

**Ανοικτά Ακαδημαϊκά Μαθήματα**

**Τεχνολογικό Εκπαιδευτικό Ίδρυμα Αθήνας**

Βάσεις Δεδομένων ΙΙ (Ε)

**Ενότητα 2:** Ανασκόπηση χρήσης της Γλώσσας SQL – θέματα υλοποίησης Βάσεων Δεδομένων - Χρήση περιορισμών (constraints) – Χρήση Views

Χ. Σκουρλάς, Α. Τσολακίδης, Δ. Βάσσης

Τμήμα Μηχανικών Πληροφορικής ΤΕ

|  |  |
| --- | --- |
| Το περιεχόμενο του μαθήματος διατίθεται με άδεια Creative Commons εκτός και αν αναφέρεται διαφορετικά | Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους. |

Περιεχόμενα

[1. Δημιουργία βάσης δεδομένων, δημιουργία πινάκων, ορισμός ευρετηρίων, αλλαγή ορισμού (προσθήκη στηλών, τροποποίηση στηλών) κ.λπ. 4](#_Toc401499819)

[2. Παραδείγματα σύνταξης και χρήσης εντολών Υπογλώσσας Ορισμού Δεδομένων (Data Definition Language -DDL). 5](#_Toc401499820)

[3. Δημιουργία ευρετηρίου - CREATE INDEX 8](#_Toc401499821)

[4. Διαχείριση Περιορισμών (Constraints) 9](#_Toc401499822)

[5. Ο περιορισμός Primary Key 12](#_Toc401499823)

[6. Ο περιορισμός Foreign Key 14](#_Toc401499824)

[7. Ο περιορισμός DEFAULT 15](#_Toc401499825)

[8. Ο περιορισμός AUTO\_INCREMENT (στο προϊόν mySQL) 16](#_Toc401499826)

[9. Ο περιορισμός Check 17](#_Toc401499827)

[10. Ανασκόπηση των όψεων (VIEW) 19](#_Toc401499828)

Στόχος του δεύτερου εργαστηρίου είναι η ανασκόπηση των εντολών της γλώσσας SQL έτσι ώστε στα επόμενα εργαστήρια ο ενδιαφερόμενος να εστιάσει απρόσκοπτα σε θέματα υλοποίησης Βάσεων Δεδομένων με χρήση SQL και να προχωρήσει σε θέματα αποθηκευμένων διαδικασιών (stored procedures), διαχείρισης cursors, διαχείρισης δοσοληψιών (transactions processing) κ.λπ.

Μετά την επεξεργασία του εργαστηρίου ο ενδιαφερόμενος θα έχει επαναλάβει τα εξής σημαντικά θέματα:

* Δημιουργία βάσης δεδομένων, δημιουργία πινάκων, ορισμός ευρετηρίων, αλλαγή ορισμού (προσθήκη στηλών, τροποποίηση στηλών) κ.λπ.
* Οι τρεις Υπογλώσσες της γλώσσας SQL
* Δημιουργία πινάκων και Διαχείριση Περιορισμών (Constraints)

Επιπλέον, θα έχει εμπεδώσει θέματα χρήσης περιορισμών:

* Ο περιορισμός Not Null
* Ο περιορισμός Unique σε στήλη ή στήλες του πίνακα
* Ο περιορισμός Primary Key
* Ο περιορισμός Foreign Key
* Ο περιορισμός DEFAULT
* Ο περιορισμός AUTO\_INCREMENT (mySQL)
* Ο περιορισμός Check

Τέλος, θα έχει κάνει ανασκόπηση των όψεων (VIEW)

**Λέξεις κλειδιά**: Δημιουργία βάσης δεδομένων, δημιουργία πινάκων, ορισμός ευρετηρίων, αλλαγή ορισμού πινάκων, Υπογλώσσες της γλώσσας SQL, Δημιουργία πινάκων και Διαχείριση Περιορισμών (Constraints), Views

# Δημιουργία βάσης δεδομένων, δημιουργία πινάκων, ορισμός ευρετηρίων, αλλαγή ορισμού (προσθήκη στηλών, τροποποίηση στηλών) κ.λπ.

##  Οι τρεις Υπογλώσσες της γλώσσας SQL

Η γλώσσα SQL περιλαμβάνει τις παρακάτω υπογλώσσες:

### Υπογλώσσα ορισμού δεδομένων - Data Definition Language (DDL).

**Εντολές**

1. CREATE DATABASE
2. ALTER DATABASE
3. CREATE TABLE
4. ALTER TABLE
5. DROP TABLE
6. CREATE INDEX
7. CREATE UNIQUE INDEX
8. DROP INDEX
9. CREATE TRIGGER
10. DROP TRIGGER
11. REPLACE TRIGGER
12. CREATE PROCEDURE
13. DROP PROCEDURE
14. CREATE FUNCTION
15. DROP FUNCTION

### Υπογλώσσα Χειρισμού Δεδομένων - Data Manipulation Language (DML)

**Εντολές**

1. SELECT
2. UPDATE
3. DELETE
4. INSERT INTO

### Υπογλώσσα ελέγχου δεδομένων - Data Control Language (DCL)

Περιλαμβάνει εντολές που σχετίζονται με δικαιώματα

1. GRANT
2. REVOKE

## Διαχείριση δοσοληψιών

Περιλαμβάνει εντολές που σχετίζονται με τη διαχείριση δοσοληψιών

1. COMMIT
2. ROLLBACK

# Παραδείγματα σύνταξης και χρήσης εντολών Υπογλώσσας Ορισμού Δεδομένων (Data Definition Language -DDL).

1.
2.

## Δημιουργία βάσης δεδομένων: CREATE DATABASE

CREATE DATABASE database\_name

## Δημιουργία πινάκων της βάσης δεδομένων: CREATE TABLE

CREATE TABLE table\_name
(
 column\_name1 data\_type,
 column\_name2 data\_type,
 column\_name3 data\_type,

);

## Ποιοί τύποι δεδομένων χρησιμοποιούνται.

Εξαρτάται από το προϊόν. Για παράδειγμα, στην περίπτωση του προϊόντος της

ORACLE χρησιμοποιούνται οι τύποι δεδομένων:

* CHAR (Width) - μέχρι 255 χαρακτήρες
* VARCHAR2(Width) - μέχρι 2000 χαρακτήρες
* NUMBER - προσημασμένοι δεκαδικοί μέχρι 38 ψηφία
* NUMBER (Width) - όπως παραπάνω για W ψηφία
* NUMBER (Width, Digits) - όπως παραπάνω για W ψηφία ,τα D δεκαδικά
* DATE - ημερομηνίες (‘2007-12-31’)

Στο προϊόν Mysql οι τύποι δεδομένων διαφέρουν λίγο πχ.

number -> numeric, int, float

 varchar2->varchar

**Βιβλιογραφία**

<http://docs.oracle.com/cd/B28359_01/server.111/b28318/datatype.htm>

<http://ss64.com/ora/syntax-datatypes.html>

<http://dev.mysql.com/doc/refman/5.7/en/data-types.html>

## Παραδείγματα δημιουργίας πινάκων σε MySql

CREATE TABLE DEPT (DEPTNO NUMERIC(2),

 DNAME VARCHAR(24),

 LOC CHAR(23));

CREATE TABLE EMP (EMPNO NUMERIC (4) NOT NULL,

 ENAME VARCHAR (20),

 JOB VARCHAR (19),

 MGR NUMERIC (4),

 HIREDATE DATE,

 SAL NUMERIC (10,2),

 COMM NUMERIC (10,2),

 DEPTNO NUMERIC (2));

CREATE TABLE PROJ (PROJNO NUMERIC (3) NOT NULL,

 PNAME VARCHAR (15),

 BUDGET NUMERIC (12,2));

CREATE TABLE ASSIGN (EMPNO NUMERIC (4) NOT NULL ,

 PROJNO NUMERIC (3) NOT NULL,

 PTIME NUMERIC (2) ,BUDGET NUMERIC (12,2));

**Θυμίζουμε ότι η εισαγωγή στοιχείων γίνεται με εντολές του τύπου:**

INSERT INTO EMP

 VALUES(7512,’ΑΝΔΡΕΟΥ’,’ΚΛΗΤΗΡΑΣ’,7890,’20/11/1988’,

 1000,NULL,30);

Η εντολή INSERT ανήκει στην Υπογλώσσα Χειρισμού Δεδομένων ((DML)

## Αλλαγή ορισμού πίνακα με εντολή ALTER TABLE – Εντολή RENAME

**Προσοχή! Η εντολή ALTER TABLE (όπως και άλλες εντολές στη συνέχεια) διαφέρει από προϊόν σε προϊόν.**

ALTER TABLE table\_name

* ADD column\_name datatype
* MODIFY column\_name datatype
* DROP [COLUMN] col\_name
* RENAME TABLE tbl\_name TO new\_tbl\_name

 [, tbl\_name2 TO new\_tbl\_name2] ...

* CHANGE old\_col\_name new\_col\_name column\_definition

RENAME TABLE old\_table TO backup\_table,

 new\_table TO old\_table;

RENAME TABLE old\_table TO tmp\_table,

 new\_table TO old\_table,

 tmp\_table TO new\_table;

alter table
   table\_name
rename to
   new\_table\_name;

## Ορισμός πίνακα βασιζόμενος σε εντολή SELECT – Παραδείγματα

CREATE TABLE temporary

 (employee\_id, start\_date, end\_date, job\_id, dept\_id)

AS SELECT employee\_id, start\_date, end\_date, job\_id, department\_id

FROM job\_history;

DROP TABLE job\_history;

RENAME temporary TO job\_history;

ALTER TABLE customers

 RENAME COLUMN credit TO credit\_amount;

ALTER TABLE *tbl\_name* CONVERT TO CHARACTER SET *charset\_name*;

**Βιβλιογραφία**

[**http://dev.mysql.com/doc/refman/5.0/en/charset-conversion.html**](http://dev.mysql.com/doc/refman/5.0/en/charset-conversion.html)

[**http://docs.oracle.com/cd/E17952\_01/refman-5.1-en/alter-table.html**](http://docs.oracle.com/cd/E17952_01/refman-5.1-en/alter-table.html)

## Παραδείγματα προσθήκης στήλης σε πίνακα και τροποποίησης στήλης

Για να προσθέσουμε στο πίνακα ΕΜΡ τη στήλη BIRTHDATE:

ALTER TABLE EMP ADD (BIRTHDATE DATE);

Για να αλλάξουμε στον πίνακα ΕΜΡ τη στήλη ΕΝΑΜΕ :

ALTER TABLE EMP MODIFY (ENAME TEXT);

## Κατάργηση πίνακα DROP TABLE

DROP TABLE EMP;

# Δημιουργία ευρετηρίου - CREATE INDEX

Tα ευρετήρια (indexes) δημιουργούνται για έναν ή περισσότερους πίνακες προκειμένου να ανακτούμε τα δεδομένα γρηγορότερα. Τα ευρετήρια δεν είναι διάφανα στον χρήστη και χρησιμοποιούνται από το ΣΔΒΔ για να επιταχύνουν τις αναζητήσεις του χρήστη (users’ queries)

CREATE INDEX index\_name ON table\_name (column\_name)

1.
2.
3.

## Ευρετήρια για απλές στήλες πίνακα

Για να ορίσουμε το ευρετήριο INAME που θα βασίζεται στη στήλη ENAME στον πίνακα EMP:

CREATE INDEX INAME ON EMP (ENAME);

## Ευρετήρια για συνδυασμό στηλών πίνακα

Για να ορίσουμε το ευρετήριο SALCOM που θα βασίζεται στο συνδυασμό στηλών Μισθού (SAL) και προμήθειας (COMM):

CREATE INDEX SALCOM ON EMP (SAL,COMM);

**Προσοχή! Τα παρακάτω ευρετήρια είναι διαφορετικά και χρησιμοποιούνται σε διαφορετικές αναζητήσεις, δηλαδή βοηθούν στη γρηγορότερη εκτέλεση διαφορετικών SELECT δηλώσεων:**

CREATE INDEX SUR\_NAM ON EMP (SURNAME, NAME);

CREATE INDEX NAM\_SUR ON EMP (NAME, SURNAME);

## Κατάργηση ευρετηρίου - Εντολή DROP INDEX

DROP INDEX SALCOM ON EMP;

# **Διαχείριση** Περιορισμών (Constraints)

**Που χρειαζόμαστε τους περιορισμούς - Παράδειγμα**

Δημιουργήστε τον πίνακα Persons

CREATE TABLE Persons

(

P\_Id int,

LastName varchar(255),

FirstName varchar(255),

Address varchar(255),

City varchar(255)

);

**Παρατηρήστε ότι για τις στήλες του πίνακα δεν ορίσαμε περιορισμούς.**

**Κάνετε Εισαγωγή δεδομένων**

insert into persons(p\_id,lastname,firstname,address,city) values (1,'Papadopoulos','Marios','Axarnwn 5','Athens'), (1,'Petrou','Nikos','Patisiwn 12','Athens');

**Εμφάνιση των δεδομένων**

select\* from persons

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **p\_id** | **LastName** | **FirstName** | **Address** | **City** |
| 1 | Papadopoulos | Marios | Axarnwn 5 | Athens |
| 1 | Petrou | Nikos | Patisiwn 12 | Athens |

**Δηλαδή, εισάγουμε δύο διαφορετικά πρόσωπα με τον ίδιο κωδικό.**

**Διαγραφή των δεδομένων**

delete from persons where p\_id=1;

1.

## Ο περιορισμός Not Null

**Όταν μια στήλη ενός πίνακα οριστεί ως NOT NULL δεν μπορεί να δεχτεί NULL τιμές (values).**

**Παρατηρήστε ότι μπορούμε να προσθέσουμε εκ των υστέρων περιορισμούς NOT NULL.**

**Προσθήκη περιορισμού:**

alter table persons change p\_id p\_id INT(11) NOT NULL

**Εισαγωγή δεδομένων**

insert into persons(p\_id,lastname,firstname,address,city) values (1,'Papadopoulos','Marios','Axarnwn 5','Athens');

**Τι θα δούμε;**

|  |  |
| --- | --- |
| Action | Message |
| insert into persons(p\_id,lastname,firstname,address,city) values (1,'Papadopoulos','Marios','Axarnwn 5','Athens') | 1 row(s) affected |

insert into persons(lastname,firstname,address,city) values ('Petrou','Nikos','Patisiwn 12','Athens');

**Προσοχή! Με τη δήλωση insert into προσπαθήσαμε να εισάγουμε τιμή NULL στη στήλη p\_id.**

**Τι θα δούμε;**

|  |  |
| --- | --- |
| Action | Message |
| insert into persons(lastname,firstname,address,city) values ('Petrou','Nikos','Patisiwn 12','Athens') | Error Code: 1364. Field 'p\_id' doesn't have a default value |

**Τι συμπέρασμα βγάζετε;**

**Διαγραφή δεδομένων**

delete from persons where p\_id=1;

## Ο περιορισμός Unique σε στήλη του πίνακα

Ο περιορισμός Unique (UNIQUE constraint) εφαρμόζεται σε στήλη του πίνακα και σημαίνει ότι κάθε τιμή στη στήλη πρέπει να υπάρχει μόνο μια φορά. Δηλαδή, κάθε συγκεκριμένη τιμή αυτής της στήλης είναι μοναδική όπως θα συνέβαινε και αν είχαμε ορίσει ένα κύριο κλειδί για αυτή τη στήλη. Οι περιορισμοί UNIQUE και PRIMARY KEY (constraints) παρέχουν εγγύηση για τη μοναδικότητα μιας τιμής για μια στήλη ή ένα σύνολο στηλών.

**Προσοχή!** Μπορεί να έχετε πολλούς περιορισμούς UNIQUE constraints για κάθε πίνακα, αλλά μόνο ένα PRIMARY KEY constraint.

**Ακολουθεί παράδειγμα**

ALTER TABLE Persons
 ADD UNIQUE (P\_Id)

**Δημιουργήστε περιορισμό δηλώνοντας και το όνομά του.**

ALTER TABLE Persons
 ADD CONSTRAINT uc\_PersonID UNIQUE (P\_Id,LastName)

**Εισαγωγή δεδομένων**

insert into persons(p\_id,lastname,firstname,address,city) values (1,'Papadopoulos','Marios','Axarnwn 5','Athens');

**OUTPUT**

|  |  |
| --- | --- |
| Action | Message |
| insert into persons(p\_id,lastname,firstname,address,city) values (1,'Papadopoulos','Marios','Axarnwn 5','Athens') | 1 row(s) affected |

insert into persons(lastname,firstname,address,city) values ('Petrou','Nikos','Patisiwn 12','Athens');

**OUTPUT**

|  |  |
| --- | --- |
| Action | Message |
| insert into persons(p\_id,lastname,firstname,address,city) values (1,'Petrou','Nikos','Patisiwn 12','Athens'); | Error Code: 1062. Duplicate entry '1' for key 'p\_id\_UNIQUE' |

**Τι συμπέρασμα βγάζετε;**

**Διαγραφή δεδομένων**

delete from persons where p\_id=1;

# Ο περιορισμός Primary Key

**Παρατηρήστε ότι μπορούμε να προσθέσουμε εκ των υστέρων περιορισμούς για κύρια και ξένα κλειδιά.**

ALTER TABLE Persons
 ADD PRIMARY KEY (P\_Id);

**Δημιουργήστε περιορισμό δηλώνοντας και το όνομά του.**

ALTER TABLE Persons

 ADD CONSTRAINT pk\_PersonID PRIMARY KEY (P\_Id,LastName);

**Κατάργηση περιορισμού PRIMARY KEY**

ALTER TABLE Persons
 DROP PRIMARY KEY

ALTER TABLE Persons
 DROP CONSTRAINT pk\_PersonID;

**Εισαγωγή δεδομένων**

insert into persons(p\_id,lastname,firstname,address,city) values (1,'Papadopoulos','Marios','Axarnwn 5','Athens');

**OUTPUT**

|  |  |
| --- | --- |
| Action | Message |
| insert into persons(p\_id,lastname,firstname,address,city) values (1,'Papadopoulos','Marios','Axarnwn 5','Athens') | 1 row(s) affected |

insert into persons(lastname,firstname,address,city) values ('Petrou','Nikos','Patisiwn 12','Athens');

**OUTPUT**

|  |  |
| --- | --- |
| Action | Message |
| insert into persons(p\_id,lastname,firstname,address,city) values (1,'Petrou','Nikos','Patisiwn 12','Athens'); | Error Code: 1062. Duplicate entry '1' for key 'PRIMARY' |

**Τι συμπέρασμα βγάζετε;**

# Ο περιορισμός Foreign Key

**Θυμίζουμε τον κλασσικό ορισμό δημιουργώντας τον πίνακα Orders.**

CREATE TABLE Orders
(
O\_Id int NOT NULL,
OrderNo int NOT NULL,
P\_Id int,
PRIMARY KEY (O\_Id),
FOREIGN KEY (P\_Id) REFERENCES Persons(P\_Id)
);

select\* from persons;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **p\_id** | **LastName** | **FirstName** | **Address** | **City** |
| 1 | Papadopoulos | Marios | Axarnwn 5 | Athens |
| 2 | Petrou | Nikos | Patisiwn 12 | Athens |

**Εισαγωγή δεδομένων**

insert into orders values(1,1000,1),(2,1000,1),(3,1000,2);

**OUTPUT**

|  |  |
| --- | --- |
| Action | Message |
| insert into orders values(1,1000,1),(2,1000,1),(3,1000,2); | 3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0 |

insert into orders values(5,1000,3);

**OUTPUT**

|  |  |
| --- | --- |
| Action | Message |
| insert into orders values(5,1000,3); | Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`db2`.`orders`, CONSTRAINT `orders\_ibfk\_1` FOREIGN KEY (`P\_Id`) REFERENCES `persons` (`p\_id`)) |

SELECT \* FROM db2.orders;

|  |  |  |
| --- | --- | --- |
| **O\_Id** | **OrderNo** | **P\_Id** |
| 1 | 1000 | 1 |
| 2 | 1000 | 1 |
| 3 | 1000 | 2 |

**Τι συμπέρασμα βγάζετε;**

# Ο περιορισμός DEFAULT

Ο περιορισμός DEFAULT εφαρμόζεται σε μία στήλη του πίνακα και ορίζει μία προεπιλεγμένη τιμή για τη στήλη. Η δήλωση insert into μπορεί να αλλάξει αυτήν την τιμή ή να την αφήσει αμετάβλητη.

Ακολουθεί παράδειγμα.

CREATE TABLE dept
(
D\_Id int NOT NULL,
department varchar(90) DEFAULT 'Development',

primary key(d\_id)
);

**Εισαγωγή δεδομένων**

insert into dept(d\_id, department) values(1,'Marketing');

**OUTPUT**

|  |  |
| --- | --- |
| Action | Message |
| insert into dept(d\_id, department) values(1,'Marketing') | 1 row(s) affected  |

insert into dept(d\_id) values(2)

**OUTPUT**

|  |  |
| --- | --- |
| Action | Message |
| insert into dept(d\_id) values(2) | 1 row(s) affected  |

SELECT \* FROM db2.dept;

|  |  |
| --- | --- |
| **d\_Id** | **Department** |
| 1 | Marketing |
| 2 | Development |

**Τι συμπέρασμα βγάζετε;**

# Ο περιορισμός AUTO\_INCREMENT (στο προϊόν mySQL)

Ο περιορισμός AUTO\_INCREMENT εφαρμόζεται σε μία στήλη του πίνακα και διαχειρίζεται αυτόματα σαν αύξοντα αριθμό την τιμή της στήλη. Η δήλωση insert into δεν μπορεί να αλλάξει τιμή αυτή.

Ακολουθεί παράδειγμα.

CREATE TABLE Project

(

P\_Id int NOT NULL AUTO\_INCREMENT,

Project varchar(255),

PRIMARY KEY (P\_Id)

);

**Εισαγωγή δεδομένων**

insert into project(p\_id, Project) values(1,'OTE');

**OUTPUT**

|  |  |
| --- | --- |
| Action | Message |
| insert into project(p\_id, Project) values(1,'OTE'); | 1 row(s) affected  |

insert into project(Project) values('TEI');

**OUTPUT**

|  |  |
| --- | --- |
| Action | Message |
| insert into project(Project) values('TEI'); | 1 row(s) affected  |

SELECT \* FROM db2. Project;

|  |  |
| --- | --- |
| **d\_Id** | **Department** |
| 1 | OTE |
| 2 | TEI |

**Τι συμπέρασμα βγάζετε;**

**Δείτε και το παρακάτω παράδειγμα**

ALTER TABLE Project AUTO\_INCREMENT=100

insert into project(Project) values('Web Site');

SELECT \* FROM db2. Project;

|  |  |
| --- | --- |
| **d\_Id** | **department** |
| 1 | OTE |
| 2 | TEI |
| 100 | Web Site |

**Τι συμπέρασμα βγάζετε;**

# Ο περιορισμός Check

CREATE TABLE Persons
(
P\_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255),
CHECK (P\_Id>0)
);

CREATE TABLE Persons
(
P\_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255),
CONSTRAINT chk\_Person CHECK (P\_Id>0 AND City='Sandnes')
);

**Οι δηλώσεις CREATE TABLE που γράψατε είναι σωστές. Δοκιμάστε και δηλώσεις INSERT INTO. Τελικά ο περιορισμός CHECK δουλεύει στην περίπτωση της mySQL΄**

Η απάντηση είναι αρνητική μέχρι σήμερα για τις τρέχουσες versions του προιόντος! Αποδείξτε το με παραδείγματα!

Άσκηση

Καταγράψτε στον παρακάτω πίνακα τα συμπεράσματά σας:

|  |  |
| --- | --- |
| Περιορισμός | Διαχείριση Περιορισμών (Constraints) και συμπεράσματα στο προϊόν mySQL |
| Ο περιορισμός Not Null | Μπορούμε να προσθέσουμε εκ των υστέρων περιορισμούς NOT NULL για όποιες στήλες του πίνακα επιθυμούμε. Αν με τη δήλωση insert into προσπαθήσουμε να εισάγουμε τιμή NULL σε μία στήλη στην οποία έχουμε εφαρμόσει τον περιορισμό βλέπουμε ότι το προϊόν δεν το επιτρέπει.Αν με τη δήλωση update προσπαθήσουμε … Αν με τη δήλωση delete προσπαθήσουμε … |
| Ο περιορισμός Unique σε στήλη ή στήλες του πίνακα |  |
| Ο περιορισμός Primary Key |  |
| Ο περιορισμός Foreign Key |  |
| Ο περιορισμός DEFAULT |  |
| Ο περιορισμός AUTO\_INCREMENT |  |
| Ο περιορισμός Check |  |

# Ανασκόπηση των όψεων (VIEW)

Πότε έχουμε Updatable και Insertable Views

Αν μπορούμε να χρησιμοποιήσουμε UPDATE, DELETE, or INSERT για να ενημερώσουμε τον πίνακα που «υποκρύπτεται».

Πότε μια view δεν είναι ενημερώσιμη (is not updatable):

Όταν περιλαμβάνει

1. Aggregate functions (SUM(), MIN(), MAX(), COUNT(), …)
2. DISTINCT
3. GROUP BY
4. HAVING
5. UNION
6. Subquery (πράξεις κ.λπ.) in the select list
7. Certain joins
8. Nonupdatable view in the FROM clause
9. A subquery in the WHERE clause that refers to a table in the FROM clause

Η υποπρόταση WITH CHECK OPTION (clause) μας προστατεύει στην περίπτωση updatable view από inserts, updates που παραβιάζουν την υποπρόταση WHERE (clause) στην εντολή select που χρησιμοποιήσαμε για να ορίσουμε τη view.

|  |
| --- |
| **Ανοικτά Ακαδημαϊκά Μαθήματα****Τεχνολογικό Εκπαιδευτικό Ίδρυμα Αθήνας** |
| **Τέλος Ενότητας** |
| **Χρηματοδότηση*** Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
* Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Αθήνας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
* Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

 |

**Σημειώματα**

**Σημείωμα Αναφοράς**

Copyright ΤΕΙ Αθήνας, Χ. Σκουρλάς, Α. Τσολακίδης, Δ. Βάσσης 2014. Χ. Σκουρλάς, Α. Τσολακίδης, Δ. Βάσσης. «Βάσεις Δεδομένων ΙΙ (Ε). Ενότητα 2: Ανασκόπηση χρήσης της Γλώσσας SQL – θέματα υλοποίησης Βάσεων Δεδομένων - Χρήση περιορισμών (constraints) – Χρήση Views». Έκδοση: 1.0. Αθήνα 2014. Διαθέσιμο από τη δικτυακή διεύθυνση: [ocp.teiath.gr](https://ocp.teiath.gr/).

**Σημείωμα Αδειοδότησης**

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό. Οι όροι χρήσης των έργων τρίτων επεξηγούνται στη διαφάνεια «Επεξήγηση όρων χρήσης έργων τρίτων».

Τα έργα για τα οποία έχει ζητηθεί άδεια αναφέρονται στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] http://creativecommons.org/licenses/by-nc-sa/4.0/

Ως **Μη Εμπορική** ορίζεται η χρήση:

* που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
* που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
* που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

**Επεξήγηση όρων χρήσης έργων τρίτων**

|  |  |
| --- | --- |
| © | Δεν επιτρέπεται η επαναχρησιμοποίηση του έργου, παρά μόνο εάν ζητηθεί εκ νέου άδεια από το δημιουργό. |
| διαθέσιμο με άδεια CC-BY | Επιτρέπεται η επαναχρησιμοποίηση του έργου και η δημιουργία παραγώγων αυτού με απλή αναφορά του δημιουργού. |
| διαθέσιμο με άδεια CC-BY-SA | Επιτρέπεται η επαναχρησιμοποίηση του έργου με αναφορά του δημιουργού, και διάθεση του έργου ή του παράγωγου αυτού με την ίδια άδεια. |
| διαθέσιμο με άδεια CC-BY-ND | Επιτρέπεται η επαναχρησιμοποίηση του έργου με αναφορά του δημιουργού. Δεν επιτρέπεται η δημιουργία παραγώγων του έργου. |
| διαθέσιμο με άδεια CC-BY-NC | Επιτρέπεται η επαναχρησιμοποίηση του έργου με αναφορά του δημιουργού. Δεν επιτρέπεται η εμπορική χρήση του έργου. |
| διαθέσιμο με άδεια CC-BY-NC-SA | Επιτρέπεται η επαναχρησιμοποίηση του έργου με αναφορά του δημιουργού και διάθεση του έργου ή του παράγωγου αυτού με την ίδια άδεια. Δεν επιτρέπεται η εμπορική χρήση του έργου. |
| διαθέσιμο με άδεια CC-BY-NC-ND | Επιτρέπεται η επαναχρησιμοποίηση του έργου με αναφορά του δημιουργού. Δεν επιτρέπεται η εμπορική χρήση του έργου και η δημιουργία παραγώγων του. |
| διαθέσιμο με άδεια CC0 Public Domain | Επιτρέπεται η επαναχρησιμοποίηση του έργου, η δημιουργία παραγώγων αυτού και η εμπορική του χρήση, χωρίς αναφορά του δημιουργού. |
| διαθέσιμο ως κοινό κτήμα | Επιτρέπεται η επαναχρησιμοποίηση του έργου, η δημιουργία παραγώγων αυτού και η εμπορική του χρήση, χωρίς αναφορά του δημιουργού. |
| χωρίς σήμανση | Συνήθως δεν επιτρέπεται η επαναχρησιμοποίηση του έργου. |

**Διατήρηση Σημειωμάτων**

* Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:
* το Σημείωμα Αναφοράς
* το Σημείωμα Αδειοδότησης
* τη δήλωση Διατήρησης Σημειωμάτων
* το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει) μαζί με τους συνοδευόμενους υπερσυνδέσμους.