

Εργαστήριο 2

Διαμόρφωση δομής πίνακα Περιορισμοί (constraints)

Ανάπτυξη και Επιμέλεια: Ζ. Γαροφαλάκη, Α. Τσολακίδης, Π. Ανδρίτσος

Στόχος

Αλλαγές στη δομή πινάκων μετά τη δημιουργία, εξοικείωση με τον ορισμό περιορισμών.

Εργασία

A. Διαμόρφωση δομής πίνακα – Εντολές ALTER

Υπάρχει η δυνατότητα της διαμόρφωσης της δομής ενός πίνακα, μετά την αρχική δημιουργία του και ενώ αυτός περιέχει και δεδομένα, χωρίς την ανάγκη διαγραφής και εκ νέου δημιουργίας του. Η δυνατότητα αυτή παρέχεται με τις ακόλουθες παραλλαγές της εντολής ALTER.

1. Η **προσθήκη στήλης** με όνομα NEASTILI και τύπο δεδομένων varchar(20) σε πίνακα με όνομα PINAKAS, γίνεται με την εντολή:

```
alter table PINAKAS add NEASTILI varchar(20);  
  
# Έλεγχος αποτελέσματος, εμφάνιση δομής πίνακα PINAKAS  
describe PINAKAS;
```

2. Η **αλλαγή τύπου δεδομένων στήλης** με όνομα NEASTILI από varchar(20) σε varchar(30), γίνεται με την εντολή:

```
alter table PINAKAS modify NEASTILI varchar(30);
```

3. Η **διαγραφή στήλης** με όνομα NEASTILI, γίνεται με την εντολή:

```
alter table PINAKAS drop NEASTILI;
```

4. Η **μετονομασία στήλης** με όνομα NEASTILI σε NEWCLM, γίνεται με την ακόλουθη εντολή. Σημειώνεται πως με αυτή την παραλλαγή της ALTER μπορούμε να μετονομάσουμε ή και να αλλάξουμε τύπο δεδομένων ταυτόχρονα σε μία στήλη.

```
alter table PINAKAS change NEASTILI NEWCLM varchar(30);  
  
# Μετονομασία στήλης και ταυτόχρονη αλλαγή τύπου δεδομένων της  
alter table PINAKAS change NEASTILI NEWCLM char(10);
```

5. Η **μετονομασία πίνακα** με όνομα PINAKAS σε PNKS, γίνεται με την ακόλουθη εντολή:

```
alter table PINAKAS rename to PNKS;  
  
# Έλεγχος αποτελέσματος, εμφάνιση όλων των πινάκων  
show tables;
```

B. Περιορισμοί

Οι περιορισμοί χρησιμοποιούνται για τον περιορισμό των δεδομένων που μπορούν να εισαχθούν σε έναν πίνακα, καθώς και τη διασφάλιση της ακρίβειας και αξιοπιστίας των δεδομένων στον πίνακα. Εάν υπάρχει παραβίαση του περιορισμού από κάποια ενέργεια σχετική με τα δεδομένα, η

ενέργεια απορρίπτεται. Οι περιορισμοί μπορεί να είναι σε επίπεδο στήλης ή πίνακα. Οι συνηθέστερα χρησιμοποιούμενοι περιορισμοί είναι:

- **NOT NULL**: μια στήλη δεν μπορεί να έχει τιμή NULL σε καμία εγγραφή
- **UNIQUE**: όλες οι τιμές σε μια στήλη είναι διαφορετικές
- **PRIMARY KEY**: (συνδυασμός **NOT NULL** και **UNIQUE**) προσδιορίζει μοναδικά κάθε εγγραφή
- **FOREIGN KEY**: αποτρέπει ενέργειες που θα καταστρέψουν συνδέσεις μεταξύ πινάκων
- **CHECK**: οι τιμές σε μια στήλη ικανοποιούν μια συγκεκριμένη συνθήκη
- **DEFAULT**: προεπιλεγμένη τιμή για μια στήλη (όταν δεν ορίζεται τιμή)
- **CREATE INDEX**: για τη γρήγορη δημιουργία/ανάκτηση δεδομένων από τη ΒΔ

1. Η προσθήκη **περιορισμού πρωτεύοντος κλειδιού** (PRIMARY KEY) σε πίνακα με όνομα PINAKAS και ορισμό ως PK τη στήλη PID του πίνακα, γίνεται με την ακόλουθη εντολή. Σημειώνεται θα οριστεί όνομα pk_PID για τον περιορισμό.

```
alter table PINAKAS add constraint pk_PID primary key (PID);
```

2. Η κατάργηση **περιορισμού πρωτεύοντος κλειδιού** (PRIMARY KEY) σε πίνακα με όνομα PINAKAS και ορισμό ως PK τη στήλη PID του πίνακα, γίνεται με την ακόλουθη εντολή:

```
alter table PINAKAS drop primary key;

# Έλεγχος αποτελέσματος, εμφάνιση δομής πίνακα PINAKAS
describe PINAKAS;

# Σύνταξη της create table εντολής για τη δημιουργία του πίνακα PINAKAS
show create table PINAKAS;
```

3. Η προσθήκη **περιορισμού ξένου κλειδιού** (FOREIGN KEY) σε πίνακα με όνομα PINAKAS και ορισμό ως FK τη στήλη P2ID του πίνακα (η στήλη PINAKAS.P2ID θα συνδέει τα στοιχεία του πίνακα με τα στοιχεία του PINAKAS2), γίνεται με την ακόλουθη εντολή. Σημειώνεται θα οριστεί όνομα fk_P2ID για τον περιορισμό.

```
alter table PINAKAS add constraint fk_P2ID foreign key(P2ID) references PINAKAS2(P2ID);
```

4. Η κατάργηση **περιορισμού ξένου κλειδιού** (FOREIGN KEY) σε πίνακα με όνομα PINAKAS και ορισμό ως FK με όνομα fk_P2ID τη στήλη P2ID του πίνακα, γίνεται με την ακόλουθη εντολή:

```
alter table PINAKAS drop foreign key fk_P2ID;
```

5. Η προσθήκη **περιορισμού προεπιλεγμένης τιμής** (DEFAULT) σε πίνακα με όνομα PINAKAS, θέτοντας προεπιλεγμένη τιμή 'UniWA' για τη στήλη UNI του πίνακα, γίνεται με την ακόλουθη εντολή:

```
alter table PINAKAS alter UNI set default 'UniWA';
```

6. Η κατάργηση **περιορισμού DEFAULT** σε πίνακα με όνομα PINAKAS για τη στήλη UNI του πίνακα, γίνεται με την ακόλουθη εντολή:

```
alter table PINAKAS alter UNI drop default;
```

7. Ο **περιορισμός AUTO_INCREMENT** εφαρμόζεται σε στήλη και διαχειρίζεται αυτόματα με αύξοντα αριθμό την τιμή της στήλης. Σημειώνεται πως οι δηλώσεις insert δεν μπορούν να αλλάξουν τιμή σε στήλη με αυτόν τον περιορισμό. Η προσθήκη **περιορισμού**

AUTO_INCREMENT στη στήλη PID (πρέπει να είναι ή να οριστεί ως PK η στήλη) πίνακα με όνομα PINAKAS, γίνεται με την ακόλουθη εντολή:

```
alter table PINAKAS change PID PID int not null auto_increment primary key;
```

```
# Εναλλακτικά, προσθήκη περιορισμού κατά τη δημιουργία πίνακα
```

```
create table PINAKAS  
(PID int(11) not null auto_increment,  
P2ID int(11),  
UNI varchar(10),  
primary key (PID))  
engine=InnoDB default charset=utf8mb4;
```

```
# Αλλαγή της αρχικής τιμής του AUTO_INCREMENT (η default είναι 1)
```

```
alter table PINAKAS auto_increment = 100;
```

8. Η κατάργηση του **περιορισμού AUTO_INCREMENT** μπορεί να προκαλέσει μηνύματα λάθους, καθώς ο περιορισμός εφαρμόζεται σε στήλες που φέρουν και άλλους περιορισμούς (όπως του πρωτεύοντος κλειδιού). Για το λόγο αυτό απαιτείται η εκτέλεση της δήλωσης κατάργησης, ως ακολούθως:

```
# Απενεργοποίηση ελέγχων ξένων κλειδιών
```

```
set foreign_key_checks=0;
```

```
alter table PINAKAS modify PID int not null;
```

```
# Ενεργοποίηση ελέγχων ξένων κλειδιών
```

```
set foreign_key_checks=1;
```

9. Η λίστα των περιορισμών στη ΒΔ personnel εμφανίζεται με την ακόλουθη εντολή:

```
select * from INFORMATION_SCHEMA.TABLE_CONSTRAINTS where CONSTRAINT_SCHEMA='personnel';
```

Δραστηριότητες

Υλοποιήστε τις ακόλουθες δραστηριότητες. Η εντολή ή οι εντολές που απαιτούνται για την υλοποίηση του κάθε βήματος, καθώς και το αποτέλεσμα της εκτέλεσής της/τους θα πρέπει να ενταχθεί/-ούν σε ένα παραδοτέο αρχείο με τη μορφή κειμένου ή με τη μορφή στιγμιοτύπου (screenshot). Το αρχείο ή τα αρχεία με τις απαντήσεις σας, θα πρέπει να συμπιεστούν σε ένα **xx_ZZZZZ_ΕΡΩΝΥΜΟ.zip**, όπου: (α) xx ο αριθμός του τμήματος στο οποίο ανήκετε (π.χ. για την ομάδα [02] ΔΕΥΤΕΡΑ 12:00-13:00, **xx = 02**) και (β) ZZZZZ ο Αριθμός Μητρώου σας. Το τελικό αυτό αρχείο θα υποβάλλεται στο χώρο του e-class -> ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ II -> Εργασίες.

1. **Συνδεθείτε** στην MySQL του συστήματός σας με όποιον από τους προαναφερόμενους τρόπους επιθυμείτε.
2. Ελέγξτε αν υπάρχει ΒΔ με την ονομασία **personnel**. Αν υπάρχει, διαγράψτε την και δημιουργήστε την εκ νέου.

```
# Εμφάνιση ΒΔ
show databases;

# Διαγραφή ΒΔ personnel
drop database if exists personnel;

# Δημιουργία ΒΔ personnel
create database personnel;
```

3. Επιλέξτε την ΒΔ personnel για **χρήση**.

```
# Επιλογή personnel για χρήση
use personnel;
```

4. Βεβαιωθείτε πως η personnel **δεν έχει** περιεχόμενους πίνακες. Αν έχει, διαγράψτε τους.

```
# Εμφάνιση πινάκων
show tables;

# Διαγραφή πίνακα με όνομα onoma_pinaka
drop table onoma_pinaka;
```

5. Δημιουργήστε τους πίνακες **DEPT**, **JOB** και **EMP** χωρίς την προσθήκη κλειδιών. Να ληφθούν υπόψιν οι τύποι δεδομένων που περιγράφονται στον [Πίνακα 1](#).

```
# Δημιουργία πίνακα DEPT
create table DEPT(DEPTNO numeric(2), DNAME varchar(24), LOC char(23));

# Έλεγχος αποτελέσματος, εμφάνιση δομής πίνακα DEPT
describe DEPT;
```

6. Εισάγεται **δεδομένα** στους πίνακες, ώστε τα περιεχόμενά τους να είναι αυτά που εμφανίζονται στην [Εικόνα 16](#).

```
# Εισαγωγή μιας εγγραφής στον πίνακα EMP
insert into EMP(EMPNO,NAME,JOBNO,DEPTNO,COMM) values (10, 'Codd', 100, 50, NULL);

# Έλεγχος αποτελέσματος, εμφάνιση περιεχομένων πίνακα EMP
select * from DEPT;
```

7. Προσθέστε στον πίνακα EMP τη **στήλη HIREDATE** με τύπο δεδομένων DATE ('YYYY-MM-DD'). Εμφανίστε εκ νέου τη δομή του πίνακα EMP.

```
# Εισαγωγή στήλης HIREDATE στον πίνακα EMP
alter table EMP add HIREDATE date;

# Έλεγχος αποτελέσματος, εμφάνιση δομής πίνακα EMP
describe EMP;
```

8. Ενημερώστε με **τιμές** τη στήλη HIREDATE του πίνακα EMP λαμβάνοντας υπόψιν πως οι ημερομηνίες πρόσληψης των υπαλλήλων είναι: CODD 10-Ιαν-2001, NAVATHE 25-Φεβ-1999, ELMASRI 17-Μαρ-2000 και DATE 7-Ιουν-1989.

```
# Ενημέρωση στήλης HIREDATE με την ημ/νια πρόσληψης του CODD
update EMP set HIREDATE='2001-01-10' where empno=10;

# Έλεγχος αποτελέσματος, εμφάνιση περιεχομένων πίνακα EMP
select * from DEPT;
```

9. Ορίστε **τύπο δεδομένων** varchar(30) στη στήλη JOB_DESCR του πίνακα JOB.

10. Καταργήστε τη **στήλη LOC** του πίνακα DEPT.

11. Μετονομάστε τη **στήλη DNAME** του πίνακα DEPT σε DEPT_NAME. Ορίστε και **τύπο δεδομένων** varchar(25) στη στήλη DEPT_NAME. Σημειώστε πως οι αλλαγές αυτές θα πρέπει να προκύψουν με την εκτέλεση μιας μόνο δήλωσης.

12. Προσθέστε **περιορισμό πρωτεύοντος κλειδιού** (PRIMARY KEY) με το όνομα pk_DEPTNO στον πίνακα DEPT.

```
alter table DEPT add constraint pk_DEPTNO primary key (DEPTNO);
```

13. Καταργήστε τον **περιορισμό πρωτεύοντος κλειδιού** (PRIMARY KEY) στον πίνακα DEPT. Δείξτε το αποτέλεσμα της κατάργησης (α) εμφανίζοντας τη δομή του πίνακα DEPT και (β) εμφανίζοντας την εντολή create table για τον πίνακα DEPT ([βήμα B.2](#)).

14. Προσθέστε **περιορισμό πρωτεύοντος κλειδιού** (PRIMARY KEY) και **περιορισμούς ξένου κλειδιού** (FOREIGN KEY) σε όλους τους πίνακες.

15. Καταργήστε τους **περιορισμούς ξένου κλειδιού** (FOREIGN KEY) στον πίνακα EMP. Δείξτε το αποτέλεσμα της κάθε κατάργησης (α) εμφανίζοντας τη δομή του πίνακα EMP και (β) εμφανίζοντας την εντολή create table για τον πίνακα EMP ([βήμα B.2](#)).

16. Προσθέστε **περιορισμό προεπιλεγμένης τιμής** (DEFAULT) στον πίνακα DEPT, θέτοντας προεπιλεγμένη τιμή 'Development' για τη στήλη DNAME του πίνακα ([βήμα B.5](#)). Δείξτε την επίδραση του περιορισμού κάνοντας χρήση των ακόλουθων εντολών και δείχνοντας το αποτέλεσμα της εκτέλεσης κάθε εντολής.

```
insert into EMP(EMPNO,NAME,JOBNO,DEPTNO,COMM) values (10, 'CODD', 100, 50, NULL);

insert into DEPT(DEPTNO) values (10), (20);

insert into DEPT values (30, 'Sales');
```

17. Εμφανίστε τη λίστα των περιορισμών της ΒΔ personnel.

18. Δημιουργήστε πίνακα PROJECT με στήλες P_ID και P_NAME. Να ληφθούν υπόψιν οι τύποι δεδομένων που περιγράφονται στον [Πίνακα 1](#). Ορίστε **πρωτεύον κλειδί** και θέστε **περιορισμό AUTO_INCREMENT** για την στήλη P_ID του πίνακα.
19. Εκτελέστε τουλάχιστον μια **εντολή καταχώρησης εγγραφής**, ώστε να δείξετε τη λειτουργία του περιορισμού. Εμφανίστε και το αποτέλεσμα της εκτέλεσης της εντολής.
20. Ορίστε **αρχική τιμή 200** για τον περιορισμό AUTO_INCREMENT και επαναλάβετε το προηγούμενο [βήμα 19](#).

ΒΔ personnel

Οι περιεχόμενοι πίνακες της ΒΔ personnel θα πρέπει να έχουν την ακόλουθη δομή και περιεχόμενα:

Στήλες	Τύπος δεδομένων
DEPT.DEPTNO, EMP.DEPTNO	numeric(2)
DNAME, JOB_DESCR	varchar(24)
LOC	char(23)
JOBCODE, JOBNO	numeric(3)
SAL, COMM	numeric(10,2)
EMPNO	numeric(4)
PROJECT.P_ID	int
PROJECT.P_NAME	varchar(255)

Πίνακας 1. Τύποι δεδομένων πινάκων EMP, JOB, DEPT

EMP

EMPNO	NAME	JOBNO	DEPTNO	COMM
10	CODD	100	50	
20	NAVATHE	200	50	450
30	ELMASRI	300	60	
40	DATE	100	50	

JOB

JOBCODE	JOB_DESCR	SAL
100	SALESMAN	2000
200	ANALYST	2000
300	DBA	3000

DEPT

DEPTNO	DNAME	LOC
50	SALES	ATHENS
60	ACCOUNTING	ATHENS
70	PAYROL	VOLOS

Εικόνα 1. Δεδομένα πινάκων EMP, JOB, DEPT