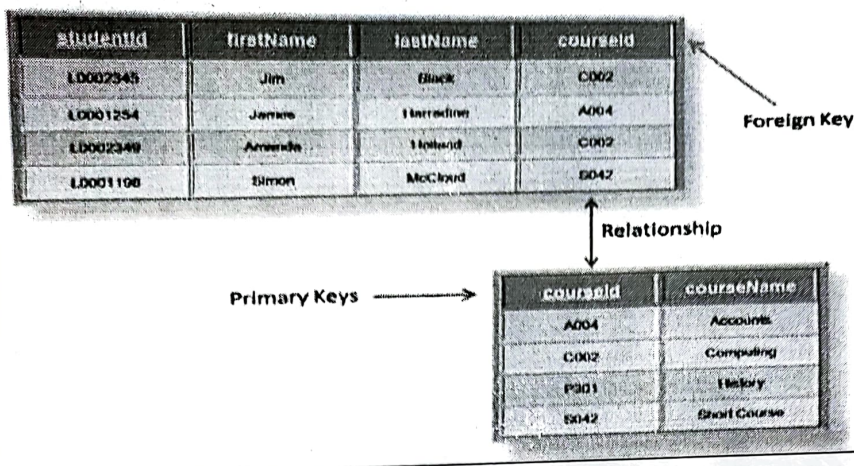
HALF YEARLY EXAM MARKING SCHEME
(2023-24)
CLASS XII
COMPUTER SCIENCE (083)

	SECTION A	
1.	FALSE	1
2.	b. 6,20	1
3.	c. -244.0	1
4.	a. 2	1
5.	b. Domain	1
6.	b. -	1
7.	a. Day={1:'monday',2:'tuesday',3:'wednesday'}	1
8.	b. ceiePO	1
9.	b. tup.insert(2,3)	1
10.	d. sum()	1
11.	TRUE	1
12.	a. 0	
13.	c. A candidate key that is not a primary key is a foreign key.	1
14.	c. Distinct	1
15.	c. global b	1
16	(d)Real	1
17.	c	1
18.	a	1

	SECTION B	
19.	e. All of these	2
20.	**def** .... **while** num >0: rem=num%10 return rev	2
21.	A foreign key is the one that is used to link two tables together via the primary key. It means the columns of one table points to the primary key attribute of the other table.	2

		
22.	(a) @20 otnmx SC@ (b) dict_items([('name', 'Aman'), ('age', 27), ('address', 'Delhi')])	2
23.	(a) print(my_exam[: :2] (b) {'age': 26} Aman	2
24.	Count(class) will count number of non null entries in column class; however count(*) will count total number of rows in a table. OR DDL : alter,drop DML : insert ,update	2
25.	13# OR (22, 44, 66)	2

	SECTION C	

26.	`PLACES={1:"Delhi",2:"London",3:"Paris",4:"New York",5:"Dubai"}`
	`def countNow(PLACES):`
	`  for place in PLACES.values():`
	`  if len(place)>5:`
	`  print(place.upper())`
	`countNow(PLACES)`
		3

27.

(a)

Acode	Name	Type	City
A01	Amrita	Savings	Delhi
A01	Amrita	Savings	Nagpur
A02	Parthodas	Current	Mumbai

1

(b) (i)

DISTINCT TID
101
NULL
102
104
103

2

(ii)

TID	COUNT(*)	MIN(FEES)
101	2	12000

(iii)

CNAME
Digital marketing
Mobile Application Development

(iv) 15500

28.

```
def COUNTLINES() :
    file = open ('TESTFILE.TXT', 'r')
    lines = file.readlines()
    count=0
    for w in lines :
        if (w[0]).lower() not in 'aeoiu'
            count = count + 1
    print ("The number of lines not starting with
vowel: ", count)
    file.close()

COUNTLINES()
```

3

29.

(i)

```
UPDATE Personal
SET Salary=Salary + Salary*0.5
WHERE Allowance IS NOT NULL;
```

(ii)

```
SELECT Name, Salary + Allowance AS "Total Salary" FROM Personal;
```

(iii)

```
DELETE FROM Personal WHERE Salary>25000;
```

3

30.

```
def Push_element(cust):
  if cust[2]=="Goa":
    L1=[cust[0],cust[1]]
    status.append(L1)

def Pop_element ():
  num=len(status)
  while len(status)!=0:
```

3

```
    dele=status.pop()
    print(dele)
    num=num-1
else:
    print("Stack Empty")
```

	SECTION D	

31.	stackItem=[]
	def Push(SItem):
	count=0
	for k in SItem:
	if (SItem[k]>=75):
	stackItem.append(k)
	count=count+1
	print("The count of elements in the stack is : ", count)
		3+1

32.	(i) pickle (ii) fout=open('temp.dat', 'wb') (iii) Statement 3: pickle.load(fin) Statement 4: pickle.dump(rec,fout)	1 1 2

	SECTION E	
33.	(i) All keys that have the properties to become a primary key are candidate keys. The candidate keys that do not become primary keys are alternate keys. (ii) import mysql.connector as mysql con1 = mysql.connect(host="localhost",user="root", password="tiger", database="sample2023") mycursor=con1.cursor() query = "SELECT * FROM student where fee>{}".format(5000) mycursor.execute(query) data=mycursor.fetchall() for rec in data: print(rec) con1.close()	1 4
34.	Advantage of a csv file: It is human readable – can be opened in Excel and Notepad applications It is just like text file	1

```
import csv
def ADD():
 fout=open("record.csv","a",newline="\n")
 wr=csv.writer(fout)
 empid=int(input("Enter Employee id :: "))
 name=input("Enter name :: ")
 mobile=int(input("Enter mobile number :: "))
 lst=[empid,name,mobile] ---------1/2 mark
 wr.writerow(lst) ---------1/2 mark
 fout.close()

def COUNTR():
 fin=open("record.csv","r",newline="\n")
 data=csv.reader(fin)
 d=list(data)
 print(len(d))
 fin.close()

ADD()
COUNTR()
```

2

2

35. (i)

Text files:

Extension is .txt

Data is stored in ASCII format that is human readable

Has EOL character that terminates each line of data stored in the text files

Binary Files

Extension is .dat

Data is stored in binary form (0s and 1s), that is not human readable.

1

(ii)

```
def Searchtype(mtype):
    fObj = open("CINEMA.DAT", "rb")
    try:
        while True:
            data = pickle.load(fObj)
            if data[2] == mtype:
                print("Movie number:",data[0])
                print("Movie Name:",data[1])
                print("Movie Type:",data[2])
    except EOFError:
        fObj.close()
```

4