HTML

1.

\*HTTPS

ANS :- hypertext transfer protocol secure.

Transfer – link to other text

Protocol – set of rules

\*WWW

ANS :- word wide web.

Ex : search -> network -> server

\*SEO

ANS :- search engine optimization.

Ex : google… yahoo…

\*SERVER

ANS :- data store karva mate.

\*WEBSIDE

ANS :- (1) static – not change

(2) dynamic – change thay sake.

\*URL

ANS :- uniform resorce locator

-web side nu address referred kare.

2.

\*HTML

ANS :- hypertext markup language.

provide to Structure.

Used to Frontend developer.

\*<html lang =”en”>

ANS :- (lang) attribute – change to behaviour .

1 .Meta – change to standard.

2.meta - open to browser .

3.meta – change to view.

4.meta – description.

5. meta – keyword.

\*TAG

ANS :- head – head in not show to code.

Body – body in show to code.

3.

\*TAG

Ans :- <p> - paragraph

<h1> - heading (h1 to h6)

<br> - line break (span)

<b> - bold – (strong)

<u> - under line – (int)

<i> - italic –(mark)

<del> - under line to word (strike)

\*Chack in 3 types of tag

Ans :- (1) open tag ke closed tag

(2) block ke inline– tag in print thay ae element

(3) types of element

4

\*TYPE OF LIST

ANS :- order list(ol)

(1)Capital

(2)small

(3)roman

(4)number

:- unorder list (ul)

(1)dot.

(2)circle

(3)square

:- description list (dl)

(1)dt – description term

(2)dd – description data

\*TABLE

ANS :- Th – table head.

Tb- table body.

Tf – table foot.

Tr – table row.

Td – table data.

Aling – center ,left ,right.

Width – provide to width.

Border – provide to border.

5

\*Rowspan

ANS :- space to row in table

\*Colspan

ANS :- space to col in table

\*Cellspacing

ANS :- one line border in table

\*cellpadding

ANS :- box in table

6

\*img

ANS :- used to image

\*Bg color

ANS :- used to color

\*Src – (sorce)

ANS :- (1) Absalute path – google ne bhi address na khayal hoy.

(2)Relative path – folder in folder.

\*Alt – (alternetiw)

ANS :- src rong hoy tyare Alt run thay.

\*Anchor tag

ANS :- (a) provide to link.

7

\*Form

ANS:- action –

Method – 1. Post (secure)

2.gate(an secure)

Input – blank space

Label – input nu name idendife kare.

User name – text

Password – text,password

Email – text

Phone number – tel

Adress – textarea

Gender – radio

Select – chack box , option.

DOB – datetime-local

Email – Disabled

Submit – button

CSS

1

\*css

ANS :- cascading style sheets

* Used to style an HTML document.
* Provide to color.

\*Types of css.

ANS :- (1) Inline

- Right to in tag.

(2) Internal

- open to same page.

(3) External

- open to next page.

2

\*Selector

ANS :- (1) element selector

* Siddha tag pe css dena .

(2) id selector (#)

- id apia aema j css lage. Id uniq hoy.

(3) class selector (.)

- class apia ama css lage.

(4) universal selector (\*)

-sare stucter pe kam krega .(margin ,padding)

(5) grouping selector (,)

- group kri ne aek sathe akha group ne color thay.

(6) Descendant selector (space)

- class or tag ke bich ki jagah.

(7) child selector (>)

- parents ke andar me jo hota he wo he child .

(8) adjacent sibling selector (+)

- parents ke bad jo tag ata he us me hi lage ga.

(9) general sibling selector (~)

- parents ke bad jo tag ata he wo sab tag me lage

ga.

3

\*Background

ANS :- (1) Background-color (color)

(2) Background-image - URL (image)

(3) Background-position (top,left,center,bottom,righ)

(4) Background-size (contain,cover)

(5) Background-repeat (repeat-x,space,round)

(6) Background-origin (border-box,padding,content)

(7) Background-clip (path)

(8) Background-attechment (scroll,local,fixed)

(9) Background

4

\*Border

ANS :- Contain ki aju baju ki chiz (type,size,color)

(1) solid – common border use bola jata he solid.

(2) border - (top-right-bottem-left)

(3) Box-dashed

(4) Box-dotted

(5) Box-inset - 3D look thoda sa deta he inset .

(6) Box-outset – 3D look thoda sad eta he outsid.

(7) Box-groove – 3D look proper

(8) Box-ridge

* Margin -> bhar ki jagah
* Padding -> andar ki jagah.

5

\*Font

Types of font:

1.Text colore

2.Text alignment

3.Text decoration

4.Text transformation

5.Text indentation

6. letter spacing

7. Word spacing

ANS :- font size – px , rem , cm , in , pt .

Font weight – bold , bolder

Font family – sans serif , serif .

\*CDN

ANS :- content delivery network .

\*Nth-child

ANS :- nth-child

(1) first-chil

(2) last-child

6

\*Display

ANS :- (1) inline – (no used to marging - padding)

(2) inline-block – (marging – padding)

(3) block – (puri jagah cover kre,margin, padding)

(4) flax – (align-item , justify-content only used to flax

Horizontal karne ke liye)

(5) grid – (gallery banane ke li

7

\*position

ANS: Types of position

1. Absolute – khud ki position na leke perents ki jage se saru hota he.
2. Relative ­­– jo khud ki jageh ko banaye rakhta he or koy bhi jagah ja sakta he.
3. Sticky – jab tak pura section khatam na ho tab tak rahene vali wo he sticky.

1. Fixed – jab tak webside khatam na ho tab tak rehta he wo he fixed.
2. Static – by default .

C LANGUGE

* C developed by Dennis Ritchie in 1972

At bell laboratories, California .

Language – Grammar

* C has 32 keywords and c++ has 63 keywords.
* C is a high level programming language.

1. Low-level Language

ANS :- Machine Language (Binary Language – 0 or 1)

2.High-Level Language

ANS:- Human Understable Language (All

Programming Languages)

3 . what is program

ANS :- it is set of rules (instruction)

4 . #includ <stdio.h>

ANS:-(1) # - predefined , link karne ke lia (locator)

(2) Include – To add something .

(3) stdio.h – standard input output.header

(wo aek library he jaha bahut sare data store he)

5. int main() (main function)

ANS:- int main() jaha se program start hota he or

usi ke sath last me return 0; karna padta he.

6. { }

ANS:- { } open gat and close gat . usi ke andar

program likh na he.

7. print

ANS:- To print something on console screen

8. scanf

ANS:- User input.

User ke pass value lene ke liye.

9. &

ANS:- variable ke Address bta ne ke liye .

10 . %

ANS:- Defined karne ke liye.

11. Data types

ANS:- there are two types of data types

1. Primitive data types-pridefined(int,float,char,double)
2. Non-primitive data types

User defined.

12. Access Specifier:-

ANS:- int:- %d or %i

float:- %f

double:- %lf

char:- %c

13. variable

ANS:- To store any values

Ex : a=50. Here a is variable and 50 is value

manav=100.Here manav is variable and

100 is value.

14. Rules of Variables:-

ANS:- 1) Variables should not start with digits.

Ex:- 1num x

num1

2) Varibles should not contain the white

Spaces.

Ex:- no of books x

no\_of\_books

3) Variables should not contain the

keywords.

4) Variables must start with a-z or A-Z or

b=60

tops=500

15 . keyword

ANS:- keywords:- predefined or reserved

Word.

- 32 keywords

Ex:- if…. For…..

16.Static Value and dynamic

ANS:- static value:- predefined value

dynamic value:- user input value.

11 . Formula

ANS :- (1) Rectangle :- a= l\*w ;

A = area

L = length

W = width

(2) Triangle :- a = b\*h/2 ;

A = area

B = base

H = hight

(3) Circle :- pi = 3.14 ;

(4) Square :- area = side \* side ;

(5) Simple Interest :-

si= amount\*rate\*time/100 ;

(6)Days conversion into years and week:-

Years = (days/365) ;

Weeks = (days % 365) / 7 ;

12 . C Programming Operators .

ANS:- To Perform Operations Between two

operands is called operators.

1. Arithmetic Opretor .

+ , - , \*, / , %

1. Increment And Decrement Operetor .

++ , --

1. Assignment Operator .

= , += , -= , \*= , /= , %=

1. Relational Operators .

== , > , < , != , >= , <=

1. Logical Operator.

&& , || , !

Logical AND:- It returns true if both the statements were true.&&

1 1 1

1 0 0

0 1 0

0 0 0

Logical OR:- It returns true if one of the statements were true.:- ||

1 1 1

1 0 1

0 1 1

0 0 0

Logical NOT:- It reverse the result, it returns true if both the statements were false.!

1 1 0

1 0 1

0 1 1

0 0 1

1. Bitwise Operator.

& , | , ^ , << , >>

1. Special Operator.

& , sizeof

1. Conditional Operator

(condtion) ? statement:statement ;

13.Conditional Statements.

1) if statement

ANS:- syntax:-

if(condition)

{

True statement;

}

2) if...else statement

ANS:- syntax:-

if(condition)

{

True statement;

}

else

{

False statement;

}

3) else if statement or else if ladder

ANS: syntax:-

if(condition)

{

statement;

}

else if(condition)

{

statement;

}

else if(condition)

{

statement;

}

else

{

statement;

}

4) Nested if statement:-

ANS : syntax:-

if(condition)

{

if(condition)

{

statement;

}

statement;

else

{

}

}

else

{

}

5) Switch Statement:-

ANS :-syntax:-

switch(varname)

{

case 1:

statement;

break;

case 2:

statement;

break;

case 3:

statement;

break;

default:

statement

}

Loops

Looping Structures:-

Loop:- It executes statement repeatedly

Types of Loops:-

1. For Loop:- sequence controlled Loop

syntax:-

for(initialization;condition;updation)

{

statement;

}

2) Nested For Loop

ANS :-

Syntax:-

for(initialization;condition;updation)

{

for(initialization;condition;updation)

{

statement

}

statement

}

3) While Loop:- Entry Controlled Loop Or Condition Based Loop.

ANS :-

syntax:

intitialization

while(condition)

{

statement;

i++;

}

4) Do...While Loop:-Exit Controlled Loop

syntax:-

intialization

do

{

statement;

updation;

}while(condition);

**Array**

Arrays:-

Array is a user defined data type which is used to store multiple values in a single element.

Array index value starts from 0.

window + 0

window + 1

window + 2

syntax:-

datatype arrname[size]={elements};

Types of Array:-

1) Single Dimensional Array(1-D):-datatype arrname[size]

2) Multi Dimensional Array(2-D):- datatype arrname[row][col]

Matrix:-

1 2 4

6 7 8

Strings

Strings:-

A series of characters.

Ex:- python

string:- %s

syntax:- char varname[20];

index value starts from 0.

gets()

string inbuilt methods:-

#include<string.h>

strlwr:- lowercase

strupr:- uppercase

strlen:- length

strrev:- reverse

strcmp:- compare

strcpy:- copy

Function

Functions:-

Function is a block of code or a piece of code that we can use again and again.

With the help of function we can reuse the same code by

calling the function again and again

Function Provides reusability because we can call same function as many times as we want.

Types of Function:-

1) Inbuilt Function or Library Functions:-

printf(),scanf(),main(),strlen()...

2) User Defined Functions:- Which is defined by user:-

add(),greetings(),factorial()...

Function Declaration:-

We will declare the function

Function Calling:-

We will call the function

syntax:-

datatype functionname(int a, int b) -- Function Declaration

{

block of code...

}

main()

{

functionname(10,20); -- Function calling

}

ategories of Function:-

There are 4 categories of Functions:-

1) Function Without Parameters and Without Return Value

2) Function With Parameters and Without Return Value

3) Function Without Parameters and With Return Value

4) Function With Parameters and With Return Value

Example :-

login()- 1

login(email,mobile). - 2

Here email and mobile are parameters.

Click on mobile button and it will return the mobile page. - 3

login(email,mobile)

Enter email:- manav@gmail.com

if it matches:

return home page

if it not matches

return invalid email or password

Interview Question:-

Difference Between Parameters and Arguments.

Parameters:-

The variables which we will write during function declaration

is called parameters.

Arguments:-

The values which we will write during function calling is

called arguments.

Return Value:-

It will return some value or it will return something

int main() --int--return type

{

return 0; -- return value

}

C++

C++ Programming:-

C++ is a high-level,

object oriented programming language.

OOPS:-

Object Oriented Programming System

POP vs OOP

POP:- Procedural Oriented Programming Language:- C Language

OOP:- Object Oriented Programming:-

C++

It is based on objects

Topics:-

Class

Objects

Constructor

Encapsulation

Abstraction

Inheritance

Polymorphism

Function Overloading

Function Overriding

History of C++ :-

C++ was created Bjarne Stroustrup in 1983 at bell laboratories,

California.

C++ Syntax:-

# - predefined

Include - To add something

Iostream - input output stream- same as stdio.h

using namespace std; - To use variables as names

main()

{

code...

}

cout<<

user input value - cin>>

OOPS:-

Object Oriented Programming System or Structure

--> Security

Class:-

Class is a logical entity that has some specific attributes and

methods.

OR

Class is a collection of data member and member function.

Data member:- Variables

Member Function:- Methods

Object:-

Object is an instance or a variable of a class.

Access Modifiers:-

1) public:- It can access anywhere in the class.

2) private:- It can only access within the class.

3) protected:-

syntax:-

class classname

{

Access modifiers;

Data members - variable

member function - methods

};

main()

{

classname obj; - object creation

}

Bydefault your class is in private mode.

Interview Question:-

Difference Between Method and Function:-

Function:-

Function will declare outside the class.

Method:-

Method will declare inside the class.

Constructor:-

Constructor is a special method that will call

automatically

when we create object.

We don't need to call the constructor as it will call automatically.

Constructor will always declare in public mode.

Types of Constructor:-

1. Default Constructor:-

A constructor which does not have

parameters.

1. Parameterized Constructor:-

A constructor which have parameters.

Note:- Class name and Constructor name always be same.

this pointer:-

To specify the variables and parameters

Encapsulation:-

Encapsulation is a process of wrapping the data in a

single entity

It provides security to the data.

setter and getter

---> setter is used to set the data and getter is used to get the

data.

Abstraction:-

Abstraction is a process to hiding unnecessary details

from the user and only show the necessary information to the user.

---> In Abstraction we will show only necessary information to the user.

There are two types of abstraction:-

1) Abstraction through header files

2) Abstraction through class

Polymorphism:-

Polymorphism derives from greek word. Poly means many and morphism means forms. So Polymorphism means

many forms.

---> Polymorphism is defines as one class having many forms

There are two types of Polymorphism:-

1. Method Overloading:-

When one class having same method name but different parameters is called

Method Overloading

1. Method Overriding:-

When Parent class and child class having same method name is called Method

Overriding.

Inheritance:-

1) Single-level Inheritance:-

class A(papa)

|

class B(son)

2) Multilevel Inheritance:-

class A(dada

|

class B(papa)

|

class C(beta)

3) Multiple Inheritance:-

class A class B

|

class C

4) Hierarchical Inheritance:-

class A

|

class B class C

5) Hybrid Inheritance:-

---> It is a combination of different Inheritance.

Inline Function:-

---> Inline function is used to increase the efficiency or speed of program

---> Inline function will replace function declaration by function calling

---> So it will call the function when we declare the function.

---> We have to use the inline keyword for using the inline function.

Friend Function:-

---> By using friend function we can access all the private members outside

the class.

---> To use friend function we will use friend keyword.

Templates in C++ :-

---> Templates is a generic programming which is used to perform programming

of different datatypes in same class.

---> By using templates we can perform different datatypes in the same class

---> To use templates we will use template keyword.

syntax:-

template <class T >

void funname(T &a, T &b)

{

code ...

}

DATABASE

DBMS:-

Database Management System

Database:-

To store the large amount of data

oracle

mysql

mongodb

postgresql

SQL:-

Structured Query Language.

---> It stores your data in a structured format

using rows and columns(tables)

mysql:- server

host:- localhost(IP:- 127.0.0.1)

Port:- 3306

per software/company there is only one database

but there are more than one table in one database

Database creation:-

CREATE DATABASE 12may\_db

To view all databases:-

SHOW DATABASES

int:-11

varchar-0 to 255

Table creation:-

CREATE TABLE student

(

stu\_id int,

stu\_name varchar(40),

stu\_subject varchar(40)

)

Insertion of data in table:-

INSERT INTO student(stu\_id,stu\_name,stu\_subject) VALUES(1,"Milan","Python");

INSERT INTO student(stu\_id,stu\_name,stu\_subject) VALUES(2,"Raj","Java");

INSERT INTO student(stu\_id,stu\_name,stu\_subject) VALUES(3,"Nisha","Android");

INSERT INTO student(stu\_id,stu\_name,stu\_subject) VALUES(4,"Mansi","PHP");

INSERT INTO student(stu\_id,stu\_name,stu\_subject) VALUES(5,"Jayraj","HTML");

To change any column name:-

ALTER TABLE student CHANGE stu\_name student\_name varchar(40)

To add new column:-

ALTER TABLE student ADD stu\_address varchar(40)

To update particular data in table:-

UPDATE student SET student\_name="nisha" WHERE stu\_id=3

UPDATE student SET stu\_subject="Flutter" WHERE stu\_id=4

To delete particular data from table:-

DELETE FROM student WHERE stu\_id=2

To delete the whole data:-

TRUNCATE TABLE student

To delete the whole table:-

DROP TABLE student

Difference between drop and truncate:-

Truncate will delete all the data from table while drop will

delete the whole table

Key in SQL

1. Primary Key:-

It is a constraint in sql that identifies a

unique value in a table.

---> In one table, only one primary key is allowed

auto\_increment:-

It increases the value automatically

CREATE DATABASE myntra\_db

1. Foriegn Key:-

It is a constriant in sql that refers to primary

key of another table

---> We can take more than one foriegn key in one table

---> It is used to link between the two or more tables.

create table category

(

cate\_id int primary key,

cate\_name varchar(40)

)

CREATE TABLE subcategory

(

subcate\_id int PRIMARY KEY AUTO\_INCREMENT,

subcate\_name varchar(40),

cate\_id\_fk int,

FOREIGN KEY(cate\_id\_fk) REFERENCES category(cate\_id)

)

CREATE TABLE product

(

prd\_id int PRIMARY KEY AUTO\_INCREMENT,

prd\_name varchar(40),

prd\_price int,

prd\_desc\_price int,

cate\_id\_fk int,

subcate\_id\_fk int,

FOREIGN KEY(cate\_id\_fk) REFERENCES category(cate\_id),

FOREIGN KEY(subcate\_id\_fk) REFERENCES

subcategory(subcate\_id)

)

CREATE DATABASE bollywood

CREATE TABLE movies

(

movie\_id int PRIMARY KEY AUTO\_INCREMENT,

movie\_name varchar(40)

)

CREATE TABLE celebrity

(

celebrity\_id int PRIMARY KEY AUTO\_INCREMENT,

celebrity\_name varchar(40),

movie\_id\_fk int,

FOREIGN KEY(movie\_id\_fk) REFERENCES movies(movie\_id)

)

1. Unique Key:-

It is a constraint in sql that identifies a unique

a value in a table.

primary key unique key

--> It cannot accept null value It can accept the

null value.

--> Only one primary key is allowed more than one unique

in one table key is allowed in table

--> It supports auto\_increment. It does not support

auto\_increment

CREATE TABLE Person

(

person\_id int PRIMARY KEY AUTO\_INCREMENT,

person\_name varchar(40),

person\_address varchar(40),

person\_mobile varchar(10) UNIQUE NOT null

)

ALTER TABLE person DROP COLUMN person\_address

Fetching or reading data

select queries

\* - universal

Where Clause

SELECT \* FROM product

SELECT \* FROM product WHERE prd\_id=2

SELECT \* FROM product WHERE prd\_id=3

SELECT \* FROM product WHERE prd\_name="Dress"

SELECT \* FROM product WHERE prd\_price=2000

SELECT prd\_name FROM product

SELECT prd\_name,prd\_price FROM product

SELECT prd\_name,prd\_price,prd\_desc\_price FROM product

SELECT \* FROM product WHERE prd\_price>1500

SELECT \* FROM product WHERE prd\_desc\_price<1500

SELECT \* FROM product WHERE prd\_price BETWEEN 1000 and 3000

Aggregate Functions:-

MIN:-

SELECT MIN(prd\_price) FROM product

SELECT prd\_name, MIN(prd\_price) FROM product

MAX:-

SELECT prd\_name,MAX(prd\_price) FROM product

SUM:-

SELECT SUM(prd\_price) FROM product

AVG:-

SELECT AVG(prd\_price) FROM product

COUNT:-

SELECT COUNT(prd\_id) FROM product

Aliases:-

To change the column name temporary-as

SELECT SUM(prd\_price) AS total FROM product

ORDER BY:-

SELECT \* FROM product ORDER BY prd\_price ASC

SELECT \* FROM product ORDER BY prd\_desc\_price DESC

SELECT \* FROM product ORDER BY prd\_name ASC

Date-time Inbuilt Functions:-

current date:-SELECT curdate()

current time:-SELECT curtime()

date and time:-SELECT now()

DISTINCT:-

SELECT DISTINCT age FROM student

string inbuilt functions:-

1. concat:-

SELECT concat("Manav"," ","Gathani")

SELECT concat(fname," ",lname) AS fullname FROM

student

1. reverse:-

SELECT reverse("sohan")

3) length:- SELECT length("Manav")

4) upper:- SELECT upper("manav")

5) lower:- SELECT lower("MANAV")

GROUP BY:-

SELECT age,COUNT(id) FROM student GROUP BY age

SELECT fname,COUNT(id) FROM student GROUP BY fname

Joins:-

Joins is used to join two or more tables.

1. Inner Join:-

Inner Join will join all the data based on

related columns

SELECT category.cate\_name,subcategory.subcate\_name FROM category

INNER JOIN subcategory ON category.cate\_id=subcategory.cate\_id\_fk

SELECT category.cate\_name,subcategory.subcate\_name,product.prd\_name,product.prd\_price,product.prd\_desc\_price FROM category

JOIN subcategory ON category.cate\_id=subcategory.cate\_id\_fk

JOIN product ON category.cate\_id=product.subcate\_id\_fk

1. Left Join:-

It returns all the records from left table

(1st table) and only matching records from right table.

---> If it does not matches it will return null values.

SELECT category.cate\_name,subcategory.subcate\_name

FROM category

LEFT JOIN subcategory ON category.cate\_id=subcategory.cate\_id\_fk

1. Right Join:-

It returns all the records from right table

and only matching records from left table

SELECT category.cate\_name,subcategory.subcate\_name

FROM category

RIGHT JOIN subcategory ON

category.cate\_id=subcategory.cate\_id\_fk

LIKE and Wildcards:-

SELECT \* FROM product WHERE prd\_name LIKE 'D%'

SELECT \* FROM product WHERE prd\_name LIKE '%s'

SELECT \* FROM product WHERE prd\_name LIKE '\_r%'

SELECT \* FROM product WHERE prd\_name LIKE '\_\_a%'

SELECT \* FROM product WHERE prd\_name LIKE 'D\_\_%'

SELECT \* FROM product WHERE prd\_name LIKE 'D%s'

SELECT \* FROM product WHERE prd\_name LIKE '%Dre%'

SELECT \* FROM product WHERE prd\_name NOT LIKE 'D%'

Advance SQL:-

Procedures:-

It is used to execute a same query again and again.

create procedure procedurename()

CREATE DATABASE procedure\_db

CREATE TABLE candidates

(

c\_id int,

c\_name varchar(40)

)

DELIMITER $$

CREATE PROCEDURE insertdata()

BEGIN

INSERT INTO candidates(c\_id,c\_name) VALUES(1,"Manav");

END

CALL insertdata();

CALL insertdata();

CALL insertdata();

CALL insertdata();

CALL insertdata();

Procedure with Parameters:-

DELIMITER $$

CREATE PROCEDURE insertDiffData(i int, j varchar(40))

BEGIN

INSERT INTO candidates(c\_id,c\_name) VALUES(i,j);

END

CALL insertDiffData(2,"Nisha");

CALL insertDiffData(3,"Milan");

CALL insertDiffData(4,"Sohan");

CALL insertDiffData(5,"Dipak");

Triggers in SQL:-

CREATE DATABASE trigger\_db

CREATE TABLE candidates

(

c\_id int,

c\_name varchar(40)

)

CREATE TABLE test

(

id int,

name varchar(40),

date\_time timestamp,

action\_performed varchar(40)

)

DELIMITER $$

CREATE TRIGGER insert\_trigger AFTER INSERT ON candidates

FOR EACH ROW

BEGIN

INSERT INTO test(id,name,action\_performed)

VALUES(new.c\_id,new.c\_name,"Record Inserted!");

END

DELIMITER $$

CREATE TRIGGER update\_trigger AFTER UPDATE ON candidates FOR EACH ROW

BEGIN

INSERT INTO test(id,name,action\_performed) VALUES(new.c\_id,new.c\_name,"Record Updated!");

END

UPDATE candidates SET c\_name="Akshay" WHERE c\_id=1

DELIMITER $$

CREATE TRIGGER delete\_trigger AFTER DELETE ON candidates

FOR EACH ROW

BEGIN

INSERT INTO test(id,name,action\_performed)

VALUES(old.c\_id,old.c\_name,"Record Deleted!");

END

DELETE FROM candidates WHERE c\_id=2

Sql select distinct statement

ANS:- SELECT DISTINCT \* FROM Employee;

SELECT DISTINCT emp\_no FROM employee;

AND, OR ,NOT

ANS:- SELECT \* FROM EMPLOYEE WHERE salary<25000 OR emp\_no > 5;

SELECT \* FROM EMPLOYEE WHERE salary<25000 AND emp\_no > 5;

SELECT \* FROM EMPLOYEE WHERE NOT salary=25000;

Combining AND , OR , NOT

SELECT \* FROM employee WHERE emp\_no >=5 AND (salary < 2000 OR salary > 50000);

PYTHON

\*OPEN SORCE

\*Platform Independent

\*case sensitive

\*sort programming language

\*Python Version

-> 3.12.0

\*IDLE

-> Integrated Development learning

Environment.

\*COMMENT

-> ‘’’ ‘’’

#

\*print

-> print krva mate.

\*check The Datatype

-> type

Ex – a=10

Print(“A : ”,a)

Print(type(a))

\*String

-> Double and single inverted coma (“ “ …. ‘ ’)

\*Boolean

-> True. (1)

Folse. (0)

\*User pase thi value gate krva mate.

-> input.

\*IF CONDITION

-> if condition :

statement

(4 space -> indentation)

\*IF/ELSE

-> if condition:

Statement

Else:

statement

\*NESTED IF

-> if condition :

If condition :

statement

else:

statement

else:

statement

\*LADDER IF

-> it condition:

Statement

elif condition:

Statement

elif condition:

statement

else:

statement

\*LOOP

-> CYCLE

\*FOR LOOP

-> for i in range(start and jump):

Statement()

Start - 0

And - -1

Jump - +1

\*string function

(space will be count and index start for 0)

\*Library function (type of {} breaket)

-> len() – find to length.

-> capitalize() – capitel to small first word.

-> casefold() – total word small

-> upper() – total word capitel

-> lower() – total word small

-> swapcase() – capitel to small and small to

Capitel.

->Center() – center ma leva mate.

-> count – count karne ke liye . capitel to capitel

and small to small.

->Endswith()-(Boolean fuction,true and false)

Last ma true ya false.

->startswith()- start ma true ya false.

->find()- right to find which number .

-> index() – find to index in position

number.

>Isalnum() – number,string …… (not space

count )

* Isalpha() – alpha mate .
* Isalnu() – number mate.
* Istitle() – title in first letter cappitel .
* Isupper() – total word capitel.
* Islower() – total word small .
* Isspace() – space counte kar ne ke liye.
* Replace() – replay word.

\* INDEX

-> start -> and -> jump

(negative or positive value)

Ex :- 012345678910(plues)

Tops technologis

109876543210(munes)

\*LIST (type of [] breacket)

-> list is group of diffrent type of data.

-> append – last me single data add karne

ke liye.

* Clear – list clear kar ne ke liye.
* Copy – copy karne ke liye.
* Count – count karne ke liye.

\*TUPLE (type in () breaket)

-> (data all type allowoud and not method

Change)

\*DICTIONARY( TYPEIN {}BREACKET)

-> key and values

->Get - value gate

->items - items the show

->keys – keys the show

->pop – remove

->popitem – last item remove

->update – jod ne ke liye.