???? How to connect mysql with python?

#To start in linux:

service mysql start

mysql -u root -p

#To access a file:

source E:/Personal/Learning/MySQL/sample.sql;

#To see current database:

Select database();

#Creating a new database:

Create database book\_shop;

#To change database:

Use book\_shop;

#create table:

create table dogs (name char(5), breed varchar(10));

#To see current tables:

Show tables;

#Insert data into tables:

Insert into dogs (name, breed) values (‘Bob’, ‘german’);

#To see details of a table:

Desc books;

#To see inside a table:

Select \* from books;

#To see specific columns from a table:

Select author\_fname, author\_lname from books;

#Concatenate two columns:

Select concat(author\_fname, “ “,author\_lname) as ‘full name’ from books;

#concatenate columns with common separator:

Select concat\_ws(“ “,title, author\_fname, author\_lname) from books;

#Selecting a part of the string:

Select substring(‘Hello World’, 1, 4) # it will give first 4 letters

Select substring(‘Hello World’, 7) # it will start from 7th letter and will go till end

Select substring(‘Hello World’, -3) # it will give 3 letters from last

Select substring(title, 1, 10) as ‘Short title’ from books; # from a column of a table

Substr() works too

#Combining two functions:

Select concat(substring(title, 1, 10), “…”) as ‘short titles’ from books;

#Replacing letters/ words

Select Replace(‘Hello World’, ‘Hell’, ‘!@#$’); #case sensitive

Select replace(title, ‘e’,’3’) from books; # for a column in a table

#updating data

update flights set duration = 430 where origin = ‘New York’ and destination = ‘London’;

#combining two functions:

Select substring(replace(title, ‘e’,’3’),1,10) from books;

#reversing the string:

Select reverse(‘Hello World’);

Select reverse(author\_fname) from books;

#group by with conditions

select origin, count(\*) from flights group by origin having count(\*) >1;

#Count characters in string:

Select char\_length(‘Hello World’);

select char\_length(author\_lname) from books;

select concat(author\_lname, “ is “, char\_length(author\_lname),” character long”) from books;

#Change string case:

Select upper(‘Hello World’);

Select lower(‘Hello World’);

#Getting unique entries:

Select distinct author\_lname from books;

select distinct concat(author\_lname, " ", author\_fname) from books;

select distinct author\_lname, author\_fname from books; #get separate columns

# ordering

select author\_lname from books order by author\_lname; #Ascending by default

select author\_lname from books order by author\_lname desc; # Descending order

select title, released\_year, author\_lname from books order by 2; # order by 2nd column

select title, released\_year, author\_lname from books order by released\_year, author\_lname; # order by 1st and then 2nd column

#Limiting the results:

select title from books limit 3;

select title, released\_year from books order by released\_year limit 4; # limit results with order

select title, released\_year from books order by released\_year limit 2,4; # limit to 4 results with order starting from 3rd result (2+1)

#Approx match % for missing data:

select author\_fname from books where author\_fname like %da%;

#matches for length = 3;

select title, stock\_quantity from books where stock\_quantity like '\_\_\_';

#Matching % or \_ within a text:

select title from books where title like '%**\%**%';

#Counting rows:

Select count(\*) from books;

Select count(author\_fname) from books;

#Counting distinct names:

select count(distinct author\_fname) from books;

select count(title) from books where title like '%the%';

#Formulas by grouping:

select title, author\_lname from books group by author\_lname;

select author\_lname, count(\*) from books group by author\_lname;

select author\_lname, count(\*) from books group by author\_lname, author\_fname; #two conditions

#Min and Max

select min(released\_year) from books;

select max(released\_year) from books;

select title, pages from books where pages = (select max(pages) from books); # with condition – slow

select title, pages from books order by pages desc limit 1; fast

#Grouping max, min

select author\_fname, author\_lname, min(released\_year) from books group by author\_lname, author\_fname;

#Sum

select sum(pages) from books;

#Group by sum

select author\_fname, author\_lname, sum(pages) from books group by author\_fname, author\_lname;

#Average:

select avg(pages) from books;

#Average group by:

select author\_fname, author\_lname, avg(pages) from books group by author\_lname, author\_fname;

Changing settings for Strict\_Trans\_Tables:

select @@global.sql\_mode;

set @@global.sql\_mode ='NO\_ENGINE\_SUBSTITUTION';

select @@global.sql\_mode;

#Numbers:

Decimal(5,2)

INT

Float

Double

#Text

Char(5)

Varchar(5)

#data/ time

name varchar(100), birthdate date, birthtime time, birthdt datetime);

insert into people (name, birthdate, birthtime, birthdt) values ('Padma','1983-11-11','10:07:00','1983-11-11 10:07:00');

curdate()

curtime()

now()

select date\_format(birthdt,'%H %i %s') from people;

select date\_format(birthdt,'%D %M %Y') from people;

SELECT DATEDIFF(NOW(), birthdate) FROM people;

create table comments (content varchar(100), created\_at timestamp default now());

create table comments2( content varchar(100), changed\_at Timestamp default now() on update current\_timestamp);

INSERT INTO comments2 (content) VALUES('I LIKE CATS AND DOGS');

UPDATE comments2 SET content='THIS IS NOT GIBBERISH' WHERE content='dasdasdasd';

#Logical operators

#Not equal to:

select title from books where released\_year != 2017;

#Not like:

select title from books where title not like '%stories%';

select title, released\_year from books where released\_year>=2000;

#And operator;

select title, concat(author\_fname, ' ' ,author\_lname) from books where concat(author\_fname, ' ' ,author\_lname) = 'Dave Eggers' && released\_year >2010;

#or operator:

select title, concat(author\_fname, ' ' ,author\_lname) from books where concat(author\_fname, ' ' ,author\_lname) || released\_year >2010;

#Between:

select title, released\_year from books where released\_year >=2004 && released\_year <=2015;

select title, released\_year from books where released\_year between 2004 and 2015;

#Not between

select title, released\_year from books where released\_year not between 2004 and 2015;

#Changing the format

select cast('2017-05-02' as datetime);

#Multiple or match – vector:

select title, author\_lname from books where author\_lname in ('carver', 'lahiri','smith');

select title, author\_lname from books where author\_lname not in ('carver', 'lahiri','smith');

#Modulo/ remainder operator:

select title, released\_year from books where released\_year>=2000 and released\_year %2 != 0;

#If else:

select title, released\_year, case when released\_year>=2000 then 'M' else 'Old' end as genre from books;

#Cross join

select \* from orders where customer\_id =

(select id from customers

where last\_name = 'George');

select \* from customers, orders; # do not use

#Implicit Inner Join

select \* from customers, orders where customer\_id = customers.id;

select first\_name, last\_name, order\_date, amount from customers, orders where orders.customer\_id = customers.id;

#Explicit Inner join

Select \* from customers

Join orders

On orders.customer\_id = customers.id;

Select first\_name, last\_name, order\_date, amount from customers

Join orders

On orders.customer\_id = customers.id; # Join can be replaced with Inner Join

#left join

Select \* from customers

left Join orders

On orders.customer\_id = customers.id; #from left to right; right join works from right to left

#If Null.. replace with something:

Select first\_name, last\_name, ifnull(sum(amount),0) as total from customers

left Join orders

On orders.customer\_id = customers.id group by customer\_id order by total desc;

#Creating dependent tables

create table customers(

id int auto\_increment primary key,

first\_name varchar(100),

last\_name varchar(100),

email varchar(100)

);

create table orders(

id int auto\_increment primary key,

order\_date date,

amount decimal(8,2),

customer\_id int,

foreign key(customer\_id) references customers(id)

);

#Deleting parent table/row along with the dependent line

create table customers(

id int auto\_increment primary key,

first\_name varchar(100),

last\_name varchar(100),

email varchar(100)

);

create table orders(

id int auto\_increment primary key,

order\_date date,

amount decimal(8,2),

customer\_id int,

foreign key(customer\_id) references customers(id)

on delete cascade

);