

# Η γλώσσα ορισμού δεδομένων της SQL

Οι εντολές CREATE TABLE, ALTER TABLE, CREATE KEY, ALTER KEY

Αθανάσιος Σταυρακούδης

<http://stavrakoudis.econ.uoi.gr>

astavrak@uoi.gr

@ASTavrakoudis

Άνοιξη 2016



- 1 Δημιουργία πινάκων με την εντολή CREATE TABLE
- 2 Περιορισμοί ξένου κλειδιού





# Σκοπός της διάλεξης

- 1 Να δημιουργείτε πίνακες και να ορίζετε τα πεδία που περιέχουν (**CREATE TABLE**).



# Σκοπός της διάλεξης

- 1 Να δημιουργείτε πίνακες και να ορίζετε τα πεδία που περιέχουν (**CREATE TABLE**).
- 2 Να εφαρμόζετε τους περιορισμούς πρωτεύοντος (**PRIMARY KEY**) και ξένου κλειδιού (**FOREIGN KEY**).



# Σκοπός της διάλεξης

- 1 Να δημιουργείτε πίνακες και να ορίζετε τα πεδία που περιέχουν (**CREATE TABLE**).
- 2 Να εφαρμόζετε τους περιορισμούς πρωτεύοντος (PRIMARY KEY) και ξένου κλειδιού (FOREIGN KEY).
- 3 Να εφαρμόζετε τα εναύσματα (πυροδοτήσεις) διαδοχικής ενημέρωσης και διαγραφής (ON UPDATE CASCADE, ON DELETE CASCADE).



# Σκοπός της διάλεξης

- 1 Να δημιουργείτε πίνακες και να ορίζετε τα πεδία που περιέχουν (**CREATE TABLE**).
- 2 Να εφαρμόζετε τους περιορισμούς πρωτεύοντος (PRIMARY KEY) και ξένου κλειδιού (FOREIGN KEY).
- 3 Να εφαρμόζετε τα εναύσματα (πυροδοτήσεις) διαδοχικής ενημέρωσης και διαγραφής (ON UPDATE CASCADE, ON DELETE CASCADE).
- 4 Να τροποποιείτε τη σχεδίαση των πινάκων με την εντολή **ALTER TABLE**.



# Σκοπός της διάλεξης

- 1 Να δημιουργείτε πίνακες και να ορίζετε τα πεδία που περιέχουν (**CREATE TABLE**).
- 2 Να εφαρμόζετε τους περιορισμούς πρωτεύοντος (PRIMARY KEY) και ξένου κλειδιού (FOREIGN KEY).
- 3 Να εφαρμόζετε τα εναύσματα (πυροδοτήσεις) διαδοχικής ενημέρωσης και διαγραφής (ON UPDATE CASCADE, ON DELETE CASCADE).
- 4 Να τροποποιείτε τη σχεδίαση των πινάκων με την εντολή ALTER TABLE.
- 5 Να δημιουργείτε δείκτες με την εντολή **CREATE INDEX**.





# Η εντολή CREATE TABLE

## Γενική μορφή

```
1 CREATE TABLE πίνακας  
2 (  
3     λίστα πεδίων,  
4     λίστα περιορισμών,  
5     άλλες επιλογές,  
6 );
```



# Η εντολή CREATE TABLE

## Γενική μορφή

```
1 CREATE TABLE πίνακας  
2 (  
3     λίστα πεδίων,  
4     λίστα περιορισμών,  
5     άλλες επιλογές,  
6 );
```

## Παράδειγμα (μόνο με λίστα πεδίων με ενσωματωμένους περιορισμούς)

```
1 CREATE TABLE friends  
2 (  
3     id    INTEGER    NOT NULL PRIMARY KEY,  
4     name  CHAR(20)   NOT NULL,  
5     age   INTEGER,  
6     phone CHAR(10)  
7 );
```



## Περιορισμοί

- 1 Περιορισμοί ως προς το πρωτεύον κλειδί (PRIMARY KEY)
- 2 Περιορισμοί ως προς τα ξένα κλειδιά (FOREIGN KEY)
- 3 Περιορισμοί για έλεγχο μοναδικότητας (UNIQUE)
- 4 Περιορισμοί για έλεγχο των εισαγόμενων τιμών (CHECK)



## Περιορισμοί

- 1 Περιορισμοί ως προς το πρωτεύον κλειδί (PRIMARY KEY)
- 2 Περιορισμοί ως προς τα ξένα κλειδιά (FOREIGN KEY)
- 3 Περιορισμοί για έλεγχο μοναδικότητας (UNIQUE)
- 4 Περιορισμοί για έλεγχο των εισαγόμενων τιμών (CHECK)

## Παράδειγμα PRIMARY KEY

```
1 CONSTRAINT pk PRIMARY KEY(id)
```



## Περιορισμοί

- 1 Περιορισμοί ως προς το πρωτεύον κλειδί (PRIMARY KEY)
- 2 Περιορισμοί ως προς τα ξένα κλειδιά (FOREIGN KEY)
- 3 Περιορισμοί για έλεγχο μοναδικότητας (UNIQUE)
- 4 Περιορισμοί για έλεγχο των εισαγόμενων τιμών (CHECK)

## Παράδειγμα PRIMARY KEY

```
1 CONSTRAINT pk PRIMARY KEY(id)
```

## Παράδειγμα FOREIGN KEY

```
1 CONSTRAINT fk_depid FOREIGN KEY(depid)  
2 REFERENCES departments (depid)
```



## Παράδειγμα PRIMARY KEY

```
1 CREATE TABLE friends
2 (
3     name CHAR(20) NOT NULL,
4     age  INTEGER,
5     phone CHAR(10),
6     CONSTRAINT pk PRIMARY KEY(name, phone)
7 );
```



## Παράδειγμα PRIMARY KEY

```
1 CREATE TABLE friends
2 (
3     name CHAR(20) NOT NULL,
4     age  INTEGER,
5     phone CHAR(10),
6     CONSTRAINT pk PRIMARY KEY(name, phone)
7 );
```

## Προσοχή

- Ο πίνακας έχει ένα πρωτεύον κλειδί.
- Το πρωτεύον κλειδί αποτελείται από δύο πεδία του πίνακα.
- Είναι λάθος να πούμε πως «δύο πεδία είναι πρωτεύοντα κλειδιά».



# Παράδειγμα ξένου κλειδί: depid





# Παράδειγμα ξένου κλειδί: depid

## *departments*

```
1 CREATE TABLE departments (  
2     depid      INTEGER NOT NULL,  
3     depname    CHAR(30) NOT NULL,  
4     manager    INTEGER  
5     CONSTRAINT pk PRIMARY KEY(depid)  
6 );
```



# Παράδειγμα ξένου κλειδί: depid

## departments

```
1 CREATE TABLE departments (  
2     depid      INTEGER NOT NULL,  
3     depname    CHAR(30) NOT NULL,  
4     manager    INTEGER  
5     CONSTRAINT pk PRIMARY KEY(depid)  
6 );
```

## employees

```
1 CREATE TABLE employees (  
2     empid      INTEGER NOT NULL,  
3     firstname  CHAR(30) NOT NULL,  
4     lastname   CHAR(30) NOT NULL,  
5     depid      INTEGER NOT NULL,  
6     salary     DECIMAL(6,2),  
7     hiredate   DATE,  
8     CONSTRAINT pk PRIMARY KEY(empid),
```



# Παράδειγμα ξένου κλειδί: depid

## *departments*

```
1 CREATE TABLE departments (  
2     depid      INTEGER NOT NULL,  
3     depname    CHAR(30) NOT NULL,  
4     manager    INTEGER  
5     CONSTRAINT pk PRIMARY KEY(depid)  
6 );
```

## *employees*

```
1 CREATE TABLE employees (  
2     empid      INTEGER NOT NULL,  
3     firstname  CHAR(30) NOT NULL,  
4     lastname   CHAR(30) NOT NULL,  
5     depid      INTEGER NOT NULL,  
6     salary     DECIMAL(6,2),  
7     hiredate   DATE,  
8     CONSTRAINT pk PRIMARY KEY(empid),  
9     CONSTRAINT fk_depdeid FOREIGN KEY(depid)  
10        REFERENCES departments(depid)  
11 );
```



# Παρατηρήσεις για τα ξένα κλειδιά

- 1 Γενικά τα ξένα κλειδιά θα πρέπει να είναι ορισμένα με την επιλογή **NOT NULL** προς αποφυγή διαφόρων **ανωμαλιών στην ενημέρωση της βάσης**.



# Παρατηρήσεις για τα ξένα κλειδιά

- 1 Γενικά τα ξένα κλειδιά θα πρέπει να είναι ορισμένα με την επιλογή **NOT NULL** προς αποφυγή διαφόρων **ανωμαλιών στην ενημέρωση της βάσης**.
- 2 Τα ξένα κλειδιά δεν έχουν καμία επίδραση στις εντολές **SELECT**.



# Παρατηρήσεις για τα ξένα κλειδιά

- 1 Γενικά τα ξένα κλειδιά θα πρέπει να είναι ορισμένα με την επιλογή **NOT NULL** προς αποφυγή διαφόρων **ανωμαλιών στην ενημέρωση της βάσης**.
- 2 Τα ξένα κλειδιά δεν έχουν καμία επίδραση στις εντολές **SELECT**.
- 3 Η επίδραση που έχουν οι περιορισμοί ξένου κλειδιού αφορά τις πράξεις ενημέρωσης της βάσης (**INSERT, DELETE και UPDATE**).



# Παρατηρήσεις για τα ξένα κλειδιά

- 1 Γενικά τα ξένα κλειδιά θα πρέπει να είναι ορισμένα με την επιλογή **NOT NULL** προς αποφυγή διαφόρων **ανωμαλιών στην ενημέρωση της βάσης**.
- 2 Τα ξένα κλειδιά δεν έχουν καμία επίδραση στις εντολές **SELECT**.
- 3 Η επίδραση που έχουν οι περιορισμοί ξένου κλειδιού αφορά τις πράξεις ενημέρωσης της βάσης (**INSERT, DELETE και UPDATE**).
- 4 Αν το ξένο κλειδί παίρνει **πολλαπλές τιμές**, π.χ. εμφανίζεται πολλές φορές ο ίδιος κωδικός τμήματος για διαφορετικούς υπαλλήλους, τότε ο περιορισμός του ξένου κλειδιού ισοδυναμεί με τη συσχέτιση **ένα προς πολλά**.



# Παρατηρήσεις για τα ξένα κλειδιά

- 1 Γενικά τα ξένα κλειδιά θα πρέπει να είναι ορισμένα με την επιλογή **NOT NULL** προς αποφυγή διαφόρων **ανωμαλιών στην ενημέρωση της βάσης**.
- 2 Τα ξένα κλειδιά δεν έχουν καμία επίδραση στις εντολές **SELECT**.
- 3 Η επίδραση που έχουν οι περιορισμοί ξένου κλειδιού αφορά τις πράξεις ενημέρωσης της βάσης (**INSERT, DELETE και UPDATE**).
- 4 Αν το ξένο κλειδί παίρνει **πολλαπλές τιμές**, π.χ. εμφανίζεται πολλές φορές ο ίδιος κωδικός τμήματος για διαφορετικούς υπαλλήλους, τότε ο περιορισμός του ξένου κλειδιού ισοδυναμεί με τη συσχέτιση **ένα προς πολλά**.
- 5 Αν το ξένο κλειδί έχει **περιορισμό μοναδικότητας και αναφέρεται σε πεδίο άλλου πίνακα με περιορισμό μοναδικότητας** (πχ πρωτεύον κλειδί) τότε αυτό ισοδυναμεί με τη συσχέτιση **ένα προς ένα**.





# Επίδραση ξένου κλειδιού



## Εισαγωγή τμήματος

```
1 INSERT INTO departments (depid, depname)
2 VALUES (1, 'Διοίκησης');
```



## Εισαγωγή τμήματος

```
1 INSERT INTO departments (depid, depname)
2 VALUES (1, 'Διοίκησης');
```

## Εισαγωγή υπαλλήλου

```
1 INSERT INTO employees (empid, firstname, lastname, depid)
2 VALUES (101, 'Μαρία', 'Αθανασίου', 1);
```



## Εισαγωγή τμήματος

```
1 INSERT INTO departments (depid, depname)
2 VALUES (1, 'Διοίκησης');
```

## Εισαγωγή υπαλλήλου

```
1 INSERT INTO employees (empid, firstname, lastname, depid)
2 VALUES (101, 'Μαρία', 'Αθανασίου', 1);
```

## Αποτυχία

```
1 INSERT INTO employees (empid, firstname, lastname, depid)
2 VALUES (201, 'Κρινιώ', 'Παπαδοπούλου', 2);
```



# Συσχέτιση πολλά προς πολλά



# Συσχέτιση πολλά προς πολλά

## projects

```
1 CREATE TABLE projects (  
2     proid      INTEGER      NOT NULL,  
3     title      VARCHAR(120) NOT NULL,  
4     budget     DECIMAL(9,2) NOT NULL DEFAULT 0,  
5     startdate  DATE,  
6     enddate    DATE,  
7     progress   DECIMAL(3,1) NOT NULL DEFAULT 0,  
8     CONSTRAINT pk_projects PRIMARY KEY (proid) );
```



# Συσχέτιση πολλά προς πολλά

## *projects*

```
1 CREATE TABLE projects (  
2     proid      INTEGER      NOT NULL,  
3     title     VARCHAR(120) NOT NULL,  
4     budget    DECIMAL(9,2) NOT NULL DEFAULT 0,  
5     startdate DATE,  
6     enddate   DATE,  
7     progress  DECIMAL(3,1) NOT NULL DEFAULT 0,  
8     CONSTRAINT pk_projects PRIMARY KEY (proid) );
```

## *workson*

```
1 CREATE TABLE workson (  
2     empid      INTEGER      NOT NULL,  
3     proid      INTEGER      NOT NULL,
```



# Συσχέτιση πολλά προς πολλά

## projects

```
1 CREATE TABLE projects (  
2     proid      INTEGER      NOT NULL,  
3     title      VARCHAR(120) NOT NULL,  
4     budget     DECIMAL(9,2) NOT NULL DEFAULT 0,  
5     startdate  DATE,  
6     enddate    DATE,  
7     progress   DECIMAL(3,1) NOT NULL DEFAULT 0,  
8     CONSTRAINT pk_projects PRIMARY KEY (proid) );
```

## workson

```
1 CREATE TABLE workson (  
2     empid      INTEGER      NOT NULL,  
3     proid      INTEGER      NOT NULL,  
4     CONSTRAINT pk_workson PRIMARY KEY (empid, proid),
```





# Συσχέτιση πολλά προς πολλά

## projects

```
1 CREATE TABLE projects (  
2     proid      INTEGER      NOT NULL,  
3     title      VARCHAR(120) NOT NULL,  
4     budget     DECIMAL(9,2) NOT NULL DEFAULT 0,  
5     startdate  DATE,  
6     enddate    DATE,  
7     progress   DECIMAL(3,1) NOT NULL DEFAULT 0,  
8     CONSTRAINT pk_projects PRIMARY KEY (proid) );
```

## workson

```
1 CREATE TABLE workson (  
2     empid      INTEGER      NOT NULL,  
3     proid      INTEGER      NOT NULL,  
4     CONSTRAINT pk_workson PRIMARY KEY (empid, proid),  
5     CONSTRAINT fk_emp FOREIGN KEY (empid)  
6         REFERENCES employees(empid),
```



# Συσχέτιση πολλά προς πολλά

## projects

```
1 CREATE TABLE projects (  
2     proid      INTEGER      NOT NULL,  
3     title     VARCHAR(120) NOT NULL,  
4     budget    DECIMAL(9,2) NOT NULL DEFAULT 0,  
5     startdate DATE,  
6     enddate   DATE,  
7     progress  DECIMAL(3,1) NOT NULL DEFAULT 0,  
8     CONSTRAINT pk_projects PRIMARY KEY (proid) );
```

## workson

```
1 CREATE TABLE workson (  
2     empid      INTEGER      NOT NULL,  
3     proid      INTEGER      NOT NULL,  
4     CONSTRAINT pk_workson PRIMARY KEY (empid, proid),  
5     CONSTRAINT fk_emp FOREIGN KEY (empid)  
6         REFERENCES employees(empid),  
7     CONSTRAINT fk_pro FOREIGN KEY (proid)  
8         REFERENCES projects(proid) );
```



## UNIQUE index, δείκτης μοναδικότητας

```
1 CREATE TABLE employees
2 (
3     empid      INTEGER      NOT NULL,
4     firstname CHAR(50)     NOT NULL,
5     lastname  CHAR(50)     NOT NULL,
6     amka      CHAR(12)     NOT NULL,
7     depid     INTEGER      NOT NULL,
8     salary    NUMERIC(6,2),
9     hiredate  DATE,
10    CONSTRAINT pk_employees PRIMARY KEY (empid),
```



## UNIQUE index, δείκτης μοναδικότητας

```
1 CREATE TABLE employees
2 (
3     empid      INTEGER      NOT NULL,
4     firstname  CHAR(50)     NOT NULL,
5     lastname   CHAR(50)     NOT NULL,
6     amka       CHAR(12)     NOT NULL,
7     depid      INTEGER      NOT NULL,
8     salary     NUMERIC(6,2),
9     hiredate   DATE,
10    CONSTRAINT pk_employees PRIMARY KEY (empid),
11    INDEX i_depid (depid),
```



## UNIQUE index, δείκτης μοναδικότητας

```
1 CREATE TABLE employees
2 (
3     empid      INTEGER      NOT NULL,
4     firstname  CHAR(50)     NOT NULL,
5     lastname   CHAR(50)     NOT NULL,
6     amka       CHAR(12)     NOT NULL,
7     depid      INTEGER      NOT NULL,
8     salary     NUMERIC(6,2),
9     hiredate   DATE,
10    CONSTRAINT pk_employees PRIMARY KEY (empid),
11    INDEX i_depid (depid),
12    UNIQUE u_amka (amka),
```



## UNIQUE index, δείκτης μοναδικότητας

```
1 CREATE TABLE employees
2 (
3     empid      INTEGER      NOT NULL,
4     firstname  CHAR(50)     NOT NULL,
5     lastname   CHAR(50)     NOT NULL,
6     amka       CHAR(12)     NOT NULL,
7     depid      INTEGER      NOT NULL,
8     salary     NUMERIC(6,2),
9     hiredate   DATE,
10    CONSTRAINT pk_employees PRIMARY KEY (empid),
11    INDEX i_depid (depid),
12    UNIQUE u_amka (amka),
13    CONSTRAINT fk_depid FOREIGN KEY (depid)
14        REFERENCES departments(depid)
15 );
```



- 1 Δημιουργία πινάκων με την εντολή CREATE TABLE
- 2 Περιορισμοί ξένου κλειδιού







## Διαγραφή

```
1 ON DELETE [CASCADE | RESTRICT | SET NULL | SET DEFAULT]
```



# Πυροδοτήσεις ενημέρωσης

## Διαγραφή

```
1 ON DELETE [CASCADE | RESTRICT | SET NULL | SET DEFAULT]
```

## Τροποποίηση

```
1 ON UPDATE [CASCADE | RESTRICT | SET NULL | SET DEFAULT]
```



# Πυροδοτήσεις ενημέρωσης

## Διαγραφή

1 | ON DELETE [CASCADE | RESTRICT | SET NULL | SET DEFAULT]

## Τροποποίηση

1 | ON UPDATE [CASCADE | RESTRICT | SET NULL | SET DEFAULT]

## Επεξηγήσεις



# Πυροδοτήσεις ενημέρωσης

## Διαγραφή

1 | ON DELETE [CASCADE | RESTRICT | SET NULL | SET DEFAULT]

## Τροποποίηση

1 | ON UPDATE [CASCADE | RESTRICT | SET NULL | SET DEFAULT]

## Επεξηγήσεις

- 1 **CASCADE** Ενεργοποιεί τη διαδοχική πράξη ενημέρωσης ή διαγραφής.



# Πυροδοτήσεις ενημέρωσης

## Διαγραφή

1 | ON DELETE [CASCADE | RESTRICT | SET NULL | SET DEFAULT]

## Τροποποίηση

1 | ON UPDATE [CASCADE | RESTRICT | SET NULL | SET DEFAULT]

## Επεξηγήσεις

- 1 **CASCADE** Ενεργοποιεί τη διαδοχική πράξη ενημέρωσης ή διαγραφής.
- 2 **RESTRICT** Αποτρέπει την ενημέρωση ή διαγραφή.



# Πυροδοτήσεις ενημέρωσης

## Διαγραφή

```
1 | ON DELETE [CASCADE | RESTRICT | SET NULL | SET DEFAULT]
```

## Τροποποίηση

```
1 | ON UPDATE [CASCADE | RESTRICT | SET NULL | SET DEFAULT]
```

## Επεξηγήσεις

- 1 **CASCADE** Ενεργοποιεί τη διαδοχική πράξη ενημέρωσης ή διαγραφής.
- 2 **RESTRICT** Αποτρέπει την ενημέρωση ή διαγραφή.
- 3 **SET NULL** Ενημερώνει το ξένο κλειδί σε τιμή **NULL**.



# Πυροδοτήσεις ενημέρωσης

## Διαγραφή

```
1 | ON DELETE [CASCADE | RESTRICT | SET NULL | SET DEFAULT]
```

## Τροποποίηση

```
1 | ON UPDATE [CASCADE | RESTRICT | SET NULL | SET DEFAULT]
```

## Επεξηγήσεις

- 1 **CASCADE** Ενεργοποιεί τη διαδοχική πράξη ενημέρωσης ή διαγραφής.
- 2 **RESTRICT** Αποτρέπει την ενημέρωση ή διαγραφή.
- 3 **SET NULL** Ενημερώνει το ξένο κλειδί σε τιμή **NULL**.
- 4 **SET DEFAULT** Ενημερώνει το ξένο κλειδί στην προεπιλεγμένη τιμή (**DEFAULT**).



## Παράδειγμα διαδοχικής διαγραφής στον πίνακα *workson*

```
1 CREATE TABLE workson
2 (
3     empid      INTEGER      NOT NULL,
4     proid      INTEGER      NOT NULL,
5     CONSTRAINT pk_workson PRIMARY KEY (empid, proid),
6     INDEX i_empid (empid),
7     INDEX i_proid (proid),
8     CONSTRAINT fk_empid FOREIGN KEY (empid)
9         REFERENCES employees(empid)
10        ON DELETE CASCADE,
11     CONSTRAINT fk_proid FOREIGN KEY (proid)
12        REFERENCES projects(proid)
13 );
```





# Πυροδότηση διαδοχικής διαγραφής

```
1 SELECT *  
2   FROM workson  
3   WHERE empid = 419;
```

```
4  
5 empid  proid  
6 -----
```

```
7 419    12
```

```
8 419    21
```

```
9 419    43
```



# Πυροδότηση διαδοχικής διαγραφής

```
1 SELECT *  
2   FROM workson  
3   WHERE empid = 419;
```

```
1 DELETE FROM employees  
2   WHERE empid = 419;
```

empid	proid
419	12
419	21
419	43



# Πυροδότηση διαδοχικής διαγραφής

```
1 SELECT *  
2   FROM workson  
3   WHERE empid = 419;
```

empid	proid
419	12
419	21
419	43

```
1 SELECT *  
2   FROM employees  
3   WHERE empid = 419;
```

Empty set (0.00 sec)

```
1 DELETE FROM employees  
2   WHERE empid = 419;
```



# Πυροδότηση διαδοχικής διαγραφής

```
1 SELECT *  
2   FROM workson  
3   WHERE empid = 419;
```

empid	proid
419	12
419	21
419	43

```
1 SELECT *  
2   FROM employees  
3   WHERE empid = 419;
```

Empty set (0.00 sec)

```
1 DELETE FROM employees  
2   WHERE empid = 419;
```

```
1 SELECT *  
2   FROM workson  
3   WHERE empid = 419;
```

Empty set (0.00 sec)



# ON UPDATE CASCADE

## Παράδειγμα διαδοχικής ενημέρωσης στον πίνακα *workson*

```
1 CREATE TABLE workson
2 (
3     empid      INTEGER      NOT NULL,
4     proid      INTEGER      NOT NULL,
5     CONSTRAINT pk_workson PRIMARY KEY (empid, proid),
6     INDEX i_empid (empid),
7     INDEX i_proid (proid),
8     CONSTRAINT fk_empid FOREIGN KEY (empid)
9         REFERENCES employees(empid)
10            ON DELETE CASCADE
11            ON UPDATE CASCADE,
12     CONSTRAINT fk_proid FOREIGN KEY (proid)
13         REFERENCES projects(proid)
14 );
```



# ON UPDATE CASCADE



# ON UPDATE CASCADE

Υπάλληλος με κωδικό 419

```
1 SELECT *  
2   FROM employees  
3   WHERE empid = 419;
```

```
4  
5 empid  firstname  lastname  depid  salary  hiredate  
6 -----  
7 419    Πέτρος      Αρβανιτάκης  2     1323.80  2000-07-17
```



# ON UPDATE CASCADE

Υπάλληλος με κωδικό 419

```
1 SELECT *  
2   FROM employees  
3   WHERE empid = 419;
```

empid	firstname	lastname	depid	salary	hiredate
419	Πέτρος	Αρβανιτάκης	2	1323.80	2000-07-17

Έργα που συμμετέχει

```
1 SELECT *  
2   FROM workson  
3   WHERE empid = 419;
```

empid	proid
419	12
419	21
419	43





# Τροποποίηση κωδικού (πρωτεύον κλειδί)



# Τροποποίηση κωδικού (πρωτεύον κλειδί)

Αλλαγή του από κωδικού 419 σε 406

```
1 UPDATE employees
2   SET empid = 406
3   WHERE empid = 419;
```



# Τροποποίηση κωδικού (πρωτεύον κλειδί)

Αλλαγή του από κωδικού 419 σε 406

```
1 UPDATE employees
2   SET empid = 406
3   WHERE empid = 419;
```

Νέος κωδικός

```
1 SELECT *
2   FROM employees
3   WHERE empid = 406;
```

empid	firstname	lastname	depid	salary	hiredate
419	Πέτρος	Αρβανιτάκης	2	1323.80	2000-07-17



# Τροποποίηση κωδικού (ξένο κλειδί)



# Τροποποίηση κωδικού (ξένο κλειδί)

419

```
1 SELECT *  
2   FROM workson  
3   WHERE empid = 419;
```

```
4  
5 Empty set (0.00 sec)
```



# Τροποποίηση κωδικού (ξένο κλειδί)

419

```
1 SELECT *  
2 FROM workson  
3 WHERE empid = 419;
```

4  
5 Empty set (0.00 sec)

406

```
1 SELECT *  
2 FROM workson  
3 WHERE empid = 406;
```

```
4  
5 empid  proid  
6 -----  
7 406    12  
8 406    21  
9 406    43
```



## UPDATE workson

```
1 UPDATE workson
2   SET empid = 406
3   WHERE empid = 419;
```



## Το τμήμα 2

```
1 SELECT *  
2   FROM departments  
3   WHERE depid = 2;
```

```
4  
5 depid  depname                manager  
6 -----
```

```
7      2  Οικονομολόγων/Λογιστών      153
```

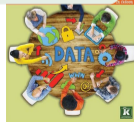




## Υπάλληλοι του τμήματος 2

```
1 SELECT *  
2   FROM departments  
3   WHERE depid = 2;
```

empid	firstname	lastname	depid	salary	hiredate
153	Μαρία	Αλεβιζάτου	2	1321.92	2001-05-15
243	Δέσποινα	Παπαδοπούλου	2	1609.52	1999-03-05
419	Πέτρος	Αρβανιτάκης	2	1323.80	2000-07-17
503	Μαριλένα	Κρέσπα	2	1105.04	2001-03-07



## Απασχόληση των υπαλλήλων του τμήματος 2

```
1 SELECT w.*  
2 FROM employees e INNER JOIN workson w ON e.empid=w.empid  
3 WHERE e.depid = 2;
```

empid	proid
-------	-------

-----

153	14
-----	----

153	38
-----	----

243	21
-----	----

243	38
-----	----

419	12
-----	----

419	21
-----	----

419	43
-----	----

503	21
-----	----

503	38
-----	----



## Διαγραφή του τμήματος 2

```
1 DELETE FROM departments
2 WHERE depid = 2;
```

## Επιβεβαίωση

```
1 SELECT *
2 FROM departments
3 WHERE depid = 2;
4
5 Empty set (0.00 sec)
```



# Καταστροφική αλυσίδα διαγραφής



## Υπάλληλοι

```
1 SELECT *  
2   FROM employees  
3   WHERE depid = 2;  
4  
5 Empty set (0.00 sec)
```



# Καταστροφική αλυσίδα διαγραφής

## Υπάλληλοι

```
1 SELECT *  
2   FROM employees  
3   WHERE depid = 2;  
4  
5 Empty set (0.00 sec)
```

## Απασχόληση σε έργα

```
1 SELECT w.*  
2   FROM employees e INNER JOIN workson w  
3   ON e.empid=w.empid  
4   WHERE e.depid = 2;  
5  
6 Empty set (0.00 sec)
```



# ON DELETE RESTRICT

Παράδειγμα περιορισμού διαδοχικής διαγραφής στον πίνακα *employees*

```
1 CREATE TABLE employees
2 (
3     empid      INTEGER      NOT NULL,
4     firstname  CHAR(30)    NOT NULL,
5     lastname   CHAR(30)    NOT NULL,
6     depid      INTEGER      NOT NULL,
7     salary     NUMERIC(6,2),
8     hiredate   DATE,
9     CONSTRAINT pk_employees PRIMARY KEY (empid),
```



# ON DELETE RESTRICT

Παράδειγμα περιορισμού διαδοχικής διαγραφής στον πίνακα *employees*

```
1 CREATE TABLE employees
2 (
3     empid      INTEGER      NOT NULL,
4     firstname  CHAR(30)    NOT NULL,
5     lastname   CHAR(30)    NOT NULL,
6     depid      INTEGER      NOT NULL,
7     salary     NUMERIC(6,2),
8     hiredate   DATE,
9     CONSTRAINT pk_employees PRIMARY KEY (empid),
10    INDEX i_depid (depid),
11    CONSTRAINT fk_depid FOREIGN KEY (depid)
12        REFERENCES departments(depid)
13        ON DELETE RESTRICT
14        ON UPDATE CASCADE
15 );
```





# ON DELETE RESTRICT

Αποτυχία διαγραφής εξαιτίας παρουσίας συσχετιζόμενων εγγραφών

```
1 DELETE
2   FROM departments
3   WHERE depid = 2;
```

```
5 ERROR 1451 (23000): Cannot delete or update a parent row:
6 a foreign key constraint fails
7 (employees, CONSTRAINT fk_depid FOREIGN KEY (depid)
8 REFERENCES departments (depid) ON UPDATE CASCADE)
```



# ON DELETE RESTRICT

Αποτυχία διαγραφής εξαιτίας παρουσίας συσχετιζόμενων εγγραφών

```
1 DELETE
2   FROM departments
3   WHERE depid = 2;
```

```
5 ERROR 1451 (23000): Cannot delete or update a parent row:
6 a foreign key constraint fails
7 (employees, CONSTRAINT fk_depid FOREIGN KEY (depid)
8 REFERENCES departments (depid) ON UPDATE CASCADE)
```

**Προσοχή!** Δεν πρόκειται για λάθος στην εντολή **SQL**.

Το **Σύστημα Διαχείρισης Βάσεων Δεδομένων** δεν εκτελεί τη διαγραφή λόγω ρυθμίσεων που απαγορεύουν τη **διαγραφή** στον **γονικό πίνακα** όταν υπάρχουν συσχετιζόμενες εγγραφές στον **θυγατρικό πίνακα**.



# Αλληλουχία διαγραφής με τη ρύθμιση ON DELETE RESTRICT



Μπορεί να γίνει έτσι

```
1 DELETE FROM workson ...  
2 DELETE FROM employees ...  
3 DELETE FROM departments ...
```



# Αλληλουχία διαγραφής με τη ρύθμιση ON DELETE RESTRICT

## Μπορεί να γίνει έτσι

```
1 DELETE FROM workson ...  
2 DELETE FROM employees ...  
3 DELETE FROM departments ...
```

## Αλλά όχι έτσι

```
1 DELETE FROM departments ...  
2 DELETE FROM employees ...  
3 DELETE FROM workson ...
```





## Παράδειγμα περιορισμού ενημέρωσης στον πίνακα *workson*

```
1 CREATE TABLE workson
2 (
3     empid      INTEGER      NOT NULL,
4     proid      INTEGER      NOT NULL,
5     CONSTRAINT pk_workson PRIMARY KEY (empid, proid),
```



# ON UPDATE RESTRICT

## Παράδειγμα περιορισμού ενημέρωσης στον πίνακα *workson*

```
1 CREATE TABLE workson
2 (
3     empid      INTEGER      NOT NULL,
4     proid      INTEGER      NOT NULL,
5     CONSTRAINT pk_workson PRIMARY KEY (empid, proid),
6     INDEX i_empid (empid),
7     INDEX i_proid (proid),
8     CONSTRAINT fk_empid FOREIGN KEY (empid)
9         REFERENCES employees(empid)
10        ON DELETE RESTRICT
11        ON UPDATE RESTRICT,
12     CONSTRAINT fk_proid FOREIGN KEY (proid)
13        REFERENCES projects(proid)
14        ON DELETE RESTRICT
15        ON UPDATE RESTRICT
16 );
```







## UPDATE *workson*

```
1 UPDATE projects
2     SET proid = 50
3     WHERE proid = 43;
```

```
4
5 ERROR 1451 (23000): Cannot delete or update a parent row:
6 a foreign key constraint fails
7 (workson, CONSTRAINT fk_proid FOREIGN KEY (proid)
8 REFERENCES projects (proid))
```



## UPDATE *workson*

```
1 UPDATE projects
2     SET proid = 50
3     WHERE proid = 43;
```

```
4
5 ERROR 1451 (23000): Cannot delete or update a parent row:
6 a foreign key constraint fails
7 (workson, CONSTRAINT fk_proid FOREIGN KEY (proid)
8 REFERENCES projects (proid))
```

**Προσοχή!** Δεν πρόκειται για λάθος στην εντολή **SQL**.

Το **Σύστημα Διαχείρισης Βάσεων Δεδομένων** δεν εκτελεί την ενημέρωση λόγω ρυθμίσεων που απαγορεύουν την **τροποποίηση πρωτεύοντος κλειδιού** στον **γονικό πίνακα** όταν υπάρχουν συσχετιζόμενες εγγραφές στον **θυγατρικό πίνακα**.



# Υπενθύμιση ( ενημέρωση πεδίων που δεν είναι πρωτεύον κλειδί)

## UPDATE *projects*

```
1 UPDATE projects
2   SET budget = 150000
3   WHERE proid = 43;
```

Query OK, 1 row affected (0.03 sec)

Rows matched: 1 Changed: 1 Warnings: 0

## SELECT *workson*

```
1 SELECT *
2   FROM workson
3   WHERE proid = 43;
```

empid	proid
-------	-------

189	43
-----	----

419	43
-----	----

593	43
-----	----

901	43
-----	----



Σας ευχαριστώ  
για την προσοχή σας.

Είμαι στη διάθεσή σας για σχόλια, απορίες και ερωτήσεις.

