



Διάλεξη 18: Προγραμματισμός Βάσης Δεδομένων II

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

- Σκανδάλες (Triggers) σε TSQL
- Ευρετήρια (Indexes) σε TSQL
- Εισαγωγή στον διαδικαστικό προγραμματισμό σε ΒΔ

Διδάσκων: Παναγιώτης Ανδρέου

Σκανδάλες σε SQL (SQL Triggers)

- Μια **Σκανδάλη (Trigger)** ορίζει μια **αντίδραση** της βάσης δεδομένων σε περίπτωση **αλλαγών πλειάδων*** (INSERT, DELETE, UPDATE) σε κάποιους **προσδιορισμένους πίνακες**.
 - **ASSERTIONS:** απαγορεύουν κάποια κατάσταση
 - Π.χ., ο μισθός του υπαλλήλου ΔΕΝ μπορεί να είναι μεγαλύτερος από αυτόν του supervisor του.
 - **TRIGGERS:** δεν σημαίνει απαραίτητα ότι απαγορεύουν μια κατάσταση, άπλα ορίζουν ακολουθία εντολών που πρέπει να ενεργοποιηθεί όταν ικανοποιηθεί μια συνθήκη
 - π.χ., όταν προστεθούν/αφαιρεθούν λεφτά από τον λογαριασμό κάποιου πελάτη, στείλε email στον πελάτη για να τον ενημερώσεις
- Το SELECT, TRUNCATE ή BULK INSERT δεν ενεργοποιούν τις σκανδάλες σε TSQL.

Κατηγορίες Σκανδάλων

- Τα triggers χωρίζονται σε δυο κατηγορίες:
 - **DDL Triggers:** Ορίζουν την **αντίδραση** σε **δομικές αλλαγές** (DROP, ALTER, κτλ).
 - **DML Triggers:** Ορίζουν την **αντίδραση** σε αλλαγές πάνω σε πλειάδες μιας σχέσης ή όψης (INSERT, UPDATE, DELETE).
 - Θα επικεντρωθούμε μόνο στα DML Triggers.
- Σημειώστε ότι τα Triggers είναι ουσιαστικά «μικρά προγράμματα σε (T)SQL» τα οποία καλούνται **ΑΥΤΟΜΑΤΑ** μόλις ενεργοποιηθεί η **ορισμένη συνθήκη** η οποία ισχύει πάνω σε **κάποιους πίνακες ή όψεις**.
 - Τα triggers ΔΕΝ μπορούμε να τα καλέσουμε (invoke) αυτόνομα (π.χ., με SELECT ή EXEC)

Παράδειγμα Trigger σε TSQL

-- Παράδειγμα σκανδάλης που παράγει μήνυμα λάθους
όποτε γίνει εισαγωγή/διαγραφή δεδομένων από τον
πίνακα PERSON

IF OBJECT_ID ('Reminder1', 'TR') IS NOT NULL

DROP TRIGGER Reminder1;

-- Διαγραφή σκανδάλης αν υπάρχει

GO

CREATE TRIGGER reminder1

ON PERSON

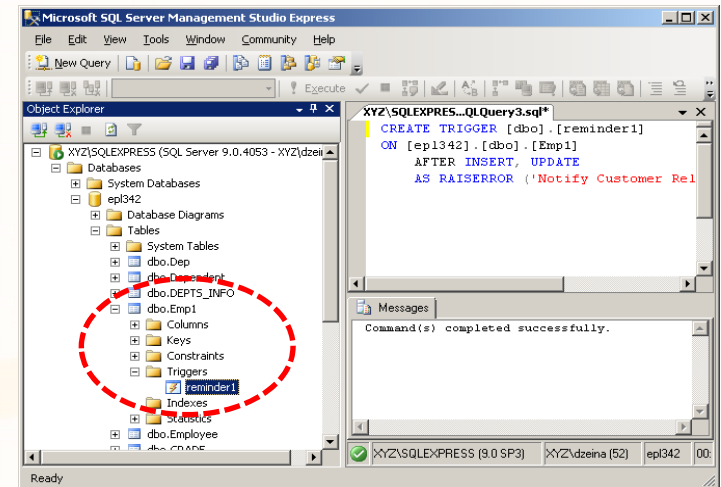
AFTER INSERT, UPDATE

AS -- ακολουθούν οι εντολές που πρέπει να εκτελεστούν

RAISERROR ('Notify Customer Relations', 16, 1); → *State: 0-255*

ROLLBACK TRANSACTION; RETURN

GO



*Severity:
0-18: Specified by User
20-25: FATAL Errors*

Παράδειγμα Trigger σε TSQL

- Κατά την εισαγωγή/ενημέρωση δεδομένων στον Πίνακα Emp1 επιστρέφεται μήνυμα λάθους διότι ενεργοποιείται η σκανδάλη.

The screenshot shows the Microsoft SQL Server Management Studio Express interface. The Object Explorer on the left displays the database structure for 'XYZ\SQLEXPRESS (SQL Server 9.0.4053 - XYZ\dzeii)', including the 'dbo.Emp1' table. The main window displays an error message: 'No row was updated. The data in row 9 was not committed. Error Source: .Net SqlClient Data Provider. Error Message: Notify Customer Relations. Correct the errors and retry or press ESC to cancel the change(s).' Below the error message, a table view shows the data in the 'dbo.Emp1' table. The table has 9 rows, with the 9th row (row 9) highlighted in red and marked with an error icon. The table data is as follows:

5	1	NULL	Wallace
6	1	A	Narayan
7	1	T	Wong
8	1	E	Jabbar
9	1	E	Smith
*	NULL	NULL	NULL

Σύνταξη Σκανδάλης σε TSQL

CREATE TRIGGER <trigger-name>

ON [schema-name>.] <table | view-name>

[WITH ENCRYPTION] -- trigger code is encrypted in DB

[EXECUTE AS <CALLER | SELF | <user>]

-- Default: Caller (of change), SELF: Creator of Trigger, user:

{{FOR | AFTER| INSTEAD OF}
<[DELETE] [,] [INSERT] [,] [UPDATE]>

AS

<sql-statements>

- FOR|AFTER: Αναφέρονται στο **ίδιο πράγμα** και προστίθεται για να είναι πιο **ευανάγνωστος** ο κώδικας,

- π.χ., AFTER DELETE AS

IF EXISTS (SELECT)

BEGIN

.....

END

*SQL
Statements*

Προσωρινοί Πίνακες σε Σκανδάλες

- Οι σκανδάλες χρησιμοποιούν δύο προσωρινούς πίνακες (inserted, deleted) για αναγνώριση επεξεργασία των ενημερώσεων
 - **Εισαγωγές:** Η πλειάδα (ή πλειάδες) που εισάγονται, καταχωρούνται στον πίνακα **inserted**
 - **Διαγραφή:** Η πλειάδα (ή πλειάδες) που διαγράφονται, καταχωρούνται στον πίνακα **deleted**
 - **Ενημερώσεις:** Ας υποθέσουμε ότι μία ενημέρωση αποτελείται από μία εισαγωγή και μία διαγραφή. Τότε παρόμοια με τις εισαγωγές και διαγραφές χρησιμοποιούνται και οι δύο πίνακες **inserted και deleted**
- Οι πίνακες inserted, deleted είναι προσβάσιμοι μόνο στα πλαίσια της σκανδάλης

Προσωρινοί Πίνακες σε Σκανδάλες (συν.)

- Παράδειγμα

ID	Name
1	John
2	Anne
3	Marius
4	Steven

```
INSERT INTO table  
VALUES(5,'Potter')
```

inserted

deleted

ID	Name
5	Potter

ID	Name
----	------

```
DELETE FROM table  
WHERE ID = 1
```

inserted

deleted

ID	Name
----	------

ID	Name
1	John

```
UPDATE table  
SET Name='Harry'  
WHERE ID = 5
```

inserted

deleted

ID	Name
5	Harry

ID	Name
5	Potter

ID	Name
1	John
2	Anne
3	Marius
4	Steven
5	Potter

ID	Name
2	Anne
3	Marius
4	Steven
5	Potter

ID	Name
2	Anne
3	Marius
4	Steven
5	Harry

Πληροφορίες για TRIGGERS

Άλλες Χρήσιμες Πληροφορίες για TRIGGERS

- Προσωρινή Απενεργοποίηση Σκανδάλης

ALTER TABLE <table-name>

<ENABLE | DISABLE> TRIGGER <ALL | <trigger-name>>

- Οι σκανδάλες μπορεί να καλούνται **αναδρομικά** μέχρι και **32 επίπεδα**.
- Η εκτέλεση μιας σκανδάλης μπορεί να προκαλέσει την **αλυσιδωτή εκτέλεση** άλλων σκανδαλών με απρόσμενα αποτέλεσμα
 - θυμηθείτε το **ON DELETE CASCADE** παράδειγμα το οποίο μπορούσε να σβήσει όλο τον πίνακα των EMPLOYEES.

Σκανδάλες και Διαδικαστικός Προγραμματισμός

- Οι Σκανδάλες σε μια βάση δεδομένων μπορεί να γίνουν αρκετά πιο ευφυείς με την χρήση **εντολών διαδικαστικού προγραμματισμού**.
- Για παράδειγμα μπορεί να ορίζονται **επαναλήψεις, συνθήκες έλεγχου, μεταβλητές, συναρτήσεις** και πάρα πολλά άλλα.
 - Αυτές οι δομές προγραμματισμού είναι μέρος επεκτάσεων της SQL (π.χ., TSQL, PL/SQL (Oracle)).
- Πρακτικά, οι σκανδάλες χρησιμοποιούνται για ελέγχους των δεδομένων ή ενημερώσεις όταν συμβεί κάποιο γεγονός
- Μερικές φορές είναι ο μόνος τρόπος να υλοποιήσουμε κάποια υπολογιζόμενα χαρακτηριστικά που συσχετίζουν πολλαπλούς πίνακες (π.χ., το ολικό ποσό σε ένα τιμολόγιο)

Σκανδάλες και Διαδικαστικός Προγραμματισμός

- **Επιβολή Σημασιολογικού Κανόνα Ακεραιότητας:**

- “Επιβεβαίωση ότι το credit rating του vendor είναι καλό εάν επιχειρήσει να γίνει εισαγωγή στον PurchaseOrderHeader πίνακα

CREATE TRIGGER *Purchasing.LowCredit* **ON** Purchasing.PurchaseOrderHeader
AFTER INSERT -- ορισμός πότε να εκτελείται το trigger αυτό

AS
IF EXISTS (SELECT *
FROM Purchasing.PurchaseOrderHeader p
JOIN **inserted** AS i ON p.PurchaseOrderID = i.PurchaseOrderID
JOIN Purchasing.Vendor AS v ON v.BusinessEntityID = p.VendorID
WHERE v.CreditRating = 5
)

Πίνακας με αντικείμενα
υπο εισαγωγή (temp
πίνακας)

BEGIN

RAISERROR ('Vendor"s credit rating is too low to accept new purchases.', 16, 1);

ROLLBACK TRANSACTION;

RETURN

END;

Ευρετήρια (Indexes) σε TSQL

- Τα **ευρετήρια (indexes)** χρησιμοποιούνται από τους DBA για να αυξήσουν την ταχύτητα των επερωτήσεων
 - **Γρήγορη εύρεση κάποιας πλειάδας** χωρίς να χρειάζεται να ψάξουμε όλες τις άλλες πλειάδες
 - Χρησιμοποιεί **δομές και αλγόριθμους** που έχουμε μάθει στο ΕΠΛ231 (binary search, balanced trees, b-trees)
- Οι επιπρόσθετες δομές που χρησιμοποιούν τα ευρετήρια χρειάζονται **επιπρόσθετο χώρο στο δίσκο**
- Δημιουργούνται με βάση **μίας ή περισσότερων στηλών** και **δημιουργούν μία σειρά δεδομένων** στην οποία γίνεται η αναζήτηση

Κατηγορίες Ευρετηρίων

- **Clustered**
 - Το πιο γρήγορο είδος ευρετηρίου
 - Αλλάζουν την φυσική σειρά αποθήκευσης στα block δεδομένων ώστε να αποθηκεύονται σύμφωνα με το ευρετήριο
 - **Μόνο ένα clustered index** μπορεί να υπάρχει για ένα πίνακα (π.χ., το PRIMARY KEY)
 - Το επόμενο και προηγούμενο record είναι πιο γρήγορα διαθέσιμα (το πιο πιθανό στο ίδιο data block) → γρήγορη ανάκτηση
- **Non-clustered**
 - Τα δεδομένα είναι αποθηκευμένα σε άλλη φυσική σειρά – **τα δευτερεύων καθορίζουν μία εναλλακτική λογική σειρά**
 - Π.χ., primary key PERSON_ID, secondary index on PERSON_PASSPORT
 - **Δεν αλλάζουν την φυσική σειρά αποθήκευσης**
 - Μπορούμε να έχουμε **πολλά non-clustered indexes**
 - Τα δημιουργούμε αν χρειάζεται να **αυξήσουμε την απόδοση επερωτήσεων**
- **Cluster**
 - Χρησιμοποιούνται όταν έχουμε πολλούς πίνακες ή πολλές βάσεις
 - Τα ευρετήρια αποθηκεύουν συσχετιζόμενα records σε κοντινά data blocks
 - Αυξάνουν την απόδοση των JOINS

Σύνταξη Ευρετηρίων

- Επιλεκτικές παράμετροι μόνο
- **CREATE** [UNIQUE] [CLUSTERED | NONCLUSTERED] **INDEX**
<index_name>
ON <object> (column [ASC | DESC] [,...n])
[INCLUDE (column_name [,...n])]
[WHERE <filter_predicate>]
[WITH (<relational_index_option> [,...n])]
- Παράδειγμα δημιουργίας Index
CREATE NONCLUSTERED INDEX IX_PERSON_PASSPORT
ON PERSON (PERSON_PASSPORT);

Διαδικαστικός Προγραμματισμός σε SQL

- Ενώ η **SQL** ξεκίνησε ως **δηλωτική γλώσσα** διατύπωσης ερωτήσεων στη συνέχεια επεκτάθηκε με **εντολές DDL** και οι σύγχρονες ΒΔ παρέχουν σήμερα και δυνατότητες για διαδικαστικό προγραμματισμό ΜΕΣΑ στην ίδια την βάση.
- Γενικά, υπάρχουν οι ακόλουθες κατηγορίες:
 - **Εσωτερικός Προγραμματισμός:** Sprocs, UDFs, Views, Triggers, Assertions, Scripts/Batches, κτλ
 - **Εξωτερικός Προγραμματισμός:** Μέσω *Γλώσσας Προγραμματισμού* (Host Language): Embedded SQL , Dynamic SQL, APIs και *Διεπαφών Βάσεων* (ODBC, JDBC, MS OLEDB, ADO.NET κτλ).

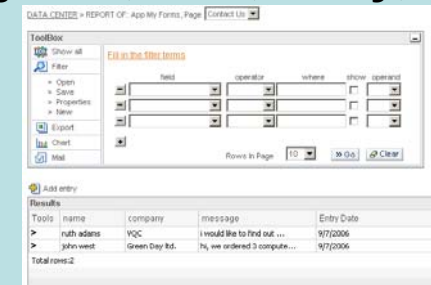
Προγραμματισμός Λειτουργιών μιας DBMS

Εξωτερικός Προγραμματισμός
(Μέσω Εφαρμογής)

Programming Language
(π.χ., JAVA, C, C#,...)

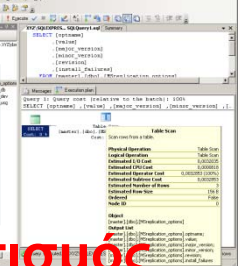
Interfaces:
JDBC,
ODBC, κτλ.

Web Languages (PHP,
Python, ASP, Ruby, Perl)



Administration Consoles

```
# SQLCMD
1> SELECT *
FROM EMPLOYEE
2> go
```



Εξωτερικός Προγραμματισμός
(Διαδραστικός)

DBMS

SProcs

UDFs

Views

Catalog/ Indexes

Tables

Triggers

Εσωτερικός Προγραμματισμός

Διαδικαστικός Προγραμματισμός σε SQL

- Για να μπορέσουμε να προγραμματίσουμε ένα DBMS χρειαζόμαστε τα κατάλληλα εργαλεία:
 - **εργαλεία που παρέχουν οι παραδοσιακές γλώσσες διαδικαστικού προγραμματισμού** (π.χ., variables, if then else, while, break, continue, try catch)
 - **και κάποια επιπρόσθετα εργαλεία που έχουν σχέση με τους πίνακες και τους περιορισμούς** (π.χ., cursors, errors)
- Χρησιμοποιώντας αυτά τα εργαλεία, επεκτείνουμε τις δυνατότητες προγραμματισμού μίας ΒΔ και συνεπώς μπορούμε να **μεταφέρουμε λογική της εφαρμογής σε επίπεδο βάσης**

Ενδιάμεσοι Πίνακες σε TSQL

- Οι **Τοπικοί Ενδιάμεσοι Πίνακες** δημιουργούνται στα πλαίσια μιας σύνδεσης με την βάση (Session) και καταστρέφονται αμέσως μετά.
 - Είναι χρήσιμοι για προσωρινή αποθήκευση ενδιάμεσων αποτελεσμάτων σε προγράμματα TSQL
 - **Δημιουργία:** CREATE TABLE #tablename (...)
 - Σημείωση: Ο πίνακας δημιουργείται στην tempdb
 - **Εισαγωγή Δεδ.:** INSERT INTO #TemporaryTable
EXEC sp_SomeStoredProcedure
- **Καθολικοί Ενδιάμεσοι Πίνακες:**
 - Έχουν αντίστοιχη λειτουργία με τους Τοπικούς, με την διαφορά ότι σε αυτούς έχουν πρόσβαση όλες οι συνδέσεις με την βάση (sessions).
 - **Δημιουργία:** CREATE TABLE ##tablename (...)