Ειδικότητα: Τεχνικός Εφαρμογών Πληροφορικής

Γ' Εξάμηνο

Μάθημα:

Βάσεις Δεδομένων Ι (2Ε)

8η-9η Ενότητα:

Συναλλαγές & Ενσωματωμένη SQL

Συναλλαγή (Transaction)

- Ένα ΣΔΒΔ διαθέτει μια γλώσσα υψηλού επιπέδου για την υποβολή αιτημάτων προσπέλασης και για την ενημέρωση των δεδομένων, στην οποία γράφονται τα αντίστοιχα προγράμματα των χρηστών του.
- Για να γίνει αντιληπτό το πώς το ΣΔΒΔ διαχειρίζεται τα αιτήματα αυτά, σε σχέση με τις έννοιες του ταυτοχρονισμού και της επαναφοράς, θεωρούμε πως η εκτέλεση του προγράμματος ενός χρήστη, ή αλλιώς **συναλλαγή**, είναι μια σειρά από ενέργειες **ανάγνωσης** και **εγγραφής** αντικειμένων της ΒΔ:
  - *Για να διαβαστεί ένα αντικείμενο της ΒΔ, πρώτα μεταφέρεται στην Κύρια Μνήμη από το δίσκο και μετά η τιμή του αντιγράφεται σε μια μεταβλητή του προγράμματος.*
  - *Για να γραφεί ένα αντικείμενο της ΒΔ, πρώτα ενημερώνεται η τιμή ενός αντιγράφου του που βρίσκεται στη μνήμη και μετά αυτό γράφεται στο δίσκο.*

Συναλλαγή (Transaction)

- Με τον όρο **συναλλαγή**, επομένως, εννοούμε ένα σύνολο λειτουργιών/ενεργειών ενημέρωσης, διαγραφής ή εισαγωγής γραμμών στη ΒΔ, το οποίο αποτελεί μια ενιαία λογική λειτουργική μονάδα.
- Συγκεκριμένα, εντάσσουμε σε μια συναλλαγή όλες εκείνες τις ενέργειες διαγραφής, ενημέρωσης ή εισαγωγής γραμμών που πρέπει να εκτελεστούν μαζί επιτυχώς.
- Αν τουλάχιστον μία ενέργεια αποτύχει, τότε καμία από τις άλλες επιτυχημένες ενέργειες δε θα γίνει αποδεκτή (δε θα γίνει **commit**). Αντίθετα, η ΒΔ θα επιστρέψει στην αρχική της κατάσταση, σαν να μην είχε συμβεί καμία ενέργεια (**rollback**).

Ιδιότητες Συναλλαγών (ACID)

- Ένα ΣΔΒΔ πρέπει να εξασφαλίζει 4 σημαντικές ιδιότητες για τις συναλλαγές, ώστε να προστατεύει τα δεδομένα του στην περίπτωση ταυτόχρονης χρήσης ή βλάβης του συστήματος.
  1. **Ατομικότητα** (**A**tomicity): είτε όλες οι ενέργειες μιας συναλλαγής ολοκληρώνονται είτε καμία.
  2. **Συνέπεια** (**C**onsistency): η εκτέλεση μιας συναλλαγής μόνη της, χωρίς την ταυτόχρονη εκτέλεση άλλων συναλλαγών, πρέπει να διατηρεί τη συνέπεια της ΒΔ.
  3. **Απομόνωση** (**I**solation): ακόμα κι αν τρέχουν πολλές συναλλαγές ταυτόχρονα, κάθε συναλλαγή πρέπει να νομίζει ότι τρέχει μόνη της, «απομονωμένη» από την επίδραση του ταυτόχρονου χρονοπρογραμματισμού.
  4. **Μονιμότητα** (**D**urability): αν μια συναλλαγή ολοκληρωθεί επιτυχώς, πρέπει το αποτέλεσμά της να επιβιώνει, ακόμα κι αν καταρρεύσει το σύστημα.

Καταστάσεις Συναλλαγών

1. **Active**: κατά την έναρξη και κατά τη διάρκειά της. Η συναλλαγή παραμένει σ' αυτή την κατάσταση ενώ εκτελείται.
2. **Partially committed**: όταν έχει εκτελεστεί η τελευταία εντολή της.
3. **Failed**: όταν το ΣΔΒΔ αντιληφθεί ότι δεν μπορεί να συνεχιστεί η ομαλή εκτέλεσή της.
4. **Aborted**: όταν η αποτυχημένη συναλλαγή έχει αναιρεθεί από το σύστημα (rolled back) και η ΒΔ έχει επαναφερθεί στην (συνεπή) κατάσταση στην οποία ήταν πριν την έναρξη της συναλλαγής.
5. **Committed**: όταν η συναλλαγή έχει ολοκληρωθεί επιτυχώς και η ΒΔ είναι σε συνεπή μορφή.

Παραδείγματα Συναλλαγών

Παράδειγμα 1:

- Στον παρακάτω πίνακα category επιχειρείται να αλλάξει η τιμή του πεδίου cat_num με την έναρξη δύο συναλλαγών.

```
START TRANSACTION;
UPDATE category
SET cat_num=1600
WHERE cat_id=3;

START TRANSACTION;
UPDATE category
SET cat_num=1200
WHERE cat_id=3;
```

cat_id	cat_name	cat_info	cat_num
1	comedy	comedy is	100
2	thriller	thriller is	150
3	adventure	adventure stands for....	1500

Αποτέλεσμα:

cat_id	cat_name	cat_info	cat_num
1	comedy	comedy is	100
2	thriller	thriller is	150
3	adventure	adventure stands for....	1600

Παραδείγματα Συναλλαγών

Παράδειγμα 1:

- Παρατηρούμε ότι η δεύτερη συναλλαγή δεν έχει πραγματοποιηθεί.
- Αυτό οφείλεται σε μια τεχνική ελέγχου του ταυτοχρονισμού που ονομάζεται **κλείδωμα**. Όταν μια συναλλαγή χρειάζεται διαβεβαίωση ότι κάποιο αντικείμενο ενδιαφέροντός της δε θα μεταβληθεί απρόβλεπτα, η συναλλαγή αποκτά ένα κλείδωμα σε αυτό το αντικείμενο.
- Το αποτέλεσμα του κλειδώματος είναι να αποκλειστούν οι άλλες συναλλαγές από το αντικείμενο (αυτόματο ROLLBACK) κι έτσι να μην μπορούν να το μεταβάλλουν.
- Για να εκτελεστεί η δεύτερη συναλλαγή, πρέπει πρώτα να κλείσει η πρώτη συναλλαγή με μια δήλωση **COMMIT** ή **ROLLBACK**. Τότε η δεύτερη θα ξεμπλοκάρει και θα εκτελεστεί κι η ίδια.

Παραδείγματα Συναλλαγών

Παράδειγμα 1:

- Δήλωση COMMIT

```
START TRANSACTION;
UPDATE category
SET cat_num=1600
WHERE cat_id=3;
COMMIT;
```

```
START TRANSACTION;
UPDATE category
SET cat_num=1200
WHERE cat_id=3;
COMMIT;
```

cat_id	cat_name	cat_info	cat_num
1	comedy	comedy is	100
2	thriller	thriller is	150
3	adventure	adventure stands for....	1500

Αποτέλεσμα:

cat_id	cat_name	cat_info	cat_num
1	comedy	comedy is	100
2	thriller	thriller is	150
3	adventure	adventure stands for....	1200

Παραδείγματα Συναλλαγών

Παράδειγμα 2:

- Στον παρακάτω πίνακα category επιχειρείται η ακύρωση της συναλλαγής που επιχειρεί να αλλάξει την τιμή του πεδίου cat_num.

```
START TRANSACTION;
UPDATE category
SET cat_num=2000
WHERE cat_id=3;
ROLLBACK;
```

cat_id	cat_name	cat_info	cat_num
1	comedy	comedy is	100
2	thriller	thriller is	150
3	adventure	adventure stands for....	1500

Αποτέλεσμα:

cat_id	cat_name	cat_info	cat_num
1	comedy	comedy is	100
2	thriller	thriller is	150
3	adventure	adventure stands for....	1500

Επίπεδο Απομόνωσης

- Ο όρος αυτός χρησιμοποιείται για να υποδηλώσει αυτό που λέμε βαθμό παρεμβολής τον οποίο μπορεί να ανεχθεί μια δεδομένη συναλλαγή όσον αφορά τις ταυτόχρονες συναλλαγές.

- Η ειδική εντολή **SET TRANSACTION** χρησιμοποιείται για να ορίσει κάποια χαρακτηριστικά της νέας συναλλαγής που θα ξεκινήσει, μεταξύ των οποίων τον **τρόπο προσπέλασης** και το **επίπεδο απομόνωσης**.

Εντολή SET TRANSACTION

Σύνταξη:

```
SET     [GLOBAL | SESSION]
        TRANSACTION
        [READ ONLY  |  READ WRITE]
        ISOLATION LEVEL
        { READ UNCOMMITTED      |
          READ COMMITTED        |
          REPEATABLE READ       |
          SERIALIZABLE }
```

Τρόπος προσπέλασης:

- **READ ONLY**: μόνο για ανάγνωση
- **READ WRITE**: για ανάγνωση και εγγραφή (Προκαθορισμένος τρόπος)

Επίπεδο απομόνωσης:

- **READ UNCOMMITTED**: επιτρέπει σε μια συναλλαγή να βλέπει ενδιάμεσα αποτελέσματα άλλων, πριν αυτές ολοκληρωθούν.
- **READ COMMITTED**: επιτρέπει σε μια συναλλαγή να βλέπει ενδιάμεσα αποτελέσματα άλλων, όταν αυτά γίνουν COMMITTED.

Εντολή SET TRANSACTION

Σύνταξη:

```
SET       [GLOBAL | SESSION]
          TRANSACTION
          [READ ONLY  |  READ WRITE]
          ISOLATION LEVEL
          { READ UNCOMMITTED        |
            READ COMMITTED          |
            REPEATABLE READ         |
            SERIALIZABLE }
```

Τρόπος προσπέλασης:

- **READ ONLY**: μόνο για ανάγνωση
- **READ WRITE**: για ανάγνωση και εγγραφή (Προκαθορισμένος τρόπος)

Επίπεδο απομόνωσης:

- **REPEATABLE READ**: επιτρέπει σε μια συναλλαγή να βλέπει μόνο αλλαγές που έγιναν από ολοκληρωμένες συναλλαγές.
- **SERIALIZABLE**: όμοια με άνω, αλλά δεν επιτρέπεται να εξακολουθεί να βλέπει «φαντάσματα» (καταστάσεις που έχουν όμως αλλάξει από άλλες συναλλαγές).

Παραδείγματα Συναλλαγών

Παράδειγμα 3:

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
START TRANSACTION;
UPDATE category
SET cat_num=3000
WHERE cat_id=3;

START TRANSACTION;
UPDATE category
SET cat_num=3500
WHERE cat_id=3;
COMMIT;
```

cat_id	cat_name	cat_info	cat_num
1	comedy	comedy is	100
2	thriller	thriller is	150
3	adventure	adventure stands for....	1500

Αποτέλεσμα:

cat_id	cat_name	cat_info	cat_num
1	comedy	comedy is	100
2	thriller	thriller is	150
3	adventure	adventure stands for....	3500

Ενσωματωμένη SQL

- Το σχεσιακό ΣΔΒΔ υποστηρίζει ένα περιβάλλον **διαδραστικής** διεπαφής, με χρήση του οποίου οι χρήστες μπορούν και διατυπώνουν εντολές σε γλώσσα SQL.
- Ανάγκη για περισσότερη ευελιξία: γενικής χρήσης προγραμματιστικός κώδικας σε συνδυασμό με την παραγωγικότητα της SQL στη διαχείριση και στην επεξεργασία της πληροφορίας.
- Π.χ. μπορεί να θέλουμε να ενισχύσουμε μια εφαρμογή ΒΔ με μία χρηστική διεπαφή η οποία είναι γραφικού τύπου και ευχάριστη στη χρήση.

Ενσωματωμένη SQL

- Προσπέλαση μιας ΒΔ μέσα από μια γλώσσα προγραμματισμού γενικού σκοπού απαιτείται τουλάχιστον γιατί:
  - υπάρχουν ερωτήσεις που δε μπορούν να διατυπωθούν σε SQL, γιατί η SQL δεν έχει όλες τις δυνατότητες μιας γλώσσας προγραμματισμού γενικού σκοπού,
  - μη-δηλωτικές εντολές (π.χ. εκτύπωση, επικοινωνία με το χρήστη) δε μπορούν να γίνουν μέσω της SQL.

- Ενσωμάτωση της SQL σε μια γλώσσα που καλείται **φιλόξενη** (host).

Ενσωματωμένη SQL

- Σε αυτήν την περίπτωση, η επεξεργασία των ερωτήσεων γίνεται από τη ΒΔ, και το αποτέλεσμα γίνεται διαθέσιμο στο πρόγραμμα **μια πλειάδα τη φορά**.
- Ένας ειδικός **προ-επεξεργαστής** (preprocessor) αντικαθιστά τον ενσωματωμένο κώδικα της SQL με δηλώσεις και κλήσεις συναρτήσεων στη host γλώσσα και μεταφράζεται το πρόγραμμα.
- Σύνταξη της μορφής:

  EXEC SQL <embedded SQL statement> END-EXEC
- Η ακριβής σύνταξη εξαρτάται από τη host γλώσσα.

Ενσωματωμένη SQL

- Χρησιμοποιούμε την εντολή **SQL INCLUDE**, για να δηλώσουμε στον προ-επεξεργαστή πού πρέπει να εισάγει τις δηλώσεις των μεταβλητών της SQL.
- Μεταβλητές της γλώσσας μπορεί να χρησιμοποιηθούν στην εντολή της SQL αν το σύμβολο : προηγείται του ονόματός τους.
- Παράδειγμα δήλωσης **δρομέα** (cursor):

```
EXEC SQL
        declare c cursor for
        select Όνομα-Πελάτη, Πόλη
        from Κατάθεση, Πελάτης
        where Κατάθεση.Όνομα-Πελάτη = Πελάτης.Όνομα-Πελάτη
                and Κατάθεση.Ποσό > :amount
END-EXEC
```

Δρομέας (Cursor)

- Πρόκειται για ένα μηχανισμό που καθιστά πραγματοποιήσιμη τη μία-προς-μία ανάκτηση των πλειάδων ενός πίνακα.
- Απαραίτητος σε host γλώσσες που δεν υποστηρίζουν συνολοειδή τύπο επεξεργασίας δεδομένων.
- Αφού δηλώσουμε ένα δρομέα, μπορούμε να:
  - τον **ενεργοποιήσουμε** (**OPEN**: τοποθετείται αμέσως πριν από την πρώτη πλειάδα),
  - **ανακτήσουμε** με αυτόν (**FETCH**: ανακτά την επόμενη πλειάδα),
  - τον **μετακινήσουμε** (σε επόμενες ή προηγούμενες θέσεις με ακέραιο αριθμό βημάτων),
  - τον **απενεργοποιήσουμε** (**CLOSE**).

Δρομέας (Cursor)

- Ενεργοποίηση δρομέα c:

  EXEC SQL OPEN c **END-EXEC**

- Ανάκτηση πλειάδων μέσα σε μεταβλητές της host γλώσσας:

  EXEC SQL FETCH c **INTO** :cn, :cc **END-EXEC**

Για την ανάκτηση όλων των πλειάδων αρκεί η επαναληπτική εκτέλεση της FETCH (π.χ. με ένα βρόχο while σε γλώσσα C).

- Απενεργοποίηση δρομέα:

  EXEC SQL CLOSE c **END-EXEC**

Ενσωματωμένη SQL

Ερωτήματα διαχείρισης:

```
EXEC SQL INSERT INTO customers VALUES (:c_sname,
    :c_sid, :c_age) END-EXEC

EXEC SQL SELECT sname, age
         INTO :c_sname, :c_age
         FROM customers
         WHERE customers.sid = :c_sid    END-EXEC

EXEC SQL DELETE FROM customers
         WHERE customers.sid = :c_sid    END-EXEC
```