|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **PB1/CSAK/1223/B 23-NOV-2023** | | | |
| **PRE BOARD EXAMINATIO I – (2023-24)**  **ANSWER KEY** | | | |
| **Subject: COMPUTER SC (083)**  **Grade: XII** | | Max. Marks: 50Time: 2 Hrs 30 Mins | |
| **Qno** | **SECTION A** | | Mark |
| **1.** | False | | **1** |
| **2.** | c. Update | | **1** |
| **3.** | b.5 | | **1** |
| **4.** | a. The number $$0$$ | | **1** |
| **5.** | a. Asc,Desc | | **1** |
| **6.** | b. Peer to Peer | | **1** |
| **7.** | a. dict\_exam.update(dict\_result) | | **1** |
| **8.** | [['hello', 'bye'], 90, 67] | | **1** |
| **9.** | b. Statement 4 | | **1** |
| **10.** | d. i and iii | | **1** |
| **11.** | b. PAN | | **1** |
| **12.** | b. 5 20 5 4 20 | | **1** |
| **13.** | c. finally | | **1** |
| **14.** | c. A candidate key which is not a primary key | | **1** |
| **15.** | a. True | | **1** |
| **16.** | Gateway | | **1** |
| **17.** | **a.** Both A and R are true and R is the correct explanation of A. | | **1** |
| **18.** | b. Both A and R are true and R is not correct explanation of A. | | **1** |
|  | SECTION B | |  |
| **19.** | a. i. NIC- Network Interface Card  ii. MAC- Media Access Control  (1/2 mark each)  b. Circuit Switching - First the complete end-to-end transmission path is established between the source and the destination computers. Then the message is transmitted through the path. The main advantage of this technique is guaranteed delivery of the message. Mostly used for voice communication.  (1 mark)  **OR**  a. Advantage: The network remains operational even if one of the nodes stops working. Disadvantage: The network stops working if the central hub stops working.( Or any other valid advantage or disadvantage. Each carries 1/2 mark)  b. | | **2** |
| **20.** | Value=30  for val in range(0,Value):  if val%4==0:  print (val\*4)  elif val%5==0:  print (val+3)  else:  print(val+10)  (1/2 mark each correction) | | **2** |
| **21.** | def vowel\_less(li):  vow="AEIOU"  for i in li:  vowelcount=0  for j in i:  if j.upper() in vow:  vowelcount+=1  if vowelcount==0:  print(i) | | **2** |
| **22.** | 2#B  87#3  56#4  9#T | | **2** |
| **23.** | a. L.remove(67)  b. text.find(‘pi’)  OR  import statistics  print( statistics.stdev(CricketScore)) | | **2** |
| **24.** | SQL Command to add primary key:  ALTER TABLE Customer ADD Ccode INTEGER PRIMARY KEY;  As the primary key is added as the last field, the command for inserting data will be:  INSERT INTO Employee VALUES("Fatima","New Delhi",3400,989);  Alternative answer:  INSERT INTO Employee(EmpId,Ename,Department,Salary)  VALUES("Fatima","New Delhi",3400,989);  1 mark each correct query  **OR**  To delete the attribute, category:  ALTER TABLE EmpDetail DROP category;  To add the attribute, Department  ALTER TABLE EmpDetail ADD Department char(10) NOT NULL;  1 mark each correct query | | **2** |
| **25.** | 25#1#25#2#  50@2@50@4@ | | **2** |
|  | **SECTION C** | |  |
| **26.** | **gR8$$#$$dAy** | | **3** |
| **27.** | **(1 mark each)** | | **3** |
| **28.** | def displaywords():  fin = open('data.txt','r')  fout=open(‘Newdata.txt’,’w’)  s = f.read()  lst = s.split()  for x in lst:  if len(x)>3:  write(x, end=" ")  fin.close()  fout.close()  (1/2 mark for opening the file in correct modes, 1 mark for correct reading and splitting, 1 mark for loop and condition,1/2 mark for closing both the files)  **OR**  def filter (oldfile,newfile) :  fin=open(oldfile,”r”)  fout=open(newfile,”w”)  while True:  text=fin.readline( )  if len(text)==0:  break  if text[0] ! = “@”:  fout.write(text)  fin.close( )  fout.close( )  filter(“source.txt”,”target.txt”)  (1½ mark for correct logic, ½ mark for function call,1 mark for opening the files and closing the files) | |  |
| **29.** | a. The Primary Key should be Productid since it uniquely identifies each row.  b. UPDATE inventory SET stock=stock+10 WHERE company = 'Parley';  c. DELETE FROM inventory WHERE RATING IS NULL;  1 mark each | | **3** |
| **30.** | stack=[ ]  def push(stack, data):  for x in data:  if data[x]>25:  stack.append(x)  def pop(stack):  while stack!=[ ]:  for i in range(-len(stack)-1,0,-1):  print(stack[i])  else:  print(“Stack Empty”)  (1½ mark for push ( ) and 1½ mark for pop() , ½ mark for correct function header and 1 mark for correct logic and statements) | | **3** |
|  | **SECTION D** | |  |
| **31.** | a. Select ClientName,City from Client where city in(‘Mumbai’,”Delhi’);  b. Select manufacturer,count(\*) from product group by Manufacturer;  c. Select ProductName,Manufacturer,ExpiryDate from product where ExpiryData<=‘2010-12-31’;  d. Select P.P\_ID,C.ClientName,C.City,P.ProductName from Product P,Client C where P.P\_ID=C.C\_CID or P.P\_ID IS NULL;  **(1 mark each)** | | **4** |
| **32.** | import csv  def ADD():  empid=input(“Enter employee ID”)  name=input(“Enter employee name”)  mobile=int(input(“Enter mobile number”))  salary=int(input(“Enter salary”))  data=[empid,name,mobile,salary]  head=[‘EmpID’,’Name’,’Mobile’,’Salary’]  fout=open(‘Record.csv’,’a’,newline=’’)  csvw=csv.writer(fout,delimiter=’,’)  csvw.writerow(head)  csvw.writerow(data)  fout.close()  def COUNT():  count=0  f=open(‘Record.csv’,’r’)  csvr=csv.reader(f,delimiter=’,’)  for data in csvr:  if data[3]>=10000:  count+=1  return count  ( 2 mark each function) | | **4** |
|  | **SECTION E** | |  |
| **33.** | a.    Star Topology- The above layout is very effective to connect all the three offices of the Mumbai branch.  b. The server should be placed in Admins block as it has maximum number of computers.  c. Switches/Hubs in all the blocks since the computers need to be connected to the network. Repeaters between all the blocks .The reason being the distance is more than 100m.  d. Voip- Voice over internet protocol.  e. Firewall.  (1 mark each) | | **5** |
| **34.** | (minimum 2 points each-2 marks )  import pickle  def HRdata():  fin=open(“Employee.Dat”,”rb”)  fout=open(“HR.Dat”,”wb”)  count=0  try:  while True:  data=pickle.load(fin)  if data[2].upper()==”HR”:  pickle.dump(data,fout)  count+=1  except:  fin.close()  fout.close()  return count  ½ mark for correctly opening and closing files, ½ mark for correct try and except block, ½ mark for correct loop, 1 mark for correctly copying data, ½ mark for correct return statement  OR  a. Absolute path and relative path:An absolute path or a full path is the path to a location in an operating system. The specialty of this pathway is that it specifies the location from the root directory and leads to the locations. A full path always includes a detailed map to the location. Absolute file paths are notated by a leading forward slash or drive label. For example: /home/example\_user/example\_directory or C:/system32/cmd.exe**.**  A relative path is a set of information that leads to a location relative to the current directory. Its related to the location from current directory. However, a relative pathway does not include the root directory and thus do not begin with the backslash or the forward slash. A relative path is also called a non-absolute path or a partial path. Relative**file paths are notated by a lack of a leading forward slash**. For example, example\_directory.  (2 points- 2marks)  import pickle  def findStudent(SGrade):  fobj=open(“STUDENT.DAT”,”rb”)  found=0  try:  while True:  data=pickle.load(fobj)  for i in data:  if data[i][1]==SGrade:  print(“Student Number:”,i)  print(“Student Name:”,data[i][0])  print(“Student Class:”,data[i][1])  found=1  except:  fobj.close()  if found==0:  print(“Record not found”)  ½ mark for correctly opening and closing files, ½ mark for correct try and except block, ½ mark for correct loop, ½ mark for correct if statement ,1/2 mark for correctly displaying data,1/2 mark for found not found | | **5** |
| **35.** | a. COUNT(\*) returns the count of all rows in the table, whereas COUNT () is used with Column\_Name passed as argument and counts the number of non-NULL values in a column that is given as argument.  b.  Assume the missing statements in the code as:        **OR**  a. Constraints are the conditions that can be enforced on the attributes of a relation. The constraints come in play whenever we try to insert, delete or update a record in a relation  **UNIQUE:** Constraint means that the values under that column are always unique.  e.g Roll\_no number(3) unique;  **PRIMARY KEY:** Constraint means that a column can not have duplicate values and not even a null value.  e.g. Roll\_no number(3) primary key;  The main difference between unique and primary key constraint is that a column specified as unique may have null value but primary key constraint does not allow null values in the column.  b.  import mysql.connector as mysql  def sql\_data():  con1=mysql.connect(host="localhost",user="root", password="tiger", database="school")  mycursor=con1.cursor()  print("Students with marks greater than 75 are : ")  mycursor.execute("select \* from student where Marks>75")  data=mycursor.fetchall()  for i in data:  print(i) | |  |

\*\*\*