

Table of Contents



- 1. Exceptions
- 2. Exception Types
- 3. Exception Blocks
- 4. Exception Handling Flow
- 5. Additional Features
- Benefits of using Oracle Exception Handling
- Related posts:

Oracle provides a robust exception handling mechanism to manage errors and unexpected situations during program execution.

This mechanism consists of the following key elements:

1. Exceptions

- Exceptions are events that signal abnormal conditions during program execution.
- They can be raised explicitly by the program using the RAISE statement or implicitly by the Oracle server itself due to various conditions, such as invalid data, insufficient privileges, or network errors.

2. Exception Types

There are two main types of exceptions in Oracle:

- **Predefined Exceptions:** These are built-in exceptions provided by Oracle for common errors. Examples include NO_DATA_FOUND and INVALID_NUMBER.
- **User-Defined Exceptions:** These are custom exceptions defined by the programmer to handle specific errors in their application.

3. Exception Blocks

- Exception blocks are code sections following the main program code that handle specific exceptions.
- They can be defined using the following keywords:
 - DECLARE: Declares local variables and exception handlers.
 - BEGIN: Contains the main program code.
 - EXCEPTION: Defines exception handlers for expected exceptions.
 - WHEN: Specifies the exception type and the corresponding code to execute when the exception occurs.
 - END: Ends the exception block.

4. Exception Handling Flow

1. The program executes the main code section.
2. If an exception occurs, the program searches for a matching exception handler in the exception block.
3. If a matching handler is found, the code in the handler block is executed, and the program continues execution after the exception block.
4. If no matching handler is found, the exception propagates to the calling environment, potentially terminating the program.

5. Additional Features

- Re-Raising Exceptions: You can re-raise an exception using the RAISE statement within an exception handler to propagate it to a higher level.
- Logging and Rollback: Exception handlers can be used to log the error details and

perform rollback operations to undo any partially completed transactions in case of an error.

- Custom Message and Code: You can define custom messages and error codes for user-defined exceptions to provide more context and information about the error.

Benefits of using Oracle Exception Handling

- Improved program stability: Catches and handles errors gracefully, preventing unexpected program termination.
- Enhanced code readability and maintainability: Clearly separates error handling logic from the main program code.
- Reduced debugging efforts: Easier identification and resolution of errors.
- Increased application robustness and reliability: Provides a safety net against unexpected situations.

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. DDL DML DCL

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. Stored Procedures and Parameters