InterSystems CACHÉ e-Learning

Programming with SQL (2006)

Leader: Al Goins

Support Specialist

InterSystems CACHÉ e-Learning

Programming Cache SQL

- Module Overview:
  - This module covers how to use SQLin your COS programs.
- Module Objectives
  - After completing this module you will be able to:
    - Use Dynamic SQL.
    - Use Embedded SQL.

InterSystems CACHÉ e-Learning

Embedded SQL

- Caché SQLsupports the ability to embed SQLstatements within Caché ObjectScript code.
- These embedded SQLstatements are converted to optimized, executable code at compilation time.

InterSystems CACHÉ e-Learning

The Macro Preprocessor

- You can used embedded SQL within class methods or within Caché ObjectScript routines.
- A routine or method is processed by the Caché Macro Preprocessor and converted to .INT (intermediate) code which is subsequently compiled to executable code.
- The Macro Preprocessor replaces all embedded SQL statements with the code that actually executes the SQL statement.

InterSystems CACHÉ e-Learning

The &sql Directive

- Embedded SQLstatements are set off from the rest of the code by the &sql() directive.
- &sql() is case insensitive
- SQLwithin the directive can use schema names or not, if not, then the package of the class is used as the default schema.

InterSystems CACHÉ e-Learning

Examples of embedded SQL

```
Method CountStudents() As %Integer {
   &sql(SELECT COUNT(*) INTO :count FROM
   MyApp.Student) Quit count
}
```

InterSystems CACHÉ e-Learning

AppShare 1:

Using embedded SQL

InterSystems CACHÉ e-Learning

Literal Values

- Embedded SQLqueries may contain literal values (strings, numbers, or dates):

```
&sql(SELECT 'Dr.' || Name INTO :name FROM
MyApp.Doctor WHERE State = 'NY')


&sql(SELECT Name INTO :name FROM
MyApp.Person WHERE Age > 50)
```

InterSystems CACHÉ e-Learning

Question 1

Which of the following are valid calls to the embedded SQLdirective?

- &sql()
- &SQL()
- &Sql()
- All of the above

InterSystems CACHÉ e-Learning

Answer 1

Which of the following are valid calls to the embedded SQLdirective?

- &sql()
- &SQL()
- &Sql()
- ***All of the above***

InterSystems CACHÉ e-Learning

Host Variables

- Host variables can be used in most places that a literal value can be used or within an INTO clause.
- A host variable is the name of a local variable, preceded by a ":" character
- Input host variables are never valid after embedded SQL.
- Output host variables are only reliably valid after embedded SQLwhen SQLCODE = 0.

InterSystems CACHÉ e-Learning

Using Host Variables

```
&sql(SELECT Name INTO :name FROM MyApp.Person WHERE
  %ID = 1)

Set minval = 10000
Set maxval = 50000
&sql(SELECT Name,Salary INTO :name, :salary FROM
  MyApp.Employee WHERE Salary > :minval AND Salary <
  :maxval)

&sql(SELECT Name, Title INTO :val(1), :val(2) FROM
  MyApp.Employee WHERE %ID = :emp("ID") )

&sql(SELECT Name, Title INTO :obj.Name, :obj.Title
  FROM MyApp.Employee WHERE %ID = :id )
```

InterSystems CACHÉ e-Learning

AppShare 2:

Using Host Variables in Embedded SQL

InterSystems CACHÉ e-Learning

SQL Cursors

- Used to retrieve multiple rows from the result of embedded SQL
- An SQLCursor is DECLAREd and given a name. You then use this name to OPEN, FETCH data from, and CLOSE the cursor
- A cursor name must be unique within a class or routine.
- The DECLARE statement must occur within a routine before any statements that use the cursor

InterSystems CACHÉ e-Learning

Using SQL Cursors

```
&sql(DECLARE C1 CURSOR FOR SELECT %ID,Name
  INTO :id, :name FROM Sample.Person ORDER
  BY Name )
&sql(OPEN C1)
&sql(FETCH C1)
While (SQLCODE = 0) {
  Write id, ": ", name,!
  &sql(FETCH C1)
}
&sql(CLOSE C1)
```

InterSystems CACHÉ e-Learning

Question 2

SQLCursors can be reused throughout your method or routine.

- True
- False

InterSystems CACHÉ e-Learning

Answer 2

SQLCursors can be reused throughout your method or routine.

- True
- **False**

InterSystems CACHÉ e-Learning

Dynamic SQL

- Queries are prepared at runtime rather than compile time.
- Allows you to build the query based on user input or runtime status
- Slightly less efficient because of runtime preparation
- Queries are cached in order to speed up reuse.

InterSystems CACHÉ e-Learning

The %Library.ResultSet Class

- Dynamic SQL is supported via the %Library.ResultSet class.

- Applications create an instance of the %Library.ResultSet class and use it to prepare, execute, and iterate over queries.

InterSystems CACHÉ e-Learning

Using the %Library.ResultSet Class

```
Set result=##class(%ResultSet).%New()

Set sc=result.Prepare("SELECT ID, Name,
  Salary FROM Employee WHERE Salary > ?")

Set sc=result.Execute(10000)
```

InterSystems CACHÉ e-Learning

Fetching the Data and Closing

```
While result.Next(.sc) {
  If $$$ISERR(sc)
  Quit
  Write
  result.Data("Name"),result.Data("Salary"),!
}

do result.Close()
```

InterSystems CACHÉ e-Learning

AppShare 3:

Using Dynamic SQL

InterSystems CACHÉ e-Learning

Module Summary

- This module covered:
  - How to program with embedded SQL.
  - How to program with dynamic SQL.

InterSystems CACHÉ e-Learning

References

This module is part of the following learning track(s):

(1) Cache SQL:

- Module 1: Introduction to SQL
- **Module 2: Programming with SQL**
- Module 3: SQLConnectivity
- Module 4: SQLGateway
- Module 5: SQLSecurity
- Module 6: Performance and Debugging

InterSystems CACHÉ e-Learning

Questions

Programming with SQL

Leader:Al Goins

Support Specialist