**Composite Data Types (Questions 1-50)**

1. Which of the following is a user-defined composite data type in PL/SQL that can hold multiple values of potentially different data types? (a) Scalar variable (b) BOOLEAN **(c) RECORD** (d) VARCHAR2 **Answer: (c)**
2. A PL/SQL TABLE is similar to which of the following data structures? (a) Linked list **(b) Array** (c) Stack (d) Queue **Answer: (b)**
3. What is the index for elements in a PL/SQL TABLE if no specific index is defined? (a) It starts from 0 and increments by 1. (b) It starts from 1 and increments by 1. **(c) It can be any scalar data type.** (d) It is implicitly defined by the order of insertion. **Answer: (c)**
4. Which of the following statements correctly declares a VARRAY named phone\_numbers that can hold up to 5 VARCHAR2(20) values? **(a) TYPE phone\_numbers IS VARRAY(5) OF VARCHAR2(20);** (b) TYPE phone\_numbers IS ARRAY(5) OF VARCHAR2(20); (c) DECLARE phone\_numbers VARRAY(5) OF VARCHAR2(20); (d) DECLARE TYPE phone\_numbers AS VARRAY(5) OF VARCHAR2(20); **Answer: (a)**
5. How do you access the third element of a VARRAY named phone\_numbers? (a) phone\_numbers[2] (b) phone\_numbers(3) **(c) phone\_numbers[3]** (d) phone\_numbers.GET(3) **Answer: (c)**
6. What is the purpose of the %TYPE attribute in PL/SQL? (a) To define a new composite data type. **(b) To declare a variable with the same data type as a database column or another variable.** (c) To handle different types of exceptions. (d) To specify the return type of a function. **Answer: (b)**
7. What is the purpose of the %ROWTYPE attribute in PL/SQL? **(a) To define a new record type based on a cursor or table structure.** (b) To declare a variable with the same data type as a single column. (c) To handle errors related to rows in a table. (d) To specify the parameter types of a procedure. **Answer: (a)**
8. Given a table named employees with columns employee\_id (NUMBER) and first\_name (VARCHAR2), how would you declare a record variable emp\_rec to hold a row from this table? (a) TYPE emp\_rec IS RECORD OF employees; (b) TYPE emp\_rec IS ROWTYPE OF employees; **(c) emp\_rec employees%ROWTYPE;** (d) DECLARE emp\_rec employees%RECORD; **Answer: (c)**
9. Which of the following is true about initializing composite data types? (a) RECORD types cannot be initialized directly during declaration. **(b) TABLE and VARRAY types must be initialized before being used.** (c) All composite data types are automatically initialized to NULL. (d) You can initialize elements of a TABLE or VARRAY using a constructor. **Answer: (b)**
10. What happens if you try to access an element in a VARRAY beyond its defined limit? (a) It returns NULL. **(b) It raises a SUBSCRIPT\_OUTSIDE\_LIMIT exception.** (c) It automatically extends the size of the VARRAY. (d) It returns the last element of the VARRAY. **Answer: (b)**
11. Which of the following is NOT a characteristic of a PL/SQL RECORD? (a) It can contain fields of different data types. (b) Its structure is user-defined. (c) You can access individual fields using dot notation. **(d) Its size is dynamically allocated.** **Answer: (d)**
12. What is the purpose of extending a PL/SQL TABLE using the EXTEND method? **(a) To add more elements to the table.** (b) To retrieve the last element of the table. (c) To remove elements from the table. (d) To check if the table is empty. **Answer: (a)**
13. What is the return type of the COUNT method for a PL/SQL TABLE or VARRAY? **(a) NUMBER** (b) BOOLEAN (c) INTEGER (d) PLS\_INTEGER **Answer: (a)**
14. Which of the following is the correct way to declare and initialize a PL/SQL TABLE of VARCHAR2(30) indexed by BINARY\_INTEGER? (a) TYPE name\_table IS TABLE OF VARCHAR2(30); name\_table nt; **(b) TYPE name\_table IS TABLE OF VARCHAR2(30) INDEX BY BINARY\_INTEGER; nt name\_table;** (c) DECLARE name\_table TABLE OF VARCHAR2(30) INDEX BY BINARY\_INTEGER; (d) TYPE name\_table IS ARRAY OF VARCHAR2(30) INDEX BY BINARY\_INTEGER; nt name\_table; **Answer: (b)**
15. How do you remove all elements from a PL/SQL TABLE? (a) DELETE nt; **(b) nt.DELETE;** (c) REMOVE nt; (d) nt.TRUNCATE; **Answer: (b)**
16. What is the significance of the index in an associative array (TABLE indexed by a scalar type other than BINARY\_INTEGER)? **(a) It acts as a key to access the elements.** (b) It determines the order of elements. (c) It specifies the maximum number of elements. (d) It indicates the data type of the elements. **Answer: (a)**
17. Which of the following is NOT a method associated with PL/SQL TABLEs and VARRAYs? (a) COUNT (b) EXISTS (c) EXTEND **(d) SORT** **Answer: (d)**
18. What does the EXISTS(n) method of a TABLE or VARRAY return? (a) The value of the element at index n. **(b) TRUE if an element exists at index n, FALSE otherwise.** (c) The total number of elements up to index n. (d) The index of the first occurrence of the value n. **Answer: (b)**
19. When would you typically use a VARRAY over a TABLE? **(a) When you know the maximum number of elements beforehand and the order is important.** (b) When you need a flexible size and non-integer indexing. (c) When you need to store a large number of records. (d) When you need to perform frequent insertions and deletions at arbitrary positions. **Answer: (a)**
20. What is the initial size of a VARRAY after declaration but before initialization? **(a) NULL** (b) 0 (c) The maximum size defined during declaration. (d) 1 **Answer: (a)**
21. How do you initialize a VARRAY during declaration? **(a) Using a constructor.** (b) By assigning values to individual elements directly. (c) Using the EXTEND method. (d) It cannot be initialized during declaration. **Answer: (a)**
22. What happens if you assign a NULL value to a field in a RECORD? **(a) The field holds a NULL value.** (b) It raises an exception. (c) The field is set to its default value (if any). (d) The entire record becomes NULL. **Answer: (a)**
23. Can you compare two RECORD variables for equality? **(a) Yes, if they are of the same RECORD type.** (b) No, direct comparison is not allowed. (c) Only if all fields are of scalar data types. (d) Only if they have been explicitly initialized. **Answer: (a)**
24. What is the purpose of the TRIM method for a PL/SQL TABLE or VARRAY? (a) To remove leading and trailing spaces from string elements. **(b) To remove elements from the end of the collection.** (c) To remove specific elements based on their value. (d) To reduce the allocated memory for the collection. **Answer: (b)**
25. How does TRIM(n) differ from TRIM without any argument for a collection? **(a) TRIM(n) removes the last n elements, while TRIM removes only the last element.** (b) TRIM(n) removes the first n elements, while TRIM removes the last element. (c) TRIM(n) removes elements with the value n, while TRIM removes NULL elements. (d) There is no difference. **Answer: (a)**
26. What is the return type of the FIRST method for a PL/SQL TABLE or VARRAY? **(a) The index of the first element.** (b) The value of the first element. (c) BOOLEAN (TRUE if the collection is not empty). (d) NUMBER (1 if the collection is not empty, NULL otherwise). **Answer: (a)**
27. What is the return type of the LAST method for a PL/SQL TABLE or VARRAY? **(a) The index of the last element.** (b) The value of the last element. (c) BOOLEAN (TRUE if the collection is not empty). (d) NUMBER (the total number of elements). **Answer: (a)**
28. What is the return type of the NEXT(index) method for a PL/SQL TABLE? **(a) The index of the next element after the given index.** (b) The value of the next element. (c) BOOLEAN (TRUE if there is a next element). (d) NUMBER (the difference between the current and next index). **Answer: (a)**
29. What is the return type of the PRIOR(index) method for a PL/SQL TABLE? **(a) The index of the element before the given index.** (b) The value of the previous element. (c) BOOLEAN (TRUE if there is a previous element). (d) NUMBER (the difference between the current and previous index). **Answer: (a)**
30. Can you use a PL/SQL TABLE or VARRAY as a parameter to a stored procedure or function? **(a) Yes, they can be passed as IN, OUT, or IN OUT parameters.** (b) No, only scalar data types are allowed as parameters. (c) Only if they are declared as global types. (d) Only if the procedure or function is in the same PL/SQL block. **Answer: (a)**
31. Can you store SQL query results directly into a PL/SQL TABLE of RECORDS using a single statement? **(a) Yes, using the BULK COLLECT INTO clause with a cursor.** (b) No, you have to fetch row by row in a loop. (c) Only if the table structure exactly matches the query result. (d) Only for simple queries with a single column. **Answer: (a)**
32. What is the difference between a nested table and a VARRAY in terms of storage? **(a) Nested tables can be sparse and are stored out-of-line if large, while VARRAYs are always dense and stored in-line (up to a certain size).** (b) VARRAYs can be sparse, while nested tables are always dense. (c) There is no significant difference in their storage. (d) Nested tables are stored in temporary segments, while VARRAYs are permanent. **Answer: (a)**
33. When would you prefer a nested table over a VARRAY? **(a) When the number of elements is not known beforehand or can vary significantly.** (b) When the order of elements is critical and the size is fixed. (c) When you need faster access to elements by index. (d) When you are working with very small collections. **Answer: (a)**
34. What is the data type of the index for a TABLE declared as TYPE my\_table IS TABLE OF VARCHAR2(50);? **(a) BINARY\_INTEGER** (b) NUMBER (c) PLS\_INTEGER (d) VARCHAR2 **Answer: (a)**
35. How do you check if a specific index exists in a PL/SQL TABLE? **(a) Using the EXISTS(index) method.** (b) By iterating through the table and checking the index. (c) By using the COUNT method and comparing with the index. (d) It's not possible to directly check if an index exists. **Answer: (a)**
36. What happens if you try to delete an element at an index that does not exist in a PL/SQL TABLE? **(a) Nothing happens; the DELETE operation is ignored for that index.** (b) It raises an exception. (c) The table is cleared. (d) The closest existing element is deleted instead. **Answer: (a)**
37. Can you have a TABLE of RECORDS where each RECORD has different fields? **(a) No, all records in a TABLE must have the same structure (defined by a RECORD type).** (b) Yes, PL/SQL allows heterogeneous tables. (c) Only if the different fields are all of compatible data types. (d) Only in certain versions of PL/SQL. **Answer: (a)**
38. What is the maximum number of elements a VARRAY can hold? **(a) Defined during the VARRAY type declaration.** (b) Limited by the available memory. (c) 2 billion (like PL/SQL tables). (d) 32767. **Answer: (a)**
39. What is the maximum number of elements a PL/SQL TABLE can effectively hold (theoretically)? **(a) Limited by available memory, practically up to the limits of BINARY\_INTEGER indexing.** (b) Defined during the TABLE type declaration. (c) A fixed number like 65535. (d) Smaller than VARRAYs. **Answer: (a)**
40. When comparing a NULL VARRAY with another VARRAY, what is the result of the equality comparison? **(a) NULL** (b) TRUE if the other VARRAY is also NULL, FALSE otherwise. (c) FALSE (d) It raises an exception. **Answer: (b)**
41. How do you create a copy of a PL/SQL TABLE? **(a) By assigning one table variable to another.** (b) Using a COPY method (which doesn't exist directly). (c) By iterating through the original table and adding elements to the new table. (d) Using the CLONE method (which doesn't exist directly). **Answer: (a)**
42. What happens to the indices of a PL/SQL TABLE after deleting some elements? **(a) The existing indices remain the same; there might be gaps.** (b) The indices are automatically renumbered to be contiguous. (c) It depends on which DELETE method is used. (d) The table becomes NULL. **Answer: (a)**
43. Can you use a cursor to populate a PL/SQL TABLE of RECORDS? **(a) Yes, using BULK COLLECT INTO.** (b) No, cursors can only fetch into scalar variables or single records. (c) Only if the cursor returns a single column. (d) Only with explicit loops and row-by-row fetching. **Answer: (a)**
44. What is the difference between %TYPE and %ROWTYPE when declaring variables based on a cursor? **(a) %TYPE is used for a single column, while %ROWTYPE is for an entire row (a RECORD).** (b) %ROWTYPE is used for a single column, while %TYPE is for an entire row. (c) %TYPE is used for scalar variables, while %ROWTYPE is for composite variables. (d) There is no functional difference. **Answer: (a)**
45. Which of the following is a valid index type for an associative array? (a) BOOLEAN (b) NUMBER **(c) VARCHAR2** (d) DATE **Answer: (c)**
46. What is the purpose of the SET operator when working with collections (though not a direct PL/SQL method

Working with

Composite Data Types 20 . Using Explicit Cursors 20. Handling Exceptions 20. Introducing Stored Procedures and Functions 20 MAKE PL/SQL MCQ  GIVE ANSWER

Okay, here are 80 PL/SQL MCQs (20 for each topic) with answers:

**Composite Data Types (Questions 1-20)**

1. Which of the following is a user-defined composite data type in PL/SQL that can hold multiple values of potentially different data types? (a) Scalar variable (b) BOOLEAN **(c) RECORD** (d) VARCHAR2 **Answer: (c)**
2. A PL/SQL TABLE is similar to which of the following data structures? (a) Linked list **(b) Array** (c) Stack (d) Queue **Answer: (b)**
3. What is the index for elements in a PL/SQL TABLE if no specific index is defined? (a) It starts from 0 and increments by 1. (b) It starts from 1 and increments by 1. **(c) It can be any scalar data type.** (d) It is implicitly defined by the order of insertion. **Answer: (c)**
4. Which of the following statements correctly declares a VARRAY named phone\_numbers that can hold up to 5 VARCHAR2(20) values? **(a) TYPE phone\_numbers IS VARRAY(5) OF VARCHAR2(20);** (b) TYPE phone\_numbers IS ARRAY(5) OF VARCHAR2(20); (c) DECLARE phone\_numbers VARRAY(5) OF VARCHAR2(20); (d) DECLARE TYPE phone\_numbers AS VARRAY(5) OF VARCHAR2(20); **Answer: (a)**
5. How do you access the third element of a VARRAY named phone\_numbers? (a) phone\_numbers[2] (b) phone\_numbers(3) **(c) phone\_numbers[3]** (d) phone\_numbers.GET(3) **Answer: (c)**
6. What is the purpose of the %TYPE attribute in PL/SQL? (a) To define a new composite data type. **(b) To declare a variable with the same data type as a database column or another variable.** (c) To handle different types of exceptions. (d) To specify the return type of a function. **Answer: (b)**
7. What is the purpose of the %ROWTYPE attribute in PL/SQL? **(a) To define a new record type based on a cursor or table structure.** (b) To declare a variable with the same data type as a single column. (c) To handle errors related to rows in a table. (d) To specify the parameter types of a procedure. **Answer: (a)**
8. Given a table named employees with columns employee\_id (NUMBER) and first\_name (VARCHAR2), how would you declare a record variable emp\_rec to hold a row from this table? (a) TYPE emp\_rec IS RECORD OF employees; (b) TYPE emp\_rec IS ROWTYPE OF employees; **(c) emp\_rec employees%ROWTYPE;** (d) DECLARE emp\_rec employees%RECORD; **Answer: (c)**
9. Which of the following is true about initializing composite data types? (a) RECORD types cannot be initialized directly during declaration. **(b) TABLE and VARRAY types must be initialized before being used.** (c) All composite data types are automatically initialized to NULL. (d) You can initialize elements of a TABLE or VARRAY using a constructor. **Answer: (b)**
10. What happens if you try to access an element in a VARRAY beyond its defined limit? (a) It returns NULL. **(b) It raises a SUBSCRIPT\_OUTSIDE\_LIMIT exception.** (c) It automatically extends the size of the VARRAY. (d) It returns the last element of the VARRAY. **Answer: (b)**
11. Which of the following is NOT a characteristic of a PL/SQL RECORD? (a) It can contain fields of different data types. (b) Its structure is user-defined. (c) You can access individual fields using dot notation. **(d) Its size is dynamically allocated.** **Answer: (d)**
12. What is the purpose of extending a PL/SQL TABLE using the EXTEND method? **(a) To add more elements to the table.** (b) To retrieve the last element of the table. (c) To remove elements from the table. (d) To check if the table is empty. **Answer: (a)**
13. What is the return type of the COUNT method for a PL/SQL TABLE or VARRAY? **(a) NUMBER** (b) BOOLEAN (c) INTEGER (d) PLS\_INTEGER **Answer: (a)**
14. Which of the following is the correct way to declare and initialize a PL/SQL TABLE of VARCHAR2(30) indexed by BINARY\_INTEGER? (a) TYPE name\_table IS TABLE OF VARCHAR2(30); name\_table nt; **(b) TYPE name\_table IS TABLE OF VARCHAR2(30) INDEX BY BINARY\_INTEGER; nt name\_table;** (c) DECLARE name\_table TABLE OF VARCHAR2(30) INDEX BY BINARY\_INTEGER; (d) TYPE name\_table IS ARRAY OF VARCHAR2(30) INDEX BY BINARY\_INTEGER; nt name\_table; **Answer: (b)**
15. How do you remove all elements from a PL/SQL TABLE? (a) DELETE nt; **(b) nt.DELETE;** (c) REMOVE nt; (d) nt.TRUNCATE; **Answer: (b)**
16. What is the significance of the index in an associative array (TABLE indexed by a scalar type other than BINARY\_INTEGER)? **(a) It acts as a key to access the elements.** (b) It determines the order of elements. (c) It specifies the maximum number of elements. (d) It indicates the data type of the elements. **Answer: (a)**
17. Which of the following is NOT a method associated with PL/SQL TABLEs and VARRAYs? (a) COUNT (b) EXISTS (c) EXTEND **(d) SORT** **Answer: (d)**
18. What does the EXISTS(n) method of a TABLE or VARRAY return? (a) The value of the element at index n. **(b) TRUE if an element exists at index n, FALSE otherwise.** (c) The total number of elements up to index n. (d) The index of the first occurrence of the value n. **Answer: (b)**
19. When would you typically use a VARRAY over a TABLE? **(a) When you know the maximum number of elements beforehand and the order is important.** (b) When you need a flexible size and non-integer indexing. (c) When you need to store a large number of records. (d) When you need to perform frequent insertions and deletions at arbitrary positions. **Answer: (a)**
20. What is the initial size of a VARRAY after declaration but before initialization? **(a) NULL** (b) 0 (c) The maximum size defined during declaration. (d) 1 **Answer: (a)**

**Using Explicit Cursors (Questions 21-40)**

1. Which step is mandatory when working with an explicit cursor? (a) Declaring a cursor variable. (b) Fetching data into a variable. (c) Closing the cursor. **(d) All of the above.** **Answer: (d)**
2. What does the %ISOPEN attribute of a cursor indicate? (a) Whether the cursor has fetched all rows. **(b) Whether the cursor is currently open.** (c) The number of rows fetched by the cursor. (d) Whether the last fetch operation was successful. **Answer: (b)**
3. What does the %FOUND attribute of a cursor return after a FETCH operation? (a) The number of rows fetched. **(b) TRUE if a row was fetched, FALSE otherwise.** (c) TRUE if the cursor is open, FALSE otherwise. (d) The value of the first column fetched. **Answer: (b)**
4. What does the %NOTFOUND attribute of a cursor return? **(a) The opposite of %FOUND.** (b) The number of rows remaining to be fetched. (c) TRUE if no rows were found by the cursor's query. (d) TRUE if the cursor has not been opened yet. **Answer: (a)**
5. Which statement retrieves the next row from an open cursor into specified variables or a record? (a) GET (b) NEXT **(c) FETCH** (d) RETRIEVE **Answer: (c)**
6. Why should you always close an explicit cursor after you are finished with it? **(a) To release the resources associated with the cursor.** (b) To prevent data inconsistencies. (c) To ensure that subsequent operations on the cursor are valid. (d) All of the above. **Answer: (a)**
7. Can you use a FOR UPDATE clause with an explicit cursor? If so, what is its purpose? (a) No, FOR UPDATE can only be used in implicit cursors. **(b) Yes, to lock the rows retrieved by the cursor for subsequent updates or deletions.** (c) Yes, to automatically update the fetched rows with new values. (d) Yes, to optimize the performance of the cursor. **Answer: (b)**
8. If you declare a cursor with parameters, when do you provide the values for these parameters? (a) During cursor declaration. **(b) When you open the cursor.** (c) During the FETCH operation. (d) When you close the cursor. **Answer: (b)**
9. What happens if you try to fetch from a cursor that has not been opened? **(a) It raises an INVALID\_CURSOR exception.** (b) It returns NULL values. (c) It automatically opens the cursor. (d) It does nothing and proceeds to the next statement. **Answer: (a)**
10. Which of the following is an advantage of using explicit cursors over implicit cursors for multi-row queries? (a) Implicit cursors are more efficient for large datasets. **(b) Explicit cursors provide more control over the processing of each row.** (c) Implicit cursors automatically handle exceptions. (d) Explicit cursors are simpler to write for basic queries. **Answer: (b)**
11. What is the purpose of the CURSOR keyword in the DECLARE section? **(a) To define a named cursor associated with a SQL statement.** (b) To open an existing cursor. (c) To fetch data from a cursor. (d) To close a cursor. **Answer: (a)**
12. What is the scope of an explicitly declared cursor? **(a) The PL/SQL block in which it is declared.** (b) The entire database session. (c) All sub-blocks within the declaring block. (d) Only the BEGIN and EXCEPTION sections of the declaring block. **Answer: (a)**
13. Which of the following is the correct syntax to declare a cursor named dept\_cursor that selects all rows from the departments table? **(a) CURSOR dept\_cursor IS SELECT \* FROM departments;** (b) DECLARE CURSOR dept\_cursor SELECT \* FROM departments; (c) TYPE dept\_cursor IS CURSOR SELECT \* FROM departments; (d) CURSOR dept\_cursor = SELECT \* FROM departments; **Answer: (a)**
14. What happens if a cursor's SELECT statement returns no rows? (a) The cursor attributes %FOUND and %NOTFOUND are never set. (b) The first FETCH operation will set %FOUND to FALSE and %NOTFOUND to TRUE. (c) It raises a NO\_DATA\_FOUND exception when the cursor is opened. (d) The cursor cannot be opened. **Answer: (b)**
15. Can you use a cursor within another cursor (nested cursors)? **(a) Yes, this is possible and can be useful for processing related data.** (b) No, cursors cannot be nested. (c) Only if the outer cursor is a FOR loop cursor. (d) Only within stored procedures. **Answer: (a)**
16. What is a cursor FOR loop in PL/SQL? **(a) A simplified way to iterate through the rows of a cursor, automatically opening, fetching, and closing the cursor.** (b) A loop that can only be used with implicit cursors. (c) A loop that requires manual opening, fetching, and closing of the cursor. (d) A loop that iterates a fixed number of times based on the cursor's query. **Answer: (a)**
17. Which of the following is the correct syntax for a cursor FOR loop using the cursor emp\_cursor? **(a) FOR emp\_rec IN emp\_cursor LOOP ... END LOOP;** (b) LOOP FETCH emp\_cursor INTO emp\_rec; EXIT WHEN emp\_cursor%NOTFOUND; ... END LOOP; (c) WHILE emp\_cursor%FOUND LOOP FETCH emp\_cursor INTO emp\_rec; ... END LOOP; (d) FOR EACH emp\_rec IN emp\_cursor LOOP ... END LOOP; **Answer: (a)**
18. What happens implicitly when a cursor FOR loop is executed? (a) The cursor is only opened once at the beginning of the loop. (b) The cursor is opened and closed for each iteration. **(c) The cursor is opened before the loop starts and closed automatically after the loop finishes.** (d) The cursor must be opened and closed manually within the loop. **Answer: (c)**
19. Can you pass parameters to a cursor used in a cursor FOR loop? **(a) Yes, the cursor must be declared with parameters, and the loop will use the same parameter values for all iterations.** (b) No, cursor FOR loops cannot use parameterized cursors. (c) Yes, you can specify different parameter values for each iteration of the loop. (d) Only if the parameters are IN OUT parameters. **Answer: (a)**
20. What is the primary advantage of using a cursor FOR loop? **(a) Reduced code complexity and automatic cursor management.** (b) Faster execution speed compared to manual cursor handling. (c) More control over the fetch operations. (d) Ability to use different parameter values for each row. **Answer: (a)**

**Handling Exceptions (Questions 41-60)**

1. What is an exception in PL/SQL? (a) A normal condition that occurs during program execution. **(b) An error condition that occurs during program execution.** (c) A user-defined data type for error handling. (d) A block of code that is always executed. **Answer: (b)**
2. Which of the following is a predefined exception in PL/SQL? (a) USER\_ERROR **(b) DIVISION\_BY\_ZERO** (c) FILE\_NOT\_FOUND\_ERROR (d) CUSTOM\_EXCEPTION **Answer: (b)**
3. Which section of a PL/SQL block is used to handle exceptions? (a) DECLARE (b) BEGIN **(c) EXCEPTION** (d) END **Answer: (c)**
4. What happens if an exception is raised in the BEGIN block and there is no corresponding handler in the EXCEPTION block? (a) The program continues execution from the next statement. **(b) The PL/SQL block terminates, and the exception is propagated to the calling environment.** (c) The system automatically tries to handle the exception. (d) A warning message is displayed, but the program continues. **Answer: (b)**
5. Which keyword is used to define a user-defined exception? **(a) EXCEPTION** (b) DECLARE EXCEPTION (c) RAISE EXCEPTION (d) DEFINE EXCEPTION **Answer: (a)**
6. How do you raise a user-defined exception named invalid\_operation? (a) THROW invalid\_operation; (b) SIGNAL invalid\_operation; **(c) RAISE invalid\_operation;** (d) EXCEPTION invalid\_operation; **Answer: (c)**
7. What is the purpose of the WHEN OTHERS exception handler? (a) To handle only predefined exceptions. **(b) To handle all exceptions that are not explicitly handled by other handlers.** (c) To handle only user-defined exceptions. (d) To specify actions to be taken if no exception occurs

MORE NEED

Okay, let's keep the PL/SQL MCQ train rolling! Here are the remaining questions to bring the total to 80, focusing on Handling Exceptions and Introducing Stored Procedures and Functions:

**Handling Exceptions (Questions 48-60)**

1. Which function can be used within an exception handler to retrieve the error code of the most recent exception? (a) SQLERRM **(b) SQLCODE** (c) ERROR\_CODE (d) EXCEPTION\_CODE **Answer: (b)**
2. Which function can be used within an exception handler to retrieve the error message of the most recent exception? (a) SQLCODE **(b) SQLERRM** (c) ERROR\_MESSAGE (d) EXCEPTION\_MESSAGE **Answer: (b)**
3. Can you have multiple exception handlers in a single EXCEPTION block? (a) No, only one WHEN OTHERS handler is allowed. **(b) Yes, you can have multiple WHEN clauses to handle specific exceptions.** (c) Only if the exceptions are related to the same type of error. (d) Only in nested PL/SQL blocks. **Answer: (b)**
4. What is the order in which exception handlers are evaluated? (a) Random order. **(b) The order in which they are defined in the EXCEPTION block.** (c) Alphabetical order of the exception names. (d) Based on the severity of the exception. **Answer: (b)**
5. What happens if an exception is raised within an exception handler? (a) The program terminates immediately. (b) The outer exception handler is executed. (c) The exception is ignored. **(d) An unhandled exception error is raised.** **Answer: (d)**
6. What is the purpose of the RAISE statement without any exception name within an exception handler? (a) To clear the current exception. **(b) To re-raise the currently active exception.** (c) To raise a generic OTHERS exception. (d) To exit the PL/SQL block. **Answer: (b)**
7. Which of the following is NOT a common use case for exception handling? (a) Providing user-friendly error messages. (b) Preventing program termination due to errors. **(c) Implementing complex business logic.** (d) Performing cleanup operations (e.g., closing cursors, releasing locks). **Answer: (c)**
8. What is the scope of an exception declared in the DECLARE section of a PL/SQL block? (a) It is global to the entire database. **(b) It is local to the PL/SQL block where it is declared.** (c) It is accessible in all sub-blocks within the PL/SQL block. (d) It is only accessible within the EXCEPTION block. **Answer: (b)**
9. What is the difference between predefined and user-defined exceptions in terms of how they are raised? (a) Predefined exceptions are raised explicitly using RAISE, user-defined are implicit. **(b) Predefined exceptions are raised implicitly by the system, user-defined are raised explicitly using RAISE.** (c) There is no difference in how they are raised. (d) Predefined exceptions cannot be raised explicitly. **Answer: (b)**
10. Which predefined exception is raised when a SELECT INTO statement returns no rows? **(a) NO\_DATA\_FOUND** (b) TOO\_MANY\_ROWS (c) INVALID\_CURSOR (d) ZERO\_DIVIDE **Answer: (a)**
11. Which predefined exception is raised when a SELECT INTO statement returns more than one row? (a) NO\_DATA\_FOUND **(b) TOO\_MANY\_ROWS** (c) INVALID\_CURSOR (d) DUP\_VAL\_ON\_INDEX **Answer: (b)**
12. Which predefined exception is raised when you try to perform an operation on a cursor that is not in the correct state (e.g., fetching from a closed cursor)? (a) NO\_DATA\_FOUND (b) TOO\_MANY\_ROWS **(c) INVALID\_CURSOR** (d) CURSOR\_ALREADY\_OPEN **Answer: (c)**
13. What is the purpose of the PRAGMA EXCEPTION\_INIT statement? **(a) To associate a user-defined exception name with a specific Oracle error number.** (b) To declare a predefined exception. (c) To raise an exception with a specific error number. (d) To ignore a specific Oracle error. **Answer: (a)**

**Introducing Stored Procedures and Functions (Questions 61-80)**

1. What is a stored procedure in PL/SQL? **(a) A block of PL/SQL code that can be executed by name and may accept parameters.** (b) A database table that stores PL/SQL code. (c) A predefined function in the Oracle database. (d) A trigger that executes automatically in response to database events. **Answer: (a)**
2. What is a stored function in PL/SQL? **(a) A block of PL/SQL code that can be executed by name, may accept parameters, and must return a value.** (b) A database view that contains PL/SQL logic. (c) A built-in operator in SQL. (d) A package that groups related procedures and functions. **Answer: (a)**
3. Which SQL statement is used to create a stored procedure? (a) CREATE FUNCTION **(b) CREATE PROCEDURE** (c) DEFINE PROCEDURE (d) MAKE PROCEDURE **Answer: (b)**
4. Which SQL statement is used to create a stored function? (a) CREATE PROCEDURE **(b) CREATE FUNCTION** (c) DEFINE FUNCTION (d) MAKE FUNCTION **Answer: (b)**
5. How do you execute a stored procedure named calculate\_bonus? (a) SELECT calculate\_bonus(); **(b) EXECUTE calculate\_bonus;** (c) CALL calculate\_bonus(); (d) RUN calculate\_bonus; **Answer: (b)**
6. How do you call a stored function named get\_employee\_name that takes an employee ID as input and returns the employee name? (a) EXECUTE get\_employee\_name(100); (b) CALL get\_employee\_name(100); **(c) SELECT get\_employee\_name(100) FROM dual;** (d) RUN get\_employee\_name(100); **Answer: (c)**
7. What are IN parameters in a stored procedure or function? (a) Parameters that are used to return values from the procedure or function. **(b) Parameters that pass values into the procedure or function.** (c) Parameters that can both pass values in and return values out. (d) Optional parameters that may or may not be provided when calling the procedure or function. **Answer: (b)**
8. What are OUT parameters in a stored procedure? (a) Parameters that pass values into the procedure. **(b) Parameters that are used to return one or more values from the procedure.** (c) Parameters that are used internally within the procedure. (d) Parameters that are automatically assigned default values. **Answer: (b)**
9. What are IN OUT parameters in a stored procedure or function? (a) Parameters that can only return values. (b) Parameters that can only accept input values. **(c) Parameters that can both accept an initial value and return a modified value.** (d) Parameters that are used for error handling. **Answer: (c)**
10. What is the purpose of the RETURN statement in a stored function? (a) To exit the function without returning a value. (b) To specify the data type of the value that the function will return. **(c) To return a value from the function to the calling environment.** (d) To handle exceptions within the function. **Answer: (c)**
11. Can a stored procedure return a value directly like a function? (a) Yes, using the RETURN statement. (b) Yes, by assigning a value to the procedure name. **(c) No, procedures typically return values through OUT or IN OUT parameters.** (d) Only if the procedure is defined with a return type. **Answer: (c)**
12. What are the advantages of using stored procedures and functions? (a) Improved performance due to reduced network traffic. (b) Enhanced security through controlled data access. (c) Increased code reusability and maintainability. **(d) All of the above.** **Answer: (d)**
13. Where are stored procedures and functions stored in the Oracle database? (a) In the client's local file system. (b) Within individual database tables. **(c) In the data dictionary of the database schema where they are created.** (d) In a separate server dedicated to PL/SQL code. **Answer: (c)**
14. Which privilege is required to create a stored procedure or function in your own schema? (a) EXECUTE ANY PROCEDURE **(b) CREATE PROCEDURE** (c) CREATE ANY PROCEDURE (d) CREATE FUNCTION **Answer: (b)**
15. Which privilege is required to execute a stored procedure or function owned by another user? (a) CREATE PROCEDURE (b) EXECUTE ANY PROCEDURE **(c) EXECUTE ON object\_name TO user\_name** (d) USAGE ON PROCEDURE **Answer: (c)**
16. What is the purpose of the CREATE OR REPLACE clause when creating stored procedures and functions? **(a) To update the definition if the object already exists, or create it if it doesn't.** (b) To create a copy of an existing procedure or function. (c) To specify that the object should be created in a different schema. (d) To add comments to the procedure or function definition. **Answer: (a)**
17. Can a stored procedure call another stored procedure or function? **(a) Yes, this allows for modularization of complex logic.** (b) No, procedures cannot call other PL/SQL blocks. (c) Only if they are in the same package. (d) Only if the called procedure or function is owned by the same user. **Answer: (a)**
18. Can a stored function call a stored procedure? **(a) Generally discouraged and subject to restrictions due to potential side effects within a function called from SQL.** (b) Yes, without any restrictions. (c) Only if the procedure does not modify database state. (d) Only within PL/SQL blocks, not directly from SQL. **Answer: (a)**
19. How do you remove a stored procedure named calculate\_bonus from the database? (a) DELETE PROCEDURE calculate\_bonus; (b) REMOVE calculate\_bonus; **(c) DROP PROCEDURE calculate\_bonus;** (d) ERASE PROCEDURE calculate\_bonus; **Answer: (c)**
20. How do you remove a stored function named get\_employee\_name from the database? (a) DELETE FUNCTION get\_employee\_name; (b) REMOVE get\_employee\_name; **(c) DROP FUNCTION get\_employee\_name;** (d) ERASE FUNCTION get\_employee\_name; **Answer: (c)**