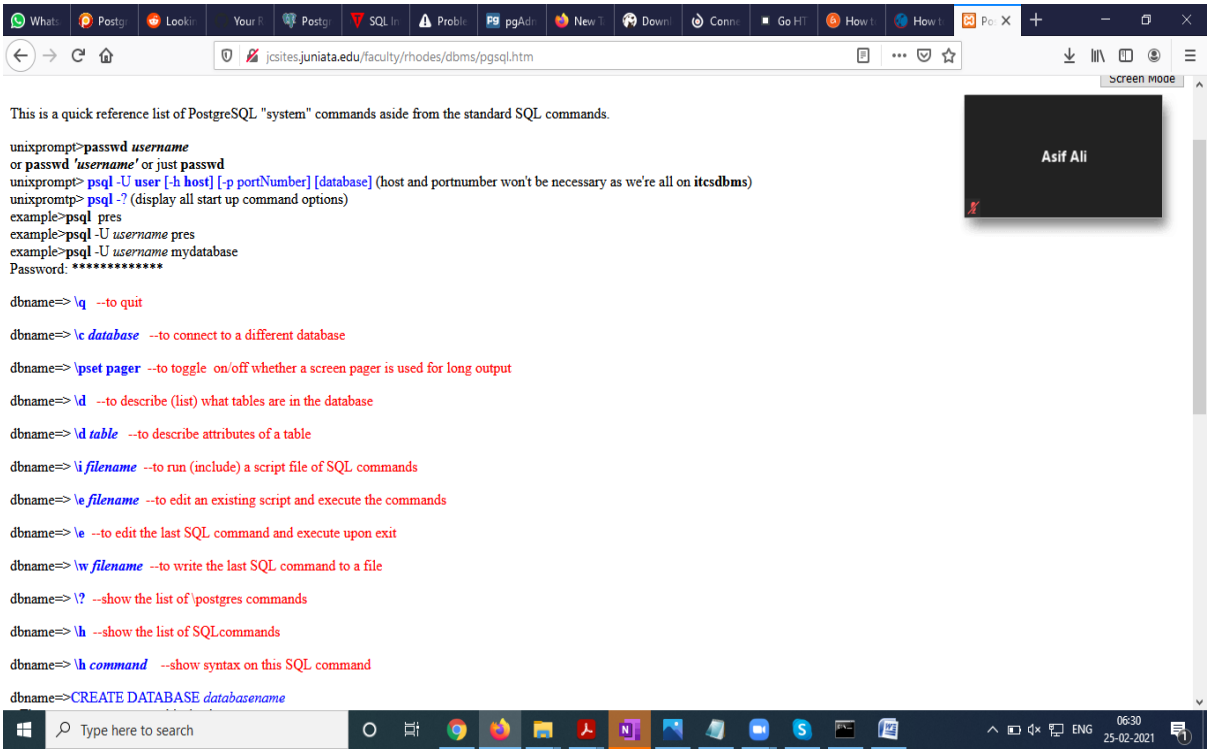
https://phoenixnap.com/kb/how-to-install-postgresql-on-ubuntu

window: https://www.guru99.com/download-install-postgresql.html



PostgreSQL Command :

http://jcsites.juniata.edu/faculty/rhodes/dbms/pgsql.htm



jcsites.juniata.edu/faculty/rhodes/dbms/pgsql.htm

This is a quick reference list of PostgreSQL "system" commands aside from the standard SQL commands.

unixprompt>**passwd** ***username***
or **passwd** ***'username'*** or just **passwd**
unixprompt> **psql** -U **user** [-h **host**] [-p portNumber] [database] (host and portnumber won't be necessary as we're all on **itcsdbms**)
unixpromtp> **psql** -? (display all start up command options)
example>**psql** pres
example>**psql** -U *username* pres
example>**psql** -U *username* mydatabase
Password: **********

dbname=> \q --to quit

dbname=> \c *database* --to connect to a different database

dbname=> \pset pager --to toggle on/off whether a screen pager is used for long output

dbname=> \d --to describe (list) what tables are in the database

dbname=> \d *table* --to describe attributes of a table

dbname=> \i *filename* --to run (include) a script file of SQL commands

dbname=> \e *filename* --to edit an existing script and execute the commands

dbname=> \e --to edit the last SQL command and execute upon exit

dbname=> \w *filename* --to write the last SQL command to a file

dbname=> \? --show the list of \postgres commands

dbname=> \h --show the list of SQLcommands

dbname=> \h *command* --show syntax on this SQL command

dbname=>CREATE DATABASE *databasename*

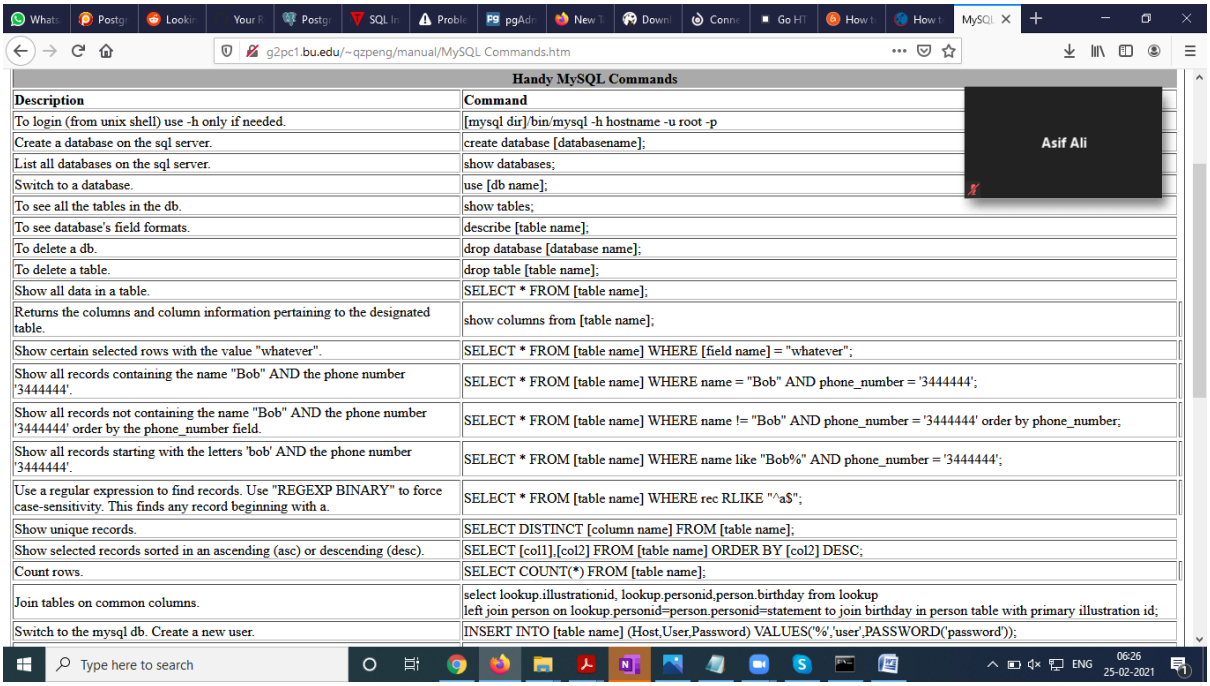
Asif Ali

Type here to search

ENG 06:30 25-02-2021

MySQL Commands

http://g2pc1.bu.edu/~qzpeng/manual/MySQL%20Commands.htm



g2pc1.bu.edu/~qzpeng/manual/MySQL Commands.htm

Handy MySQL Commands	
Description	**Command**
To login (from unix shell) use -h only if needed.	[mysql dir]/bin/mysql -h hostname -u root -p
Create a database on the sql server.	create database [databasename];
List all databases on the sql server.	show databases;
Switch to a database.	use [db name];
To see all the tables in the db.	show tables;
To see database's field formats.	describe [table name];
To delete a db.	drop database [database name];
To delete a table.	drop table [table name];
Show all data in a table.	SELECT * FROM [table name];
Returns the columns and column information pertaining to the designated table.	show columns from [table name];
Show certain selected rows with the value "whatever".	SELECT * FROM [table name] WHERE [field name] = "whatever";
Show all records containing the name "Bob" AND the phone number '3444444'.	SELECT * FROM [table name] WHERE name = "Bob" AND phone_number = '3444444';
Show all records not containing the name "Bob" AND the phone number '3444444' order by the phone_number field.	SELECT * FROM [table name] WHERE name != "Bob" AND phone_number = '3444444' order by phone_number;
Show all records starting with the letters 'bob' AND the phone number '3444444'.	SELECT * FROM [table name] WHERE name like "Bob%" AND phone_number = '3444444';
Use a regular expression to find records. Use "REGEXP BINARY" to force case-sensitivity. This finds any record beginning with a.	SELECT * FROM [table name] WHERE rec RLIKE "^a$";
Show unique records.	SELECT DISTINCT [column name] FROM [table name];
Show selected records sorted in an ascending (asc) or descending (desc).	SELECT [col1],[col2] FROM [table name] ORDER BY [col2] DESC;
Count rows.	SELECT COUNT(*) FROM [table name];
Join tables on common columns.	select lookup.illustrationid, lookup.personid,person.birthday from lookup left join person on lookup.personid=person.personid=statement to join birthday in person table with primary illustration id;
Switch to the mysql db. Create a new user.	INSERT INTO [table name] (Host,User,Password) VALUES('%','user',PASSWORD('password'));

Asif Ali

Type here to search

ENG 06:26 25-02-2021

Types of Keys in Database Management System

Super Key ,Primary Key, Candidate Key,Alternate Key,Foreign Key,Composite Key,Compound Key

*PostgreSQL database using SQL

Creating a Postgres database

```
CREATE DATABASE test;
\c test
```

PostgreSQL constraints

```
CREATE TABLE users (
  id SERIAL PRIMARY KEY,
  age INT,
  first_name TEXT,
  last_name TEXT,
  email TEXT UNIQUE NOT NULL
);
```

SQL insert statement

```
INSERT INTO users (age, email, first_name, last_name)
VALUES (24, 'able@gmail.com', 'Ram', 'Kumar');
INSERT INTO users (age, email, first_name, last_name)
VALUES (24, 'ravi@gmail.com', 'Ravi', 'Kumar');
INSERT INTO users (age, email, first_name, last_name)
VALUES (24, 'dinesh@gmail.com', 'dinesh', 'Kumar');
```

SELECT statement

ORDER BY, GROUP BY, GROUP BY,the HAVING,UNION, INTERSECT, and EXCEPT, LEFT JOIN, INNER JOIN, CROSS JOIN, FULL OUTER JOIN.

```
SELECT * FROM users;

SELECT email FROM users;

Example of the (=) equal operator using WHERE clause
    SELECT first_name,last_name
    FROM employee
    WHERE last_name = 'smith';
Example of OR operator using WHERE clause
    SELECT first_name, last_name
    FROM
    employee
    WHERE
    first_name = 'megan' OR last_name = 'will';
Examples of the LIKE operator using the WHERE clause
    SELECT first_name,last_name
    FROM employee
    WHERE last_name LIKE 'smi%';

Examples of PostgreSQL ORDER BY clause

    SELECT first_name, last_name
    FROM employee
    ORDER BY first_name ASC;


SELECT *
FROM users
```

```
WHERE email = 'able@gmail.com';
```

Combining conditionals with AND and OR

```
SELECT *
FROM users
WHERE last_name > 'kumar'
AND age > 16;
```

Updating SQL records

```
UPDATE users
SET first_name = 'Mohan', last_name = 'gupta' WHERE email =
'able@gmail.com';
```

```
UPDATE users
SET first_name = 'arun', last_name = 'k'
WHERE id = 1;
```

Deleting SQL records

```
DELETE FROM users
WHERE id = 1;
```

PostgreSQL Schema.

```
CREATE SCHEMA Company;

  CREATE TABLE Employee(
    Emp_id SERIAL PRIMARY KEY,
    first_name VARCHAR(30) NOT NULL,
    last_name VARCHAR(30) NOT NULL,
    Age integer NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    address CHARACTER(50),
    salary REAL
    );

SELECT * FROM Employee;
    Or
SELECT * FROM Company.Employee;
SELECT * FROM public.Employee;
```

Syntax to Drop Schema:

DROP SCHEMA schema_name;

or

DROP SCHEMA [IF EXISTS] schema_name [CASCADE | RESTRICT];

ALTER SCHEMA command:

ALTER SCHEMA schema_name

RENAME TO new_name;

Triggers

Stored Procedures

Connecting to a PostgreSQL database with Go's database/sql package

Install the github.com/lib/pq package

```
go get -u github.com/lib/pq
```