> *Stored procedures are pre-compiled modules containing SQL statements and logic stored on the database server.*

They can be invoked multiple times with different inputs, improving performance and code reuse.

Parameters play a crucial role in passing data to and from stored procedures.

Types of Parameters

- IN parameter: Used to pass data from the calling program to the stored procedure. The value is assigned to the parameter before the procedure is executed.
- OUT parameter: Used to return data from the stored procedure to the calling program. The procedure assigns a value to the parameter before it finishes executing.
- IN OUT parameter: Combines the functionality of both IN and OUT parameters. It allows passing data to the procedure and receiving data back after execution.

Usage of Parameters

Parameters are declared within the stored procedure definition using the syntax:

```
CREATE OR REPLACE PROCEDURE procedure_name (
  parameter_name1 IN parameter_type1,
  parameter_name2 OUT parameter_type2,
  ...
)
```

```
...
```

- parameter_type specifies the data type of the parameter.
- IN, OUT, or IN OUT keywords indicate the parameter mode.

In the procedure body, parameters can be accessed using the parameter_name within SQL statements and logic.

Benefits of using parameters

- Increased security: Prevents SQL injection attacks by separating data and code.
- Improved performance: Avoids recompiling the entire procedure for different input values.
- Enhanced code clarity: Makes the procedure logic easier to understand and maintain.
- Flexibility: Allows passing different data sets to the same procedure.

Examples of using parameters in stored procedures

Calculating a sum

```
CREATE OR REPLACE PROCEDURE calculate_sum (
  number1 IN NUMBER,
  number2 IN NUMBER,
  result OUT NUMBER
)
BEGIN
  result := number1 + number2;
END;
```

Related posts:

1. SQL Functions
2. History of DBMS
3. Introduction to DBMS
4. Introduction to Database
5. Advantages and Disadvantages of DBMS
6. SQL | DDL, DML, DCL Commands
7. Domain
8. Entity and Attribute
9. Relationship among entities
10. Attribute
11. Database Relation
12. DBMS Keys
13. Schema
14. Twelve rules of CODD
15. Normalization
16. Functional Dependency
17. Transaction processing concepts
18. Schedules
19. Serializability
20. OODBMS vs RDBMS
21. RDBMS
22. SQL Join
23. SQL Functions
24. Trigger
25. Oracle cursor

26. Introduction to Concurrency control
27. Net 11
28. NET 3
29. NET 2
30. GATE, AVG function and join DBMS | Prof. Jayesh Umre
31. GATE 2014 DBMS FIND Maximum number of Super keys | Prof. Jayesh Umre
32. GATE 2017 DBMS Query | Prof. Jayesh Umre
33. Data types
34. Entity
35. Check Constraint
36. Primary and Foreign key
37. SQL join
38. DDLDMLDCL
39. Database applications
40. Disadvantages of file system data management
41. RGPV DBMS Explain the concepts of generalization and aggregation with appropriate examples
42. RGPV solved Database approach vs Traditional file accessing approach
43. Find all employees who live in the city where the company for which they work is located
44. Concept of table spaces, segments, extents and block
45. Triggers: mutating errors, instead of triggers
46. Dedicated Server vs Multi-Threaded Server
47. Distributed database, database links, and snapshot
48. RDBMS Security
49. SQL queries for various join types
50. Cursor management: nested and parameterized cursors

51. Oracle exception handling mechanism