Key words in Python:

**True-** This  is a boolean value that represents the logical truth. It is a reserved keyword we cant assign a value to it. (True=7 Syntax error).

**False-** This is a boolean value that represents the logical false.

**Boolean**- It is a python datatype which is used to compare, conditional operations.

EG: print(10<100) #true

Print(3==1) #False

**And :** This keyword is used in logical operators where if both the conditions are true then it returns True.

Eg: x = 10

Y = 15

Print ( if x>5 and y>10) # true

**OR** : This keyword is used in logical operators where if any one condition is true then it returns True. If both the conditions are false then it returns false.

Eg: print (if x>15 or y<30)

**None:** It represents the absence of a value or null value. It indicates that variable is not assigned to any value.

**Not:** This keyword is used in logical operators to invert a value. If the output is true a value then by using this not keyword output will be false.

**IS:** It is used in identity operator. That checks whether the two variables refers to same object in memory or not.

EG: x= [1,2,3]

Y= x

Print(y is x) #True

**Break:** It is used in loops to exit a loop when specific conditions is met.

EG: for i in range(5):

If i == 3:

Break

Print(i) # 0,1,2

**Continue:** It is used in loops to skip the code in the current iteration and continue to next iteration of the loop.

EG: for i in range(5):

If i == 3:

continue

Print(i) # 0,1,2,4

**Pass:** It is a null statement nothing happens when it is executed. It is used to prevent indentation errors.

EG: n=10

for i in range(n):

pass o/p: ---

**Try:** It is a exception handling mechanism. Here try block of code is attempt for execution. This try block is used to handle run time errors.

EG: try:

Number=10/0:

Except:

Print( ‘ you cant divide by zero’)

**Except:** It is also exception handling mechanism. It allows program to continue running even if error occurs.

**Finally:** It runs after try and except block of code executed. Even if their an any unexpected error in above blocks also it gets executed.

EG: try:

Number=10/0:

Except:

Print( ‘ you cant divide by zero’)

Finally:

Print(‘ finally executed’).

CREATE DATABASE SampleDB;

USE SampleDB;

CREATE TABLE Employee (EmployeeID INT(30),name varchar(30),Age INT,Department VARCHAR(50),Salary DECIMAL(10,2));

INSERT INTO Employee (EmployeeID, name, Age, Department, Salary) VALUES

(121,'John', 30, 'IT', 60000.00),

(1213,'Jane', 25, 'HR', 55000.00),

(9878,'Robert', 35, 'Finance', 70000.00),

(2312,'Emily', 28, 'Marketing', 58000.00),

(7687,'Michael', 40, 'IT', 80000.00),

(0965,'Sarah', 32, 'HR', 62000.00),

(7654,'David', 29, 'Finance', 67000.00),

(1231,'Laura', 27, 'Marketing', 56000.00),

(6543,'James', 45, 'IT', 90000.00),

(7654,'Sophia', 26, 'HR', 54000.00),

(6532,'Daniel', 33, 'Finance', 72000.00),

(1897,'Olivia', 31, 'Marketing', 60000.00);

select \* from Employee;

use SampleDB;

select \* from Employee;

select \* from Employee where Age>28;

select \* from Employee where Age<50 and Age > 28;

select \* from Employee where Age is not null;

update Employee set Department='non-IT' where EmployeeID=9878;

select \* from Employee;

delete from Employee where name='Sophia';

select EmployeeID from Employee limit 8;

select min(salary) from Employee;

select max(salary) from Employee;

select count(EmployeeID) from Employee;

select sum(salary) from Employee;

select avg(salary) from Employee;

use SampleDB;

truncate table Employee;

select \* from Employee;