

Programming in SQL

Variables

A **variable** is a named allocation of memory that we can use to store information. The type of information that the variable can store is defined when the variable is declared.

Declaration

```
DECLARE @<variable name>    <type>
```

Examples

```
DECLARE @quantity INT
```

```
DECLARE @acctBalance MONEY
```

```
DECLARE @clientRef VARCHAR(50)
```

Programming in SQL

Variables

Assignment

```
SET @<variable name>      = <expression>
SELECT @<variable name> = <expression> FROM...
```

Examples

```
SET @quantity = 50
SET @acctBalance = @acctBalance - @debit
SET @acctBalance -= @debit
SELECT    @clientRef = clientID
          FROM clients
          WHERE name = 'Tremblay'
DECLARE @grade INT = 76
```

Programming in SQL

Conditional Statements

Test

```
IF <expression>
BEGIN
    <instructions>
END
```

```
IF <expression>
BEGIN
    <instructions>
END
ELSE
BEGIN
    <instructions>
END
```

Examples

```
IF @sales > @previousSales
BEGIN
    PRINT 'sales are on the rise!'
END
```

```
IF @sales > @previousSales
BEGIN
    PRINT 'sales are on the rise!'
END
ELSE
BEGIN
    PRINT 'sales are down since
    last quarter!'
END
```

Programming in SQL

Conditional Statements

Choice

```
CASE <expression>
  WHEN <expression> THEN <expression>
  ...
  WHEN <expression> THEN <expression>
  ELSE <expression>
END
```

```
CASE
  WHEN <condition> THEN <expression>
  ...
  WHEN <condition> THEN <expression>
  ELSE <expression>
END
```

Examples

```
DECLARE @id VARCHAR(20) = 'CL'
DECLARE @company VARCHAR(20)
SET @company =
  CASE @id
    WHEN 'AF' THEN 'Air France'
    WHEN 'AM' THEN 'Royal Air Morocco'
    WHEN 'CL' THEN 'Comair Limited'
    WHEN 'AA' THEN 'Air Algeria'
    WHEN 'T' THEN 'Tasair'
  ELSE 'unknown'
END
```

```
DECLARE @grade INT = 97
DECLARE @evaluation VARCHAR(20)
SET @evaluation =
  CASE
    WHEN @grade < 20 THEN 'very low'
    WHEN @grade < 40 THEN 'low'
    WHEN @grade < 60 THEN 'medium'
    WHEN @grade < 80 THEN 'good'
    WHEN @grade < 90 THEN 'very good'
    ELSE 'excellent!'
  END
```

Programming in SQL

Loops

WHILE Loop

```
WHILE    <condition>
BEGIN
    <instructions>
END
```

Example

```
DECLARE    @counter INT = 0
WHILE      @counter < 10
BEGIN
    PRINT 'loops ' + CONVERT(VARCHAR, @cpt)
    SET @counter += 1
END
```