ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

ΜΕΡΟΣ ΠΕΜΠΤΟ

Triggers, Stored procedures

Γιώργος Μαρκομανώλης

Περιεχόμενα

Triggers

Ενημέρωση δεδομένων άλλων πινάκων

Αν θέλουμε κατά την καταχώρηση ενός φοιτητή στον πίνακα students, τότε να καταχωρούνται αυτόματα σαν βαθμολογίες του στον πίνακα grades μηδέν, σε ένα trigger που λειτουργεί για καταχώρηση και ενημέρωση στοιχείων, τότε φτιάχνουμε την εξής trigger:

```
CREATE TRIGGER STUDENTS_BIU0 FOR STUDENTS
ACTIVE BEFORE INSERT OR UPDATE POSITION 0
AS
declare variable mitroo integer;
begin
IF (INSERTING) then
BEGIN
mitroo=new.id;
END
insert into grades values (:mitroo, 111,0);
end^
```

Παρατήρηση: Ο παραπάνω κώδικας δουλεύει σωστά όταν δημιουργείται την trigger από το πρόγραμμα **ib expert**.

ΠΡΟΣΟΧΗ: Βάζουμε **:** πριν από κάθε μεταβλητή όταν:

- Η μεταβλητή χρησιμοποιείται σε SQL ερώτημα.
- Η μεταβλητή παίρνει τιμή από εντολή του τύπου [FOR] SELECT … INTO.

Εξήγηση της trigger:
Το όνομα της trigger είναι ***STUDENTS_BIU0*** , εφαρμόζεται στον πίνακα ***STUDENTS***, ενεργοποιείται πριν την καταχώρηση και την ενημέρωση του πίνακα.
Δηλώνουμε την μεταβλητή ***mitroo*** που είναι ακέραιος.

Κύριο σώμα

Αν κάνουμε καταχώρηση δεδομένων τότε η μεταβλητή mitroo είναι ίση με τον καινούριο αριθμό μητρώου του φοιτητή που θέλουμε να καταχωρήσουμε .
Τέλος κύριου σώματος.

Καταχώρηση στον πίνακα **grades** της μεταβλητής μητρώο, τον κωδικό 111 και τον βαθμό μηδέν.
Τέλος.

Για exceptions θα μιλήσουμε αργότερα απλώς πρέπει να τις χρησιμοποιήσουμε για ένα παράδειγμα. Exceptions είναι τα μηνύματα σε περίπτωση που γίνουν κάποια λάθη στην βάση.

Έστω ότι θέλουμε να εμφανίζουμε ένα μήνυμα όταν προσπαθούμε να καταχωρήσουμε έναν φοιτητή στην βάση με αριθμό μητρώου που υπάρχει ήδη καταχωρημένο.

Δημιουργούμε ένα exception με το μήνυμα: Ο αριθμός μητρώου χρησιμοποιείται ήδη.

CREATE EXCEPTION ID_IS_IN_USE
'Ο αριθμός μητρώου χρησιμοποιείται ήδη';

Έπειτα προσαρμόζουμε την exception στην παραπάνω trigger:

```
CREATE TRIGGER STUDENTS_BIU0 FOR STUDENTS
ACTIVE BEFORE INSERT OR UPDATE POSITION 0
AS
declare variable mitroo integer;
begin
if (inserting) then
begin
if (exists(select 1 from students where id=new.id)) then
exception ID_IS_IN_USE;
else
mitroo=new.id;
end
insert into grades values (:mitroo, 111,0);
end^
```

Η εξήγηση είναι ίδια όπως πριν με την μόνη διαφορά ότι στο κύριο σώμα έχουμε μια εντολή **SELECT** που επιλέγουμε μόνο μια γραμμή αν υπάρχει φοιτητής με τον αριθμό μητρώου που προσπαθούμε να καταχωρήσουμε και αν υπάρχει, τότε εμφανίζεται το μήνυμα ***Ο αριθμός μητρώου χρησιμοποιείται ήδη*** και φυσικά τερματίζει η trigger.

Ασφάλεια

Αν θέλουμε να ορίσουμε μόνο τον χρήστη John να καταχωρεί και να κάνει διαγραφή δεδομένων από τον πίνακα students, τότε:

Δημιουργούμε την exception

CREATE EXCEPTION NO_AUTHORITY
'Δεν έχετε δικαιοδοσία';

Έπειτα φτιάχνουμε την trigger

```
CREATE TRIGGER STUDENTS_BID3 FOR STUDENTS
```

ACTIVE BEFORE INSERT OR DELETE POSITION 3
AS
begin
if (current_user<>'John') then
exception no_authority;
end^

Επεξήγηση:
Το όνομα της trigger είναι ***STUDENTS_BID3*** , εφαρμόζεται στον πίνακα ***STUDENTS***, ενεργοποιείται πριν την καταχώρηση και την διαγραφή από τον πίνακα.

Κύριο σώμα

Αν ο χρήστης που είναι συνδεμένος δεν είναι ο John τότε εμφανίζεται το μήνυμα ***Δεν έχετε δικαιοδοσία***
Τέλος.

Αλλαγή trigger

Για να αλλάξουμε το περιεχόμενο μιας trigger, δίνουμε την εντολή:

ALTER TRIGGER όνομα_trigger
και το περιεχόμενό της.

Διαγραφή trigger

DROP TRIGGER όνομα_trigger;

Stored Procedures

Μια αποθηκευμένη διαδικασία (stored procedure) είναι ένα πρόγραμμα γραμμένο σε PSQL και αποθηκευμένο σαν εκτελέσιμος κώδικας. Όταν μεταφραστεί μπορούμε να τον εκτελέσουμε από μια εφαρμογή ή από άλλο PSQL πρόγραμμα χρησιμοποιώντας EXECUTE PROCEDURE ή SELECT αναλόγως πως έχει δηλωθεί.
Οι αποθηκευμένες διαδικασίες δέχονται παραμέτρους εισόδου από εφαρμογές σαν στοιχεία και επιστρέφουν ένα σύνολο τιμών σαν αποτέλεσμα.

Executable stored procedures

Οι διαδικασίες που καλούνται με EXECUTE PROCEDURE μπορούν να επιστρέψουν επιλεκτικά μια γραμμή. Χρησιμοποιούνται συνήθως για διαδικασίες εισαγωγής, ενημέρωσης και διαγραφής δεδομένων.

Selectable stored procedures

Αυτές οι διαδικασίες επιστρέφουν αποτελέσματα πολλών γραμμών τα οποία εμφανίζονται με SELECT από τον εικονικό πίνακα.

Δημιουργία stored procedure

Σύνταξη εντολής:

CREATE PROCEDURE όνομα
[(λίστα παράμετρων εισόδου)]
[RETURNS (λίστα στοιχείων εξόδου)]
AS
<σώμα διαδικασίας>
<σώμα διαδικασίας>=
[DECLARE [VARIABLE] var data-type;
BEGIN
Εντολές
END

Στοιχεία κεφαλής

Στην κεφαλή συμπληρώνουμε:

- Το όνομα της αποθηκευμένης διαδικασίας που είναι μοναδικό για κάθε βάση
- Λίστα παράμετρε εισόδου αν υπάρχει μαζί με τον τύπο δεδομένων τους.

```
CREATE PROCEDURE test
(number1 integer, start_date date)
```

- Το όνομα κάθε στοιχείου πρέπει να είναι μοναδικό σε κάθε αποθηκευμένη διαδικασία.

- Η λίστα εξόδου αποτελείται από τα ονόματα των στοιχείων με τον τύπο δεδομένων τους μετά από την εντολή RETURNS.

```
CREATE PROCEDURE test
(number1 integer, start_date date)
RETURNS (number2 integer, name varchar(20))
```

- Η λέξη AS που απαιτείται πριν το κύριος σώμα της αποθηκευμένης διαδικασίας.

<u>Στοιχεία σώματος</u>

- Δηλώνουμε όλες τις τοπικές μεταβλητές.

```
DECLARE VARIABLE number3 INTEGER DEFAULT 0;
```

- Ο κύριος κώδικας γράφεται μεταξύ του BEGIN και END.

Μεταξύ του BEGIN και END μπορούμε να έχουμε:

1. Εντολές που να αναθέτουν τιμές στις τοπικές μεταβλητές.

2. SELECT εντολές για να βρούμε τιμές από κάποιες στηλές και να καταχωρηθούν σε κάποιες μεταβλητές.

3. Βρόγχους όπως FOR SELECT….DO και WHILE…DO.

4. IF…THEN…ELSE.

5. EXECUTE PROCEDURE για να καλέσουμε άλλες procedures.

6. EXCEPTIONS.

7. WHEN εντολή για το χειρισμό συγκεκριμένων λαθών

8. POST_EVEN εντολή για την προσθήκη μιας υπενθύμισης ενός γεγονότος.

9. SUSPEND για να σταματήσει η διαδικασία και να επιστραφούν οι τιμές.

Όταν δημιουργούμε μια stored procedure στο ib expert για να την εκτελέσετε πιέστε με το ποντίκι με το δεξί πλήκτρο πάνω στην διαδικασία και επιλέξτε Run Procedure.

Παράδειγμα

Έστω ότι θέλουμε, να πληκτρολογούμε τον αριθμό μητρώου ενός φοιτητή και να εμφανίζεται ένα κείμενο με τον αριθμό των μαθημάτων που έχει περάσει.

Δημιουργούμε 4 exceptions για τα μηνύματα από 0 μαθήματα ως 3.

```
CREATE EXCEPTION 0_LESSONS
'Έχει περάσει 0 μαθήματα';

CREATE EXCEPTION 1_LESSONS
'Έχει περάσει 1 μάθημα';

CREATE EXCEPTION 2_LESSONS
'Έχει περάσει 2 μαθήματα';

CREATE EXCEPTION 3_LESSONS
'Έχει περάσει 3 μαθήματα';
```

Η διαδικασία είναι η ακόλουθη:

```
CREATE PROCEDURE NEW_PROCEDURE (
   ID1 INTEGER)
AS
DECLARE VARIABLE lessons INTEGER;
begin
SELECT COUNT(GRAD) FROM grades2
WHERE ID=:id1 AND GRAD>=5
INTO :lessons;
if (lessons=0) then
exception 0_LESSONS;
if (lessons=1) then
exception 1_LESSONS;
if (lessons=2) then
exception 2_LESSONS;
if (lessons=3) then
exception 3_LESSONS;
  suspend;
end
```

Επεξήγηση:

Αρχικά δημιουργούμε τα μηνύματα που εμφανίζονται αναλόγως τον αριθμό μαθημάτων που έχουν περάσει οι φοιτητές.
Έπειτα δημιουργούμε την stored procedure με το όνομα ***NEW_PROCEDURE*** και δηλώνουμε ως μεταβλητές εισόδου την ***ID1*** *που είναι ακέραιος. Δηλώνουμε την μεταβλητή* ***lessons*** *που είναι και αυτή ακέραιος.*

Κύριο σώμα

Εντολή **SELECT** η οποία βρίσκει τον αριθμό των μαθημάτων για ένα συγκεκριμένο αριθμό μητρώου για τα οποία έχει βαθμό από 5 και άνω και τον αποθηκεύει τον αριθμό στην μεταβλητή ***lessons*** και στην συνέχεια ελέγχουμε αναλόγως με τον αριθμό των μαθημάτων που έχει περάσει ο φοιτητής να βγαίνει το κατάλληλο μάθημα.
Με την εντολή **suspend** σταματάει η διαδικασία και επιστρέφονται οι τιμές.
Τέλος.

Παράδειγμα

Έστω ότι θέλουμε να πληκτρολογούμε τον αριθμό μητρώου του φοιτητή. και να εμφανίζεται η βαθμολογία του σε κάθε μάθημα.

Δημιουργούμε την αποθηκευμένη διαδικασία:

```
CREATE PROCEDURE bathmoi (
   IDN INTEGER)
RETURNS (
   mathimata INTEGER,
   bathmoi INTEGER)
AS
begin
 FOR SELECT code,grade
  FROM grades
  where ID=:idn
  INTO :mathimata, :bathmoi
 DO
  SUSPEND;
End
```

Επεξήγηση:

Δημιουργούμε μια stored procedure με το όνομα bathmoi, μεταβλητή εισόδου ***IDN*** ακέραιος και μεταβλητές που επιστρέφονται από την διαδικασία οιι ***mathimata, bathmoi*** που είναι ακέραιες και οι δύο.

Κύριο σώμα

Εντολή **SELECT** που επαναλαμβάνεται και επιλέγει τον κωδικό του μαθήματος και τον βαθμό αντίστοιχα από τον πίνακα με τις βαθμολογίες για συγκεκριμένο αριθμό μητρώου και τα αποθηκεύει στις μεταβλητές ***mathimata, bathmoi***

DO (κάνει)
Suspend και επιστρέφονται οι τιμές κάθε φορά που γίνεται loop.
Τέλος.

Εμφάνιση των βαθμών σε μια συμβολοσειρά

```
CREATE PROCEDURE BATH (
   QW VARCHAR(40),
   ER VARCHAR(40))
RETURNS (
   LINE VARCHAR(1000))
AS
DECLARE VARIABLE ONE_LINE SMALLINT;
DECLARE VARIABLE STOP_ME SMALLINT;
begin
LINE='';
STOP_ME=1;
FOR EXECUTE statement 'SELECT '||QW||' FROM '||ER
INTO:ONE_LINE
DO BEGIN
IF(STOP_ME>2) THEN
SUSPEND;
EXIT;
IF(ONE_LINE IS NOT NULL) THEN
LINE=LINE||ONE_LINE||'';
STOP_ME=STOP_ME+1;
END
  suspend;
end
```

Επεξήγηση:

Δημιουργούμε διαδικασία με το όνομα ***BATH***.και δηλώνουμε ως μεταβλητές εισόδου τις ***QW*** και ***ER*** που είναι συμβολοσειρές μεταβλητού μήκους 40 χαρακτήρων. Η διαδικασία επιστρέφει την μεταβλητή ***LINE*** που είναι συμβολοσειρά μεταβλητού μήκους 1000 χαρακτήρων. Δηλώνουμε τις μεταβλητές ***ONE_LINE*** και ***STOP_ME*** σαν μικρός ακέραιος.

Κύριο σώμα

Θέτουμε αρχικές τιμές στο ***LINE*** την κενή συμβολοσειρά ‘’ και στην ***STOP_ME*** το 1.
Αρχίζει ένας βρόγχος που επαναλαμβάνει την εντολή ***SELECT*** και αποθηκεύει αυτό που βρίσκει στην μεταβλητή ***ONE_LINE***.
Επαναλαμβάνεται η διαδικασία:
Έλεγχος αν η τιμή της ***STOP_ME*** είναι μεγαλύτερη της 2, τότε
Επιστρέφει τα αποτελέσματα και σταματάει την διαδικασία.
Αν η μεταβλητή ***ONE_LINE*** δεν είναι κενή, τότε
Θέτουμε την ***LINE*** ίση με την ένωση της μεταβλητής ***LINE*** και ***ONE_LINE*** και αυξάνουμε την ***STOP_ME*** κατά ένα.
Τέλος κύριου σώματος.

Επιστροφή αποτελεσμάτων.

Τέλος.