

VELAMMAL MAT. HR. SEC. SCHOOL, SURAPET, CHENNAI – 66
XII - COMPUTER SCIENCE
QEX REVISION
CHAPTER 9 LISTS, TUPLES, SETS AND DICTIONARY

PART-II

1. What is a list in Python? Give example.

- A list in Python is known as a “sequence data type” like strings.
- It is an ordered collection of values enclosed within square brackets [].
- Each value of a list is called as element.
- It can be of any type such as numbers, characters, strings and even the nested lists as well.
- The elements can be modified or mutable which means the elements can be replaced, added or removed.
- **Example**
Marks = [10, 23, 41, 75]
Fruits = ["Apple", "Orange", "Mango", "Banana"]
MyList = []

2. How will you access the list elements in reverse order? (OR) What is reverse indexing? Give Example.

- Python enables reverse or negative indexing to access the list elements in reverse order
- The python sets -1 as the index value for the last element in list and -2 for the preceding element and so on. This is called as **Reverse Indexing**.
- **Example :**
Marks = [10, 23, 41, 75]
i = -1
while i >= -4:
 print (Marks[i])
 i = i + -1

Output
75
41
23
10

3. What will be the value of x in following python code?

```
List1 = 2,4,6,[1,3,5]
x=len(List)
```

output:
x value is 4

4. Differentiate del with remove() function of List

| del | remove() |
|--|--|
| del statement is used to delete known elements | remove() function is used to delete elements of a list if its index is unknown |
| The del statement can also be used to delete entire list | remove() function cannot be used to delete entire list |

5. Write the syntax of creating a Tuple with n number of elements.

- **Syntax:**
Tuple_Name = (E1, E2, E2 En) # Tuple with n number elements
Tuple_Name = E1, E2, E3 En # Elements of a tuple without parenthesis

Example :
>>> MyTup1 = (23, 56, 89, 'A', 'E', 'T', "Tamil")
>>> MyTup2 = 23, 56, 89, 'A', 'E', 'T', "Tamil"

6. What is a set in Python?

- A set is another type of collection data type.
- A Set is a mutable and an unordered collection of elements without duplicates.
- That means the elements within a set cannot be repeated.

PART – III

1. What are the advantages of Tuples over list

| Tuples | List |
|---|--|
| 1.The elements of a tuple are unchangeable | The elements of list are changeable |
| 2.The elements of tuple are enclosed by parenthesis | The elements of a list are enclosed within square brackets |
| 3.Iterating tuples is faster | Iterating list is not faster |

2. Write a short note about sort().

| sort () | Sorts the element in list | List.sort(reverse=True False, key=myFunc) |
|--|---------------------------|--|
| Both arguments are optional If reverse is set as True, list sorting is in descending order. Ascending is default. Key=myFunc; “myFunc” - the name of the user defined function that specifies the sorting criteria. | | MyList=['Thilothamma', 'Tharani', 'Anitha', 'SaiSree', 'Lavanya'] MyList.sort() print(MyList) MyList.sort(reverse=True) print(MyList) Output: ['Anitha', 'Lavanya', 'SaiSree', 'Tharani', 'Thilothamma'] ['Thilothamma', 'Tharani', 'SaiSree', 'Lavanya', 'Anitha'] |
| Note: sort() will affect the original list. | | |

3. What will be the output of the following code?

```
list = [2**x for x in range(5)]
print(list)
```

output:
[1,2,4,8,16]

4. Explain the difference between del and clear() in dictionary with an example.

| del command | clear() command |
|---|---|
| del keyword is used to delete a particular element | clear() command is used to delete all the elements in a dictionary |
| del keyword is used to remove the dictionary name from memory | clear () command cannot remove the dictionary name from memory |
| <ul style="list-style-type: none"> • Syntax: <i>del dictionary_name[key]</i> • <i>del dictionary_name</i> • Example: | <ul style="list-style-type: none"> • Syntax: • <i>dictionary_name.clear()</i> • Example: |

- del Dict['Mark1']
- del Dict

- Dict.clear()

5. List out set operations supported by python.

- Different set operations are
- **Union**
- **Intersection**
- **Difference and**
- **Symmetric difference.**

6. What is the difference between List and Dictionary

| LIST | DICTIONARY |
|---|---|
| List is an ordered set of elements | Dictionary is a data structure that is used for matching one element (Key) with another (Value) |
| Index values can be used to access a particular element | In Dictionary key represents index. Key may be a number or a string |
| List are used to look up a value | Dictionary is used to take one value and look up another value |

PART -IV

1. What are the different ways to insert an element in a list. Explain with suitable example?

- append() function is used to add a single element
- extend() function is used to add more than one element to an existing list.
- insert() function is used to insert an element at any position of a list.

- **(i) append() function is used to add a single element**

Syntax:

List.append (element to be added)

Example

```
>>> Mylist=[34, 45, 48]
>>> Mylist.append(90)
>>> print(Mylist)
[34, 45, 48, 90]
```

- **(ii) extend() function is used to add more than one element to an existing list.**

Syntax:

List.extend ([elements to be added])

Example

```
>>> Mylist.extend([71, 32, 29])
>>> print(Mylist)
[34, 45, 48, 90, 71, 32, 29]
```

- (iii) If you want to include an element at your desired position, you can use insert() function.
- **The insert() function is used to insert an element at any position of a list.**

Syntax:

List.insert (position index, element)

Example :

```
>>> MyList=[34,98,47,'Kannan', 'Gowri', 'Lenin', 'Sree']
>>> print(MyList)
```

```
[34, 98, 47, 'Kannan', 'Gowri', 'Lenin', 'Sree']
>>> MyList.insert(3, 'Ram')
>>> print(MyList)
[34, 98, 47, 'Ram', 'Kannan', 'Gowri', 'Lenin', 'Sree']
```

2. What is the purpose of range() ? Explain with an example.

- The range() is a function used to generate a series of values in Python.
- Using range() function, you can create list with series of values.
- The range() function has three arguments.

Syntax of range() function:

range (start value, end value, step value)

where,

start value – beginning value of series. Zero is the default beginning value.

end value – upper limit of series. Python takes the ending value as upper limit – 1.

step value – It is an optional argument, which is used to generate different interval of values.

Example :

```
for x in range (1, 6):
    print(x)
```

Output

```
1
2
3
4
5
```

3. What is nested tuple? Explain with Example.

- A tuple can be defined inside another tuple is called Nested tuple.
- In a nested tuple, each tuple is considered as an element.
- The for loop will be useful to access all the elements in a nested tuple.

Example

```
Toppers = (("Vinodini", "XII-F", 98.7), ("Soundarya", "XII-H", 97.5),
           ("Tharani", "XII-F", 95.3), ("Saisri", "XII-G", 93.8))
```

```
for i in Toppers:
    print(i)
```

Output:

```
('Vinodini', 'XII-F', 98.7)
('Soundarya', 'XII-H', 97.5)
('Tharani', 'XII-F', 95.3)
('Saisri', 'XII-G', 93.8)
```

25. Explain the different set operations with example.

- The python is also supports the set operations such as **Union, Intersection, difference and Symmetric difference.**

(i) Union: It includes all elements from two or more sets

- In python, the operator | is used to union of two sets. The function union() is also used to join two sets in python.

Example:

```
setA={2,4,6,8}
setB={'A', 'B', 'C', 'D'}
print(setA|setB)
print(setA.union(setB))
```

Output:

```
{2, 4, 6, 8, 'A', 'D', 'C', 'B'}
{2, 4, 6, 8, 'A', 'D', 'C', 'B'}
```

(ii) Intersection:

- It includes the common elements in two sets SetA and SetB
- The operator & is used to intersect two sets in python.
- The function intersection() is also used to intersect two sets in python.

Example :

```
setA={'A', 2, 4, 'D'}
setB={'A', 'B', 'C', 'D'}
print(setA & setB)
print(setA.intersection(setB))
Output:
{'A', 'D'}
{'A', 'D'}
```

(iii) Difference :

- It includes all elements that are in first set (say setA) but not in the second set (say setB)
- The minus (-) operator is used to difference set operation in python.
- The function difference() is also used to difference operation.

Example :

```
setA={'A', 2, 4, 'D'}
setB={'A', 'B', 'C', 'D'}
print(setA - setB)
print(setA.difference(setB))
Output:
{2, 4}
{2, 4}
```

(iv) Symmetric difference :

- It includes all the elements that are in two sets (say sets A and B) but not the one that are common to two sets.
- The caret (^) operator is used to symmetric difference set operation in python.
- The function symmetric_difference() is also used to do the same operation.

Example:

```
setA={'A', 2, 4, 'D'}
setB={'A', 'B', 'C', 'D'}
print(setA ^ setB)
print(setA.symmetric_difference(setB))
```

Output:

```
{2, 4, 'B', 'C'}
{2, 4, 'B', 'C'}
```

EXTRA QUESTION ANSWER:**1. How will you create a list in Python?**

- In python, a list is simply created by using square bracket.
- The elements of list should be specified within square brackets.

- Syntax: *Variable = [element-1, element-2, element-3 element-n]*

➤ Example

```
Marks = [10, 23, 41, 75]
Fruits = ["Apple", "Orange", "Mango", "Banana"]
MyList = [ ]
Mylist = [ "Welcome", 3.14, 10, [2, 4, 6] ] - Nested List
```

2. What is nested list? Give an example.

- Nested list is a list containing another list as an element.
- Example:
Mylist = ["Welcome", 3.14, 10, [2, 4, 6]]

3. How will you access the list elements?

- Python assigns an automatic index value for each element of a list begins with zero.
- Index value can be used to access an element in a list.
- In python, index value is an integer number which can be positive or negative.
- **Example**
Marks = [10, 23, 41, 75]

| | | | | |
|------------------|----|----|----|----|
| Marks | 10 | 23 | 41 | 75 |
| Index (Positive) | 0 | 1 | 2 | 3 |
| Index (Negative) | -4 | -3 | -2 | -1 |

```
>>> Marks = [10, 23, 41, 75]
```

```
>>> print (Marks[0])
```

```
10
```

4. How will you access all the elements of a list? Give Example.

- Loops are used to access all elements from a list.
- The initial value of the loop must be zero. Zero is the beginning index value of a list.
- **Example**

```
Marks = [10, 23, 41, 75]
```

```
i = 0
```

```
while i < 4:
```

```
    print (Marks[i])
```

```
    i = i + 1
```

Output

```
10
```

```
23
```

```
41
```

```
75
```

| Iteration | i | while i < 4 | print (Marks[i]) | i = i + 1 |
|-----------|---|-------------|------------------|-----------|
| 1 | 0 | 0 < 4 True | Marks [0] = 10 | 0 + 1 = 1 |
| 2 | 1 | 1 < 4 True | Marks [1] = 23 | 1 + 1 = 2 |
| 3 | 2 | 2 < 4 True | Marks [2] = 41 | 2 + 1 = 3 |
| 4 | 3 | 3 < 4 True | Marks [3] = 75 | 3 + 1 = 4 |
| 5 | 4 | 4 < 4 False | -- | -- |

5. Write short notes on len() function.

- The len() function is used to find the length of a list. (i.e., the number of elements in a list).
- The len() function is used to set the upper limit in a loop to read all the elements of a list.
- If a list contains another list as an element, len() returns that inner list as a single element.
- **Example :Accessing single element**

```
>>> MySubject = ["Tamil", "English", "Comp. Science", "Maths"]
>>> len(MySubject)
4
```

6. How will you access elements from list using for loop?

- In Python, the **for** loop is used to access all the elements in a list one by one.

Syntax:

```
for index_var in list:
    print(index_var)
```

- Here, **index_var** represents the index value of each element in the list.

Example :

```
Marks=[23, 45, 67, 78, 98]
```

```
for x in Marks:
```

```
    print(x)
```

Output

```
23
45
67
78
98
```

7. How to change the list elements? Write the syntax for changing the list elements. Give an example.

- The lists are mutable, which means they can be changed.
- A list element or range of elements can be changed or altered by using simple assignment operator (=).
- **Syntax:**

List_Variable [index of an element] = Value to be changed

List_Variable [index from : index to] = Values to be changed

- Where, **index from** is the beginning index of the range;
- **index to** is the upper limit of the range which is excluded in the range.
- For example, if you set the range [0:5] means, Python takes only 0 to 4 as element index.

```
Example 1:  MyList = [2, 4, 5, 8, 10]
            MyList[2] = 6          - changing the 2nd index value as 6
            After changing list becomes
            MyList = [2, 4, 6, 8, 10]
```

```
Example 2:  MyList = [2, 4, 5, 8, 10]
            To update the range of elements from 0 to 4
            MyList[0:5] = 2,4,6,8,10
            After changing list becomes
            Mylist = [2, 4, 6, 8, 10]
```

8. Explain the different methods to delete an element from a list?

- There are two ways to delete an element from a list viz. del statement and remove() function.
- del statement is used to delete known elements whereas
- remove() function is used to delete elements of a list if its index is unknown.

- The del statement can also be used to delete entire list.

- **Syntax:**

```
del List [index of an element]    # to delete a particular element
del List [index from : index to]  # to delete multiple elements
del List                          # to delete entire list
```

- **Example1:**

```
>>> MySubjects = ['Tamil', 'Hindi', 'Telugu', 'Maths']
>>> print(MySubjects)
['Tamil', 'Hindi', 'Telugu', 'Maths']
>>> del MySubjects[1]
>>> print(MySubjects)
['Tamil', 'Telugu', 'Maths']
```

- **Example2:**

- The upper limit of index is specified within square brackets, will be taken as -1 by the python.

```
>>> del MySubjects[1:3]
>>> print(MySubjects)
['Tamil']
```

- **Example3:**

- The del statement deletes entire list.
- ```
>>>del MySubjects
```

- **remove()** function can also be used to delete one or more elements if the index value is not known.

- **Syntax:**

```
List.remove(element) # to delete a particular element
>>> MyList=[12,89,34,'Kannan', 'Gowrisankar', 'Lenin']
>>> MyList.remove(89)
>>> print(MyList)
[12, 34, 'Kannan', 'Gowrisankar', 'Lenin']
```

- pop() function can also be used to delete an element using the given index value.
- pop() function deletes and returns the last element of a list if the index is not given.

- **Syntax:**

*List.pop(index of an element)*

- **Example for pop() function:**

```
>>> MyList=[12,34,'Kannan', 'Gowrisankar', 'Lenin']
>>> MyList.pop(1)
34
>>> print(MyList)
[12, 'Kannan', 'Gowrisankar', 'Lenin']
```

- The function **clear()** is used to delete all the elements in list, it deletes only the elements and retains the list.

- **Syntax**

*List.clear()*

- **Example for clear() function:**

```
>>> MyList=[12,34,'Kannan', 'Gowrisankar', 'Lenin']
>>> MyList.clear()
>>> print(MyList)
[]
```

### 9. How will you create a list with series of values?

- The range() function, you can create a list with series of values.
- To convert the result of range() function into list, we need one more function called list().
- The list() function makes the result of range() as a list.
- **Syntax:**
- **List\_Varibale = list ( range ( ) )**
- **Example :**  

```
>>> Even_List = list(range(2,11,2))
>>> print(Even_List)
[2, 4, 6, 8, 10]
```

### 10. What is list comprehension?

- List comprehension is a simplest way of creating sequence of elements that satisfy a certain condition.
- **Syntax:**  
List = [ expression for variable in range ]
- **Example :**  

```
>>> squares = [x ** 2 for x in range(1,11)]
>>> print (squares)
Output:
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

### 11. Tabulate Other important list functions.

| Function                                                                                                                                                                                                                     | Description                                                 | Syntax                                    | Example                                                                                                                                                                                                  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| copy ( )                                                                                                                                                                                                                     | Returns a copy of the list                                  | List.copy ( )                             | <pre>MyList=[12, 12, 36] x = MyList.copy() print(x) <b>Output:</b> [12, 12, 36]</pre>                                                                                                                    |
| count ( )                                                                                                                                                                                                                    | Returns the number of similar elements present in the last. | List.count(value)                         | <pre>MyList=[36 ,12 ,12] x = MyList.count(12) print(x) <b>Output:</b> 2</pre>                                                                                                                            |
| index ( )                                                                                                                                                                                                                    | Returns the index value of the first recurring element      | List.index(element)                       | <pre>MyList=[36 ,12 ,12] x = MyList.index(12) print(x) <b>Output:</b> 0</pre>                                                                                                                            |
| reverse ( )                                                                                                                                                                                                                  | Reverses the order of the element in the list.              | List.reverse ( )                          | <pre>MyList=[36 ,23 ,12] MyList.reverse() print(MyList) <b>Output:</b> [12 ,23 ,36]</pre>                                                                                                                |
| sort ( )                                                                                                                                                                                                                     | Sorts the element in list                                   | List.sort(reverse=True False, key=myFunc) |                                                                                                                                                                                                          |
| Both arguments are optional<br>If reverse is set as True, list sorting is in descending order.<br>Ascending is default.<br>Key=myFunc; “myFunc” - the name of the user defined function that specifies the sorting criteria. |                                                             |                                           | <pre>MyList=['Thiru', 'Tharani', 'Anitha', 'Sai', 'Lavanya'] MyList.sort ( ) print(MyList) MyList.sort(reverse=True) print(MyList) <b>Output:</b> ['Anitha', 'Lavanya', 'Sai', 'Tharani', 'Thiru']</pre> |

|                                                    |                                      |           |                                                                       |
|----------------------------------------------------|--------------------------------------|-----------|-----------------------------------------------------------------------|
| <b>Note:</b> sort() will affect the original list. |                                      |           | ['Thiru', 'Tharani', 'Sai', 'Lavanya', 'Anitha']                      |
| max ( )                                            | Returns the maximum value in a list. | max(list) | <pre>MyList=[21,76,98,23] print(max(MyList)) <b>Output:</b> 98</pre>  |
| min ( )                                            | Returns the minimum value in a list. | min(list) | <pre>MyList=[21,76,98,23] print(min(MyList)) <b>Output:</b> 21</pre>  |
| sum ( )                                            | Returns the sum of values in a list. | sum(list) | <pre>MyList=[21,76,98,23] print(sum(MyList)) <b>Output:</b> 218</pre> |

### 12. What is a Tuple? How will you create a tuple?

- Tuples consists of a number of values separated by comma and enclosed within parentheses.
- Values in a tuple cannot be changed.
- **Creating Tuple:**
- Creating tuples is similar to list. The elements of a tuple can be even defined without parenthesis.
- The elements defined within parenthesis or without parenthesis, there is no difference in it's function.
- **Syntax:**  

```
Empty tuple Tuple_Name = ()
Tuple with n number elements Tuple_Name = (E1, E2, E2 En)
Elements of a tuple without parenthesis Tuple_Name = E1, E2, E3 En
```

**Example :**

```
>>> MyTup1 = (23, 56, 89, 'A', 'E', 'I', "Tamil")
>>> print(MyTup1)
(23, 56, 89, 'A', 'E', 'I', 'Tamil')
>>> MyTup2 = 23, 56, 89, 'A', 'E', 'I', "Tamil"
>>> print (MyTup2)
(23, 56, 89, 'A', 'E', 'I', 'Tamil')
```

### 13. How to Create tuples using tuple() function

- The tuple() function is used to create Tuples from a list.
- When you create a tuple, from a list, the elements should be enclosed within square brackets.
- **Syntax:**  

```
Tuple_Name = tuple([list elements])
```

**Example:**

```
>>> MyTup3 = tuple([23, 45, 90])
>>> print(MyTup3)
(23, 45, 90)
```

### 14. How to Create Single element tuple?

- While creating a tuple with a single element, add a comma at the end of the element.
- In the absence of a comma, Python will consider the element as an ordinary data type; not a tuple.
- Creating a Tuple with one element is called “Singleton” tuple.
- **Example:**  

```
>>> MyTup5 = (10,)
>>> type(MyTup5)
<class 'tuple'>
```

### 15. How to Accessing values in a Tuple?

- Each element of tuple has an index number starting from zero.
- The elements of a tuple can be easily accessed by using index number.

#### **Example:**

```
>>> Tup1 = (12, 78, 91, "Tamil", "Telugu", 3.14, 69.48)
to access all the elements of a tuple
>>> print(Tup1)
(12, 78, 91, 'Tamil', 'Telugu', 3.14, 69.48)
#accessing selected elements using indices
>>> print(Tup1[2:5])
(91, 'Tamil', 'Telugu')
#accessing from the first element up to the specified index value
>>> print(Tup1[:5])
(12, 78, 91, 'Tamil', 'Telugu')
accessing from the specified element up to the last element.
>>> print(Tup1[4:])
('Telugu', 3.14, 69.48)
accessing from the first element to the last element
>>> print(Tup1[:])
(12, 78, 91, 'Tamil', 'Telugu', 3.14, 69.48)
```

### 16. How will you update and delete a tuple?

- Tuple is immutable, the elements in a tuple cannot be changed.
- Instead of altering values in a tuple, joining two tuples or deleting the entire tuple is possible.

#### **Example:**

```
Tup1 = (2,4,6,8,10)
Tup2 = (1,3,5,7,9)
Tup3 = Tup1 + Tup2
print(Tup3)
Output
(2, 4, 6, 8, 10, 1, 3, 5, 7, 9)
```

- To delete an entire tuple, the **del** command can be used.

#### **Syntax:**

```
del tuple_name
```

#### **Example:**

```
Tup1 = (2,4,6,8,10)
print("The elements of Tup1 is ", Tup1)
del Tup1
```

### 17. What is a set? How will you create a set?

- A set is another type of collection data type.
- A Set is a mutable and an unordered collection of elements without duplicates.
- That means the elements within a set cannot be repeated.
- A set is created by placing all the elements separated by comma within a pair of curly brackets.
- The set( ) function can also be used to create sets in Python.

#### • **Syntax:**

```
Set_Variable = {E1, E2, E3 En}
```

#### • **Example**

```
>>> S1={1,2,3,'A',3.14}
>>> print(S1)
{1, 2, 3, 3.14, 'A'}
```

#### • **Creating Set using List or Tuple**

- A list or Tuple can be converted as set by using set( ) function.
  - This is very simple procedure.
  - First you have to create a list or Tuple then, substitute its variable within set( ) function as argument.
- ```
MyList=[2,4,6,8,10]
MySet=set(MyList)
print(MySet)
```

Output:

```
{2, 4, 6, 8, 10}
```

18. Write short note on Tuple Assignment?

- Tuple assignment is a powerful feature in Python.
- It allows a tuple variable on the left of the assignment operator to be assigned to the values on the right side of the assignment operator.
- Each value is assigned to its respective variable
- Example:

```
>>> (a, b, c) = (34, 90, 76)
>>> print(a,b,c)
34 90 76
```

19. What is a Dictionary? How to create it?

- Dictionary is a mixed collection of elements.
- The dictionary type stores a key along with its element.
- The keys in a Python dictionary is separated by a colon (:) while the commas work as a separator for the elements.
- The key value pairs are enclosed with curly braces { }.
- **Syntax of defining a dictionary:**

```
Dictionary_Name = { Key_1: Value_1,Key_2:Value_2,.....,Key_n: Value_n}
```
- **Key in the dictionary must be unique case sensitive and can be of any valid Python type.**
- **Creating a Dictionary :**

```
# Empty dictionary
Dict1 = { }
# Dictionary with Key
Dict_Stud = { RollNo: 1234, Name:Murali, Class:XII, Marks:451}
```

20. What is Dictionary Comprehension?

- In Python, comprehension is another way of creating dictionary.
- Syntax of creating dictionary.
- Syntax

```
Dict = { expression for variable in sequence [if condition] }
```
- The if condition is optional and if specified, only those values in the sequence are evaluated using the expression which satisfy the condition.

- **Example :**
Dict = { x : 2 * x for x in range(1,10)}
- **Output:**
{1: 2, 2: 4, 3: 6, 4: 8, 5: 10, 6: 12, 7: 14, 8: 16, 9: 18}

21. How will you access elements in a dictionary?

- Accessing all elements from a dictionary is very similar as Lists and Tuples.
- Simple print function is used to access all the elements.
- If you want to access a particular element, square brackets can be used along with key.
- Example:

```
MyDict = { 'Reg No': '1221', 'Name': 'Tamilselvi', 'School': 'CGHSS',
           'Address': 'Rotler St., Chennai 112' }

print(MyDict)
print("Register Number: ", MyDict['Reg_No'])
print("Name of the Student: ", MyDict['Name'])
print("School: ", MyDict['School'])
print("Address: ", MyDict['Address'])
```

Output:

```
{'Reg_No': '1221', 'Name': 'Tamilselvi', 'School': 'CGHSS', 'Address': 'Rotler St., Chennai 112'}
Register Number: 1221
Name of the Student: Tamilselvi
School: CGHSS
Address: Rotler St., Chennai 112
```

Choose the correct Answer (1 Mark)

- Pick odd one in connection with collection data type
(a) List (b) Tuple (c) Dictionary **(d) Loop**
- Let list1=[2,4,6,8,10], then print(List1[-2]) will result in
(a) 10 **(b) 8** (c) 4 (d) 6
- Which of the following function is used to count the number of elements in a list?
(a) count() (b) find() **(c) len()** (d) index()
- If List=[10,20,30,40,50] then List[2]=35 will result
(a) [35,10,20,30,40,50] (b) [10,20,30,40,50,35]
(c) [10,20,35,40,50] (d) [10,35,30,40,50]
- How many collection of data types in python?
a) 3 **b) 4** c) 5 d) 2
- Which of the following is not data type in python?
a) List b) Tuples **c) Variable** d) Set
- In list, the negative index number begin with
a) 0 b) 1 **c) -1** d) -0.1
- Which function is used to set the upper limit in a loop to read all elements of a list?
a) upper() b) limit **c) len()** d) loop
- Which of the following operator can be used to alter the range of elements in the list?
a) == b) : c) :: **d) =**
- a = ['A', 2,3,[4,5,6]], the number elements in a list
a) 4 b) 6 c) 5 d) 3
- Which of the following Python function can be used to add more than one element within an existing list?

- (a) append() (b) append_more() **(c) extend()** (d) more()

12. What will be the result of the following Python code?

```
S=[x**2 for x in range(5)]
```

```
print(S)
```

- (a) [0,1,2,4,5] **(b) [0,1,4,9,16]** (c) [0,1,4,9,16,25] (d) [1,4,9,16,25]

13. How many ways to delete the elements from a list?

- a) 2** b) 3 c) 4 d) 5

14. Which of the following statement is used to delete an element from the list?

- a) remove **b) del** c) delete d) case

15. Which of the following is used to delete known elements where the index value is known?

- a) del b) delete **c) remove()** d) del()

16. How many arguments used in the range() function?

- a) 3** b) 4 c) 2 d) 1

17. Which of the following function is used to convert the result of range() function in to list?

- a) convert() b) range() **c) list()** d) listrange()

18. What is the output for the following?

```
A = list(range(2,5,2))
```

```
print(A)
```

- a)[2,4]** b)[2,3,4] c)[2,3,4,5] d)[2,3]

19. Which of the following function used to include an element in a list at a desired position?

- a) append() b) extend() **c) insert()** d) format()

20. Which function is not used to delete the element in a list?

- a) del **b) delete** c) remove d) pop

21. What is the use of type() function in python?

- (a) To create a Tuple (b) To know the type of an element in tuple.

(c) To know the data type of python object.

- (d) To create a list.

22. Which of the following statement is not correct?

- (a) A list is mutable (b) A tuple is immutable.

(c) The append() function is used to add an element.

(d) The extend() function is used in tuple to add elements in a list.

23. Which of the following function returns the number of similar elements present in the list?

- a. elements() **b. count()** c. return() d. find()

24. Using sort(), the default sorting is

- a. ascending** b. descending c. no sorting d. criteria sorting

25. What is the output for the following?

```
mylist = 36,12,12]
```

```
x = mylist.index(12)
```

```
print(x)
```

- a.0 **b.1** c.2 d.1.5

26. Which of the following can be defined with or without parenthesis() ?

- a. list b. set c. dictionary **d. tuple**

27. While creating a tuple from the list, the element should be enclosed with in

- a. []** b. () c. { } d. [()]

28. Which command is used to delete entire tuple?

- a. delete b. remove c. rem **d. del**

29. Creating a tuple with one element is called as
a. int **b. singleton** c. oneton d. one element tuple
30. Which function is used to return minimum value in a list?
a. minimum() **b. min()** c. lowest() d. minvalue()
31. Let setA={3,6,9}, setB={1,3,9}. What will be the result of the following snippet?
print(setA|setB)
(a) {3,6,9,1,3,9} (b) {3,9} (c) {1} **(d) {1,3,6,9}**
32. Which of the following set operation includes all the elements that are in two sets but not the one that are common to two sets?
(a) Symmetric difference (b) Difference (c) Intersection (d) Union
33. What is the output for the following?
(a,b) = (34)
a. a=34 b=34 a=34, b=34 Error
34. Which of the following is an unordered collection of elements without duplicates?
a. List b. Tuple **c. Set** d. All of these
35. A set is created using
a. [] **b. {}** c. () d. [[]]
36. What is output of the following? a = { 1, 2, 2, 3}
a. { 1, 2 } b. { 1, 2, 3, 4 } **c. { 1, 2, 3 }** d. { 1, 3 }
37. Which of the following operator is used to join two sets in python?
a. & b. ^ **c. |** d. %
38. A = B | C is equivalent to
a. A = B.set(C) b. A = B.join(C) **c. A = B.union(C)** d. A = C. set(B)
39. Which of the following is a powerful feature in python
a. Tuple assignment b. List assignment c. Assignment statement d. set operations
40. Which of the following is faster than list?
a. Set **b. Tuples** c. Dictionary d. Pairs
41. The keys in Python, dictionary is specified by
a. = b. ; c. + **d. :**
42. Which of the following can be used to access a particular element in a dictionary?
a. { } **b. []** c. () d. <>
43. Choose the incorrect pair from the following
a. List = [] **b. Tuple = { }** c. set = { } d. Dictionary = { }
44. Which command is used to delete a element from dictionary?
a. delete b. remove **c. del** d. clear()
45. Which command is used to delete entire element from dictionary?
a. delete b. remove c. del **d. clear()**
46. Which one of the following is a mixed collection of elements?
a. List b. Tuples c. Set **d. Dictionary**
47. Which operator work as a separator for the elements in dictionary?
a. comma(,) b. semicolon (;) c. colon (:) d. dot (.)
48. Accessing all elements from a dictionary is very similar as
a. List b. Tuples c. Set **d. Lists and Tuples**
49. Which function is used to access all the elements from a dictionary?
a. list() b. set() **c. print()** d. input()

50. Which command is used to delete dictionary?
a. del b. delete c. remove d. clear()

CHAPTER 10 - PYTHON CLASSES AND OBJECTS

PART - II

1. What is a class in Python?

- Class is the main building block in Python.
- Class is a template for the object.

2. Define class instantiation.

- The process of creating object is called as "Class Instantiation".
- Syntax:
`Object_name = class_name()`
- Example:
`S = Sample()`

3. What is the output of the following program?

```
class Sample:
    __num=10
    def disp(self):
        print(self.__num)

S=Sample()
S.disp()
print(S.__num)
```

```
output :
10
Traceback (most recent call last):
File "D:/Python/Class-Test-04.py", line 7, in <module>
print(S.__num)
AttributeError: 'Sample' object has no attribute '__num'
```

4. What is a Constructor? Give example. (Or) How will you create constructor in Python?

- Constructor is the special function that is automatically executed when an object of a class is created.
- In Python, there is a special function called "init" which act as a Constructor.
- It must begin and end with **double underscore**.
- This constructor function can be defined **with or without arguments**.
- This method is used to **initialize the class variables**.

General format :

```
def __init__(self, [args .....]):
    <statements>
```

Example :

```
class Sample:
    def __init__(self, num):
        print("Constructor of class Sample...")
        self.num=num
        print("The value is :", num)

S=Sample(10)
```

OUTPUT:

Constructor of class Sample ...

The value is: 10

5. What is a Destructor? Give example. (or) What is the purpose of Destructor?

- Destructor is also a special method gets executed automatically when an object exit from the scope.
- It is just opposite to constructor.
- It is used to remove the memory of an object.
- In Python, `__del__()` method is used as destructor.

Example :

```
class Sample:
    def __del__(self):
        print("Destructor of class Sample")

S=Sample()
```

Output:

Destructor of class Sample

PART-III

1. What are class members? How do you define it?

- Class variable and methods are together known as members of the class.
- Variables defined inside a class are called as "Class Variable"
- Functions defined inside a class are called as "Methods".
- The class members should be accessed through objects or instance of class.
- Example

```
class Student:
    mark1, mark2 = 50, 90 #class variable
    def process(self):      # class method
        sum = Student.mark1 + Student.mark2
        avg = sum/2
        print("Total Marks = ", sum)
        print("Average Marks = ", avg)
        return
```

```
S=Student()
S.process()
```

2. Write a class with two private class variables and print the sum using a method.

```
class Sample:
    __n1=10
    __n2=20
    def disp(self):
        print(Sample.__n1 + Sample.__n2)
```

```
S=Sample()
```

```
S.disp()
```

Output:

30

3. Find the error in the following program to get the given output?

```
class Fruits:
    def __init__(self, f1, f2):

        self.f1=f1
```

```
        self.f2=f2
    def display(self):
        print("Fruit 1 = %s, Fruit 2 = %s" %(self.f1, self.f2))
F = Fruits ('Apple', 'Mango')
del F.display
F.display()
Output
Fruit 1 = Apple, Fruit 2 = Mango
```

Error : Remove the line `del F.display()`

4. What is the output of the following program?

```
class Greeting:
    def __init__(self, name):
        self.__name = name
    def display(self):
        print("Good Morning ", self.__name)
obj=Greeting('Bindu Madhavan')
obj.display()
```

Output:

Good Morning Bindu Madhavan

5. How do you define constructor and destructor in Python?

- Constructor is the special function that is automatically executed when an object of a class is created.
- In Python, there is a special function called "**init**" which act as a Constructor.
- It must begin and end with **double underscore**.
- This constructor function can be defined **with or without arguments**.
- This method is used **to initialize the class variables**.

General format :

```
def __init__(self, [args .....]):
    <statements>
```

Example :

```
class Sample:
    def __init__(self, num):
        print("Constructor of class Sample...")
        self.num=num
        print("The value is :", num)

S=Sample(10)
```

OUTPUT:

Constructor of class Sample ...

The value is: 10

Destructor:

- Destructor is also a special method gets executed automatically when an object exit from the scope.
- It is used to remove the memory of an object.
- In Python, `__del__()` method is used as destructor.

Example :

```
class Sample:
    def __init__(self):
        print("Constructor of class Sample");
```

```

        def __del__(self):
            print("Destructor of class Sample")

S=Sample()

```

OUTPUT:

Constructor of class Sample
Destructor of class Sample

PART-IV

1. Write a menu driven program to add or delete stationary items. You should use dictionary to store items and the brand.

CODE:

```

stationary={}
print("\n1. Add Item \n2.Delete item \n3.Exit")
ch=int(input("\nEnter your choice: "))
while(ch==1)or(ch==2):
    if(ch==1):
        n=int(input("\nEnter the Number of Items to be added in the Dictionary: "))
        for i in range(n):
            item=input("\nEnter an Item Name: ")
            brand=input("\nEnter the Brand Name: ")
            stationary[item]=brand
        print(stationary)

    elif(ch==2):
        ritem=input("\nEnter the item to be removed from the Dictionary: ")
        stationary.pop(ritem)
        print(stationary)
ch=int(input("\nEnter your choice: "))

```

OUTPUT

```

>>>
===== RESTART: C:/Users/SANJANASRI.SANJANASRI-PC/Desktop/Python/menu.py =====
1. Add Item
2.Delete item
3.Exit

Enter your choice: 1

Enter the Number of Items to be added in the Dictionary: 2

Enter an Item Name: Pen

Enter the Brand Name: Rorito

Enter an Item Name: Pencil

Enter the Brand Name: Camlin
{'Pen': 'Rorito', 'Pencil': 'Camlin'}

Enter your choice: 2

Enter the item to be removed from the Dictionary: Pen
{'Pencil': 'Camlin'}

Enter your choice: 3
>>> |

```

EXTRA QUESTION ANSWER:

1. What is an object in Python?

- Object is a collection of data and function that act on those data.
- Objects are also called as instances of a class or class variable.
- In Python, everything is an object.

2. How will you define class?

- In Python, a class is defined by using the keyword **class**.
- Every class has a unique name followed by a colon (:).

• Syntax:

```

class class_name:
    statement_1
    statement_2
    .....
    .....
    statement_n

```

- **Example:** Program to define a class

```

class Sample:
    x, y = 10, 20 # class variables

```

3. How will you create an object?

- Once a class is created, next you should create an object or instance of that class.
- The process of creating object is called as “Class Instantiation”.

• Syntax:

```

Object_name = class_name( )

```

- Example:

```

S = Sample()

```

4. How will you access the class members?

- Any class member ie. class variable or method (function) can be accessed by using object with a dot (.) operator.

• Syntax:

```

Object_name . class_member

```

- Example :

```

class Sample:
    x, y = 10, 20
S=Sample( ) # class instantiation
print("Value of x = ", S.x)
print("Value of y = ", S.y)

```

- Output :

```

Value of x = 10
Value of y = 20

```

5. What are class methods?

- Functions defined inside a class are called as “Methods”.
- Class Method is similar to ordinary function with a small difference that, the class method must have the first argument named as **self**.
- No need to pass a value for this argument when we call the method.
- Python provides its value automatically.
- Even if a method takes no arguments, it should be defined with the first argument called **self**.

- If a method is defined to accept only one argument it will take it as two arguments ie. **self and the defined argument.**

- **Example:**

```
class Student:
    mark1, mark2 = 50, 90 #class variable
    def process(self):      # class method
        sum = Student.mark1 + Student.mark2
        avg = sum/2
        print("Total Marks = ", sum)
        print("Average Marks = ", avg)
        return
S=Student()
S.process()
```

6. Write short notes on Public and Private Data Members.

- The variables which are defined inside the class is public by default.
- These variables can be accessed anywhere in the program using dot operator.
- A variable prefixed with double underscore becomes private in nature.
- These variables can be accessed only within the class.

- Example :

```
class Sample:
    def __init__(self, n1, n2):
        self.n1=n1      # public member
        self.__n2=n2    # private member
    def display(self):
        print("Class variable 1 = ", self.n1)
        print("Class variable 2 = ", self.__n2)
S=Sample(12, 14)
S.display()
```

- **Output**

Class variable 1 = 12
Class variable 2 = 14
Value 1 = 12

7. Write a program to calculate area and circumference of a circle

```
class Circle:
    pi=3.14
    def __init__(self,radius):
        self.radius=radius
    def area(self):
        return Circle.pi*(self.radius**2)
    def circumference(self):
        return 2*Circle.pi*self.radius
r=int(input("Enter Radius: "))
C=Circle(r)
print("The Area =",C.area())
print("The Circumference =", C.circumference())
```

Output:

Enter Radius: 5

The Area = 78.5

The Circumference = 31.400000000000002

8. Write a program to accept a string and print the number of uppercase, lowercase, vowels, consonants and spaces in the given string

```
class String:
    def __init__(self):
        self.uppercase=0
        self.lowercase=0
        self.vowels=0
        self.consonants=0
        self.spaces=0
        self.string=""
    def getstr(self):
        self.string=str(input("Enter a String: "))
    def count_upper(self):
        for ch in self.string:
            if (ch.isupper()):
                self.uppercase+=1
    def count_lower(self):
        for ch in self.string:
            if (ch.islower()):
                self.lowercase+=1
    def count_vowels(self):
        for ch in self.string:
            if (ch in ('A', 'a', 'e', 'E', 'i', 'I', 'o', 'O', 'l', 'L')):
                self.vowels+=1
    def count_consonants(self):
        for ch in self.string:
            if (ch not in ('A', 'a', 'e', 'E', 'i', 'I', 'o', 'O', 'l', 'L')):
                self.consonants+=1
    def count_space(self):
        for ch in self.string:
            if (ch==" "):
                self.spaces+=1
    def execute(self):
        self.count_upper()
        self.count_lower()
        self.count_vowels()
        self.count_consonants()
        self.count_space()
    def display(self):
        print("The given string contains...")
        print("%d Uppercase letters"%self.uppercase)
        print("%d Lowercase letters"%self.lowercase)
        print("%d Vowels"%self.vowels)
        print("%d Consonants"%self.consonants)
        print("%d Spaces"%self.spaces)
S = String()
S.getstr()
```

```
S.execute()
S.display()
```

Output :

```
Enter a String: Welcome
The given string contains...
1 Uppercase letters
6 Lowercase letters
3 Vowels
4 Consonants
0 Spaces
```

9. What is the output of the following program?

```
class Sample:
    __num=10
    def disp(self):
        print(self.__num)

S=Sample()
S.disp()
print(S.__num)
```

```
output :
10
Traceback (most recent call last):
File "D:/Python/Class-Test-04.py", line 7, in <module>
print(S.__num)
AttributeError: 'Sample' object has no attribute '__num'
```

Choose the correct Answer:

- Which of the following are the key features of an Object Oriented Programming language?
(a) Constructor and Classes (b) Constructor and Object
(c) Classes and Objects (d) Constructor and Destructor
- Functions defined inside a class:
(a) Functions (b) Module **(c) Methods** (d) Section
- Class members are accessed through which operator?
(a) & **(b) .** (c) # (d) %
- Which of the following method is automatically executed when an object is created?
(a) __object__() (b) __del__() (c) __func__() **(d) __init__()**
- A private class variable is prefixed with
(a) __ (b) && (c) ## (d) **
- Which of the following method is used as destructor?
(a) __init__() (b) __dest__() (c) __rem__() **(d) __del__()**
- Which of the following class declaration is correct?
(a) class class_name (b) class class_name<>
(c) class class_name: (d) class class_name[]
- Which of the following is the output of the following program?
class Student:
 def __init__(self, name):
 self.name=name
S=Student("Tamil")
(a) Error or (No output) (b) Tamil (c) name (d) self
- Which of the following is the output of the following program?
class Student:

```
def __init__(self, name):
    self.name=name
    print(name)
```

S=Student("Tamil")

- (a) Error **(b) Tamil** (c) name (d) self
10. Which of the following is the private class variable?
(a) __num (b) ##num (c) \$\$num (d) &&num
11. The process of creating an object is called as:
(a) Constructor (b) Destructor (c) Initialize **(d) Instantiation**
12. Which of the following is not an object oriented language?
a. C b. C++ c. Java d. Python
13. Which argument doesn't need a value when we call the method?
a. this **b. self** c. var d. first
14. Write the output for the following
class test
 x,y = 10,5
s = test()
print(s.x+s.y)
a.10 b. 5 **c. 15** d.105
15. In python constructor must begin and end with
a. double underscore b. double plus c. double hyphen d. double minus
16. Which position of the argument named self in python class method?
a. last **b. first** c. second d. third
17. Which of the following can be accessed by using object with dot(.) operator?
a. list b. tuples c. dictionary **d. none of these**
18. By default the class variables are
a. private **b. public** c. protected d. method
19. class is a template for the
a. method b. members **c. object** d. constructor
20. Which one of the following may be a variable declaration, decision control, loop or even a function definition
a. class members b. class instantiation c. class method **d. class definition**
21. Which of the following is an incorrect statement?
a. Constructor executed automatically when the object is created
b. In Python, 'init' act as a constructor
c. Constructor is used to initialize the class variables
d. In python, constructor can be defined only with arguments

CHAPTER – 11 DATABASE CONCEPTS

Choose the correct Answer:

- What is the acronym of DBMS?
a) Database Management Symbol b) Database Managing System
c) Database Management System d) Databasic Management System
- A table is known as _____.
a) tuple b) attribute **c) relation** d) entity
- Which database model represents parent-child relationship?
a) Relational b) Network **c) Hierarchical** d) Object
- Relational database model was first proposed by
a) E F Codd b) E E Codd c) E F Cadd d) E F Codder

5. What type of relationship does hierarchical model represents?
a) one-to-one **b) one-to-many** c) many-to-one d) many-to-many
6. Who is called Father of Relational Database from the following?
a) Chris Date b) Hugh Darween **c) Edgar Frank Codd** d) Edgar Frank Cadd
7. Which of the following is an RDBMS?
a) Dbase b) Foxpro
c) Microsof Access **d) SQLite**
8. What symbol is used for SELECT statement?
a) σ b) Π
c) X d) Ω
9. A tuple is also known as
a) table **b) row**
c) attribute d) field
10. Who developed ER model?
a) Chen b) EF Codd
c) Chend d) Chand
11. Which of the following is an organized collection of data?
a. Word Processor b. Spreadsheet
c. Programming Language **d. Database**
12. Which of the following characteristics of DBMS becomes challenge?
a. Data redundancy b. Data security
c. Data Consistency d. Data Integrity
13. Which of the following provides protection and security to the databases?
a. MySQL **b. DBMS**
c. Oracle d. CSV
14. How many major components are there in DBMS?
a. 5 b. 4
c. 3 d. 2
15. Each row in a table represents
a. Fields **b. Record**
c. Data d. File
9. Which of the following groups data among records specific categories or types of data?
a. Table b. File
c. Field d. Relation
10. Which of the following is not an example of DBMS?
a. Foxpro b. Dbase
c. COBOL d. MS-Access
12. How many types of relationships used in a database?
a. 3 b. 5
c. 2 **d. 4**
13. Which of the following defines a relation that contains a vertical subset of relation?
a. select b. product

- c. intersection **d. project**
14. Relational Algebra is used to query the database tables using
a. Dbase **b. SQL**
c. ER – Model d. Relational model
15. Which of the following represents Unary relational operations PROJECT
a) σ **b) Π**
c) X d) Ω
16. SQL stands for
a. Structured Question Language b. Selection Query Language
c. Structured Query Language d. Standard Query Language
17. Relational Algebra was first created by
a. E F Codd b. Chen
c. Chris Date d. Hugh Darwen
18. Choose the correct pair from the following
a. Product - \cup b. Intersection - X
c. Project – Π d. Product - σ
19. Which is a way of combining two relations
a. Intersection **b. Product**
c. Union d. Set difference
20. Which of the following relation which includes all tuples that are in A but not in B
a. Intersection b. Product
c. Union **d. Set difference**
21. In which year Relational Database model was first proposed?
a. 1960 b. 1964
c. 1974 **d. 1970**
22. ER model was developed in the year
a. 1960 b. 1964
c. 1976 d. 1970
23. Who manages the complete database management system?
a. End user **b. DBA**
c. DB designer d. Software developer
24. Which of the following is not a RDBMS software?
a. Oracle b. My SQL
c. Maria DB **d. Foxpro**

PART - II

1. Mention few examples of a database.

- Foxpro, dbase.

2. List some examples of RDBMS.

1. SQL server

2. Oracle
3. MySQL
4. MariaDB
5. SQLite

3. What is Data consistency?

Data Consistency means that data values are the same at all instances of a database

4. What is the difference between Hierarchical and Network data model?

- The difference between hierarchical and Network data model is:
 - In hierarchical model, a child record has only one parent node,
 - Hierarchical model represents the data in one to many relationships
 - In a Network model, a child may have many parent nodes.
 - Network model represents the data in many to-many relationships.
 - Network model is easier and faster to access the data

5. What is normalization?

- Normalization is an integral part of RDBMS in order to reduces data redundancy and improves data integrity
- Database Normalization was proposed by Dr. Edger F Codd

PART - C

1.What is the difference between Select and Project command

SELECT (symbol : σ)

- General form $\sigma_c(R)$ with a relation R and a condition C on the attributes of R.
- The SELECT operation is used for selecting a subset with tuples according to a given condition.
- Select filters out all tuples that do not satisfy C.

STUDENT TABLE

| Studno | Name | Course | Year |
|--------|---------------|--------------------|------|
| cs1 | Kannan | Big Data | II |
| cs2 | Gowri Shankar | R language | I |
| cs3 | Lenin | Big Data | I |
| cs4 | Padmaja | Python Programming | I |

$\sigma_{\text{course} = \text{"Big Data"}}(\text{STUDENT})$

| Studno | Name | Course | Year |
|--------|--------|----------|------|
| cs1 | Kannan | Big Data | II |
| cs3 | Lenin | Big Data | I |

PROJECT (symbol : Π)

- The projection eliminates all attributes of the input relation but those mentioned in the projection list.
- The projection method defines a relation that contains a vertical subset of Relation.

Example

$\Pi_{\text{course}}(\text{STUDENT})$

Result

| Course |
|--------------------|
| Big Data |
| R language |
| Python Programming |

- Duplicate row is removed in the result

2. What is the role of DBA?

- Database Administrator or DBA is the one who manages the complete database management system.
- DBA takes care of the security of the DBMS, managing the license keys, managing user accounts and access etc.

3. Explain Cartesian Product with a suitable example?

PRODUCT OR CARTESIAN PRODUCT (Symbol : \times)

- Cross product is a way of combining two relations.
- The resulting relation contains, both relations being combined.
- $A \times B$ means A times B, where the relation A and B have different attributes.
- This type of operation is helpful to merge columns from two relations.

| Table A | | Table B | |
|---------|---------------|---------|------------|
| Studno | Name | studno | subject |
| cs1 | Kannan | cs28 | Big Data |
| cs2 | Gowri Shankar | cs62 | R language |

Cartesian product :Table A x Table B

| studno | Name | Course | subject |
|--------|--------|--------|------------|
| cs1 | Kannan | cs28 | Big Data |
| cs1 | Kannan | cs62 | R language |

| | | | |
|-----|---------------|------|------------|
| cs2 | Gowri Shankar | cs28 | Big Data |
| cs2 | Gowri Shankar | cs62 | R language |

4. Explain Object Model with example.

- Object model stores the data in the form of objects, attributes and methods, classes and Inheritance.
- This model handles more complex applications, such as Geographic information System (GIS), scientific experiments, engineering design and manufacturing.
- It is used in file Management System.
- It represents real world objects, attributes and behaviors. It provides a clear modular structure.
- It is easy to maintain and modify the existing code.

5. Write a note on different types of DBMS users.

1. Database Administrators

- Database Administrator or DBA is the one who manages the complete database management system.
- DBA takes care of the security of the DBMS, managing the license keys, managing user accounts and access etc.

2. Application Programmers or Software Developers

- This user group is involved in developing and designing the parts of DBMS.

3. End User

- All modern applications, web or mobile, store user data.
- Applications are programmed in such a way that they collect user data and store the data on DBMS systems running on their server.
- End users are the one who store, retrieve, update and delete data.

4. Database designers:

- They** are responsible for identifying the data to be stored in the database for choosing appropriate structures to represent and store the data.

PART- D

1.Explain different types of Data Model.

Types of Data Model

- Hierarchical Model
- Relational Model
- Network Database Model
- Entity Relationship Model
- Object Model

1. Hierarchical Model

Hierarchical model was developed by IBM as Information Management System.

- In Hierarchical model, data is represented as a simple tree like structure form.
- This model represents a one-to-many relationship ie parent-child relationship.
- One child can have only one parent but one parent can have many children.
- This model is mainly used in IBM Main Frame computers.

2. Relational Model

- The Relational Database model was first proposed by E.F. Codd in 1970 .
- It is the most widespread data model used for database applications around the world.
- The basic structure of data in relational model is tables (relations).
- All the information's related to a particular type is stored in rows of that table.
- Hence tables are also known as relations in a relational model.
- A relation key is an attribute which uniquely identifies a particular tuple (row in a relation (table)).

3. Network Model

- Network database model is an extended form of hierarchical data model.
- The difference between hierarchical and Network data model is:
 - In hierarchical model, a child record has only one parent node,
 - In a Network model, a child may have many parent nodes.
 - It represents the data in many to-many relationships.
- This model is easier and faster to access the data

4. Entity Relationship Model. (ER model)

- Relationships are created by dividing the object into entity and its characteristics into attributes.
- It was developed by Chen in 1976.
- This model is useful in developing a conceptual design for the database.
- It is very simple and easy to design logical view of data.
- The developer can easily understand the system by looking at ER model constructed.
- Rectangle represents the entities. E.g. Doctor and Patient
- Ellipse represents the attributes E.g. D-id, D-name, P-id, P-name.
- Attributes describes the characteristics and each entity becomes a major part of the data stored in the database.
- Diamond represents the relationship in ER diagrams
E.g. Doctor diagnosis the Patient



5. Object Model

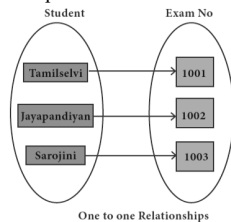
- Object model stores the data in the form of objects, attributes and methods, classes and Inheritance.
- This model handles more complex applications, such as Geographic information System (GIS), scientific experiments, engineering design and manufacturing.
- It is used in file Management System.
- It represents real world objects, attributes and behaviors. It provides a clear modular structure.
- It is easy to maintain and modify the existing code.

2. Explain different Types of Relationships

1. One-to-One Relationship
2. One-to-Many Relationship
3. Many-to-One Relationship
4. Many-to-Many Relationship

1. One-to-One Relationship

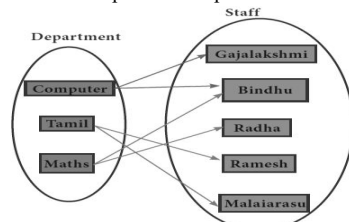
- In One-to-One Relationship, one entity is related with only one other entity.
- One row in a table is linked with only one row in another table and vice versa.
- For example: A student can have only one exam number



One to one Relationships

2. One-to-Many Relationship

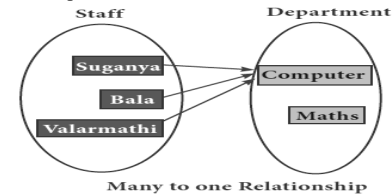
- In One-to-Many relationship, one entity is related to many other entities.
- One row in a table A is linked to many rows in a table B, but one row in a table B is linked to only one row in table
- For example: One Department has many staff members



One to Many Mapping

3. Many-to-One Relationship

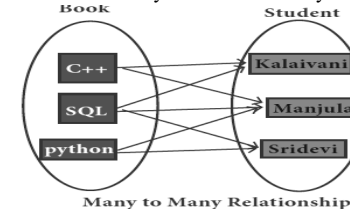
- In Many-to-One Relationship, many entities can be related with only one in the other entity.
- For example: A number of staff members working in one Department.
- Multiple rows in staff members table is related with only one row in Department table.



Many to one Relationship

4. Many-to-Many Relationship

- A many-to-many relationship occurs when multiple records in a table are associated with multiple records in another table.
- **Example : Books and Student.**
Many Books in a Library are issued to many students.



Many to Many Relationship

3. Differentiate DBMS and RDBMS

| Basis of Comparison | DBMS | RDBMS |
|---------------------|---|--|
| Expansion | Database Management System | Relational DataBase Management System |
| Data storage | Navigational model ie data by linked records | Relational model (in tables). ie data in tables as row and column |
| Data redundancy | Exhibit | Not Present |
| Normalization | Not performed | RDBMS uses normalization to reduce redundancy |
| Data access | Consumes more time | Faster, compared to DBMS. |
| Keys and indexes | Does not use | used to establish relationship. Keys are used in RDBMS. |

| | | |
|------------------------|---------------------------------------|--|
| Transaction management | Inefficient, Error prone and insecure | Efficient and secure |
| Distributed Databases | Not supported | Supported by RDBMS |
| Example | Dbase, FoxPro | SQL server, Oracle, mysql, MariaDB, SQLite |

4. Explain the different operators in Relational algebra with suitable examples

Relational Algebra is divided into various groups

1. Unary Relational Operations

- SELECT (symbol : σ)
- PROJECT (symbol : π)

2. Relational Algebra Operations from Set Theory

- UNION (\cup)
- INTERSECTION (\cap)
- DIFFERENCE ($-$)
- CARTESIAN PRODUCT (\times)

SELECT (symbol : σ)

- General form $\sigma_c(R)$ with a relation R and a condition C on the attributes of R.
- The SELECT operation is used for selecting a subset with tuples according to a given condition.
- Select filters out all tuples that do not satisfy C.

STUDENT TABLE

| Studno | Name | Course | Year |
|--------|---------------|--------------------|------|
| cs1 | Kannan | Big Data | II |
| cs2 | Gowri Shankar | R language | I |
| cs3 | Lenin | Big Data | I |
| cs4 | Padmaja | Python Programming | I |

$\sigma_{\text{course}} = \text{"Big Data"}(\text{STUDENT})$

| Studno | Name | Course | Year |
|--------|--------|----------|------|
| cs1 | Kannan | Big Data | II |
| cs3 | Lenin | Big Data | I |

PROJECT (symbol : π)

- The projection eliminates all attributes of the input relation but those mentioned in the projection list.
- The projection method defines a relation that contains a vertical subset of Relation.

Example

$\pi_{\text{course}}(\text{STUDENT})$

Result

| Course |
|--------------------|
| Big Data |
| R language |
| Python Programming |

- Duplicate row is removed in the result

2. Relational Algebra Operations from Set Theory

1. UNION (Symbol : \cup)

- It includes all tuples that are in tables A or in B. It also eliminates duplicates. Set A Union Set B would be expressed as $A \cup B$

- **Example 3**

Consider the following tables

| Table A | | Table B | | Table $A \cup B$ | |
|---------|---------|---------|--------------|------------------|--------------|
| Studno | Name | Studno | Name | Studno | Name |
| cs1 | Kannan | cs1 | Kannan | cs1 | Kannan |
| cs3 | Lenin | cs2 | GowriShankar | cs2 | GowriShankar |
| cs4 | Padmaja | cs3 | Lenin | cs3 | Lenin |
| | | | | cs4 | Padmaja |

2. SET DIFFERENCE (Symbol : $-$)

- The result of $A - B$, is a relation which includes all tuples that are in A but not in B.
- The attribute name of A has to match with the attribute name in B.
- Example 4 (using ABOVE TABLE)

| Table $A - B$ | |
|---------------|---------|
| cs4 | Padmaja |

3. INTERSECTION (symbol : \cap) $A \cap B$

- Defines a relation consisting of a set of all tuple that are in both in A and B.
- However, A and B must be union-compatible.

| A ∩ B | |
|-------|--------|
| cs1 | Kannan |
| cs3 | Lenin |

4. PRODUCT OR CARTESIAN PRODUCT (Symbol : X)

- Cross product is a way of combining two relations.
- The resulting relation contains, both relations being combined.
- A x B means A times B, where the relation A and B have different attributes.
- This type of operation is helpful to merge columns from two relations.

| Table A | | Table B | |
|---------|---------------|---------|------------|
| Studno | Name | studno | subject |
| cs1 | Kannan | cs28 | Big Data |
| cs2 | Gowri Shankar | cs62 | R language |

Cartesian product :Table A x Table B

| studno | Name | Course | subject |
|--------|---------------|--------|------------|
| cs1 | Kannan | cs28 | Big Data |
| cs1 | Kannan | cs62 | R language |
| cs2 | Gowri Shankar | cs28 | Big Data |
| cs2 | Gowri Shankar | cs62 | R language |

5. Explain the Characteristics of Database Management System

1. Data stored into Tables

- Data is never directly stored into the database.
- Data is stored into tables, created inside the database.
- DBMS also allows to have relationship between tables which makes the data more meaningful and connected.

2. Reduced Redundancy

- In the modern world hard drives are very cheap, but earlier when hard drives were too expensive, unnecessary repetition of data in database was a big problem But DBMS follows Normalization which divides the data in such a way that repetition is minimized.

3. Data Consistency

- On live data, it is being continuously updated and added, maintaining the consistency of data can become a challenge.
- But DBMS handles it by itself.

4. Support Multiple user and Concurrent Access

- DBMS allows multiple users to work on it(update, insert, delete data) at the same time and still manages to maintain the data consistency.

5. Query Language

- DBMS provides users with a simple query language, using which data can be easily fetched, inserted, deleted and updated in a database.

6. Security

- The DBMS also takes care of the security of data, protecting the data from unauthorized access.
- In a typical DBMS, we can create user accounts with different access permissions, using which we can easily secure our data by restricting user access.

7. DBMS Supports Transactions

- It allows us to better handle and manage data integrity in real world applications where multi-threading is extensively used

Extra Question Answers:

Introduction- Database:

- A database is an organized collection of data, generally stored and accessed electronically from a computer system.
- The term "database" is also used to refer to any of the DBMS, the database system or an application associated with the database.

1. Define Data.

Data:

- Data are raw facts stored in a computer.
- A data may contain any character, text, word or a number.
- **Example :** 600006, DPI Campus, SCERT, Chennai, College Road

2. Define Information.

Information:

- Information is formatted data, which allows to be utilized in a significant way.
- **Example**
SCERT
College Road
DPI Campus
Chennai 600006
- When the data is processed, organized and formatted, it gives a meaningful information about the SCERT institution contact address.

3. Define Database.

- Database is a repository collection of related data organized in a way that data can be easily accessed, managed and updated.
- Database can be a software or hardware based, with one sole purpose of storing data.

4. Write short note on DBMS.

Data Base Management System (DBMS)

- A DBMS is software that allows us to create, define and manipulate database, allowing users to store, process and analyze data easily.
- DBMS provides us with an interface or a tool, to perform various operations to create a database, storing of data and for updating data, etc.
- DBMS also provides protection and security to the databases.
- It also maintains data consistency in case of multiple users.
- **Examples of DBMS software's are :** Foxpro, dbase.

5. What are the Advantages of DBMS

- Segregation of application program
- Minimal data duplication or Data Redundancy
- Easy retrieval of data using the Query Language
- Reduced development time and maintenance

6. Explain the Components of DBMS

1. Hardware
2. Software
3. Data
4. Procedures / Methods
5. Database Access Languages

1. Hardware:

- The computer, hard disk, I/O channels for data, and any other physical component involved in storage of data

2. Software:

- This main component is a program that controls everything.
- The DBMS software is capable of understanding the Database Access Languages and interprets into database commands for execution.

3. Data:

- It is that resource for which DBMS is designed.
- DBMS creation is to store and utilize data.

4. Procedures/Methods:

- They are general instructions to use a database management system such as installation of DBMS, manage databases to take backups, report generation, etc.

5. Data Base Access Languages:

- They are the languages used to write commands to access, insert, update and delete data stored in any database.
- **Examples of popular DBMS:** Dbase, FoxPro

7. Define Table, Record and Field.

Table :

- Table is the entire collection of related data in one table, referred to as a File or Table where the data is organized as row and column.
- A Table is known as a RELATION

Record:

- Each row in a table represents a record, which is a set of data for each database entry.
- A Row is known as a TUPLE

Field:

- Each table column represents a Field, which groups each piece or item of data among the records into specific categories or types of data.
- A column is known as an ATTRIBUTE
- Eg. StuNo., StuName, StuAge, StuClass, StuSec.

CHAPTER – 12 STRUCTURED QUERY LANGUAGE (SQL)

PART - II

1. Write a query that selects all students whose age is less than 18 in order wise.

SELECT * FROM Student WHERE Age<18 ORDER BY Age;

2. Differentiate Unique and Primary Key constraints.

| Unique Key Constraints | Primary Key Constraints |
|--|---|
| Unique constraint ensures that no two rows have the same value in the specified columns. | This constraint declares a field as a Primary key which helps to uniquely identify a record. |
| The UNIQUE constraint can be applied only to fields that have also been declared as NOT NULL . | The primary key does not allow NULL values and therefore a field declared as primary key must have the NOT NULL constraint |
| | It is similar to unique constraint except that only one field of a table can be set as primary key . |

3. Write the difference between table constraint and column constraint?

Column constraint - Column constraint apply only to individual column.

Table constraint - Table constraint apply to a group of one or more columns

4. Which components of SQL lets insert values in tables and which lets to create a Table?

- i. To insert values in tables - Data Manipulation Language (DML) can be used
- ii. To create a Table - Data Definition Language (DDL) can be used

5.What is the difference between SQL and MySQL

SQL:

- Structured Query Language is a Language used for accessing databases
- It is a language that helps to create and operate relational databases

MySQL:

- MySQL is a database management system
- MySQL is a RDBMS

PART-III

1. What is a constraint? Write short note on Primary key constraint.

Constraint:

- Constraint is a condition applicable on a field or set of fields.

Uses of Constraints:

- Constraints are used to limit the type of data that can go into a table.
- This ensures the accuracy and reliability of the data in the database.
- Constraints could be either on a column level or a table level.
- Constraints ensure database integrity

Primary Key Constraint

- This constraint declares a field as a Primary key which helps to uniquely identify a record.
- It is similar to unique constraint except that only one field of a table can be set as primary key.
- The primary key does not allow **NULL** values and therefore a field declared as primary key must have the **NOT NULL** constraint.

2. Write a SQL statement to modify the student table structure by adding a new field.

- The **ALTER** command is used to alter the table structure like adding a column, renaming the existing column, change the data type of any column or size of the column or delete the column from the table.

Syntax:

ALTER TABLE <table-name> ADD <column-name><data type><size>;

Example:

ALTER TABLE Student ADD Address char(25);

3. Write any three DDL commands.

1. **Create:** To create tables in the database.
2. **Alter:** Alters the structure of the database.
3. **Drop:** Delete tables from database.
4. **Truncate:** Remove all records from a table, also release the space occupied by those records.

2. Write the use of Savepoint command with an example.

SAVEPOINT command

- The **SAVEPOINT** command is used to temporarily save a transaction so that you can rollback to the point whenever required.
- The different states of our table can be saved at anytime using different names and the rollback to that state can be done using the **ROLLBACK** command.

SAVEPOINT savepoint_name;

Example:

INSERT INTO Student VALUES (107, 'Beena', 'F', 20, 'Cochin');

COMMIT;

UPDATE Student SET Name = 'Mini' WHERE Admno=105;

SAVEPOINT A;

INSERT INTO Student VALUES(106, 'Jisha', 'F', 19, 'Delhi');

SAVEPOINT B;

ROLLBACK TO A;

5. Write a SQL statement using DISTINCT keyword.

DISTINCT Keyword

- The **DISTINCT** keyword is used along with the **SELECT** command to eliminate duplicate rows in the table.
- This helps to eliminate redundant data.

Example:

SELECT DISTINCT Place FROM Student;

PART-IV

1. Write the different types of constraints and their functions.

Constraint:

- Constraint is a condition applicable on a field or set of fields.

Uses of Constraints:

- Constraints are used to limit the type of data that can go into a table.
- This ensures the accuracy and reliability of the data in the database.
- Constraints could be either on a column level or a table level.
- Constraints ensure database integrity

Type of Constraints

1. Unique Constraint
2. Primary Key Constraint
3. Default Constraint
4. Check Constraint

(1)Unique Constraint

- This constraint ensures that no two rows have the same value in the specified columns.
- The **UNIQUE** constraint can be applied only to fields that have also been declared as **NOT NULL**.
- Example: **UNIQUE** constraint applied on **Admno** of student table ensures that no two students have the same admission number

CREATE TABLE Student

```
(  
  Admno integer NOT NULL UNIQUE, → Unique constraint  
  Name char (20) NOT NULL,  
  Gender char (1),  
  Age integer,  
  Place char (10),  
);
```

Multiple Constraints:

- When two constraints are applied on a single field, it is known as multiple constraints.

(2)Primary Key Constraint:

- This constraint declares a field as a Primary key which helps to uniquely identify a record.
- It is similar to unique constraint except that only one field of a table can be set as primary key.
- The primary key does not allow **NULL** values and therefore a field declared as primary key must have the **NOT NULL** constraint.
- Example:

CREATE TABLE Student

```
(  
  Admno integer NOT NULL PRIMARY KEY, → Primary Key constraint  
  Name char(20)NOT NULL,  
  Gender char(1),  
  Age integer,  
  Place char(10),  
);
```

(3) DEFAULT Constraint

- The **DEFAULT** constraint is used to assign a default value for the field.
- When no value is given for the specified field having **DEFAULT** constraint, automatically the default value will be assigned to the field.
- Example:

CREATE TABLE Student

```
(  
  Admno integer NOT NULL PRIMARY KEY,  
  Name char(20)NOT NULL,
```

```
  Gender char(1),  
  Age integer DEFAULT = "17", → Default Constraint  
  Place char(10),  
);
```

(4) Check Constraint:

- This constraint helps to set a limit value placed for a field.
- When we define a check constraint on a single column, it allows only the restricted values on that field.
- The check constraint may use relational and logical operators for condition.
- Example

CREATE TABLE Student

```
(  
  Admno integer NOT NULL PRIMARY KEY  
  Name char(20)NOT NULL,  
  Gender char(1),  
  Age integer CHECK Age<=19, → Check Constraint  
  Place char(10)  
);
```

TABLE CONSTRAINT

- When the constraint is applied to a group of fields of the table, it is known as Table constraint.
- The table constraint is normally given at the end of the table definition.

Example:

CREATE TABLE Student 1

```
(  
  Admno integer NOT NULL,  
  Firstname char(20),  
  Lastname char(20),  
  Gender char(1),  
  Age integer,  
  Place char(10),  
  PRIMARY KEY (Firstname, Lastname) → Table constraint  
);
```

2. Consider the following employee table. Write SQL commands for the qtns.(i) to (v).

| EMP CODE | NAME | DESIG | PAY | ALLOWANCE |
|----------|-----------|------------|-------|-----------|
| S1001 | Hariharan | Supervisor | 29000 | 12000 |
| P1002 | Shaji | Operator | 10000 | 5500 |
| P1003 | Prasad | Operator | 12000 | 6500 |
| C1004 | Manjima | Clerk | 8000 | 4500 |
| M1005 | Ratheesh | Mechanic | 20000 | 7000 |

(i) To display the details of all employees in descending order of pay.

SELECT * from employee ORDER BY pay DESC;

(ii) To display all employees whose allowance is between 5000 and 7000.

SELECT * from employee WHERE allowance BETWEEN 5000 and 7000;

(iii) To remove the employees who are mechanic.

DELETE from employee WHERE desig = "Mechanic";

(iv) To add a new row.

INSERT INTO employee VALUES ("M1006", "Kumar", "Mechanic", 25000, 8000);

(v) To display the details of all employees who are operators.

SELECT * FROM employee WHERE desig = "Operator";

3. What are the Components of SQL? Write the commands in each.

1. DML – Data Manipulation Language
2. DDL – Data Definition Language
3. DCL – Data Control Language
4. TCL – Transaction Control Language
5. DQL – Data Query Language

SQL commands - Data Definition Language :

1. Create : To create tables in the database.
2. Alter: Alters the structure of the database.
3. Drop: Delete tables from database.
4. Truncate: Remove all records from a table, also release the space occupied by those records.

SQL commands - Data Manipulation Language:

1. Insert - Inserts data into a table
2. Update - Updates the existing data within a table.
3. Delete - Deletes all records from a table, but not the space occupied by them.

SQL commands - Data Control Language :

1. Grant - Grants permission to one or more users to perform specific tasks.

2. Revoke - Withdraws the access permission given by the GRANT statement.

SQL command - Transactional Control Language :

1. Commit - Saves any transaction into the database permanently.
2. Roll back - Restores the database to last commit state.
3. Save point - Temporarily save a transaction so that you can rollback.

SQL command in Data Query Language is

1. Select - It displays the records from the table

4. Construct the following SQL statements in the student table-

(i) SELECT statement using GROUP BY clause.

(ii) SELECT statement using ORDER BY clause.

(i) SELECT statement using GROUP BY clause.

GROUP BY clause

- The **GROUP BY** clause is used with the **SELECT** statement to group the students on rows or columns having identical values or divide the table in to groups.
- The syntax for the **GROUP BY** clause is
- **SELECT <column-names> FROM <table-name> GROUP BY <column-name>HAVING <condition>;**

Example:

- **SELECT Gender FROM Student GROUP BY Gender;**
SELECT Gender, count(*) FROM Student GROUP BY Gender;
- The * is used with the COUNT to include the NULL values.
- The GROUP BY applies the aggregate functions independently to a series of groups that are defined by having a field value in common.

(ii) SELECT statement using ORDER BY clause.

ORDER BY clause

- The **ORDER BY** clause in SQL is used to sort the data in either ascending or descending based on one or more columns.
 1. By default **ORDER BY** sorts the data in ascending order.
 2. We can use the keyword **DESC** to sort the data in descending order and the keyword **ASC** to sort in ascending order.

The ORDER BY clause is used as :

SELECT <column-name>[,<column-name>,....] FROM <table-name>ORDER BY <column1>,<column2>,...ASC| DESC ;

Example :

SELECT * FROM Student ORDER BY Name;

The ORDER BY clause does not affect the original table.

WHERE clause

- The **WHERE** clause is used to filter the records.
- It helps to extract only those records which satisfy a given condition.

Example

```
SELECT * FROM Student WHERE Age>=18 ORDER BY Name;
SELECT * FROM Student WHERE Age>=18 ORDER BY Name DESC;
Sorting can be done on multiple fields
```

5. Write a SQL statement to create a table for employee having any five fields and create a table constraint for the employee table.

CREATE TABLE Employee

```
(
    Empno integer NOT NULL,
    Firstname char(20),
    Lastname char(20),
    Design char(20),
    Pay integer,
    PRIMARY KEY (Firstname, Lastname)
);
```

EXTRA QUESTION ANSWER:

1. SQL Commands and their Functions

- **Keywords** - They have a special meaning in SQL. They are understood as instructions.
- **Commands** - They are instructions given by the user to the database also known as statements.
- **Clauses** - They begin with a keyword and consist of keyword and argument
- **Arguments** - They are the values given to make the clause complete.

DDL Commands

CREATE TABLE Command

- You can create a table by using the **CREATE TABLE** command.
- When a table is created, its columns are named, data types and sizes are to be specified.
- Each table must have at least one column.

The syntax for a table created with constraint:

```
CREATE TABLE <table-name>
    (<column name><data type>[<size>]<column constraint>,
    <column name><data type>[<size>]<column constraint> .....
    <table constraint>(<column name>,[<column name>....]).....
);
```

Example:

```
CREATE TABLE Student
(
```

```
Admno integer NOT NULL PRIMARY KEY, → Primary Key constraint
Name char(20) NOT NULL,
Gender char(1),
Age integer,
Place char(10),
);
```

2. What is SQL?

- The Structured Query Language (SQL) is a standard programming language to access and manipulate databases.
- SQL allows the user to create, retrieve, alter, and transfer information among databases.
- It is a language designed for managing and accessing data in a Relational Data Base Management System (RDBMS).

3. List out different versions of SQL?

- The original version was developed at IBM's Research Centre and originally called as **Sequel** in early **1970's**.
- Later the language was changed to SQL.
- In **1986**, ANSI (American National Standard Institute) published an SQL standard that was updated again in 1992,
- The latest SQL was released in 2008 and named as **SQL 2008**.

4. List out RDBMS packages.

- RDBMS stands for Relational DataBase Management System.
- RDBMS packages are
- Oracle,
- MySQL,
- MS SQL Server,
- IBM DB2 and
- Microsoft Access.

5. What are the functions of RDBMS?

- RDBMS is a type of DBMS with a row-based table structure that connects related data elements and includes functions related to Create, Read, Update and Delete operations, collectively known as CRUD.

6. Define Table, Field and Record.

Tables:

- The data in RDBMS, is stored in database objects, called Tables. A table is a collection of related data entries and it consists of rows and columns.

Field

- A field is a column in a table that is designed to maintain specific related information about every record in the table.
- The fields in a student table may be of the type AdmnNo, StudName, StudAge, StudClass, Place etc.

Record:

- A Record is a row, which is a collection of related fields or columns that exist in a table.
- A record is a horizontal entity in a table which represents the details of a particular student in a student table.

7. Explain Processing Skills of SQL

1. Data Definition Language (DDL) :

- The SQL DDL provides commands for defining relation schemas (structure), deleting relations, creating indexes and modifying relation schemas.

2. Data Manipulation Language (DML) :

- The SQL DML includes commands to insert, delete, and modify tuples in the database.

3. Embedded Data Manipulation Language :

- The embedded form of SQL is used in high level programming languages.

4. View Definition:

- The SQL also includes commands for defining views of tables.

5. Authorization :

- The SQL includes commands for access rights to relations and views of tables.

6. Integrity :

- The SQL provides forms for integrity checking using condition.

7. Transaction control :

The SQL includes commands for transaction processing and control over transaction processing

8. How to create new Database?

Creating Database

- Syntax: CREATE DATABASE database_name;
- Example : CREATE DATABASE stud;

9. How to work with the database

- Syntax: USE DATABASE;
- Example: USE stud;

10. Expand WAMP ?

- WAMP stands for “Windows, Apache, MySQL and PHP”.

DML COMMANDS

- Once the schema or structure of the table is created, values can be added to the table.
- The DML commands consist of inserting, deleting and updating rows into the table.
- **INSERT command** The INSERT command helps to add new data to the database or add new records to the table.
- Syntax:

INSERT INTO <table-name> [column-list] VALUES (values);

Example:

INSERT INTO Student (AdmnNo, Name, Gender, Age, Place) VALUES

(100,' Ashish',' M', 17,' Chennai');

INSERT INTO Student (AdmnNo, Name, Gender, Age, Place) VALUES

(102, 'Adarsh', 'M', 18, 'Delhi');

INSERT INTO Student VALUES (102, 'Akshith', 'M', '17,' 'Bangalore');

INSERT INTO Student (AdmnNo, Name, Place) VALUES (104, 'Abinandh', 'Chennai');

Note:

- In the INSERT command the fields that are omitted will have either default value defined or NULL value
- **DELETECOMMAND**
- The DELETE command permanently removes one or more records from the table.
- It removes the entire row, not individual fields of the row, so no field argument is needed.

- Syntax:

DELETE FROM table-name *WHERE* condition;

Example:

DELETE FROM Student **WHERE** AdmnNo=104;

This record will be deleted from table student

DELETE * FROM Student;

All the record will be deleted from table student.

- **UPDATE COMMAND**
- The UPDATE command updates some or all data values in a database.
- It can update one or more records in a table.
- The **UPDATE** command specifies the rows to be changed using the **WHERE** clause and the new data using the **SET** keyword.
- Syntax:

UPDATE <table-name> **SET** column-name = value, column-name = value,... *WHERE* condition;

Example:

UPDATE Student **SET** Age = 20 **WHERE** Place = “Bangalore”;

□ To update multiple fields, multiple field assignment can be specified with the **SET** clause separated by comma.

UPDATE Student **SET** Age=18, Place = ‘Chennai’ **WHERE** AdmnNo = 102;

Additional DDL Commands:

ALTER COMMAND

- The **ALTER** command is used to alter the table structure like adding a column, renaming the existing column, change the data type of any column or size of the column or delete the column from the table.

Syntax:

ALTER TABLE <table-name> ADD <column-name><data type><size>;

Example:

ALTER TABLE Student ADD Address char;

- To modify existing column of table, the **ALTER TABLE** command can be used with **MODIFY** clause

Syntax:

ALTER <table-name> MODIFY<column-name><data type><size>;

Example:

ALTER TABLE Student MODIFY Address char (25);

- The **ALTER** command can be used to rename an existing column:

Syntax:

ALTER <table-name> RENAME old-column-name TO new-column-name;

Example:

ALTER TABLE Student RENAME Address TO City;

- **ALTER** command can also be used to remove a column or all columns

Syntax:

ALTER <table-name> DROP COLUMN <column-name>;

Example:

ALTER TABLE Student DROP COLUMN City;

TRUNCATE command

- The **TRUNCATE** command is used to delete all the rows from the table, the structure remains and the space is freed from the table.

Syntax:

TRUNCATE TABLE table-name;

Example:

TRUNCATE TABLE Student;

DROP TABLE command

- The **DROP TABLE** command is used to remove a table from the database.
- If you drop a table, all the rows in the table is deleted and the table structure is removed from the database.
- There is a condition for dropping a table; it must be an empty table.
- Once all the rows are deleted, the table can be deleted by **DROP TABLE** command

- **DROP TABLE table-name;**
DROP TABLE Student;

DELETE, TRUNCATE AND DROP statement:

DELETE

- The **DELETE** command deletes only the rows from the table based on the condition given in the where clause or deletes all the rows from the table if no condition is specified.
- But it does not free the space containing the table.

TRUNCATE

- The **TRUNCATE** command is used to delete all the rows, the structure remains in the table and free the space containing the table.

DROP

- The **DROP** command is used to remove an object from the database.
- If you drop a table, all the rows in the table is deleted and the table structure is removed from the database.
- Once a table is dropped we cannot get it back.

DQL COMMAND– SELECT command

- A Query is a command given to get a desired result from the database table.
- The **SELECT** command is used to query or retrieve data from a table in the database.
- It is used to retrieve a subset of records from one or more tables.

Syntax:

SELECT <column-list>FROM<table-name>;

- Table-name is the name of the table from which the information is retrieved.
- Column-list includes one or more columns from which data is retrieved.

Example:

SELECT Admno, Name FROM Student;

SELECT * FROM STUDENT;

DISTINCT Keyword

- The **DISTINCT** keyword is used along with the **SELECT** command to eliminate duplicate rows in the table.
- This helps to eliminate redundant data.

Example:

SELECT DISTINCT Place FROM Student;

ALL Keyword

- The **ALL** keyword retains duplicate rows.
- It will display every row of the table without considering duplicate entries.

SELECT ALL Place FROM Student;

- The **WHERE** clause in the **SELECT** command specifies the criteria for getting the desired result.

Syntax:

SELECT <column-name>[,<column-name>,....] FROM <table-name>WHERE condition;

Example:

SELECT Admno, Name, Place FROM Student WHERE Place ="Chennai";

SELECT Admno, Name, Age FROM Student WHERE Age >= 18;

- The relational operators like =, <, <=, >, >=, <> can be used to compare two values in the **SELECT** command used with **WHERE** clause.
- The logical operators **OR**, **AND** and **NOT** can also be used to connect search conditions in the **WHERE** clause.
- Example :
SELECT Admno, Name, Age, Place FROM Student WHERE (Age>=18 AND Place = "Delhi");
SELECT Admno, Name, Age, Place FROM Student WHERE (Age>=18 OR Place ="Delhi");
SELECT Admno, Name, Place FROM Student WHERE (NOT Place ="Delhi");

BETWEEN and NOT BETWEEN Keywords

- The **BETWEEN** keyword defines a range of values the record must fall into to make the condition true.
- The range may include an upper value and a lower value between which the criteria must fall into.
SELECT Admno, Name, Age, Gender FROM Student WHERE Age BETWEEN 18 AND 19;
- The **NOT BETWEEN** is reverse of the **BETWEEN** operator where the records not satisfying the condition are displayed.
SELECT Admno, Name, Age FROM Student WHERE Age NOT BETWEEN 18 AND 19;

IN Keyword

- The **IN** keyword is used to specify a list of values which must be matched with the record values.
- It is used to compare a column with more than one value.
- It is similar to an **OR** condition.
- **Example :**
 - **SELECT Admno, Name, Place FROM Student WHERE Place IN ("Chennai", "Delhi");**
- The **NOT IN** keyword displays only those records that do not match in the list.
- Example:

SELECT Admno, Name, Place FROM Student WHERE Place NOT IN ("Chennai", "Delhi");

NULL Value :

- The **NULL** value in a field can be searched in a table using the **IS NULL** in the **WHERE** clause.
SELECT * FROM Student WHERE Age IS NULL;
- Non **NULL** values in a table can be listed using **IS NOT NULL**.

HAVING clause

- The **HAVING** clause can be used along with **GROUP BY** clause in the **SELECT** statement to place condition on groups and can include aggregate functions on them.
- Example
SELECT Gender , count(*) FROM Student GROUP BY Gender HAVING Place = 'Chennai';

TCL commands

COMMIT command

- The **COMMIT** command is used to permanently save any transaction to the database.
- When any DML commands like **INSERT**, **UPDATE**, **DELETE** commands are used, the changes made by these commands are not permanent.
- It is marked permanent only after the **COMMIT** command is given from the SQL prompt.
- Once the **COMMIT** command is given, the changes made cannot be rolled back.
- The **COMMIT** command is used as
COMMIT;

ROLLBACK command

The **ROLLBACK** command restores the database to the last committed state.

- It is used with **SAVEPOINT** command to jump to a particular savepoint location.
- The syntax for the **ROLLBACK** command is :
ROLL BACK TO save point name;

Choose the correct Answer

1. Which commands provide definitions for creating table structure, deleting relations, and modifying relation schemas.

- | | | | |
|---|-------------|---------------|----------------|
| a. DDL | b. DML | c. DCL | d. DQL |
| 2. Which command lets to change the structure of the table? | | | |
| a. SELECT | b. ORDER BY | c. MODIFY | d. ALTER |
| 3. The command to delete a table is | | | |
| A) DROP | B) DELETE | C) DELETE ALL | D) ALTER TABLE |
| 4. Queries can be generated using | | | |

a. **SELECT** b. ORDER BY c. MODIFY d. ALTER

5. The clause used to sort data in a database
a. SORT BY b. **ORDER BY** c. GROUP BY d. SELECT

6. Which of the following is a standard programming language to access and manipulate databases
a. MySQL b. **SQL** c. PHP d. Python

7. Which of the following language was designed for managing and accessing data in RDBMS?
a. DBMS b. DDL c. DML d. **SQL**

8. Latest version of SQL was released in the year
a. 2018 b. 1982 c. 1992 d. **2008**

9. The specific related information about every record in the table is maintained by
a. Language b. Relation c. Tuple d. **Field**

10. DDL Stands for
a. Data Defined Language b. **Data Definition Language**
c. Definition Data Language d. Dictionary Data Language

11. Which component of SQL includes commands to insert, delete and modify data in a tables in database?
a. DCL b. TCL c. DDL d. **DML**

12. Which processing skills of SQL includes commands of access rights to creations and views of tables?
a. View definition b. Integrity c. **Authorization** d. Transaction Control

13. WAMP stands for
a. Windows, Android, MySQL, PHP b. Windows, Apache, MySQL, Python
c. Windows, APL, MySQL, PHP d. **Windows, Apache, MySQL, PHP**

14. How many categories of SQL commands are divided?
a. 4 b. **5** c. 3 d. 7

15. The data in a database is stored based on the kind of value stored is known as
a. language b. function c. record d. **datatype**

16. Which of the following data type same as real except the precision may exceed 64?
a. float b. read c. **double** d. long real

17. Which of the following begin with a keyword and consists of keyword and argument?
a. commands b. statements c. **clauses** d. data

18. Which constraint apply to a group of one or more column?
a. Column b. Default c. Uniqu d. **Table**

19. Which of the following SQL DDL command used to create a table
a. CREATE b. **CREATE TABLE** c. NEW TABLE d. DDL TABLE

20. The constraint enforces a field to always contain a value is
a. NULL b. **NOT NULL** c. YES d. ALWAYS

21. How many types of database integrity constraints are there?
a. **4** b. 5 c. 3 d. Multiple

22. Which of the following constraint does not allow NULL values?
a. Unique b. Primary key c. Check d. **Unique and Primary Key**

23. Which of the following constraint use relational and logical operators for condition?
a. Primary key b. Unique c. **Check** d. Table

24. ANSI stands for
a. **American National Standard Institute** b. American National System Institute
c. American National Structured International d. American National Standard Instruction

25. Which keyword is used to sort the records in ascending order

a. ASCD b. ASD
c. ASCE d. **ASC**

26. Which keyword is used to divide the table into groups?
a. DIVIDE BY b. ORDER BY
c. **GROUP BY** d. HAVING

27. Which clause can be used along with GROUP BY clause and SELECT statement to include aggregate function on them?
a. WHERE b. FROM
c. **HAVING** d. COMMIT

28. Which command is used to permanently save any transaction to the database?
a. **COMMIT** b. SAVE c. SAVEPOINT d. ROLLBACK

29. Which of the following are the TCL commands?
a. COMMIT b. SAVEPOINT c. ROLLBACK d. **All the above**

30. Which command is used to temporarily save a transaction
a. COMMIT b. SAVE c. **SAVEPOINT** d. ROLLBACK

31. Which command restores the database to the last committed state?
a. COMMIT b. SAVE c. SAVEPOINT d. **ROLLBACK**

32. Which provides statements for creation and deletion of tables
a. **DDL** b. DQL c. DML d. TCL

33. Which provides authorization commands to access data
a. DDL b. **DCL** c. DML d. TCL