



Περιεχόμενα

- Εντολές SQL διαχείρισης εγγραφών
- Σύνταξη επερωτήσεων
- Εκτέλεση εντολών SQL στην PHP (Βιβλιοθήκη mysqli)



Προσθήκη νέων εγγραφών Η εντολή INSERT

- Εντολή προσθήκης νέας εγγραφής
INSERT INTO <table_name> (<field1>, <field2>, ..., <fieldN>)
VALUES ('val1', 'val2', ..., 'valN')

Τα αλφαριθμητικά εμπεριέχονται μέσα σε αποστρόφους ή εισαγωγικά.

Εαν τα εισαγωγικά που περικλείουν αλφαριθμητικά περιέχονται και μέσα στην τιμή τους πρέπει να διαφοροποιούνται με τον χαρακτήρα \.

Π.χ. Προσθήκη πεδίου O'Connor says "Hi"

Με απλά εισαγωγικά: 'O'Connor says "Hi" '

Με διπλά εισαγωγικά: "O'Connor says \"Hi\" "



Προσθήκη νέων εγγραφών

Η εντολή INSERT

Δε μπορούν να παραλειφθούν τιμές για πεδία που έχουν οριστεί στον πίνακα με την παράμετρο NOT NULL.

Π.χ. Δεν πρωτεύοντα κλειδιά με περιορισμό AUTO INCREMENT αρχικοποιούνται αυτομάτως οπότε δε είναι αναγκαία η εισαγωγή τους

Χρησιμοποιούνται σε πρωτεύοντα κλειδιά. Έτσι ο χρήστης δε χρειάζεται να μεριμνήσει για την απόδοση τιμών σε αυτά.



Εκτέλεση ερωτήσεων Η εντολή SELECT

- Η απλούστερη εντολή SELECT:

```
SELECT <field1>, <field2>, ..., <fieldN> FROM <table_name>;
```

Επιστρέφει τα περιεχόμενα των πεδίων <field1>, field2,...,fieldN όλων των εγγραφών που περιέχονται στον πίνακα <table_name>.

- Για επιστροφή όλων των πεδίων χρησιμοποιείται το σύμβολο *.

```
SELECT * FROM <table_name>;
```

Επιστρέφει τις τιμές όλων τα πεδίων όλων των εγγραφών του πίνακα.
(με τη σειρά που τα πεδία είναι δηλωμένα στον πίνακα).



Εισάγοντας φίλτρα αναζήτησης

- Σχεδόν πάντα αναζητούνται εγγραφές που τα πεδία τους ικανοποιούν κάποια συνθήκη.
- Τα φίλτρα αναζήτησης εισάγεται μετά την εντολή `SELECT` με την Προσθήκη της λέξης κλειδιού `WHERE`
- Δομή φίλτρου αναζήτησης
`<SELECT statement> WHERE <condition>;`
- `<condition>` : ορίζει τις συνθήκες που πρέπει να πληρούν οι εγγραφές του πίνακα προκειμένου να συμπεριληφθούν στα επιστρεφόμενα αποτελέσματα



Συνθήκες σε φίλτρα ερωτήσεων

Αριθμητικές συγκρίσεις

- Αριθμητικές συγκρίσεις:
 1. Ισότητα: (=) π.χ. `Salary = 800`
 2. Μεγαλύτερο (>) π.χ. `Salary > 800`,
Μεγαλύτερο ή ίσο (>=) π.χ. `Salary >= 800`
 3. Μικρότερο (<), π.χ. `Salary < 800`
Μικρότερο ή ίσο (<=) π.χ. `Salary <= 800`
 4. Ανισότητα: (<>) π.χ. `Salary <> 800`



Συγκρίσεις αλφαριθμητικών

LIKE: ορίζει σχέση ισότητας αλφαριθμητικών

WHERE <field> LIKE <pattern>

- `SELECT * FROM Employees WHERE LastName LIKE 'Connor';`
Ανάκληση των εγγραφών με ακριβές επώνυμο Connor
- Χρήση wildcards `_`, `%` για τον εντοπισμό εγγραφών που εμπεριέχουν το αλφαριθμητικό στο πεδίο τους :
 - `_` οποιοσδήποτε χαρακτήρας (οπωσδήποτε όμως ένας)
 - `%` οποιαδήποτε ακολουθία χαρακτήρων (ακόμη και κανένας χαρακτήρας)
- Π.χ. Έστω εγγραφές με ονόματα Connor, O'Connor, Connors
 1. `WHERE Name LIKE 'Connor_'`
Επιστρέφει: Connors
 2. `WHERE Name Like '%Connor'`
Επιστρέφει: Connor, O'Connor
 3. `WHERE Name Like '%Connor%'`
Επιστρέφει: Connor, O'Connor, Connors



Συνθήκες σε φίλτρα επερωτήσεων

- Συνδυασμός πολλαπλών συνθηκών με τελετές AND, OR
- Διάξευση - OR:
- WHERE <condition1> OR <condition2>
- Ισχύει για εγγραφές για τις οποίες ισχύει τουλάχιστον 1 από τις condition1 , condition2
- *SELECT Name FROM Employees WHERE Salary < 800 OR Salary > 1500*
- *Επιστρέφει όλες τις εγγραφές που το πεδίο τους Salary έχει τιμή < 800 ή > 1500*
- Σύζευξη AND:
- WHERE <condition1> AND <condition2>;
- Ισχύει για εγγραφές που πληρούν και τα δύο κριτήρια condition1, condition2
- Π.χ.
- *SELECT Name FROM Employees WHERE Salary > 800 AND Salary < 1500*
- *Επιστρέφει όλες τις εγγραφές που το πεδίο τους Salary έχει τιμή >800 και <1500*



Εύρος τιμών ως συνθήκη (BETWEEN)

- BETWEEN: φίλτρο των αποτελεσμάτων
Οι τιμές του εξεταζόμενου πεδίου εμπεριέχονται σε ένα δεδομένο εύρος τιμών.
- Σύνταξη:
WHERE <field> BETWEEN <minValue> AND <maxValue>
- Π.χ.
SELECT Name from Employees WHERE Salary BETWEEN 800 AND 1500;
Επιστρέφει τιμή στο πεδίο Salary μεταξύ των τιμών 800 και 1500



Ταξινόμηση αποτελεσμάτων

- Οι επιστρεφόμενες εγγραφές μπορούν να ταξινομηθούν με άυξουσα ή φθίνουσα σειρά ως προς οποιοδήποτε πεδίο
- Τα κριτήρια ταξινόμησης ορίζονται σε μια εντολή SQL με τη δήλωση ORDER BY.
- Σύνταξη:
ORDER BY <field> ASC/DESC;
- ASC: ταξινόμηση αποτελεσμάτων ως προς το πεδίο <field> με αύξουσα σειρά
DESC: ταξινόμηση αποτελεσμάτων ως προς το πεδίο <field> με φθίνουσα σειρά

*SELECT * FROM Employees ORDER BY LastName ASC;*

Επιστροφή εγγραφών ταξινομημένων ως προς το επώνυμο με αλφαβητική σειρά



Τροποποίηση εγγραφών

Η εντολή UPDATE

- UPDATE <table_name> SET <field1>=val1 , <field2> = val2 , ..., <fieldN> = valN WHERE <condition>;
- <condition>: συνθήκη ενημέρωσης των πεδίων field1, field2, ... fieldN
- Παράδειγμα:
UPDATE <Employees> SET Salary = 1000 WHERE Salary = 800;



Διαγραφή εγγραφών Η εντολή DELETE

- DELETE FROM <table_name> WHERE <condition>;
- <condition>: συνθήκη διαγραφής εγγραφών
- Π.χ.
DELETE FROM Employees WHERE LastName = 'Smith' ;



Ανάκτηση πεδίων από πολλαπλούς πίνακες (JOIN)

- Πολλές φορές, είναι αναγκαία η ανάκτηση δεδομένων που περιέχονται σε πολλαπλούς πίνακες με μία εντολή. Αυτό γίνεται με εντολές τύπου JOIN.
- Για την ανάκτηση δεδομένων από πολλαπλούς πίνακες χρησιμοποιούνται εντολές τύπου JOIN.



Ανάκτηση πεδίων από πολλαπλούς πίνακες (JOIN)

- Θεωρούμε δύο πίνακες LeftTable και RightTable
- Τα πεδία προσδιορίζονται με τη σύνταξη Table . Field
Χρήσιμο για το διαχωρισμό συνώνυμων πεδίων που έχουν το ίδιο όνομα και βρίσκονται σε διαφορετικούς πίνακες
- Δομή επερώτησης:
SELECT
LeftTable.Leftfield1, ..., LeftTable.leftFieldN
RightTable.rightField1, ..., RightTable.RightFieldN
FROM LeftTable <JOIN statement>
- **4 διαφορετικοί τύποι δηλώσεων JOIN**



INNER JOIN / JOIN

Δομή επερώτησης:

SELECT

LeftTable.Leftfield1, ..., LeftTable.leftFieldN

RightTable.rightField1, ..., RightTable.RightFieldN

FROM LeftTable INNER JOIN RightTable

ON LeftTable.CommonField1 = RightTable.CommonField2

- Δήλωση του κοινού πεδίου:
ON LeftTable.CommonField1 = RightTable.CommonField2 :
- **INNER JOIN / JOIN:** Επιστρέφει τις εγγραφές που οι τιμές του κοινού πεδίου τους εμφανίζονται και στους 2 πίνακες.



Παράδειγμα INNER JOIN

Employees

EmployeeID	Name	Department
1	John Smith	1
2	John Doe	3
3	Jane Doe	4

Departments

DeptID	DeptName
1	Sales
2	Research
3	Customer Service

```
SELECT Employees.Name Departments.DeptName FROM Employees INNER JOIN  
Departments ON Employees.Department=Departments.DeptID;
```

Επιστρέφει:

John Smith Sales

John Doe Customer Service



LEFT JOIN

- **LEFT JOIN:** Επιστρέφει όλες τις εγγραφές από τον «αριστερό» πίνακα ακόμη κι αυτές που οι τιμές του κοινού τους πεδίου δεν εμφανίζονται στο «δεξιό» πίνακα.
- Δομή επερώτησης:
SELECT
LeftTable.Leftfield1, ..., LeftTable.leftFieldN
RightTable.rightField1, ..., RightTable.RightFieldN
FROM LeftTable LEFT JOIN RightTable ON
LeftTable.CommonField1 = RightTable.CommonField2



Παράδειγμα LEFT JOIN

Employees

EmployeeID	Name	Department
1	John Smith	1
2	John Doe	3
3	Jane Doe	4

Departments

DeptID	DeptName
1	Sales
2	Research
3	Customer Service

```
SELECT Employees.Name Departments.DeptName FROM Employees LEFT JOIN Departments  
ON Employees.Department=Departments.DeptID;
```

Επιστρέφει:

John Smith

Sales

John Doe

Customer Service

Jane Doe



RIGHT JOIN

- **RIGHT JOIN:** Επιστρέφει όλες τις εγγραφές από τον «δεξιό πίνακα» ακόμη κι αυτές που οι τιμές του συσχετισμένου τους πεδίου δεν εμφανίζονται στον αριστερό πίνακα.

SELECT

LeftTable.Leftfield1, ..., LeftTable.leftFieldN

RightTable.rightField1, ..., RightTable.RightFieldN

FROM LeftTable

RIGHT JOIN RightTable

ON LeftTable.CommonField1 = RightTable.CommonField2

WHERE <condition>;



Παράδειγμα RIGHT JOIN

Employees

EmployeeID	Name	Department
1	John Smith	1
2	John Doe	3
3	Jane Doe	4

Departments

DeptID	DeptName
1	Sales
2	Research
3	Customer Service

```
SELECT Employees.Name Departments.DeptName FROM Employees RIGHT JOIN Departments  
ON Employees.Department=Departments.DeptID;
```

Επιστρέφει:

John Smith Sales
 Research
John Doe Customer Service



FULL JOIN

FULL JOIN: Επιστρέφει όλες τις εγγραφές και των δύο πινάκων.

SELECT

LeftTable.Leftfield1, ..., LeftTable.leftFieldN

RightTable.rightField1, ..., RightTable.RightFieldN

FROM LeftTable FULL JOIN RightTable ON

LeftTable.CommonField1 = RightTable.CommonField2



Δικαιώματα πρόσβασης

GRANT <privileges> ON <resource> TO 'username' WITH <extra options>

<privilege1, privilege2,, privilegeN>: εκχωρούμενα δικαιώματα χρήσης (π.χ. SELECT, INSERT, DELETE, CREATE, ALTER, DROP)

<RESOURCE>: πόρος στον οποίο εκχωρείται δικαίωμα (π.χ. Πίνακας ή βάση δεδομένων)

Σύνταξη <database> . <table>

dbname .* : Εκχώρηση δικαιώματος χρήσης για όλους τους πίνακες της ΒΔ dbname

<username> χρήστης στον οποίο εκχωρούνται δικαιώματα χρήσης

WITH GRANT OPTION

Δίνει στο χρήστη username τη δυνατότητα να εκχωρήσει με τη σειρά του τα δικαιώματά <privileges> σε τρίτους χρήστες

GRANT SELECT ON Company.Employees TO 'john';



Ανάκληση δικαιωμάτων χρήσης

- Η αναίρεση δικαιωμάτων χρήσης εκτελείται με τη δήλωση REVOKE
Σύνταξη:

REVOKE <privilege> ON <resource> FROM 'username';

<privilege>: αφαιρούμενο δικαίωμα ή δικαιώματα (π.χ. INSERT, UPDATE, DELETE)

GRANT OPTION: αφαίρεση του δικαιώματος να εκχωρούνται δικαιώματα

<resource>: ο πόρος για τον οποίο ανακαλούνται τα δικαιώματα χρήσης

Σύνταξη: <database> . <table>

- Ο χρήστης που εκτελεί μία εντολή αναίρεσης δικαιωμάτων θα πρέπει να έχει το δικαίωμα GRANT OPTION για τον εκάστοτε πόρο καθώς και τα δικαιώματα χρήσης που ανακαλεί.



Σύνδεση σε DB στην PHP

Η βιβλιοθήκη mySQLi

Σύνδεση και αποσύνδεση από βάση δεδομένων

Διαχείριση εγγραφών (προσθήκη επερωτήσεις, ενημερώση, διαγραφή)

- Το λογισμικό Apache - mySQL - PHP συνδυάζεται για τη δημιουργία δυναμικών ιστοσελίδων. (WAMP – LAMP)
- mySQLi βιβλιοθήκη σύνδεσης σε βάσεις δεδομένων μέσω PHP
- Η PHP επιτρέπει τη διασύνδεση και αλληλεπίδραση με βάσεις δεδομένων σε mySQL με τη βιβλιοθήκη mySQLi (mySQL Improved)
- Δύο μορφές (αντικειμενοστραφής και διαδικαστική)
- Procedural



Σύνδεση σε βάση δεδομένων

```
$mysqli = mysqli_connect('servername','user','pass','database_name');
```

Servername: όνομα διακομιστή που παρέχει υπηρεσία διαχείρισης βάσεων δεδομένων

Για τοπική σύνδεση δίνεται το αλφαριθμητικό “localhost” ή “127.0.0.1”

Για απομακρυσμένη σύνδεση δίνεται η διεύθυνση IP ή το όνομα του διακομιστή (αν έχει καταχώρηση DNS)

user: χρήστης

pass: κωδικός πρόσβασης

database_name: Η βάση δεδομένων που θα χρησιμοποιήσει ο χρήστης



Έλεγχος σφάλματος κατά τη σύνδεση σε βάση δεδομένων

Έλεγχος σφάλματος κατά τη σύνδεση

`mysqli_connect_errno()` σε δομή `if`

Επιστρέφει `TRUE` σε περίπτωση σφάλματος .

Μήνυμα σφάλματος επιστρέφεται από τη συνάρτηση

`mysqli_connect_error()`

`mysqli_get_host_info($mysqli)`

Επιστρέφει πληροφορίες σχετικές με τη σύνδεση στην εκάστοτε βάση
δεδομένων

Αποσύνδεση από βάση

`mysqli_close($mysqli);`



Εκτέλεση εντολών SQL

Όλες οι εντολές SQL αποστέλλονται στη MySQL με τη συνάρτηση `mysqli_query`

```
mysqli_query( $mysqli, $sql );
```

Η `mysqli_query` επιτρέπει την εκτέλεση οποιαδήποτε εντολής SQL (INSERT, SELECT κλπ.)

Εκτέλεση επερώτησης:

```
$sql="SELECT * FROM Customers";
```

```
mysqli_query($mysqli, $sql);
```

Για να εκτελεστεί επιτυχώς η εντολή, ο χρήστης πρέπει να έχει τα απαραίτητα δικαιώματα χρήσης.



Έλεγχος σφαλμάτων

Μια εντολή ενδέχεται να αποτύχει για διάφορους λόγους.

Έλεγχος εκτέλεσης

Σε επιτυχή εκτέλεση η `mysqli_query()` επιστρέφει μη μηδενική τιμή

```
$result = mysqli_query( $mysqli, $someSQLCommand);
if ($result ==TRUE)
{
    echo "SQL command executed successfully";
}
else
{
    printf("Error: %s\n", mysqli_error($mysqli) );
}
mysqli_close($mysqli);
```



Προσθήκη, τροποποίηση, διαγραφή εγγραφών

INSERT, UPDATE, DELETE:

Εντολές που δεν επιστρέφουν αποτελέσματα, αλλά μόνο επιτυχία ή αποτυχία της εκτέλεσής τους

1. Σύνταξη δήλωσης `$sql= "<statement>"`
2. Εκτέλεση δήλωσης `mysqli_query`
3. Επιβεβαίωση εκτέλεσης



Ανάκτηση εγγραφών

Οι εντολές SELECT επιστρέφουν πολλαπλές εγγραφές

Μετά την εκτέλεση της εντολής αναζήτησης ο χρήστης προσπελαύνει τις εγγραφές σειριακά

Σειρά ενεργειών

1. Σύνταξη επερωτήματος ανάκτησης
2. Εκτέλεση επερωτήματος (mysqli_query)
3. Επιβεβαίωση εκτέλεσης
4. Ανάκτηση κάθε μεμονωμένης εγγραφής από το επιστρεφόμενο αποτέλεσμα \$resultset της επερώτησης

```
$recordArray = mysqli_fetch_array($resultSet, MYSQLI_XXXX)
```

σε κάθε διαδοχική εκτέλεσή της επιστρέφει μία εγγραφή. Οι τιμές των πεδίων αποθηκεύονται με τη μορφή αριθμητικού array (MYSQLI_NUM) ή με τη μορφή συσχετιστικού array (MYSQLI_ASSOC) ή και με τις δύο μορφές (MYSQLI_BOTH)



Διαδικασία ανάκτησης εγγραφών

```
resultSet = mysqli_query("<SELECT statement>")
```

```
If ($resultSet == TRUE)
```

```
{
```

```
    while ($retrievedRecord = mysqli_fetch_array($resultSet,MYSQLI_ASSOC))
```

```
    {
```

```
        //Αποθήκευση τιμών πεδίων σε μεταβλητές
```

```
        $field1value = $retrievedRecord['fieldName1'];
```

```
        ...
```

```
        $fieldNvalue = $retrievedRecord['fieldNameN'];
```

```
    }
```

```
}
```

```
else
```

```
{
```

```
    printf("Execution error: %s\n",mysqli_error($mysqli));
```

```
}
```




Αποδέσμευση πόρων

- Κάθε εντολή ανάκτησης δεδομένων δεσμεύει μνήμη για την αποθήκευση των αποτελεσμάτων της.
- `mysqli_free_result($resultSet);`
Ελευθερώνει τη μνήμη που δεσμεύτηκε για την αποθήκευση των αποτελεσμάτων `$resultSet` του ερωτήματος.
- Εκτελείται πριν το κλείσιμο της σύνδεσης με `mysqli_close()`.