Διάλεξη 19: Προγραμματισμός Βάσης Δεδομένων III

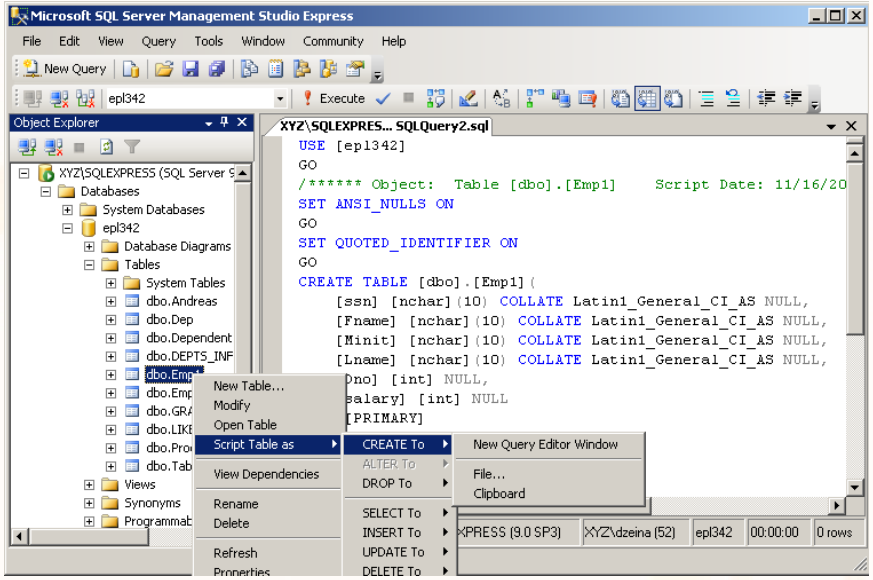
Στην ενότητα αυτή θα μελετηθούν τα εξής επιμέρους θέματα:
Εισαγωγή στις έννοιες:

- Scripts/Batches σε TSQL
- Δυναμική SQL
- Δρομείς (Cursors) σε TSQL

Διδάσκων: Παναγιώτης Ανδρέου

Scripts σε TSQL

- **Scripts: Ακολουθία (T)SQL Εντολών** η οποία **αποθηκεύεται** σε ένα **αρχείο** για **επαναχρησιμοποίηση**.
- Παραδείγματα Χρήσης:
  - **Backup / Restore** πινάκων ή ολόκληρης της DB.
    - Θυμηθείτε το Northwind.sql (Εργαστήριο 9) το οποίο χρησιμοποιήσατε για να δημιουργήσετε αυτόματα όλη την βάση.
  - **Επανάληψη Συχνών**
  - **Λειτουργιών** (Συντήρηση, κτλ.)



Μεταβλητές σε TSQL

Παράδειγμα Script σε TSQL

Set current Database (otherwise script will be executed on whatever database is currently open)

```
USE epl342;
```

Declare Variable with Name TEST of TYPE int *(default = NULL)*

```
DECLARE @TEST int
```

Assign Value to Variable TEST

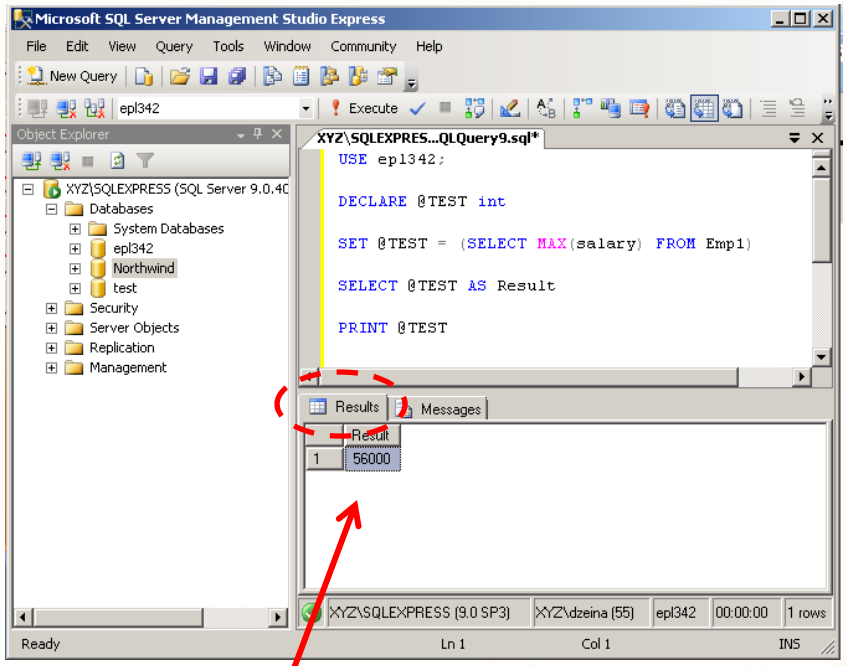
```
SET @TEST = (SELECT MAX(salary) FROM Emp1)
```

Display the TEST Variable with Column Name Result

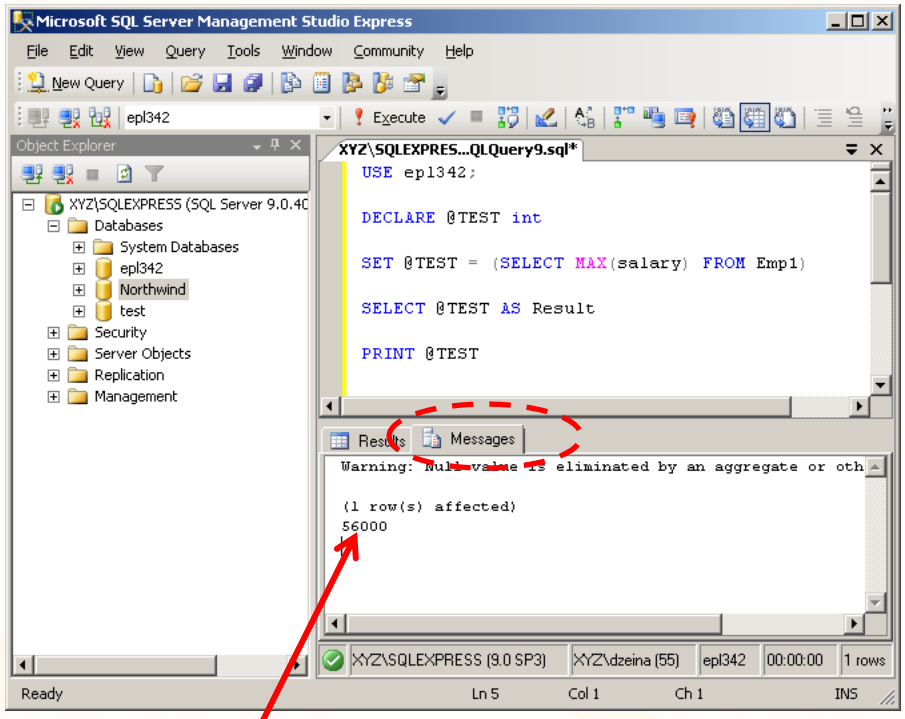
```
SELECT @TEST AS Result  -- Αντίστοιχο του STDOUT
```

PRINT @TEST – Αντίστοιχο του **STDERR.** Το μήνυμα είναι μέχρι 8000 χαρακτήρες και επιστρέφεται στον χρήστη.

Μεταβλητές σε TSQL



SELECT



PRINT (δες και RAISERROR για try…catch)

Χαρακτηριστικά των Scripts

- Χαρακτηριστικά των Scripts
  1. Τα Scripts **δημιουργούνται** και **εκτελούνται** από κάποιο **χρήστη** ή ως μέρος άλλου **script**.
  2. Το Script εκτελείται **γραμμή-γραμμή** από **πάνω** προς τα **κάτω** (η σύνταξη ελέγχετε πριν την εκτέλεση) από τον μεταφραστή της βάσης.
  3. Εάν **προκύψει λάθος (runtime λάθος)** τότε **ΑΚΥΡΩΝΕΤΑΙ** ολόκληρο το script.
     - Ένα Script εκτελείται ως μια δοσοληψία (transaction). Συνεπώς αποτελεί μια **ατομική πράξη**: "Είτε εκτελείται ολόκληρο ή καθόλου"!
     - Μετά από κάποιο λάθος **επαναφέρεται** (**ROLLBACK**) πίσω στην αρχική κατάσταση η βάση δεδομένων.

Παράδειγμα με χρήση @@IDENTITY

Παράδειγμα Εισαγωγής Συσχετιζόμενων Δεδομένων σε Δυο Πίνακες (Order και OrderDetails).

```
USE Northwind
DECLARE @NewOrderID int

INSERT INTO Orders(CustomerID, OrderDate)
VALUES (15, DATEADD(day,-1,GETDATE()))

SET @NewOrderID = @@IDENTITY
-- ή SELECT @NewOrderID = @@IDENTITY

INSERT INTO [Order Details](OrderID, ProductID, UnitPrice, Quantity)
VALUES (@NewOrderID, 1, 50, 25)

SELECT 'The OrderID of the INSERTed row is ' + CONVERT(varchar(8),
    @NewOrderID)
```

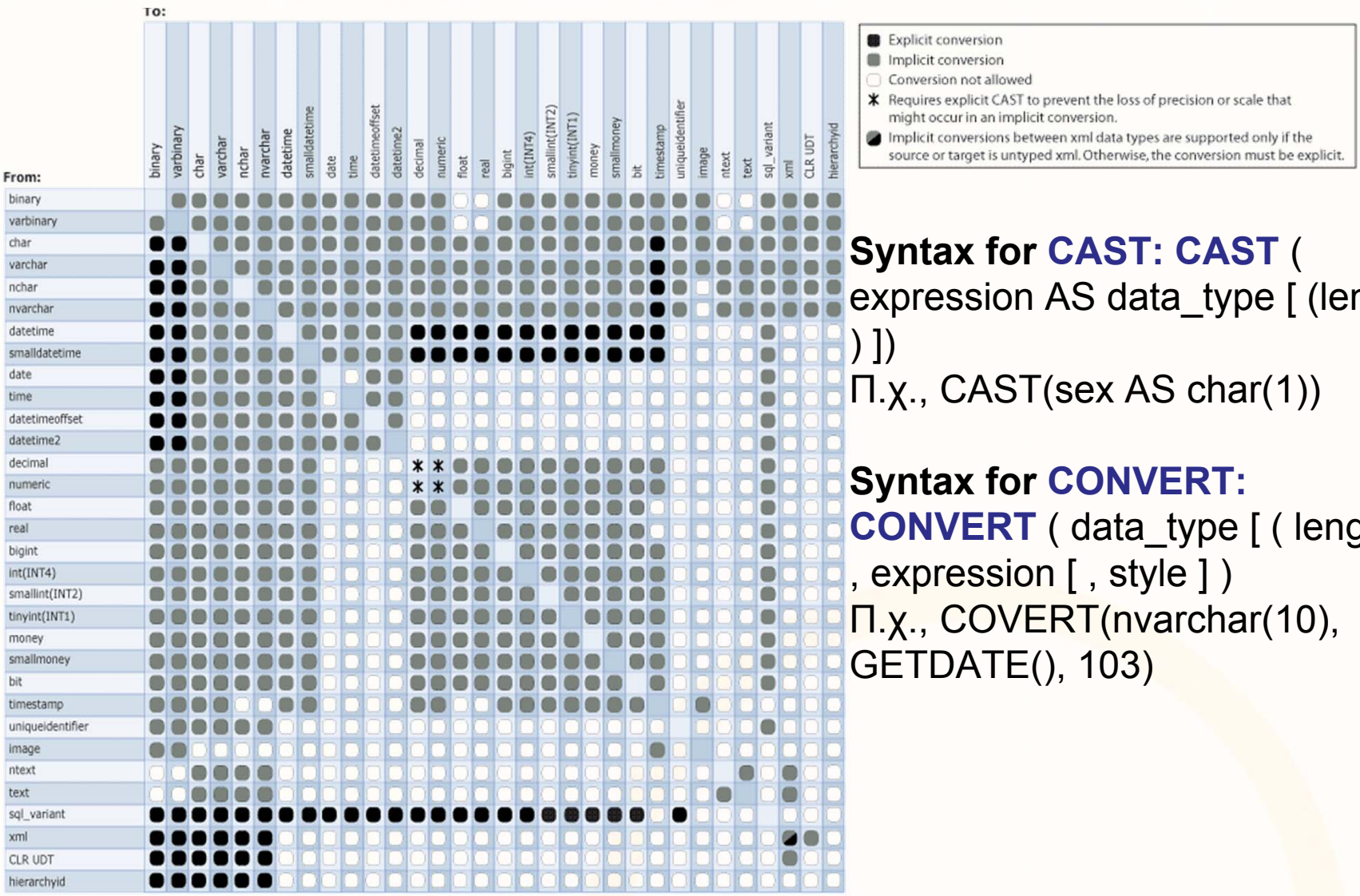
Current Date Function

Assign value to variable

@@: System Function (last recorded Identity)

Casting integer to string

Πίνακας Convert / Cast σεTSQL



Syntax for CAST: CAST (expression AS data_type [(length)])
Π.χ., CAST(sex AS char(1))

Syntax for CONVERT:
CONVERT (data_type [(length)] , expression [, style])
Π.χ., COVERT(nvarchar(10), GETDATE(), 103)

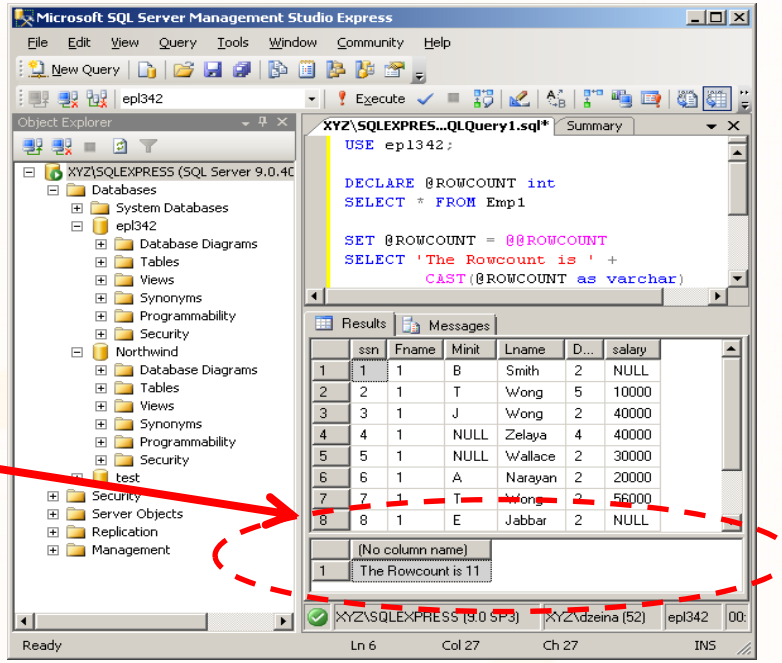
Χρήση Συνάρτησης Συστήματος @@ ROWCOUNT

- Η συνάρτηση συστήματος **@@ROWCOUNT** σας επιστρέφει τον **αριθμό των πλειάδων** που **επηρεάστηκαν** ή **διαβάστηκαν** από την τελευταία **SQL επερώτηση**.
  - Θυμηθείτε το μήνυμα: "(X row(s) affected)" που προκύπτει μετά από ανάγνωση/αλλαγή δεδομένων ενός πίνακα

```
USE epl342;

DECLARE @ROWCOUNT int
SELECT * FROM Emp1

SET @ROWCOUNT = @@ROWCOUNT
SELECT 'The Rowcount is' +
        CAST(@ROWCOUNT as varchar);
```



- Με την εντολή "**SET NOCOUNT ON**" δεν τυπώνεται το "(X row(s) affected)"

Δέσμες (Batches) σε TSQL

- Τα **Batches (Δέσμες)** είναι Scripts εντολών TSQL τα οποία διαχωρίζονται με την εντολή **GO** και τα οποία εκτελούνται ανεξάρτητα μεταξύ τους (δηλαδή όχι στα πλαίσια του ίδιου transaction)
- **Παράδειγμα:**

```
USE epl342;
GO -- αποστολή δέσμης εκφράσεων TSQL στον SQL Server
INSERT INTO Emp1(SSN) VALUES ('4411111993')
GO -- αποστολή δέσμης εκφράσεων TSQL στον SQL Server
INSERT INTO Emp1(SSN) VALUES ('3311111993')
GO -- δεν πρέπει να υπάρχουν άλλες εντολές στην ίδια γραμμή με το GO!
```

- Η εντολή **GO** λειτουργεί ΜΟΝΟ στο πλαίσιο του **SQL Management Studio** ή της **sqlcmd** (ή **osql**).
  - Σε προγράμματα γίνεται κάτι αντίστοιχο μέσω εξειδικευμένων εντολών, π.χ. στη JAVA: **stmt.executeUpdate(query);**

Δέσμες (Batches) σε TSQL

- Κάποιες **εντολές** είναι **αναγκαστικό** να είναι μέρος του **δικού** τους **Batch** (δηλαδή **πρεπει** να **ακολουθούνται** από GO).
  - Μερικές από αυτές είναι:
    - **CREATE TRIGGER**
    - **CREATE VIEW**
    - **CREATE PROCEDURE** → θα το δούμε σε λίγο
- **Συντακτικά Λάθη** ελέγχονται όπως και στα Scripts πριν την εκτέλεση ολόκληρου του Batch
- Εάν προκύψουν **Runtime λάθη** σε ένα batch στο σημείο Χ τότε δεν εκτελείται καμία εντολή μετά το Χ.
  - Εντολές πριν το Χ δεν γίνονται ROLLBACK και αλλάζουν μόνιμα την κατάσταση της βάσης.

Δυναμική SQL (Dynamic SQL) σε TSQL

- Η δυναμική SQL (dynamic SQL) επιτρέπει στους προγραμματιστές βάσεων δεδομένων να παράγουν της δηλώσεις SQL με δυναμικό τρόπο
  - Χρήσιμες εάν **δεν είναι γνωστή εκ των προτέρων η SQL έκφραση**.
  - **Σημείωση:** Όλα τα προηγούμενα παραδείγματα ήταν με στατικές εκφράσεις SQL (Static SQL)
- Εκτελούνται με **EXEC** ή με **EXECUTE sp_executesql**
- Παράδειγμα:
  Αντί για SELECT * FROM PERSON

```
DECLARE @SQL NVARCHAR(MAX)
SET @SQL = 'SELECT * FROM PERSON'
EXECUTE sp_executesql @SQL
```

Έκφραση SQL που παράγεται δυναμικά κατά την εκτέλεση

Δυναμική SQL (Dynamic SQL) σε TSQL

```
-- Δημιουργία Πίνακα Ως Χρήστης DBO (Database Owner)
USE epl342;
GO
EXECUTE ('CREATE TABLE SalesTable (SalesID int, SalesName
    varchar(10));')
AS USER = 'dbo';
GO
```

Το EXEC είναι συντομογραφία του EXECUTE

Επισημάνσεις για το EXEC

- Εκτελείται κάτω από με τα **ίδια δικαιώματα** με τον Script που το καλεί.
- Το EXEC τρέχει με τα **ίδιο connection** με το πρόγραμμα που το καλεί.
- Εάν θα γίνει σύμπτυξη με συνάρτηση, τότε αυτή πρέπει να γίνει πριν την κλήση της EXEC.
  - **ΛΑΘΟΣ:** EXEC ('SELECT * FROM Emp1 WHERE date=' **+ GetDate()**);

Εάν ήταν @DATEVAL δεν θα είχε πρόβλημα

Έλεγχος Ροής σε TSQL

- **Τελεστής Έλεγχου**
  - **IF <Boolean Expression>**
    <SQL statement> | BEGIN <code series> END
    ELSE
    <SQL statement> | BEGIN <code series> END

Επισημάνσεις

- Προφανώς επιτρέπεται και η **εμφώλευση** αυτού του τελεστή τόσο με τον εαυτό του όσο και με άλλους τελεστές.
- Εάν το <Boolean Expression> είναι NULL τότε είναι FALSE η λογική συνθήκη.
- **ΛΑΘΟΣ**: IF @myvar=NULL ➔ **ΣΩΣΤΟ:** IF @myvar IS NULL
- Υπάρχει και η **CASE** (δηλ., αντίστοιχο της SWITCH στη C) http://msdn.microsoft.com/en-us/library/ms181765.aspx

Δημιουργία Γνωρισμάτων σε TSQL

```
USE AdventureWorks;
GO
DECLARE @CATEGORY NVARCHAR(50)
SET @CATEGORY = SELECT
    CASE ProductLine
      WHEN 'R' THEN 'Road'
      WHEN 'M' THEN 'Mountain'
      WHEN 'T' THEN 'Touring'
      WHEN 'S' THEN 'Other sale items'
      ELSE 'Not for sale'
    END,  Name, SSN
FROM Production.Product
WHERE PRODUCT_ID=5;
GO
```

Η συνθήκη μπορούσε να είναι και πιο σύνθετη, π.χ., (ProductNumber % 2) = 0

Επανάληψη σε TSQL

- **Τελεστής Επανάληψης** σε TSQL

```
WHILE Boolean_expression
BEGIN
        sql_statement |
      statement_block |    -- several statements in BEGIN  … END
      BREAK |
      CONTINUE
END
```

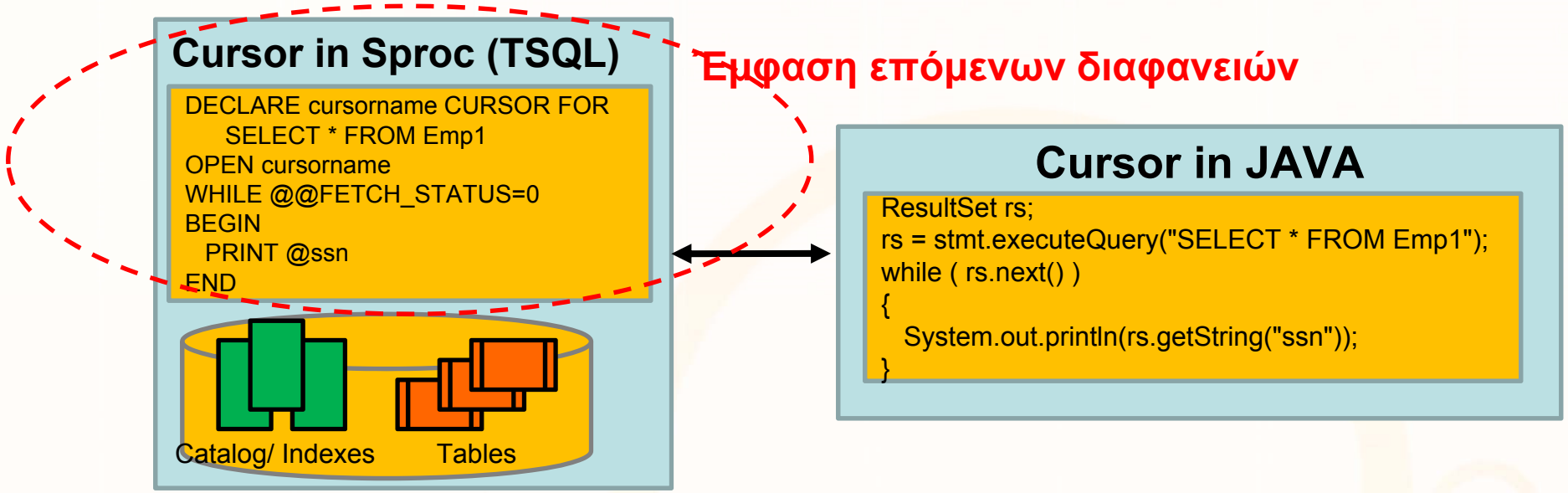
- **Παράδειγμα**

```
DECLARE @var INT

WHILE 1 = 1
BEGIN
      WAITFOR DELAY '00:00:01'
      SET @var = (SELECT MAX(salary) FROM EMP1)
END
```

Cursors (Δρομείς) σε TSQL

- Γνωρίζουμε ότι τα **αποτελέσματα** επερωτήσεων επιστρέφονται σε μια επερώτηση υπό **μορφή μιας σχέσης**
  - Π.χ., "**SELECT * FROM Emp1**" επιστρέφει ένα ολόκληρο πίνακα.
- Εάν θέλουμε να **επεξεργαστούμε** τα **αποτελέσματα** αυτά **γραμμή-γραμμή** (αντί να απλά να τυπωθούν), τότε χρησιμοποιούμε την έννοια του **Δρομέα (Cursor).**
  - **Cursors** υπάρχουν και στην **JAVA** όπως είδατε στο φροντιστήριο.



Cursors (Δρομείς) σε TSQL

```
USE epl342
DECLARE @ssn nchar(10), @fname nchar(10)
DECLARE employee_cursor CURSOR FOR          -- Δήλωση μεταβλητής τύπου Cursor
    SELECT ssn,fname FROM Emp1 -- Επερώτηση που συνδέεται με τον Cursor

OPEN employee_cursor   -- Άνοιγμα Cursor (Εκτέλεση Επερώτησης)

-- Ανάγνωση πρώτης γραμμής στις μεταβλητές id, fname
FETCH NEXT FROM employee_cursor
INTO @ssn, @fname

WHILE @@FETCH_STATUS=0  -- όσο δεν άδειασαν τα αποτελέσματα
BEGIN
  PRINT RTRIM(@ssn) + ', ' + RTRIM(@fname) -- RTRIM: φεύγει right spaces
  FETCH NEXT FROM employee_cursor  -- Ανάγνωση επόμενης γραμμής
  INTO @ssn, @fname
END

CLOSE employee_cursor -- Κλείσιμο Cursor
DEALLOCATE employee_cursor  -- Αποδέσμευση Πίνακα Ενδιάμεσων Αποτελεσμάτων που χρησιμοποιεί ο SqlServer για το Result του Cursor . Εναλλακτικά μένει στην μνήμη μέχρι το Κλεισιμο του Session.
```

Τυπώνει
1, 1
2, Franklin
3, Alicia
4, Jennifer
5, Ramesh
6, Joyce
7, Ahmad
8, James
1, 1

Χαρακτηριστικά Cursor

- **Κατεύθυνση Cursor:** Η default λειτουργία του cursor είναι να κινείται μπροστά **NEXT (default)**, εγγραφή-εγγραφή μέσα σε ένα αποτέλεσμα. Υπάρχουν ορίσματα (**SCROLL**) για να κινείται διαφορετικά **FIRST, LAST, PRIOR, RELATIVE, κτλ.**
- **Ενημέρωση Αποτελεσμάτων:** Είναι δυνατό να ενημερώνεται το αποτελέσματα που προσπελαύνεται από ένα CURSOR (**READ ONLY (default) | UPDATE**)
  - π.χ., κατά την προσπέλαση αύξησε ένα γνώρισμα κατά 1
- **Εμβέλεια Cursor:** Η εμβέλεια ενός CURSOR μπορεί να περιοριστεί τοπικά μέσα στο ίδιο batch) (**LOCAL (default) | GLOBAL**)
- Στο manual υπάρχουν αρκετές άλλες εξειδικεύσεις που προσφέρονται από την TSQL

Σύνταξη Δημιουργίας Cursor

ISO Syntax

DECLARE cursor_name [INSENSITIVE] [SCROLL] **CURSOR**
    FOR select_statement
    [FOR { READ ONLY | UPDATE [OF column_name [,...n]] }]
[;]

Transact-SQL Extended Syntax

DECLARE cursor_name **CURSOR** [LOCAL | GLOBAL]
    [FORWARD_ONLY | SCROLL]
    [STATIC | KEYSET | DYNAMIC | FAST_FORWARD]
    [READ_ONLY | SCROLL_LOCKS | OPTIMISTIC]
    [TYPE_WARNING]
    FOR select_statement
    [FOR UPDATE [OF column_name [,...n]]]
[;]

Παράδειγμα Script για Backup Βάσεων

```
DECLARE @name VARCHAR(50) -- database name
DECLARE @path VARCHAR(256) -- path for backup files
DECLARE @fileName VARCHAR(256) -- filename for backup
DECLARE @fileDate VARCHAR(20) -- used for file name

SET @path = 'C:\Backup\'   -- κατάλογος στον οποίο θα γίνει το backup

SELECT @fileDate = CONVERT(VARCHAR(20),GETDATE(),112)

DECLARE db_cursor CURSOR FOR
        SELECT name
        FROM master.dbo.sysdatabases
        WHERE name NOT IN ('master','model','msdb','tempdb')
-- Continued On Next Page
```

Μορφοποίηση Ημερ. με style 112, δηλ. σε: 20091119

Πίνακας του Καταλόγου που περιέχει τα ονόματα όλων των βάσεων

Παράδειγμα Script για Backup Βάσεων (συν.)

```
-- Continued From Previous Page

OPEN db_cursor   -- άνοιγμα cursor

FETCH NEXT FROM db_cursor -- ανάκτηση επόμενης τιμής σε τοπική var
INTO @name

WHILE @@FETCH_STATUS = 0
BEGIN
      -- δημιουργία filename
      SET @fileName = @path + @name + '_' + @fileDate + '.BAK'
      -- εντολή για backup βάσης στο directory/name @fileName
      BACKUP DATABASE @name TO DISK = @fileName
      -- Το RESTORE DATABASE κάνει restore την βάση
      FETCH NEXT FROM db_cursor INTO @name
END

CLOSE db_cursor
DEALLOCATE db_cursor
```