

# Ενημέρωση βάσης δεδομένων με SQL

## DELETE, UPDATE, INSERT

Αθανάσιος Σταυρακούδης

<http://stavrakoudis.econ.uoi.gr>  
astavrak@uoi.gr  
@AStavrakoudis

Άνοιξη 2016



- 1 Γενικά για την αποθήκευση δεδομένων και την ενημέρωση της βάσης δεδομένων
- 2 Διαγραφή δεδομένων με την εντολή DELETE
- 3 Τροποποίηση δεδομένων με την εντολή UPDATE
- 4 Εισαγωγή δεδομένων INSERT
- 5 Ανάλυση ενός προβλήματος



# Σκοπός του μαθήματος



# Σκοπός του μαθήματος

- **Ενημέρωση** είναι η **τροποποίηση** του περιεχομένου της βάσης δεδομένων.



# Σκοπός του μαθήματος

- **Ενημέρωση** είναι η **τροποποίηση** του περιεχομένου της βάσης δεδομένων.
- Η γλώσσα **SQL** διαθέτει τρεις εντολές για το σκοπό αυτό:



# Σκοπός του μαθήματος

- **Ενημέρωση** είναι η **τροποποίηση** του περιεχομένου της βάσης δεδομένων.
- Η γλώσσα **SQL** διαθέτει τρεις εντολές για το σκοπό αυτό:
  - 1 **DELETE** για τη διαγραφή εγγραφών από πίνακες,



# Σκοπός του μαθήματος

- **Ενημέρωση** είναι η **τροποποίηση** του περιεχομένου της βάσης δεδομένων.
- Η γλώσσα **SQL** διαθέτει τρεις εντολές για το σκοπό αυτό:
  - 1 **DELETE** για τη διαγραφή εγγραφών από πίνακες,
  - 2 **UPDATE** για την τροποποίηση εγγραφών σε πίνακες,



# Σκοπός του μαθήματος

- **Ενημέρωση** είναι η **τροποποίηση** του περιεχομένου της βάσης δεδομένων.
- Η γλώσσα **SQL** διαθέτει τρεις εντολές για το σκοπό αυτό:
  - 1 **DELETE** για τη διαγραφή εγγραφών από πίνακες,
  - 2 **UPDATE** για την τροποποίηση εγγραφών σε πίνακες,
  - 3 **INSERT** για την εισαγωγή εγγραφών σε πίνακες.





# Σκοπός του μαθήματος

- **Ενημέρωση** είναι η **τροποποίηση** του περιεχομένου της βάσης δεδομένων.
- Η γλώσσα **SQL** διαθέτει τρεις εντολές για το σκοπό αυτό:
  - 1 **DELETE** για τη διαγραφή εγγραφών από πίνακες,
  - 2 **UPDATE** για την τροποποίηση εγγραφών σε πίνακες,
  - 3 **INSERT** για την εισαγωγή εγγραφών σε πίνακες.



# Σκοπός του μαθήματος

- **Ενημέρωση** είναι η **τροποποίηση** του περιεχομένου της βάσης δεδομένων.
- Η γλώσσα **SQL** διαθέτει τρεις εντολές για το σκοπό αυτό:
  - 1 **DELETE** για τη διαγραφή εγγραφών από πίνακες,
  - 2 **UPDATE** για την τροποποίηση εγγραφών σε πίνακες,
  - 3 **INSERT** για την εισαγωγή εγγραφών σε πίνακες.

## Σκοπός του μαθήματος είναι

να καταλάβετε τη **λειτουργία ενημέρωσης** μιας βάσης δεδομένων μέσα από τις εντολές **DELETE, UPDATE, INSERT**.



# Διαρκής μεταβολή του περιεχομένου



# Διαρκής μεταβολή του περιεχομένου

## Τι μεταβάλλεται

- Το **πλήθος των εγγραφών** ενός ή περισσότερων πινάκων.
- Οι **τιμές** ενός ή περισσότερων πεδίων σε έναν η περισσότερους πίνακες.



# Διαρκής μεταβολή του περιεχομένου

## Τι μεταβάλλεται

- Το **πλήθος των εγγραφών** ενός ή περισσότερων πινάκων.
- Οι **τιμές** ενός ή περισσότερων πεδίων σε έναν ή περισσότερους πίνακες.

## Τι δεν μεταβάλλεται

- Το πλήθος των πινάκων της βάσης δεδομένων (**σχήμα**).
- Η δομή των πινάκων της βάσης δεδομένων (**σχήμα**).



# Σε μια απόδειξη πώλησης;



Τι καταγράφεται κατά την πώληση ενός προϊόντος;

Εικόνα από: <http://en.wikipedia.org/wiki/Receipt>



Που καταγράφεται;



Που καταγράφεται η πώληση ενός προϊόντος;



# Συναλλαγές στο ATM



Τι καταγράφεται σε μια συναλλαγή σε ένα ATM;

Εικόνα από: [http://en.wikipedia.org/wiki/Automated\\_teller\\_machine](http://en.wikipedia.org/wiki/Automated_teller_machine)





# Καταγραφή τραπεζικών συναλλαγών.



Που καταγράφονται οι τραπεζικές συναλλαγές;



## Εγγραφή

Είναι και θα είναι πάντα δωρεάν!

Ημερομηνία γέννησης

Γιατί χρειάζεται να δώσω την ημερομηνία γέννησής μου;

Γυναίκα  Άνδρας

Τι καταγράφεται κατά την εγγραφή στο facebook;



# Στο facebook;

Που καταγράφονται τα μηνύματα του facebook;



# Στο facebook;

Που καταγράφονται τα μηνύματα του facebook;



# Στο facebook;

Που καταγράφονται τα μηνύματα του facebook;



# Στο facebook;

Που καταγράφονται τα μηνύματα του facebook;



- 1 Πόσο «ρεύμα» πληρώνει η εταιρεία facebook;
- 2 Πόση ενέργεια καταναλώνεται παγκοσμίως για τις ανάγκες του διαδικτύου;
- 3 Τι ξέρετε για τη Βόρεια Καρολίνα;



# Περιεχόμενα

- 1 Γενικά για την αποθήκευση δεδομένων και την ενημέρωση της βάσης δεδομένων
- 2 Διαγραφή δεδομένων με την εντολή DELETE
- 3 Τροποποίηση δεδομένων με την εντολή UPDATE
- 4 Εισαγωγή δεδομένων INSERT
- 5 Ανάλυση ενός προβλήματος



# Διαγραφή εγγραφών με την εντολή DELETE

**DELETE FROM** πίνακας  
**WHERE** παράσταση





# Διαγραφή εγγραφών με την εντολή DELETE

**DELETE FROM** πίνακας  
**WHERE** παράσταση

## Παρατηρήσεις για την εντολή DELETE

- 1 Χρειάζεται έγκυρο όνομα πίνακα που ακολουθεί φράση **DELETE**.



# Διαγραφή εγγραφών με την εντολή DELETE

**DELETE FROM** πίνακας  
**WHERE** παράσταση

## Παρατηρήσεις για την εντολή DELETE

- 1 Χρειάζεται έγκυρο όνομα πίνακα που ακολουθεί φράση **DELETE**.
- 2 Χρειάζεται έγκυρο όνομα πεδίου που ακολουθεί φράση **WHERE**.



# Διαγραφή εγγραφών με την εντολή DELETE

**DELETE FROM** πίνακας  
**WHERE** παράσταση

## Παρατηρήσεις για την εντολή DELETE

- 1 Χρειάζεται έγκυρο όνομα πίνακα που ακολουθεί φράση **DELETE**.
- 2 Χρειάζεται έγκυρο όνομα πεδίου που ακολουθεί φράση **WHERE**.
- 3 Η διαγραφή γίνεται κατά πλειάδες (εγγραφές) και όχι μεμονωμένα.



# Διαγραφή εγγραφών με την εντολή DELETE

**DELETE FROM** πίνακας  
**WHERE** παράσταση

## Παρατηρήσεις για την εντολή DELETE

- 1 Χρειάζεται έγκυρο όνομα πίνακα που ακολουθεί φράση **DELETE**.
- 2 Χρειάζεται έγκυρο όνομα πεδίου που ακολουθεί φράση **WHERE**.
- 3 Η διαγραφή γίνεται κατά πλειάδες (εγγραφές) και όχι μεμονωμένα.
- 4 Η φράση **WHERE** είναι προαιρετική, αλλά χωρίς αυτή θα γίνει διαγραφή όλων των εγγραφών.



# Διαγραφή δεδομένων ενός πίνακα

## Διαγραφή όλων των δεδομένων ενός πίνακα

```
1 DELETE  
2 FROM projects;
```



# Διαγραφή δεδομένων ενός πίνακα

## Διαγραφή όλων των δεδομένων ενός πίνακα

```
1 DELETE  
2 FROM projects;
```

## Προσοχή:

Θα διαγραφούν όλες οι εγγραφές του πίνακα. Όλες.



# Διαγραφή δεδομένων ενός πίνακα

## Διαγραφή όλων των δεδομένων ενός πίνακα

```
1 DELETE  
2 FROM projects;
```

## Προσοχή:

Θα διαγραφούν όλες οι εγγραφές του πίνακα. Όλες.

## Σφάλμα:

```
1 DELETE *  
2 FROM projects;
```



# Επιλεκτική διαγραφή δεδομένων





# Επιλεκτική διαγραφή δεδομένων

DELETE με WHERE για τη διαγραφή υποσυνόλου των δεδομένων

```
1 DELETE
2   FROM projects
3   WHERE proid = 43;
```



# Επιλεκτική διαγραφή δεδομένων

DELETE με WHERE για τη διαγραφή υποσυνόλου των δεδομένων

```
1 DELETE
2   FROM projects
3   WHERE proid = 43;
```

- 1 Τι θα συμβεί αν δεν υπάρχει εγγραφή στον πίνακα *projects* με *proid=43*;
- 2 Τι θα συμβεί στον πίνακα *workson* για την απασχόληση των υπαλλήλων στο έργο με *proid=43*;



# Διαγραφή δεδομένων με υποερώτημα

Να διαγραφεί η συμμετοχή υπαλλήλων στο έργο με κωδικό 43 για τους υπαλλήλους με μισθό άνω των 1400 €



# Διαγραφή δεδομένων με υποερώτημα

Να διαγραφεί η συμμετοχή υπαλλήλων στο έργο με κωδικό 43 για τους υπαλλήλους με μισθό άνω των 1400 €

```
1 DELETE
2   FROM workson
3   WHERE proid = 43
4   AND empid
```



# Διαγραφή δεδομένων με υποερώτημα

Να διαγραφεί η συμμετοχή υπαλλήλων στο έργο με κωδικό 43 για τους υπαλλήλους με μισθό άνω των 1400 €

```
1 DELETE
2   FROM workson
3   WHERE proid = 43
4         AND empid IN (SELECT empid
5                        FROM employees
6                        WHERE salary > 1400);
```



# Διαγραφή δεδομένων με υποερώτημα σύζευξης

Να διαγραφούν οι υπάλληλοι που δεν συμμετέχουν σε κανένα έργο



# Διαγραφή δεδομένων με υποερώτημα σύζευξης

Να διαγραφούν οι υπάλληλοι που δεν συμμετέχουν σε κανένα έργο

```
1 DELETE
2   FROM employees
3   WHERE empid IN
```



# Διαγραφή δεδομένων με υποερώτημα σύζευξης

Να διαγραφούν οι υπάλληλοι που δεν συμμετέχουν σε κανένα έργο

```
1 DELETE
2   FROM employees
3   WHERE empid IN
4     (SELECT e.empid
5      FROM employees e LEFT JOIN workson w
6       ON e.empid = w.empid
7      WHERE w.empid IS NULL);
```





- 1 Γενικά για την αποθήκευση δεδομένων και την ενημέρωση της βάσης δεδομένων
- 2 Διαγραφή δεδομένων με την εντολή DELETE
- 3 Τροποποίηση δεδομένων με την εντολή UPDATE**
- 4 Εισαγωγή δεδομένων INSERT
- 5 Ανάλυση ενός προβλήματος



# Ενημέρωση εγγραφών με την εντολή UPDATE

**UPDATE** πίνακας

**SET** πεδίο=τιμή

**WHERE** παράσταση



# Ενημέρωση εγγραφών με την εντολή UPDATE

**UPDATE** πίνακας

**SET** πεδίο=τιμή

**WHERE** παράσταση

## Παρατηρήσεις για την εντολή UPDATE

- 1 Χρειάζεται έγκυρο όνομα πίνακα που ακολουθεί φράση **UPDATE**.



# Ενημέρωση εγγραφών με την εντολή UPDATE

**UPDATE** πίνακας

**SET** πεδίο=τιμή

**WHERE** παράσταση

## Παρατηρήσεις για την εντολή UPDATE

- 1 Χρειάζεται έγκυρο όνομα πίνακα που ακολουθεί φράση **UPDATE**.
- 2 Χρειάζεται έγκυρο όνομα πεδίου που ακολουθεί φράση **SET**.



# Ενημέρωση εγγραφών με την εντολή UPDATE

**UPDATE** πίνακας

**SET** πεδίο=τιμή

**WHERE** παράσταση

## Παρατηρήσεις για την εντολή UPDATE

- 1 Χρειάζεται έγκυρο όνομα πίνακα που ακολουθεί φράση **UPDATE**.
- 2 Χρειάζεται έγκυρο όνομα πεδίου που ακολουθεί φράση **SET**.
- 3 Τήρηση κανόνων ακεραιότητας.



# Ενημέρωση εγγραφών με την εντολή UPDATE

**UPDATE** πίνακας

**SET** πεδίο=τιμή

**WHERE** παράσταση

## Παρατηρήσεις για την εντολή UPDATE

- 1 Χρειάζεται έγκυρο όνομα πίνακα που ακολουθεί φράση **UPDATE**.
- 2 Χρειάζεται έγκυρο όνομα πεδίου που ακολουθεί φράση **SET**.
- 3 Τήρηση κανόνων ακεραιότητας.
- 4 Ταυτόχρονη αλλαγή δύο ή περισσότερων πεδίων.



# Ενημέρωση εγγραφών με την εντολή UPDATE

**UPDATE** πίνακας

**SET** πεδίο=τιμή

**WHERE** παράσταση

## Παρατηρήσεις για την εντολή UPDATE

- 1 Χρειάζεται έγκυρο όνομα πίνακα που ακολουθεί φράση **UPDATE**.
- 2 Χρειάζεται έγκυρο όνομα πεδίου που ακολουθεί φράση **SET**.
- 3 Τήρηση κανόνων ακεραιότητας.
- 4 Ταυτόχρονη αλλαγή δύο ή περισσότερων πεδίων.
- 5 Η ενημέρωση γίνεται κατά πλειάδες (εγγραφές) και όχι μεμονωμένα.



# Ενημέρωση εγγραφών με την εντολή UPDATE

**UPDATE** πίνακας

**SET** πεδίο=τιμή

**WHERE** παράσταση

## Παρατηρήσεις για την εντολή UPDATE

- 1 Χρειάζεται έγκυρο όνομα πίνακα που ακολουθεί φράση **UPDATE**.
- 2 Χρειάζεται έγκυρο όνομα πεδίου που ακολουθεί φράση **SET**.
- 3 Τήρηση κανόνων ακεραιότητας.
- 4 Ταυτόχρονη αλλαγή δύο ή περισσότερων πεδίων.
- 5 Η ενημέρωση γίνεται κατά πλειάδες (εγγραφές) και όχι μεμονωμένα.
- 6 Η φράση **WHERE** είναι προαιρετική, αλλά χωρίς αυτή θα γίνει τροποποίηση όλων των εγγραφών.





# Παράδειγμα ενημέρωσης

Αύξηση μισθού κατά 100 € του υπαλλήλου με κωδικό 153



# Παράδειγμα ενημέρωσης

Αύξηση μισθού κατά 100 € του υπαλλήλου με κωδικό 153

```
1 UPDATE employees
2   SET salary = salary + 100
3   WHERE empid = 153;
```



# Παράδειγμα ενημέρωσης

Αύξηση μισθού κατά 100 € του υπαλλήλου με κωδικό 153

```
1 UPDATE employees
2   SET salary = salary + 100
3   WHERE empid = 153;
```

Πριν την ενημέρωση

```
1 SELECT empid, salary
2   FROM employees
3   WHERE empid = 153;
```

empid	salary
153	1321.92



# Παράδειγμα ενημέρωσης

Αύξηση μισθού κατά 100 € του υπαλλήλου με κωδικό 153

```
1 UPDATE employees
2   SET salary = salary + 100
3   WHERE empid = 153;
```

## Πριν την ενημέρωση

```
1 SELECT empid, salary
2   FROM employees
3   WHERE empid = 153;
```

empid	salary
153	1321.92

## Μετά την ενημέρωση

```
1 SELECT empid, salary
2   FROM employees
3   WHERE empid = 153;
```

empid	salary
153	1421.92



# Τροποποίηση δεδομένων με υποερώτημα

Να δοθεί αύξηση 7% στους υπαλλήλους που απασχολούνται σε περισσότερα από 2 έργα



# Τροποποίηση δεδομένων με υποερώτημα

Να δοθεί αύξηση 7% στους υπαλλήλους που απασχολούνται σε περισσότερα από 2 έργα

```
1 UPDATE employees
2   SET salary = salary*1.07
3   WHERE empid IN
```



# Τροποποίηση δεδομένων με υποερώτημα

Να δοθεί αύξηση 7% στους υπαλλήλους που απασχολούνται σε περισσότερα από 2 έργα

```
1 UPDATE employees
2   SET salary = salary*1.07
3   WHERE empid IN (SELECT empid
4                   FROM workson
5                   GROUP BY empid
6                   HAVING COUNT(*) > 2);
```



# Τροποποίηση δεδομένων με υποερώτημα

Να δοθεί αύξηση 7% στους υπαλλήλους που απασχολούνται σε περισσότερα από 2 έργα

```
1 UPDATE employees
2   SET salary = salary*1.07
3   WHERE empid IN (SELECT empid
4                   FROM workson
5                   GROUP BY empid
6                   HAVING COUNT(*) > 2);
```

- 1 Θα εκτελεστεί πρώτα το υποερώτημα και θα επιστρέψει μια λίστα τιμών **empid**.
- 2 Θα εκτελεστεί ενημέρωση των εγγραφών του πίνακα **employees** για όλες τις τιμές της λίστας **empid** που επιστρέφονται από το υποερώτημα.





# Ενημέρωση σύνθετου κλειδιού

Μετάθεση υπαλλήλου 419 από το έργο 43 στο έργο 5



# Ενημέρωση σύνθετου κλειδιού

Μετάθεση υπαλλήλου 419 από το έργο 43 στο έργο 5

```
1 UPDATE workson
2     SET proid = 5
3     WHERE proid = 43 AND empid = 419;
```



# Ενημέρωση σύνθετου κλειδιού

Μετάθεση υπαλλήλου 419 από το έργο 43 στο έργο 5

```
1 UPDATE workson
2   SET proid = 5
3   WHERE proid = 43 AND empid = 419;
```

Λάθος ενημέρωση της βάσης

```
1 UPDATE workson
2   SET proid = 5
3   WHERE empid = 419;
```

Αν ο υπάλληλος με κωδικό 419, απασχολείται σε 2 έργα, τότε ο δεύτερος τρόπος θα παράξει διπλότυπες τιμές στο **πρωτεύον κλειδί** του πίνακα **workson** και η ενημέρωση θα αποτύχει.



# Περιεχόμενα

- 1 Γενικά για την αποθήκευση δεδομένων και την ενημέρωση της βάσης δεδομένων
- 2 Διαγραφή δεδομένων με την εντολή DELETE
- 3 Τροποποίηση δεδομένων με την εντολή UPDATE
- 4 Εισαγωγή δεδομένων INSERT**
- 5 Ανάλυση ενός προβλήματος



# Η εντολή INSERT για την εισαγωγή δεδομένων

## Γενική μορφή της εντολής INSERT

```
1 INSERT INTO tablename(column1, column2, ..., columnN)  
2 VALUES (value1, value2, ..., valueN);
```



# Η εντολή INSERT για την εισαγωγή δεδομένων

## Γενική μορφή της εντολής INSERT

```
1 INSERT INTO tablename(column1, column2, ..., columnN)  
2 VALUES (value1, value2, ..., valueN);
```

## Γενικές παρατηρήσεις για την εντολή INSERT

- 1 Χρειάζεται έγκυρο όνομα πίνακα/πεδίων μετά τον όρο **INSERT INTO**.



# Η εντολή INSERT για την εισαγωγή δεδομένων

## Γενική μορφή της εντολής INSERT

```
1 INSERT INTO tablename(column1, column2, ..., columnN)  
2 VALUES (value1, value2, ..., valueN);
```

## Γενικές παρατηρήσεις για την εντολή INSERT

- 1 Χρειάζεται έγκυρο όνομα πίνακα/πεδίων μετά τον όρο **INSERT INTO**.
- 2 Δίνουμε το όνομα του πίνακα και μέσα σε παρενθέσεις τη λίστα πεδίων.



# Η εντολή INSERT για την εισαγωγή δεδομένων

## Γενική μορφή της εντολής INSERT

```
1 INSERT INTO tablename(column1, column2, ..., columnN)  
2 VALUES (value1, value2, ..., valueN);
```

## Γενικές παρατηρήσεις για την εντολή INSERT

- 1 Χρειάζεται έγκυρο όνομα πίνακα/πεδίων μετά τον όρο **INSERT INTO**.
- 2 Δίνουμε το όνομα του πίνακα και μέσα σε παρενθέσεις τη λίστα πεδίων.
- 3 Με τον όρο **VALUES** δίνουμε έγκυρες τιμές στα πεδία.





# Η εντολή INSERT για την εισαγωγή δεδομένων

## Γενική μορφή της εντολής INSERT

```
1 INSERT INTO tablename(column1, column2, ..., columnN)  
2 VALUES (value1, value2, ..., valueN);
```

## Γενικές παρατηρήσεις για την εντολή INSERT

- 1 Χρειάζεται έγκυρο όνομα πίνακα/πεδίων μετά τον όρο **INSERT INTO**.
- 2 Δίνουμε το όνομα του πίνακα και μέσα σε παρενθέσεις τη λίστα πεδίων.
- 3 Με τον όρο **VALUES** δίνουμε έγκυρες τιμές στα πεδία.
- 4 Το πλήθος πεδίων πρέπει να ταυτίζεται με το πλήθος τιμών.



# Η εντολή INSERT για την εισαγωγή δεδομένων

## Γενική μορφή της εντολής INSERT

```
1 INSERT INTO tablename(column1, column2, ..., columnN)  
2 VALUES (value1, value2, ..., valueN);
```

## Γενικές παρατηρήσεις για την εντολή INSERT

- 1 Χρειάζεται έγκυρο όνομα πίνακα/πεδίων μετά τον όρο **INSERT INTO**.
- 2 Δίνουμε το όνομα του πίνακα και μέσα σε παρενθέσεις τη λίστα πεδίων.
- 3 Με τον όρο **VALUES** δίνουμε έγκυρες τιμές στα πεδία.
- 4 Το πλήθος πεδίων πρέπει να ταυτίζεται με το πλήθος τιμών.
- 5 Μπορούμε να παραλείψουμε από τη λίστα πεδίων όσα πεδία μπορούν να πάρουν τιμή **NULL**.



# Η εντολή INSERT για την εισαγωγή δεδομένων

## Γενική μορφή της εντολής INSERT

```
1 INSERT INTO tablename(column1, column2, ..., columnN)  
2 VALUES (value1, value2, ..., valueN);
```

## Γενικές παρατηρήσεις για την εντολή INSERT

- 1 Χρειάζεται έγκυρο όνομα πίνακα/πεδίων μετά τον όρο **INSERT INTO**.
- 2 Δίνουμε το όνομα του πίνακα και μέσα σε παρενθέσεις τη λίστα πεδίων.
- 3 Με τον όρο **VALUES** δίνουμε έγκυρες τιμές στα πεδία.
- 4 Το πλήθος πεδίων πρέπει να ταυτίζεται με το πλήθος τιμών.
- 5 Μπορούμε να παραλείψουμε από τη λίστα πεδίων όσα πεδία μπορούν να πάρουν τιμή **NULL**.
- 6 Αν μια τιμή αντιστοιχεί σε πεδίο κειμένου ή ημερομηνίας χρειάζονται εισαγωγικά πχ 'Δημήτριος' ή '2005-05-10'.



# Εισαγωγή δεδομένων με INSERT, 3 τρόποι



# Εισαγωγή δεδομένων με INSERT, 3 τρόποι

Τοποθέτηση του υπαλλήλου με κωδικό 811 στο έργο με κωδικό 5

```
1 INSERT INTO workson (empid, proid)
2   VALUES (811, 5);
```



# Εισαγωγή δεδομένων με INSERT, 3 τρόποι

Τοποθέτηση του υπαλλήλου με κωδικό 811 στο έργο με κωδικό 5

```
1 INSERT INTO workson (empid, proid)
2   VALUES (811, 5);
```

Τοποθέτηση του υπαλλήλου με κωδικό 811 στο έργο με κωδικό 5

```
1 INSERT INTO workson (proid, empid)
2   VALUES (5, 811);
```



# Εισαγωγή δεδομένων με INSERT, 3 τρόποι

Τοποθέτηση του υπαλλήλου με κωδικό 811 στο έργο με κωδικό 5

```
1 INSERT INTO workson (empid, proid)
2   VALUES (811, 5);
```

Τοποθέτηση του υπαλλήλου με κωδικό 811 στο έργο με κωδικό 5

```
1 INSERT INTO workson (proid, empid)
2   VALUES (5, 811);
```

Τοποθέτηση του υπαλλήλου με κωδικό 811 στο έργο με κωδικό 5

```
1 INSERT INTO workson
2   VALUES (811, 5);
```



# Τι ισχύει για την εντολή INSERT





# Τι ισχύει για την εντολή INSERT

- 1 Το **πλήθος πεδίων** στη **λίστα πεδίων** πρέπει να είναι **ακριβώς το ίδιο με το πλήθος της λίστας τιμών**.



# Τι ισχύει για την εντολή INSERT

- 1 Το **πλήθος πεδίων** στη λίστα πεδίων πρέπει να είναι **ακριβώς το ίδιο** με το **πλήθος της λίστας τιμών**.
- 2 Οι τιμές της λίστας τιμών αντιστοιχίζονται **ένα-προς-ένα** στα πεδία της λίστας πεδίων.



# Τι ισχύει για την εντολή INSERT

- 1 Το **πλήθος πεδίων** στη λίστα πεδίων πρέπει να είναι **ακριβώς το ίδιο** με το **πλήθος της λίστας τιμών**.
- 2 Οι τιμές της λίστας τιμών αντιστοιχίζονται **ένα-προς-ένα** στα πεδία της λίστας πεδίων.
- 3 Η σειρά εμφάνισης στη λίστα πεδίων δεν παίζει ρόλο.



# Τι ισχύει για την εντολή INSERT

- 1 Το **πλήθος πεδίων** στη λίστα πεδίων πρέπει να είναι **ακριβώς το ίδιο** με το **πλήθος της λίστας τιμών**.
- 2 Οι τιμές της λίστας τιμών αντιστοιχίζονται **ένα-προς-ένα** στα πεδία της λίστας πεδίων.
- 3 Η σειρά εμφάνισης στη λίστα πεδίων δεν παίζει ρόλο.
- 4 Στη λίστα πεδίων πρέπει να συμπεριλαμβάνονται **όλα** τα πεδία που, κατά τη δημιουργία του πίνακα δηλώθηκαν ως **NOT NULL**.



# Τι ισχύει για την εντολή INSERT

- 1 Το **πλήθος πεδίων** στη λίστα πεδίων πρέπει να είναι **ακριβώς το ίδιο** με το **πλήθος της λίστας τιμών**.
- 2 Οι τιμές της λίστας τιμών αντιστοιχίζονται **ένα-προς-ένα** στα πεδία της λίστας πεδίων.
- 3 Η σειρά εμφάνισης στη λίστα πεδίων δεν παίζει ρόλο.
- 4 Στη λίστα πεδίων πρέπει να συμπεριλαμβάνονται **όλα** τα πεδία που, κατά τη δημιουργία του πίνακα δηλώθηκαν ως **NOT NULL**.
- 5 Όλες οι τιμές της λίστας τιμών θα πρέπει να είναι σύμφωνες με το **πεδίου ορισμού** όλων των των πεδίων του πίνακα.



# Τι ισχύει για την εντολή INSERT - συνέχεια

- 1 Αν πρέπει να δηλωθεί η τιμή **NULL** κατά την εισαγωγή δεδομένων δίνεται ως **NULL** και όχι με εισαγωγικά “NULL”.



# Τι ισχύει για την εντολή INSERT - συνέχεια

- 1 Αν πρέπει να δηλωθεί η τιμή **NULL** κατά την εισαγωγή δεδομένων δίνεται ως **NULL** και όχι με εισαγωγικά **"NULL"**.
- 2 Αν ένα πεδίο παραληφθεί από τη λίστα πεδίων, τότε εισάγεται η προκαθορισμένη (**DEFAULT**) τιμή.



# Τι ισχύει για την εντολή INSERT - συνέχεια

- 1 Αν πρέπει να δηλωθεί η τιμή **NULL** κατά την εισαγωγή δεδομένων δίνεται ως **NULL** και όχι με εισαγωγικά **"NULL"**.
- 2 Αν ένα πεδίο παραληφθεί από τη λίστα πεδίων, τότε εισάγεται η προκαθορισμένη (**DEFAULT**) τιμή.
- 3 Σε συσχετιζόμενους πίνακες





# Τι ισχύει για την εντολή INSERT - συνέχεια

- 1 Αν πρέπει να δηλωθεί η τιμή **NULL** κατά την εισαγωγή δεδομένων δίνεται ως **NULL** και όχι με εισαγωγικά **"NULL"**.
- 2 Αν ένα πεδίο παραληφθεί από τη λίστα πεδίων, τότε εισάγεται η προκαθορισμένη (**DEFAULT**) τιμή.
- 3 Σε συσχετιζόμενους πίνακες
  - Η εισαγωγή δεδομένων στον **πατρικό πίνακα** δεν έχει καμία επίδραση στον **θυγατρικό πίνακα**.



# Τι ισχύει για την εντολή INSERT - συνέχεια

- 1 Αν πρέπει να δηλωθεί η τιμή **NULL** κατά την εισαγωγή δεδομένων δίνεται ως **NULL** και όχι με εισαγωγικά **"NULL"**.
- 2 Αν ένα πεδίο παραληφθεί από τη λίστα πεδίων, τότε εισάγεται η προκαθορισμένη (**DEFAULT**) τιμή.
- 3 Σε συσχετιζόμενους πίνακες
  - Η εισαγωγή δεδομένων στον **πατρικό πίνακα** δεν έχει καμία επίδραση στον **θυγατρικό πίνακα**.
  - Η εισαγωγή δεδομένων στο **θυγατρικό πίνακα**, επηρεάζεται από την **ακεραιότητα αναφορών**.



# Τι ισχύει για την εντολή INSERT - συνέχεια

- 1 Αν πρέπει να δηλωθεί η τιμή **NULL** κατά την εισαγωγή δεδομένων δίνεται ως **NULL** και όχι με εισαγωγικά **"NULL"**.
- 2 Αν ένα πεδίο παραληφθεί από τη λίστα πεδίων, τότε εισάγεται η προκαθορισμένη (**DEFAULT**) τιμή.
- 3 Σε συσχετιζόμενους πίνακες
  - Η εισαγωγή δεδομένων στον **πατρικό πίνακα** δεν έχει καμία επίδραση στον **θυγατρικό πίνακα**.
  - Η εισαγωγή δεδομένων στο **θυγατρικό πίνακα**, επηρεάζεται από την **ακεραιότητα αναφορών**.
- 4 Αν για οποιοδήποτε λόγο, αποτύχει η εκτέλεση μιας εντολής **INSERT** η αποτυχία ισχύει **συνολικά για την εγγραφή** και όχι μόνο για κάποια από τα πεδία.



# Αποτυχία INSERT



# Αποτυχία INSERT

Λείπει το υποχρεωτικό πεδίο lastname

```
1 INSERT INTO employees (empid, depid, hiredate)
2 VALUES (199, 5, CURRENT_DATE());
```



# Αποτυχία INSERT

Λείπει το υποχρεωτικό πεδίο lastname

```
1 INSERT INTO employees (empid, depid, hiredate)
2 VALUES (199, 5, CURRENT_DATE());
```

Λείπει η τιμή από το πεδίο lastname

```
1 INSERT INTO employees (empid, depid, lastname)
2 VALUES (199, 5)
```



# Αποτυχία INSERT

Λείπει το υποχρεωτικό πεδίο lastname

```
1 INSERT INTO employees (empid, depid, hiredate)
2 VALUES (199, 5, CURRENT_DATE());
```

Λείπει η τιμή από το πεδίο lastname

```
1 INSERT INTO employees (empid, depid, lastname)
2 VALUES (199, 5)
```

Η τιμή του πρωτεύοντος κλειδιού υπάρχει ήδη

```
1 INSERT INTO employees (empid, depid, lastname)
2 VALUES (811, 4, 'Κάτου')
```



# Εναλλακτικός τρόπος για την εντολή INSERT

Εισαγωγή νέας εγγραφής στον πίνακα employees

```
1  INSERT INTO employees
2      SET empid      = 199,
3          lastname   = 'Κάτου',
4          firstname  = 'Άννα',
5          depid      = 5,
6          salary     = 1500,
7          hiredate   = '2010-03-15';
```





# Εναλλακτικός τρόπος για την εντολή INSERT

Εισαγωγή νέας εγγραφής στον πίνακα employees

```
1  INSERT INTO employees
2      SET empid      = 199,
3          lastname   = 'Κάτου',
4          firstname  = 'Άννα',
5          depid      = 5,
6          salary     = 1500,
7          hiredate   = '2010-03-15';
```

❶ Ο τρόπος αυτός μοιάζει με την εντολή **UPDATE**.



# Εναλλακτικός τρόπος για την εντολή INSERT

Εισαγωγή νέας εγγραφής στον πίνακα employees

```
1  INSERT INTO employees
2      SET empid      = 199,
3          lastname   = 'Κάτου',
4          firstname  = 'Άννα',
5          depid      = 5,
6          salary     = 1500,
7          hiredate   = '2010-03-15';
```

- 1 Ο τρόπος αυτός μοιάζει με την εντολή **UPDATE**.
- 2 Χρησιμοποιούμε τη γραφή **SET column = value**.



# Εναλλακτικός τρόπος για την εντολή INSERT

Εισαγωγή νέας εγγραφής στον πίνακα employees

```
1  INSERT INTO employees
2      SET empid      = 199,
3          lastname   = 'Κάτου',
4          firstname  = 'Άννα',
5          depid      = 5,
6          salary     = 1500,
7          hiredate   = '2010-03-15';
```

- 1 Ο τρόπος αυτός μοιάζει με την εντολή **UPDATE**.
- 2 Χρησιμοποιούμε τη γραφή **SET column = value**.
- 3 Χωρίζουμε τη λίστα πεδίων με κόμματα.



# Εναλλακτικός τρόπος για την εντολή INSERT

Εισαγωγή νέας εγγραφής στον πίνακα employees

```
1  INSERT INTO employees
2      SET empid      = 199,
3          lastname   = 'Κάτου',
4          firstname  = 'Άννα',
5          depid      = 5,
6          salary     = 1500,
7          hiredate   = '2010-03-15';
```

- 1 Ο τρόπος αυτός μοιάζει με την εντολή **UPDATE**.
- 2 Χρησιμοποιούμε τη γραφή **SET column = value**.
- 3 Χωρίζουμε τη λίστα πεδίων με κόμματα.
- 4 Η σειρά εμφάνισης των πεδίων δεν έχει σημασία.



# Εναλλακτικός τρόπος για την εντολή INSERT

Εισαγωγή νέας εγγραφής στον πίνακα employees

```
1  INSERT INTO employees
2      SET empid      = 199,
3          lastname   = 'Κάτου',
4          firstname  = 'Άννα',
5          depid      = 5,
6          salary     = 1500,
7          hiredate   = '2010-03-15';
```

- 1 Ο τρόπος αυτός μοιάζει με την εντολή **UPDATE**.
- 2 Χρησιμοποιούμε τη γραφή **SET column = value**.
- 3 Χωρίζουμε τη λίστα πεδίων με κόμματα.
- 4 Η σειρά εμφάνισης των πεδίων δεν έχει σημασία.
- 5 **Ισχύουν όλοι οι περιορισμοί ακεραιότητας δεδομένων (κλειδιά, τιμές NULL, κτλ).**



# Εισαγωγή δεδομένων με υποερώτημα

Να τοποθετηθεί στο έργο με 5 ο υπάλληλος του τμήματος 4 που παίρνει το μεγαλύτερο μισθό

```
1 INSERT INTO workson(proid, empid)
2 SELECT 5, empid
3     FROM employees
4     WHERE salary = ( SELECT MAX(salary)
5                       FROM employees
6                       WHERE depid = 4 );
```



# Εισαγωγή δεδομένων με υποερώτημα

Να τοποθετηθεί στο έργο με 5 ο υπάλληλος του τμήματος 4 που παίρνει το μεγαλύτερο μισθό

```
1 INSERT INTO workson(proid, empid)
2 SELECT 5, empid
3     FROM employees
4     WHERE salary = ( SELECT MAX(salary)
5                       FROM employees
6                       WHERE depid = 4 );
```

- 1 Δε χρειάζεται η φράση **VALUES** σε υποερωτήματα μετά το **INSERT**.
- 2 Προσοχή στην παράσταση **SELECT 5, empid**.
- 3 Τι θα συμβεί αν όλοι οι υπάλληλοι του τμήματος 4 έχουν τον ίδιο μισθό;



# Περιεχόμενα

- 1 Γενικά για την αποθήκευση δεδομένων και την ενημέρωση της βάσης δεδομένων
- 2 Διαγραφή δεδομένων με την εντολή DELETE
- 3 Τροποποίηση δεδομένων με την εντολή UPDATE
- 4 Εισαγωγή δεδομένων INSERT
- 5 Ανάλυση ενός προβλήματος





# Διπλή εισαγωγή

Εισαγωγή επιπλέον εγγραφής για τον ίδιο υπάλληλο

```
1 INSERT INTO employees(empid, lastname, depid)
2 VALUES (781, 'Μικράκη', 4);
```



# Διπλή εισαγωγή

Εισαγωγή επιπλέον εγγραφής για τον ίδιο υπάλληλο

```
1 INSERT INTO employees(empid, lastname, depid)
2 VALUES (781, 'Μικράκη', 4);
```

Ωστόσο, η κ. Μικράκη υπάρχει ήδη στη βάση δεδομένων

```
1 SELECT empid, lastname, depid
2 FROM employees
3 WHERE lastname = 'Μικράκη';
```

```
4 empid  lastname  depid
5 -----
6 780    Μικράκη    4
7 781    Μικράκη    4
8
```



# Λανθασμένη διόρθωση του προβλήματος

Διαγραφή της υπαλλήλου με επώνυμο ΜΙΚΡΑΚΗ

```
1 DELETE FROM employees
2   WHERE lastname = 'Μικράκη';
3
4 Query OK, 2 rows affected (0.00 sec)
```



# Λανθασμένη διόρθωση του προβλήματος

Διαγραφή της υπαλλήλου με επώνυμου ΜΙΚΡΑΚΗ

```
1 DELETE FROM employees
2   WHERE lastname = 'Μικράκη';
3
4 Query OK, 2 rows affected (0.00 sec)
```

Τώρα η κ. ΜΙΚΡΑΚΗ δεν υπάρχει καθόλου στη βάση δεδομένων

```
1 SELECT empid, lastname, depid
2   FROM employees
3   WHERE lastname = 'Μικράκη';
4
5 Empty set (0.01 sec)
```

Διαγράφηκαν και οι δύο εγγραφές!!



# Πιο προσεκτική διόρθωση του προβλήματος

Διαγραφή της υπαλλήλου με κωδικό υπαλλήλου 781

```
1 DELETE FROM employees
2   WHERE empid=781;
3
4 Query OK, 1 row affected (0.00 sec)
```



# Πιο προσεκτική διόρθωση του προβλήματος

Διαγραφή της υπαλλήλου με κωδικό υπαλλήλου 781

```
1 DELETE FROM employees
2   WHERE empid=781;
3
4 Query OK, 1 row affected (0.00 sec)
```

Τώρα η κ. Μικράκη υπάρχει μία φορά στη βάση δεδομένων

```
1 SELECT empid, lastname, depid
2   FROM employees
3   WHERE lastname = 'Μικράκη';
```

```
4 empid  lastname  depid
5 -----
6 780    Μικράκη    4
```



Σας ευχαριστώ για την προσοχή σας.

Είμαι στη διάθεσή σας για σχόλια, απορίες και ερωτήσεις.

