



ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

ΜΕΡΟΣ ΠΕΜΠΤΟ

Stored procedures, Exceptions, Διακλώματα

Γιώργος Μαρκομανώλης

Περιεχόμενα

Stored Procedures	1
Exceptions	1
Εντολή When	2
Δικαιώματα	3
GRANT	3
Δικαιώματα REFERENCES σε στήλες	5
Δικαιώματα σε trigger, stored procedure και όψη	5
Roles	6
REVOKE	7

Stored Procedures

Μπορούμε μέσα από μια stored procedure, να καλέσουμε μια άλλη διαδικασία.

Δηλαδή αν θέλουμε να καλέσουμε την stored procedure **TEST** η οποία έχει ως λίστα εισόδου την μεταβλητή **mitroo** και λίστα εξόδου την μεταβλητή **onoma** δίνουμε την εντολή:

```
EXECUTE PROCEDURE TEST (:MITROO)  
RETURNING_VALUES (:ONOMA)
```

Λιαδικασία για εισαγωγή δεδομένων σε έναν πίνακα

```
CREATE PROCEDURE "INSERT" (  
NAME CHAR(10),  
SURNAME CHAR(10),  
MITROO INTEGER,  
START_DATE DATE)  
AS  
begin  
INSERT INTO STUDENTS VALUES (:name,:surname,:mitroo,:start_date);  
suspend;  
end^
```

Διαγραφή stored procedure

```
DROP PROCEDURE όνομα;
```

Exceptions

Exception είναι ένα μήνυμα που εμφανίζεται κατά την δημιουργία ενός λάθους στην βάση.

Δημιουργία exception

```
CREATE EXCEPTION όνομα 'μήνυμα':
```

Παράδειγμα

Για την δημιουργία μιας exception με όνομα `no_authority` και μήνυμα: `Δεν έχετε δικαιώματα`, δίνουμε την εντολή:

```
CREATE EXCEPTION NO_AUTHORITY 'Δεν έχετε δικαιώματα':  
COMMIT;
```

Αλλαγή exception

Για να αλλάζουμε το μήνυμα της exception **no_authority** και να έχει μήνυμα 'Δοκιμή' δίνουμε την εντολή:

```
ALTER EXCEPTION NO_AUTHORITY 'Δοκιμή';  
COMMIT;
```

Διαγραφή exception

```
DROP EXCEPTION NO_AUTHORITY;
```

Εντολή When

Η σύνταξη της **When** είναι η ακόλουθη:

```
WHEN όνομα_exception DO κώδικας
```

Η εντολή **WHEN** πρέπει να βρίσκεται μέσα στο block (block είναι η περιοχή μεταξύ ενός BEGIN και END) στο οποίο ορίζεται η exception που θέλει να χειριστεί.

Παράδειγμα

```
BEGIN  
IF (EXISTS(SELECT 1 FROM STUDENTS  
           WHERE ID=:ID1)) THEN  
  EXCEPTION TEST;
```

Κώδικας...

```
WHEN EXCEPTION TEST DO  
BEGIN  
Κώδικας...
```

```
END  
END
```

Δικαιώματα

GRANT

Με την εντολή **GRANT** δίνουμε δικαιώματα σε έναν χρήστη.
Γενική σύνταξη εντολής:

```
GRANT <δικαίωμα>  
ON <αντικείμενο>  
TO <χρήστη>;
```

Δικαίωμα	Πρόσβαση
SELECT	Ανάγνωση δεδομένων
INSERT	Δημιουργία νέων γραμμών
UPDATE	Αλλαγή δεδομένων
DELETE	Διαγραφή γραμμών
REFERENCES	Αναφορά σε ένα πρωτεύον κλειδί από ένα ξένο κλειδί. Είναι πάντα απαραίτητο για απόκτηση δικαιωμάτων σε πίνακες που περιέχουν ξένα κλειδιά.
ALL	Όλα τα παραπάνω δικαιώματα
EXECUTE	Εκτέλεση stored procedure
ROLE	Όλα τα δικαιώματα σε έναν role.

REVOKE

Με την εντολή **REVOKE** αφαιρούμε δικαιώματα από έναν χρήστη.
Γενική σύνταξη εντολής:

```
REVOKE <δικαίωμα>  
ON <αντικείμενο>  
FROM <χρήστη>;
```

ROLES

Ένας **role** είναι ένα σύνολο δικαιωμάτων, αντί να γράφουμε πολλά δικαιώματα, βάζουμε όσα θέλουμε σε ένα **role** και μετά το **role** είναι διαθέσιμο σαν δικαίωμα για τους χρήστες.

Αντικείμενο

Το αντικείμενο μπορεί να είναι πίνακας, όψη, stored procedure ή role.

Περιορισμοί Δικαιωμάτων

Τα δικαιώματα **SELECT**, **INSERT**, **UPDATE** και **DELETE** εφαρμόζονται μόνο σε αντικείμενα που είναι πίνακες ή όψεις. Το δικαίωμα **REFERENCES** εφαρμόζεται μόνο σε πίνακα και ειδικά σε αυτόν που αναφέρεται από ξένα κλειδιά. Το **EXECUTE** εφαρμόζεται μόνο σε stored procedure.

Παρατήρηση: Για να δημιουργήσουμε μια όψη πρέπει να έχουμε δικαίωμα **SELECT** για τους πίνακες που αποτελείται ή όψη.

Χρήστες

- **Roles σαν χρήστες:** Αν σε ένα role βάλουμε δικαιώματα τότε αυτόματα το role έχει τις ιδιότητες ενός δικαιώματος και μπορεί ένας χρήστης να συνδεθεί στην βάση με αυτό το role.
- **Όψεις σαν χρήστες:** Οι όψεις χρειάζονται άδεια για την πρόσβαση στους πίνακες, τις όψεις και τις stored procedures.
- **Triggers και procedures σαν χρήστες:** Οι stored procedures και οι triggers χρειάζονται δικαιώματα για τα αντικείμενα για τα οποία τα δηλώνουμε/

Γενική σύνταξη GRANT:

```
GRANT <δικαιώματα>  
ON [TABLE] <όνομα_πίνακα> | <όψη> | <αντικείμενο>  
TO <χρήστη>  
[WITH GRANT OPTION] | [WITH ADMIN OPTION];
```

Παρατήρηση: Τα δικαιώματα **UPDATE**, **REFERENCES** μπορούν να εφαρμοστούν και για συγκεκριμένες στήλες.

Η εντολή

WITH GRANT OPTION

Σημαίνει ότι ο χρήστης που αποχτάει αυτό το δικαίωμα μπορεί να το μεταβιβάσει και σε άλλους χρήστες.

Παραδείγματα

Για να δηλώσουμε ότι ο χρήστης John έχει δικαιώματα **SELECT**, **INSERT** και **UPDATE** στον πίνακα **STUDENTS**, τότε γράφουμε:

```
GRANT SELECT, INSERT, UPDATE ON STUDENTS TO JOHN;
```

Αν θέλουμε ο χρήστης Giannis να μπορεί να κάνει update μόνο την στήλη NAME και SURNAME από τον πίνακα STUDENTS, τότε δίνουμε την εντολή:

GRANT UPDATE (NAME, SURNAME) ON STUDENTS TO GIANNIS;

Παρατήρηση: Μπορούμε να αφαιρέσουμε το δικαίωμα του update από κάποια στήλη για τον χρήστη Giannis χωρίς να χρειαστεί να αφαιρεθεί εντελώς το Update από όλες τις στήλες.

Δικαιώματα REFERENCES σε στήλες

Το δικαίωμα REFERENCES χρειάζεται για να έχουμε δικαίωμα πρόσβασης σε πίνακα που περιέχει foreign key. Χρειάζεται για να μπορέσουμε να δημιουργήσουμε ένα ξένο κλειδί σε έναν πίνακα που αναφέρεται σε πίνακα για τον οποίο δεν έχουμε δικαιώματα πρόσβασης.

Για να δώσουμε στον χρήστη JOHN το δικαίωμα να δηλώσει ένα foreign key, το οποίο αναφέρεται στο πρωτεύον κλειδί του πίνακα STUDENTS, το ID, χωρίς να έχουμε δικαίωμα πρόσβασης, τότε δίνουμε την εντολή:

GRANT REFERENCES (ID) ON STUDENTS TO JOHN;

Δικαιώματα σε trigger, stored procedure και όψη

Όταν ένα αντικείμενο από τα trigger, stored procedure και όψη χρειάζεται δικαιώματα για την διαχείριση ενός πίνακα ή όψης πρέπει να δώσουμε τα κατάλληλα δικαιώματα. Μπορούμε αν θέλουμε να δώσουμε τα δικαιώματα του πίνακα σε μια stored procedure αντί για χρήστες για λόγους ασφαλείας και να δώσουμε στον χρήστη το δικαίωμα να εκτελεί την stored procedure.

Παράδειγμα

Αν έχουμε την stored procedure STUD και θέλουμε να της δώσουμε το δικαίωμα να κάνει INSERT δεδομένα στον πίνακα STUDENTS τότε δίνουμε την εντολή:

GRANT INSERT ON STUDENTS TO PROCEDURE STUD;

Δικαίωμα EXECUTE

Για να δώσουμε το δικαίωμα στον χρήστη JOHN να εκτελεί την stored procedure STUD, δίνουμε την εντολή:

GRANT EXECUTE ON PROCEDURE STUD TO JOHN;

Μπορούμε αντί για όνομα χρήστη να έχουμε άλλη stored procedure, trigger, view, role, λίστα χρηστών ή την λέξη PUBLIC.

Παραδείγματα

GRANT INSERT, UPDATE ON STUDENTS TO JOHN;

Για να μπορεί ο χρήστης JOHN να κάνει καταχώρηση και ενημέρωση των στοιχείων του πίνακα STUDENTS.

GRANT ALL ON STUDENTS TO JOHN;

Με την παραπάνω εντολή τα δικαιώματα SELECT, INSERT, UPDATE, DELETE και REFERENCES για τον πίνακα STUDENTS καταχωρούνται στον χρήστη JOHN.

Αν θέλετε η παραπάνω εντολή να καταχωρήσει δικαιώματα σε παραπάνω από έναν χρήστη, για παράδειγμα στους χρήστες JOHN, GIANNIS, δίνουμε την εντολή:

GRANT ALL ON STUDENTS TO JOHN, GIANNIS;

Ρόλοι (Roles)

Για να δημιουργήσουμε έναν ρόλο ακολουθούμε τα εξής βήματα:

1. Δημιουργούμε ένα ρόλο.
2. Καταχωρούμε δικαιώματα στον ρόλο.
3. Καταχωρούμε τον ρόλο σε ένα χρήστη.
4. Ορίζουμε τον ρόλο κατά την σύνδεση με την βάση.

Δημιουργία ρόλου

Για να δημιουργήσουμε ένα ρόλο με το όνομα TEST, δίνουμε την εντολή:

***CREATE ROLE TEST;
COMMIT;***

Για να καταχωρήσουμε δικαιώματα στον ρόλο, δίνουμε την εντολή:

GRANT ALL ON STUDENTS TO TEST;

Σαν να είναι ο TEST χρήστης.

Έπειτα καταχωρείται τον ρόλο σε έναν χρήστη, έστω τον JOHN με την εντολή:

GRANT TEST TO JOHN {WITH ADMIN OPTION};

Η εντολή WITH ADMIN OPTION σημαίνει ότι ο χρήστης μπορεί να μεταβιβάσει τον ρόλο και σε άλλον χρήστη.

Δηλαδή έχουμε WITH ADMIN OPTION σε ένα ρόλο και έχουμε WITH GRANT OPTION μόνο σε πίνακα ή όψη.

REVOKE

Παραδείγματα

1) ***REVOKE SELECT ON STUDENTS FROM JOHN;***

Αφαιρεί το δικαίωμα του SELECT για τον πίνακα STUDENTS από τον χρήστη JOHN.

2) ***REVOKE UPDATE ON STUDENTS FROM PROCEDURE STUD;***

Αφαιρεί το δικαίωμα της ενημέρωσης των στοιχείων για τον πίνακα STUDENTS από την stored procedure STUD.

3) ***REVOKE INSERT, UPDATE ON STUDENTS FROM JOHN;***

Αφαιρεί το δικαίωμα του INSERT, UPDATE για τον πίνακα STUDENTS από τον χρήστη JOHN.

4) ***REVOKE ALL ON STUDENTS FROM JOHN;***

Αφαιρεί το δικαίωμα του INSERT, DELETE, UPDATE, SELECT, REFERENCES για τον πίνακα STUDENTS από τον χρήστη JOHN.

5) ***REVOKE EXECUTE ON PROCEDURE STUD FROM JOHN;***

Αφαιρεί το δικαίωμα EXECUTE για την stored procedure από τον χρήστη JOHN.

6)
***REVOKE INSERT, UPDATE
ON STUDENTS
FROM JOHN, GIANNIS;***

Αφαιρεί το δικαίωμα UPDATE για τον πίνακα STUDENTS από τον χρήστη JOHN, GIANNI.

7)
***REVOKE UPDATE
ON STUDENTS
FROM TEST;***

Αφαιρεί το δικαίωμα UPDATE για τον πίνακα STUDENTS από τον ρόλο TEST.

8) **REVOKE TEST FROM JOHN;**

Αφαιρεί τον ρόλο TEST από τον χρήστη JOHN.

9) **REVOKE SELECT, INSERT, UPDATE
ON STUDENTS FROM PUBLIC;**

Αφαιρεί τα δικαιώματα SELECT, INSERT, UPDATE για τον πίνακα STUDENTS από όλους τους χρήστες.

10)

**REVOKE GRANT OPTION
FOR SELECT
ON STUDENTS
FROM JOHN;**

Αφαιρεί το δικαίωμα να μπορεί ο χρήστης JOHN να μεταφέρει το δικαίωμα SELECT για τον πίνακα STUDENTS σε άλλους χρήστες.