

# Python

## Interview Questions



iNeuron Intelligence Pvt Ltd

---

### **Q1. Define Dictionary.**

Ans- a) Unordered sets of objects.

b) Also known as maps, hashmaps, lookup tables, or associative array.

c) Data exists in key-value pair. Elements in dictionary have a key and a corresponding value. The key and the value are separated from each other with a colon “:” and all elements are separated by comma.

d) Elements of a dictionary are accessed by the “key” and not by index. Hence it is more or less like an associative array where every key is associated with a value and elements exist in an unordered fashion as key-value pair.

e) Dictionary literals use curly brackets ‘ {}’.

### **Q2. How can we create a dictionary object?**

Ans- A dictionary object can be created in any of the following ways:

# Initializing an empty Dictionary

```
Dictionary = {}
```

```
print("An empty Dictionary: ")
```

```
print(Dictionary)
```

# Creating a Dictionary using in-built dict() method

```
Dictionary = dict({1: 'Python', 2: 'Java', 3: 'Dictionary'})
```

```
print("\nDictionary created by using dict() method: ")
```

```
print(Dictionary)
```

# Creating dictionary by key: value pair format

```
Dictionary = dict([(1, 'Java'), (2, 'Python'), (3, 'Dictionary')])
```

```
print("\nDictionary with key: value pair format: ")
```

```
print(Dictionary)
```

### Q3. Explain from Keys () method.

Ans- The fromkeys () method returns a new dictionary that will have the same keys as the dictionary object passed as the argument. If you provide a value then all keys will be set to that value or else all keys will be set to 'None'.

```
In [7]: #forming a dictionary
subject={'english':89,'maths':90}
print(subject)
print(len(subject))
print(type(subject))

{'english': 89, 'maths': 90}
2
<class 'dict'>

In [8]: #providing no value
dict2=dict.fromkeys(subject)
dict2

Out[8]: {'english': None, 'maths': None}

In [10]: #providing a value
dict3=dict.fromkeys(subject,92)
dict3

Out[10]: {'english': 92, 'maths': 92}
```

### Q4. What is the purpose of items() function?

Ans- the items() function does not take any parameters. It returns the view object that shows the given dictionary's key value pairs.

```
In [11]: #usage of items function

subject={'english':89,'maths':90}
subject.items()

Out[11]: dict_items([('english', 89), ('maths', 90)])
```

### Q5. Define bucket sorting?

Ans- Bucket Sort is a two-step procedure for sorting data. First, the values are collected in special containers called buckets. Then, these values are transferred appropriately into a sorted list. For the algorithm to be feasible, the elements to be sorted must have a limited set of values.

**Q6. Which one of the following is correct way of declaring and initialising a variable, x with value 5?**

- A. `int x`  
    `x=5`
- B. `int x=5`
- C. `x=5`
- D. `declare x=5`

Ans: C

Explanation: One of the following is correct way of declaring and initialising a variable, x with value 5 is `x=5`.

**Q7. How many local and global variables are there in the following Python code?**

```
var1=5
def fn():
    var1=2
    var2=var1+5
var1=10
fn()
```

- A. 1 local, 1 global variables
- B. 1 local, 2 global variables
- C. 2 local, 1 global variables
- D. 2 local, 2 global variables

Ans-D. 2 local, 2 global variables

Explanation: 2 local, 2 global variables are there in the following Python code.

**Q8. Which one is false regarding local variables?**

- A. These can be accessed only inside owning function
- B. Any changes made to local variables does not reflect outside the function.
- C. These remain in memory till the program ends
- D. None of the above

Ans: C

Explanation: These remain in memory till the program ends is false regarding local variables.

**Q9. Which of the following options will give an error if set1={2,3,4,5}?**

- A. print(set1[0])
- B. set1[0]=9
- C. set1=set1+{7}
- D. All of the above

Ans : D

Explanation: All of the above option will give error if set1={2,3,4,5}.

**Q10. What will be the result of below Python code?**

```
set1= {1,2,3}
```

```
set1.add (4)
```

```
set1.add (4)
```

```
print(set1)
```

- A. {1,2,3,4}
- B. {1,2,3}
- C. {1,2,3,4,4}
- D. It will throw an error as same element is added twice

Ans: A

Explanation: The output for the following python code is {1,2,3,4}.



iNeuron Intelligence Pvt Ltd

---

**Q11. Which of the following options will give an error if set1= {2,3,4,5}?**

- A. `print(set1[0])`
- B. `set1[0] = 9`
- C. `set1=set1 + {7}`
- D. All of the above

Ans: D

Explanation: All of the above option will give error if set1= {2,3,4,5}

**Q12.What will the below Python code do?**

```
set1={"a",3,"b",3}  
set1.remove(3)
```

- A. It removes element with index position 3 from set1
- B. It removes element 3 from set1
- C. It removes only the first occurrence of 3 from set1
- D. No such function exists for set

Ans: B

Explanation: It removes element 3 from set1.

**Q13. What will be the output of following Python code?**

```
set1= {2,5,3}  
set2= {3,1}  
set3= {}  
set3=set1&set2  
print(set3)
```

- A. {3}
- B. {}
- C. {2,5,3,1}
- D. {2,5,1}

Ans: A

Explanation: The output of the following code is {3}



iNeuron Intelligence Pvt Ltd

---

**Q14. Which of the following is True regarding lists in Python?**

- A. Lists are immutable.
- B. Size of the lists must be specified before its initialization
- C. Elements of lists are stored in contiguous memory location.
- D. `size(list1)` command is used to find the size of lists.

Ans: C

Explanation: Elements of lists are stored in contiguous memory location is True regarding lists in Python.

**Q15. Which of the following will give output as [23,2,9,75]?**

If `list1 = [6,23,3,2,0,9,8,75]`

- A. `print(list1[1:7:2])`
- B. `print(list1[0:7:2])`
- C. `print(list1[1:8:2])`
- D. `print(list1[0:8:2])`

View Answer

Ans: C

Explanation: `print(list1[1:8:2])` of the following will give output as [23,2,9,75].

**Q16. The marks of a student on 6 subjects are stored in a list, `list1 = [80,66,94,87,99,95]`. How can the student's average mark be calculated?**

- A. `print(avg(list1))`
- B. `print(sum(list1)/len(list1))`
- C. `print(sum(list1)/size of(list1))`
- D. `print(total(list1)/len(list1))`

Ans: B

Explanation: the student's average mark be calculated through `print(sum(list1)/len(list1))`.



iNeuron Intelligence Pvt Ltd

---

**Q17. What will be the output of following Python code?**

```
list1= ["Python", "Java", "c", "C", "C++"]  
print(min(list1))
```

- A. c
- B. C++
- C. C
- D. min function cannot be used on string elements

Ans: C

Explanation: C will be the output of following Python code.

**Q18. What will be the result after the execution of above Python code?**

```
list1= [3,2,5,7,3,6]  
list1.pop (3)  
print(list1)
```

- A. [3,2,5,3,6]
- B. [2,5,7,3,6]
- C. [2,5,7,6]
- D. [3,2,5,7,3,6]

Ans: A

Explanation: [3,2,5,3,6] will be the result after the execution of above Python code.



**Q19. What will be the output of below Python code?**

```
list1=["tom","mary","simon"]  
list1.insert(5,8)  
print(list1)
```

- A. ["tom", "mary", "simon", 5]
- B. ["tom", "mary", "simon", 8]
- C. [8, "tom", "mary", "simon"]
- D. Error

Ans: B

Explanation: ["tom", "mary", "simon", 8] will be the result after the execution of above Python code.

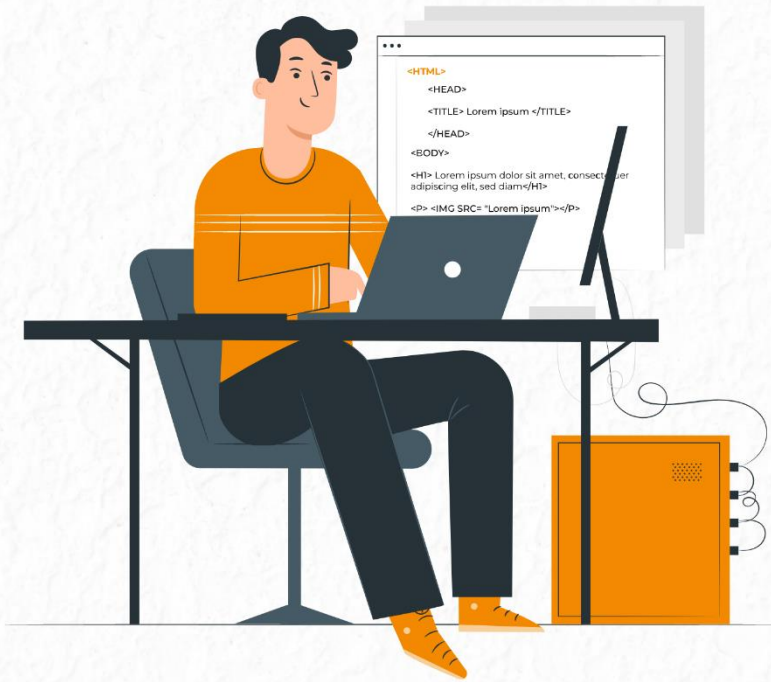
**Q20. Which among the following are mutable objects in Python?**

- (i) List
- (ii) Integer
- (iii) String
- (iv) Tuple

- A. i only
- B. i and ii only
- C. iii and iv only
- D. iv only

Ans- A

Explanation: List are mutable objects in Python.



# Python

## Interview Questions



iNeuron Intelligence Pvt Ltd

---

**Q21. What will be the output of below Python code?**

```
list1=["tom","mary","simon"]  
list1.insert(5,8)  
print(list1)
```

- A. ["tom", "mary", "simon", 5]
- B. ["tom", "mary", "simon", 8]
- C. [8, "tom", "mary", "simon"]
- D. Error

Ans: B

Explanation: ["tom", "mary", "simon", 8] will be the result after the execution of above Python code.

**Q22. The elements of a list are arranged in descending order. Which of the following two will give same outputs?**

- i. print(list\_name.sort())
- ii. print(max(list\_name))
- iii. print(list\_name.reverse())
- iv. print(list\_name[-1])

- A. i, ii
- B. i, iii
- C. ii, iii
- D. iii, iv

Ans: B

Explanation: print(list\_name.sort ()) and print(list\_name.reverse()) will give same outputs.

**Q23. What will be the output of below Python code?**

```
list1=[1,3,5,2,4,6,2]
list1.remove(2)
print(sum(list1))
```

- A. 18
- B. 19
- C. 21
- D. 22

Ans: C

Explanation: 21 will be the result after the execution of above Python code.

**Q24. Which of the following would give an error?**

- A. list1 = []
- B. list1= [] \*3
- C. list1= [2,8,7]
- D. None of the above

Ans: D

Explanation: None of the above will result in error

**Q25. What is type conversion in python?**

Ans - When we perform any operation on variables of different datatypes, the data of one variable will be converted to a higher datatype among the two variables and the operation is completed. When this conversion is done by interpreter automatically then it is known as implicit type conversion while conversions is done by user then it is called explicit type conversion.

```
num1=10
num2="20"
result=num1+int(num2)
print(result)
```

**Q26. When we want to treat some data as a group, it would not be good to create individual variables for each data. We can store them together as a collection.**

Ans- There are many collection data types which are supported by Python-

1. List- List can be used to store a group of elements together in a sequence.
2. Tuple- A tuple is an immutable sequence of Python objects. Tuples are sequences, just like lists.
3. String- In a program, not all values will be numerical. We will also have alphabetical or alpha numerical values. Such values are called strings.
4. Set- A set is an unordered group of values with no duplicate entries. Set can be created by using the keyword set or by using curly braces { }. set function is used to eliminate duplicate values in a list.
5. Dictionary- A dictionary can be used to store an unordered collection of key-value pairs. The key should be unique and can be of any data type.

**Q27. What is math module?**

Answer - math is another useful module in Python. Once you have imported the math module, you can use some of the below functions:

1. math.ceil(x) - Smallest integer greater than or equal to x
2. math.floor(x) - Largest integer smaller than or equal to x
3. math.factorial(x) - Factorial of x
4. math.fabs(x) - Gives absolute value of x

**Q28. What are the differences between python 2 and 3?**

Answer -The main differences between python 2 and python 3 are as follows

python 2	python 3
a) print statement is treated more as statement	print statement is treated more as statement
b) integer size limited to 32 bits	integer size unlimited
c) complex	Simplified
d) ASCII is used.	Unicode is used.

**Q29. What are docstrings in Python?**

Answer - Docstrings are not actually comments, but they are documentation strings. These docstrings are within triple quotes. They are not assigned to any variable and therefore, at times, serve the purpose of comments as well.

"""

Using docstring as a comment.

This code divides 2 numbers

"""

a=10

b=5

c=a/b

print(c)

Output -

2.0

**Q30. What is `__init__` in Python?**

Answer - "`__init__`" is a reserved method in python classes. It is called as a constructor in object-oriented terminology. This method is called when an object is created from a class and it allows the class to initialize the attributes of the class.

**Q31. Does Python have Opp's concepts?**

Ans - Python is an object-oriented programming language. This means that any program can be solved in python by creating an object model. However, Python can be treated as procedural as well as structural language.

**Q32. How will you capitalize the first letter of string?**

Ans- In Python, the `capitalize()` method capitalizes the first letter of a string. If the string already consists of a capital letter at the beginning, then, it returns the original string.

**Q33. Why we use Lambda Functions?**

Ans - Lambda functions are used when you need a function for a short period of time. This is commonly used when you want to pass a function as an argument to higher-order functions, i.e. functions that take other functions as their arguments.

**Q34. What is Exception handling in python?**

Ans - Sometimes the programs may misbehave or terminate/crash unexpectedly due to some unexpected events during the execution of a program. These unexpected events are called as exceptions and the process of handling them to avoid misbehaviour or crashing the program is called as exception handling.



iNeuron Intelligence Pvt Ltd

---

**Q35. What are functions in python?**

Ans - Functions are set of instructions to perform a specific task. Below is the syntax of functions in python.

```
def function_name ([arg1, ..., argn]):  
#statements  
[return value]  
variable_name = function_name ([val1, ..., valn])
```

**Q36. What is random module?**

Answer -Python has many inbuilt packages and modules. One of the most useful modules is random. This module helps in generating random numbers.

The code given below generates a random number between x and y-1 (both inclusive) using the randrange function of the random module.

```
import random  
a=20  
b=30  
print(random. randrange (a, b))  
output:  
Any random number between 20 to 30.
```

**Q37. A class variable or instance variable that holds data associated with a class and its object is known as?**

- A. Class variable
- B. Method
- C. Operator overloading
- D. Data member





iNeuron Intelligence Pvt Ltd

---

### **Q38. What is seek() function in python?**

Answer - Python provides seek() function to navigate the file object pointer to the required position specified.

Syntax: file\_object. seek(offset, [whence])

//file\_object indicates the file object pointer to be navigated

//offset indicates which position the file object pointer is to be navigated

### **Q39. What is PEP 8??**

Ans- PEP 8 is a coding convention, a set of recommendations, about how to write your Python code more readable.

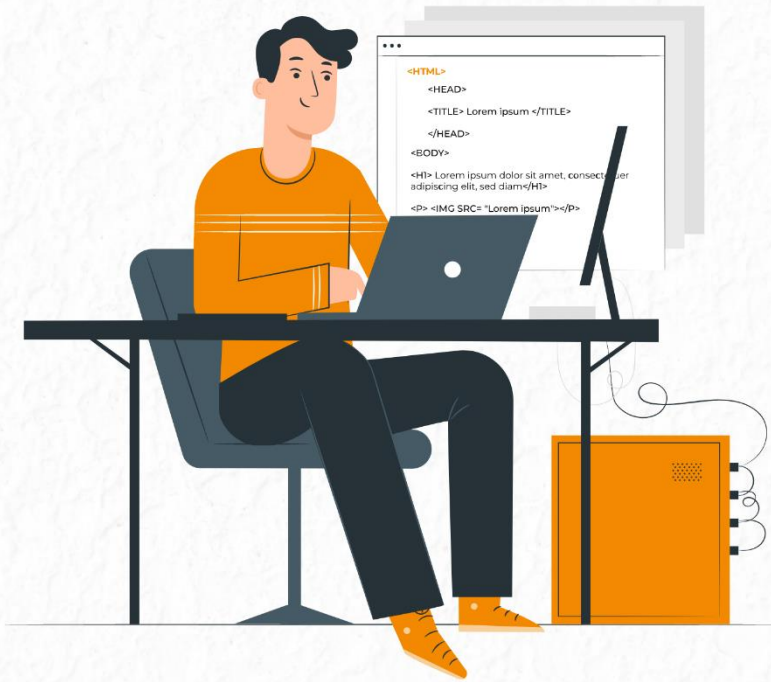
### **Q40. What is pickling and unpickling in Python?**

Ans- Pickling is a way to convert a python object (list, dict, etc.) into a character stream. Pickle has two main methods. The first one is dump, which dumps an object to a file object and the second one is load, which loads an object from a file object.

While the process of retrieving original Python objects from the stored string representation is called unpickling.

### **Q41. What is namespace in Python?**

Ans- In Python, every name introduced has a place where it lives and can be hooked for. This is known as namespace. It is like a box where a variable name is mapped to the object placed. Whenever the variable is searched out, this box will be searched, to get corresponding object.



# Python

## Interview Questions



## **Q1. What is PL/SQL?**

The Procedural Language/Structured Query Language is a programming language used to create, manage and manipulate Oracle database objects. It provides powerful procedural control of the data stored in an Oracle Database, including triggers, procedures, functions, and packages. In addition, it offers many features not available with regular SQL, such as explicit cursor handling for iterative processing, error trapping capability, and support for object-oriented programming extensions.

## **Q2. What are the different cursor attributes in PL/SQL?**

In PL/SQL, a cursor is an interface for retrieving and processing data from the database. Cursor attributes are used to manage cursors in Oracle databases. Cursor attributes give information about the execution states of SQL statements or the conditions that affect the execution of these statements within PL/SQL blocks. Some common cursor attributes in PL/SQL include %FOUND, %NOTFOUND, %ROWCOUNT, and %ISOPEN.

The first two (found and not found) indicate whether a SELECT statement successfully retrieved records as part of an explicit or implicit cursor declaration;

rowcount indicates how many rows have been selected so far;

If ISOPEN returns TRUE if a corresponding query has not yet been executed until its end or else FALSE if it already has ended with no more rows being available for selection from now on.



iNeuron Intelligence Pvt Ltd

---

### **Q3. What are the advantages of the PL/SQL in points?**

Advantages of PL/SQL:

It is faster than other conventional programming languages.

Simplified Application Development as it supports procedural, functional, and object-oriented Programming approaches.

Secure Access Management: It allows better control of data changes with the help of privileges, constraints & triggers.

Enhanced Productivity due to advanced features like Object Oriented capabilities debugging support etc.

### **Q4. How to create and use user-defined exceptions in PL/SQL?**

User-defined exceptions in PL/SQL allow developers to create custom exception handling for their applications. They can be defined using the EXCEPTION statement and raised with the RAISE statement. Once created, these user-defined exceptions can be caught within a code block using an exception handler containing WHEN clauses (for expected errors) and other statements for unexpected errors or conditions. The code then handles the error gracefully or allows it to propagate further.



iNeuron Intelligence Pvt Ltd

---

### **Q5. Explain the features of PL/SQL?**

Features of PL/SQL:

It supports both SQL (Structured Query Language) & DML (Data Manipulation commands).

Easy integration with web technology like HTML & XML via embedded call interface or external procedure feature.

Error Handling mechanism helps in the debugging process, i.e., identify code errors quickly.

Allows execution of complex business logic by means of user-defined functions without giving access to the table structure itself.

Supports package concept that enables developers separate functionality into a logical group.

Supports Procedural Programming Paradigm – conditional statements, loops, iteration, etc.

PL/SQL Features

### **Q6. Describe the compilation process for a PL/SQL program?**

The compilation process for a PL/SQL program involves converting SQL and PL/SQL code into an executable format that the Oracle Database can interpret. This is done with the help of the SQL\*Plus command-line interface, which first parses and translates all source code statements, checks syntax errors (if any), and resolves schema objects like tables & views referenced in query statements to their respective base address before finally creating a database procedure context from within which it executes those instructions.



### **Q7. What is a Rollback statement in PL/SQL?**

A PL/SQL command known as a rollback statement is used to undo database changes that have been made since the last commit or rollback was executed. It undoes all transactions that have been completed in the current session by reverting any data changes and returning to its prior state. This statement is commonly used to undo any unintentional changes or incorrect transactions that users may have committed.

Rollback statement in PL/SQL

### **Q8. How do you write a block of statements in PL/SQL?**

A block of statements in PL/SQL is a set of SQL or PL/SQL commands grouped to execute as a single unit. A basic structure for writing blocks of statements consists of declarations, an execution section, an exception-handling section and an optional end statement.

Declarations can include variable definitions, which must be on the first line followed by BEGIN keyword; this is followed by one or more lines with executable statements within it such as SELECT, INSERT etc.;

finally, any exceptions should be handled so the process completes successfully, without errors using EXCEPTION keyword before ending the block with END keyword



**Q9. How can one identify common errors while executing a set of commands using the syntaxes within a program unit defined by declarative sections, executable sections, exception handling sections, etc.?**

Common errors while executing a set of commands written using the PL/SQL syntax in a program unit can be identified by understanding its structure and meaning.

To identify common errors, users must review each line within these sections and verify their correctness with regard to data type definitions associated with existing parameters, including formal ones.;

this is also applicable when dealing with scalar datatypes for column values for tables, plus checking if correct table names have been specified, among other details and specific requirements stated beforehand, including trigger declarations, etc.

These steps will help determine where mistakes might exist, allowing issues to be addressed promptly, correcting them during the development phase itself, and preventing actual code production failures from occurring later on due to rigorous testing and earlier assessments undertaken before submission into the system environment.

**Q10. What scalar data types are available for defining variables associated with identifiers according to Oracle's specific ANSI standards-compliant PL/SQL definition?**

Scalar data types are available for defining variables associated with identifiers according to Oracle's specific ANSI standards-compliant PL/SQL definition. Scalar types, also known as elementary or primitive datatypes, represent a single value and can be either number (NUMBER), string (VARCHAR2) or date type (DATE).





**Q11. How can numeric values be manipulated using PL/SQL?**

Numeric values can be manipulated using PL/SQL in various ways. This includes performing basic arithmetic operations (such as addition, subtraction, multiplication, and division), rounding numbers to the nearest integer or decimal place, and calculating exponentials or logarithms with complex functions such as SQRT(), LOG() or POW().

**Q12. Can you explain how to use an exception block within a PL/SQL package body?**

An exception block within a PL/SQL package body can be used to handle any errors that occur during execution. It is defined by using the keyword EXCEPTION followed by one or more WHEN clause statements which check for specific error conditions and execute particular code if they are met. For example:

```
EXCEPTION WHEN OTHERS THEN INSERT INTO log_table (error)
VALUES (SQLCODE);
```

This block would catch all other exceptions not explicitly handled in an earlier statement, log them into a separate table and then continue with normal program flow. You can also define your own custom exceptions and create additional nested blocks when needed.

**Q13. What are actual parameters, and how do they work with current transactions when programming with PL/SQL?**

Actual parameters are values passed to a procedure or function during invocation. When programming with PL/SQL, they can be used within the current transaction by referencing them in SQL statements (e.g., SELECT \* FROM TABLE WHERE id =:p\_id). This enables developers to conditionally execute database operations depending on the values provided during execution.

For example: if some procedure was designed to retrieve data only from recent entries, then p\_id being not null would trigger a query that selects all records created after the given date. Having it set as NULL would cause those lines of code to be skipped altogether.





**Q14. In PL/SQL, how do you add a new table row using DML operations?**

New table rows can be added using DML (Data Manipulation Language) methods in PL/SQL. The INSERT statement, which describes the columns and values of a specific row that you want to add to your database, is used to do this.

For instance: (column1, column2,...) INSERT INTO MY TABLE VALUES ('value1', 'value2',...); This will create a new table row in "my\_table" containing the appropriate values for each provided column. You can insert several rows simultaneously using stored procedures or dynamic SQL.

**Q15. How can you use Pseudo columns to manipulate data within PL/SQL?**

Pseudo columns are virtual columns that present particular values in a SELECT statement but have no actual existence within the underlying table. In PL/SQL, pseudo-columns can manipulate data by providing additional information for operations such as sorting or replacing existing column names when selecting records from tables.

For example, you could use the ROWID pseudo-column to retrieve the unique row identifier of each record without having it explicitly defined in your SELECT

query like so: SELECT ... FROM my\_table ORDER BY ROWID;

This would result in all queried rows being sorted according to their row identifiers instead of any other previously specified order criteria.



**Q16. Explain what happens when the commit statement changes are made using DML operations within the context of a sequence of statements implemented using PL/SQL programming language?**

The commit statement is used in PL/SQL to save changes made using DML operations such as INSERT, UPDATE, and DELETE. When this commit statement is included at the end of a sequence of statements, it tells the database server that all data manipulation language (DML) commands should be permanently recorded within its tables on disk.

This means that even if an error occurs during execution or the program stops before completion for any other reason, whatever modifications have been successfully committed will remain effective and won't require manual intervention from users afterwards.

The commit statement also serves to release any resources that were previously held by a cursor or other database objects while the sequence of statements was being executed

**Q16. What is a SQL Cursor, and what purpose does it serve within the context of PL/SQL programming language?**

Within the context of PL/SQL programming language, a SQL Cursor is an internal structure used to handle and process multiple rows of data. It acts as a pointer within your program which points at each row returned from queries in sequence so that you can take appropriate action depending on the result set specified by such statement (e.g., updating records).

As a result, developers may use SQL cursors to retrieve data regularly without constantly relying on database server resources, which greatly reduces overhead when working with huge amounts of data.



iNeuron Intelligence Pvt Ltd

---

**Q17. How can you use string literals within my queries while programming with PL/SQL?**

String literals are used when writing queries in PL/SQL to represent text-based information such as names, titles and descriptions, etc., stored inside database tables or views previously by other applications using character strings instead of numerical integers and floats, etc.

You can write string literals directly into your queries simply by enclosing them within single quote marks ('): `SELECT * FROM <tablename> WHERE name = 'John Doe';`

**Q18. How can we write a multi-line comment within a PL/SQL program?**

Multi-line comments in PL/SQL are used to add remarks and notes over multiple lines of code. They begin with a double hyphen (--) followed by an asterisk (\*), then the comment text, and finally, they end with an asterisk (\*) followed by two hyphens (--).

Anything between these characters will be ignored when compiling your program or script, allowing developers to clarify their work for other readers without changing its functions.

For example, the following multi-line comment in PL/SQL explains what several lines of code do together: `--* This block checks if there is new data added to the database table since the last execution *--`



iNeuron Intelligence Pvt Ltd

---

**Q19. Can PL/SQL commands store or display graphic images?**

Yes, PL/SQL commands can store and display graphic images. Graphics files such as JPEGs or PNGs can be stored in the database within BLOB (binary large objects) fields that efficiently store digital properties. Likewise, these binary data types can then be used to output graphical elements from the results of a query via display packages like Oracle's own HTP package.

This package allows developers to store pictures directly into their application databases and then render them on webpages without any extra work by utilizing functions like `http.image` or even generating dynamic bar charts with graphs with code similar to `http.p ('');`

**Q20. How can network traffic be monitored with the help of PL/SQL commands?**

Network traffic can be monitored with the help of PL/SQL commands using database packet sniffing. Packet sniffing is a method of intercepting and analyzing network packets sent over various networks, including local area and wide-area networks.

With this technique, developers can use PL/SQL packages such as `DBMS_NETWORK_PACKET` to look at all incoming and outgoing network requests from their program or script in order to detect malicious activity or analyze performance issues like slow response times.