Meta Linguistic Abstractions (Part 1)

Praseed Pai K.T.

What is the Difference between a Stored Procedure (SP) and a Stored Function (SF)?

- SP & SF are available with RDBMS software
- Oracle (PL/SQL) , SQL Server (T-SQL) and MySQL supports it

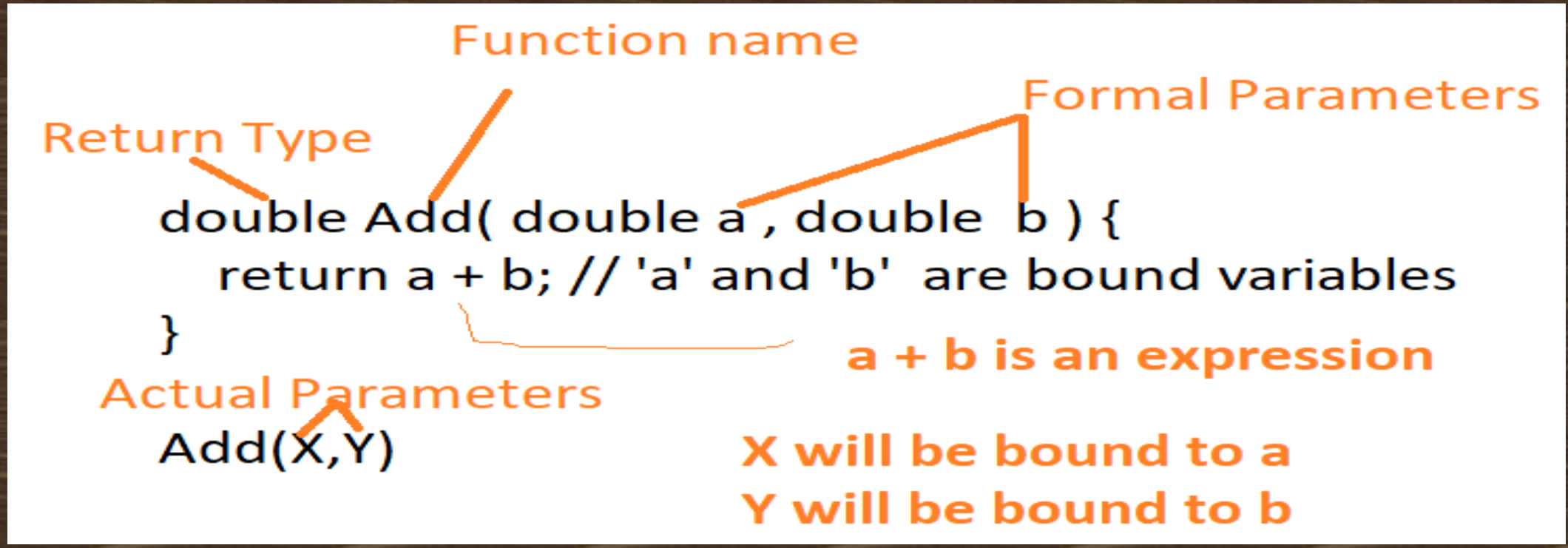
- Usual Answer
  - I do not know. I did not realize “SF” is available in RDBMS software
  - Stored Procedure is Compiled , SF is interpreted
  - In Function, It is mandatory to use RETURN value, in SP, you do not need
  - You can call a Stored Function from a Procedure, converse is not possible

The Biggest Difference is

- SF can be used in the Where Clause (Where/Having <Predicate>)
- SF can be included in a Selected Field List

```
select dbo.helloworldfunction() as regards
```

Lexicon of Function and it's invocation



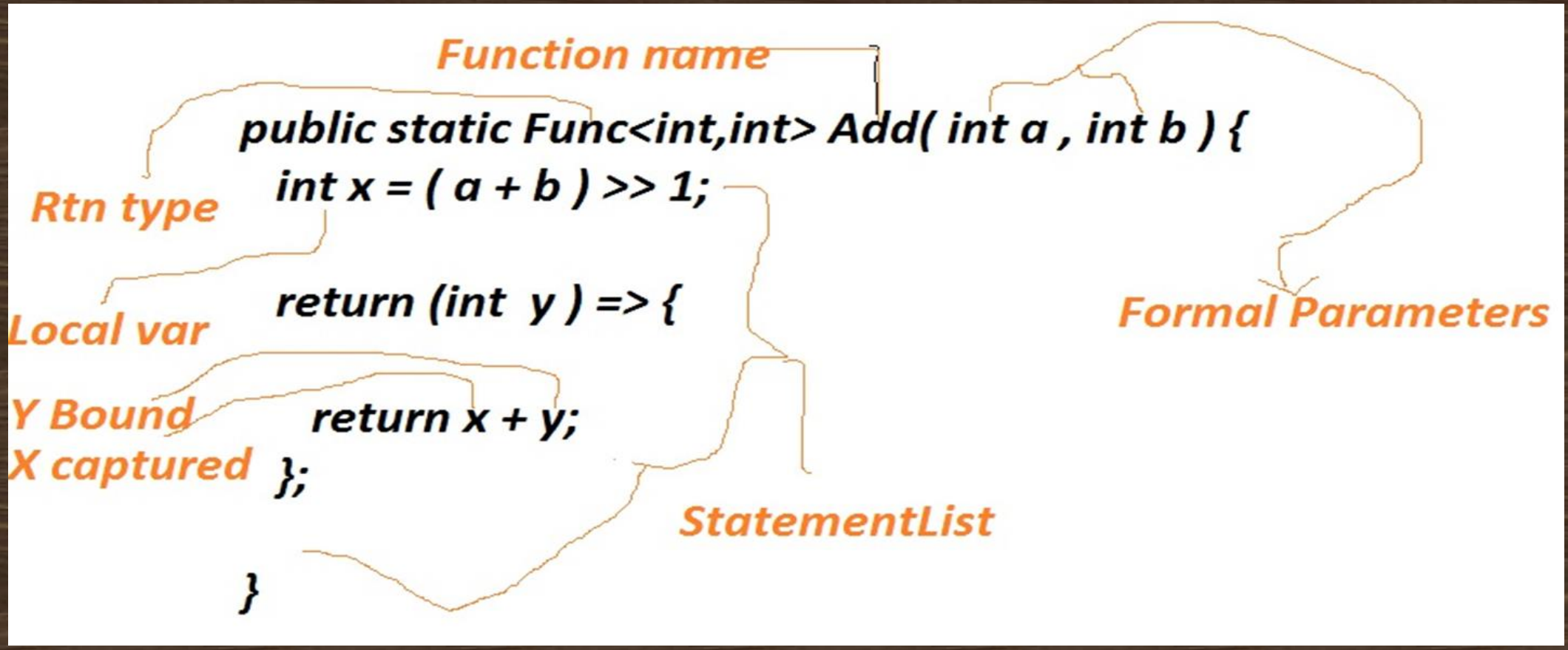
About Procedures

```
double c = 10;

double Add2( double a , double b )  {
      c = c+1;  // an assignment statement
      return a + b + c; // a and b are bound variables
                        // c is global
}
```

Global Variable

Lambda (Function) in FP



Expression/Statement

- Expression is what you evaluate for it’s value
- Statement is what you execute for it’s effects (on variables)

Functions vs Procedures

- Functions enable Expression oriented Programming
  - An expression never mutates value
- Procedures enable Statement Oriented Programming
  - A Statement can mutate more than one variable in every step

Function vs Procedure

- Pure Function returns a single return value (it can be vectored!)
- Procedure returns many values
  - Procedure uses Call by Reference to achieve this
  - It might change Global Variables as well!
- In a Pure Function
  - Return Value should be solely based on the Formal Parameters

Functions behaves more or less like procedures in Conventional Programs

- Most languages allow statements within functions
- Functions can also have a list of statements
- The major difference is in the return value
- To make function pure , we should do
  - Parameter should be the sole determinant of the return value
  - Any kind of mutation which happens should be inside the scope of function
  - We should not pass by reference
  - There should not be any I/O, Thread, Null return value

Referential Transparency of Function

- If a function always return same value for a give parameter value set, it is technically referentially transparent
- Should not do
  - I/O (Network / File)
  - Threading/Asynchrony
  - should not return NULL
- The Gold Standard for a Referentially Transparent Function
  - The Values can be memorized (without side effects)

What restrictions are there in SF?

- SF should not
  - create temporary tables
  - Issue Delete, Update and Insert statements
  - No transactions
- Why there is Restrictions?
  - SF should not have mutability and side effects
  - Any external data source or parameters should be read only