MARKING SCHEME

Section – A

Q01.	False	(1)
Q02.	(A) eval	(1)
Q03.	(D) dict_student.update(dict_marks)	(1)
Q04.	(B) False	(1)
Q05.	(C) mail2@kvsangathan**.**	(1)
Q06.	(D) close()	(1)
Q07.	(C) alter	(1)
Q08.	(B) DROP DATABASE	(1)
Q09.	(C) S4	(1)
Q10.	(C) Foreign Key	(1)
Q11.	(D) file_object.seek(offset [, reference_point])	(1)
Q12.	(A) DISTINCT	(1)
Q13.	(B) VoIP	(1)
Q14.	(C) 30.6	(1)
Q15.	(D) count(*)	(1)
Q16.	(B) connect	(1)
Q17.	(A) Both A and R are true and R is the correct explanation for A	(1)
Q18.	(C) A is True but R is False	(1)
	Section – B	

Q19.	Def checkNumber(N): # Def should be def
	status = N%2
	return # return what? Should be return status
	#main-code
	num=int(input(“ Enter a number to check :)) # Message not enclosed within quotation mark
	k=checkNumber(num)
	if k = 0: # must be k = = 0
	print(“This is EVEN number”)
	else:
	print(“This is ODD number”)
	
	(½ mark for each correct correction made and underlined.)
		(2)

Q20.	1 mark for each correct point of difference	(2)
Q21.	(A) 322ADORSF	(2)

	(B) dict_keys(['name', 'age', 'dept', 'rno']) 1 Marks for each correct answer.	
Q22.	A foreign key is used to set or represent a relationship between two relations (or tables) in a database. Its value is derived from the primary key attribute of another relation. (1 mark for explanation and 1 mark for example) (Any relevant correct example may be marked)	(2)
Q23.	(A) (i) HTTP: Hyper Text Transfer Protocol (ii) FTP: File Transfer Protocol (½ mark for every correct full form) (B) TELNET is used to access a remote computer / network. (1 mark for correct answer)	(2)
Q24.	1 20 L@ 4 60 L@M@ 9 120 L@M@N@ (½ M + ½ M + 1 M) means ½ - ½ marks for first two lines and 1 mark for last line. OR [(2, 4), (4, 16)] (½ mark for each correct pair of tuple , ½ mark for enclosing in parenthesis) means concept of tuple and list	(2)
Q25.	1 mark for the difference and 1 mark for appropriate example OR DDL- ALTER, DROP DML – INSERT, UPDATE (½ mark for each correct categorization)	(2)
	Section – C	
Q26.	½ Marks for each correct answer (i) BRAND_NAME FLAVOUR LAYS TOMATO UNCLE CHIPS SPICY HALDIRAM TOMATO (ii) BRAND_NAME FLAVOUR PRICE QUANTITY HALDIRAM TOMATO 25 30 (iii) BRAND_NAME LAYS (iv) count(distinct (BRAND_NAME)) 3 (v) PRICE PRICE*1.5	(3)

	10 15 (vi) distinct (BRAND_NAME) UNCLE CHIPS LAYS HALDIRAM	

Q27.	def countINDIA():
	f=open('d:\\myfile.txt')
	data=f.read()
	data=data.split()
	
	ctr=0
	for w in data:
	if w.upper()=='INDIA':
	ctr=ctr+1
	print('Frequency of India is ',ctr)
	
	#main
	countINDIA()
	OR
	def countVowel():
	ctr=0
	f=open('d:\\myfile.txt')
	data=f.read()
	for ch in data:
	if ch.lower() in 'aeiou':
	ctr=ctr+1
	
	print('Total number of vowels are : ', ctr)
		(3)

Q28.	(A) (i) select bname, auname, price from books where bid like “comp%”; (ii) update books set price = price + 50 where bid like “hist%”; (iii) select * from books order by price; (iv) select bid, bname, qty_issued from books, issued where books.bid = issued.bid; (½ mark for each correct SQL) (B) SHOW TABLES; (1 mark for correct answer)	(2+1)
Q29.	def lenFOURword(L):	(3)

```
    indexList=[]
    for i in range(len(L)):
        if len(L[i])==4:
            indexList.append(i)
    return indexList
```

½ mark for function header
½ mark for declaration of indexList
½ mark for loop
½ mark for checking condition
½ mark for appending
½ mark for returning

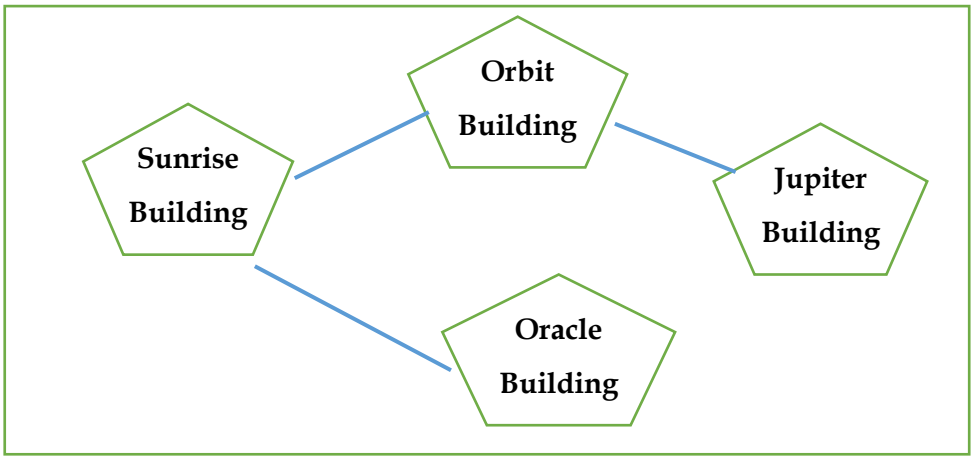
Q30. (3)

```
xiia=[]
student=[['Rajveer', '99999999999','XI', 'B'],['Swatantra', '8888888888','XII', 'A'],
['Sajal','77777777777','VIII','A'],['Yash', '1010101010','XII','A']]

def pushElement(student):
    for d in student:
        if d[2]=='XII' and d[3]=='A':
            xiia.append([d[0],d[1]])

def popElement():
    while len(xiia)!=0:
        print(xiia.pop())
    else:
        print('Stack Empty')

pushElement(student)
print(xiia)
popElement()
```

(1.5 marks for correct pushElement() and 1.5 marks for correct popElement())

OR

```
stackItem=[]
def Push(SItem):
    count=0
    for k in SItem:
        if (SItem[k]>=25):
```

	stackItem.append(k) count=count+1 print("The count of elements in the stack is : ", count) (1 mark for correct function header 1 mark for correct loop ½ mark for correct If statement ½ mark for correct display of count)	
	Section – D	
Q31.	i) Suggest a cable layout of connections between the buildings.  ii) Orbit Building iii) a. Internet Connecting Device/Modem- Orbit Building b. Switch- Each Building iv) MAN, it is formed to connect various locations of the city via various communication media. v) PAN is "Personal Area Network", basically configured at home area.	(5)
Q32.	(A) 1005 12 (B) Statement 1: con1.cursor() Statement 2: mycursor.execute(querry) Statement 3: con1.commit() (1 mark for each correct answer) OR (A) c&&vVpP (B) Statement 1:con1.cursor() Statement 2: mycursor.execute("select * from student where Marks>75")	(5)

	Statement 3: mycursor.fetchall() (1 mark for each correct statement)	

Q33.	Advantage of a csv file:
	* It is human readable – can be opened in Excel and Notepad applications
	* It is just like text file
	Program:
	import csv
	def ADD():
	fout=open("teacher.csv","a",newline="\n")
	wr=csv.writer(fout)
	tid=int(input("Enter teacher id :: "))
	name=input("Enter name :: ")
	mobile=int(input("Enter mobile number :: "))
	lst=[tid, name, mobile] --------- ½ mark
	wr.writerow(lst) --------- ½ mark
	fout.close()
	
	def COUNTR():
	fin=open("teacher.csv","r",newline="\n")
	data=csv.reader(fin)
	d=list(data)
	print("No of records :",len(d))
	fin.close()
	
	ADD()
	COUNTR()
	(1 mark for advantage
	½ mark for importing csv module
	1 ½ marks each for correct definition of ADD() and COUNTR()
	½ mark for function call statements
	)
	
	OR
	
	Difference between binary file and csv file: (Any one difference may be given)
	Binary file:
		(5)

* Extension is .dat
* Not human readable
* Stores data in the form of 0s and 1s

CSV file
* Extension is .csv
* Human readable
* Stores data like a text file

Program:

```
import csv
def add():
    fout=open("employee.csv","a",newline='\n')
    wr=csv.writer(fout)
    eid=int(input("Enter Employee Id :: "))
   name=input("Enter employee name :: ")
   salary =int(input("Enter salary :: "))
   FD=[eid, name, salary]
   wr.writerow(FD)
   fout.close()

def search():
    fin=open("employee.csv","r",newline='\n')
    data=csv.reader(fin)
    found=False
    print("The Details are")
    for i in data:
        if int(i[2])>40000:
              found=True
               print(i[0], i[1], i[2])
     if found==False:
          print("Record not found")
fin.close()
add()
print("Now displaying")
search()
```

(1 mark for difference
½ mark for importing csv module
1 ½ marks each for correct definition of add() and search()

	½ mark for function call statements)		
Section – E			
Q34.	(i)	stockid	(1+
	(ii)	degree = 8, cardinality = 3	½+½
	(iii)	(a) insert into stock values(201,'2022-10-18','neckphone','boat',500); (b) update stock set price=price*0.95 where year(dopurchase)=2020; OR (a) delete from stock where year(dopurchase) < 2015; (b) alter table stock add column STATUS char(1);	2)
Q35.	(i)	pickle (1 mark for correct module)	(4)
	(ii)	fout=open('temp.dat', 'wb') (1 mark for correct statement)	
	(iii)	Statement 3: pickle.load(fin) Statement 4: pickle.dump(rec,fout) (1 mark for each correct statement)	

0-O-o- End of Paper –o-O-0