

# בסיס נתונים

[ההרחבות MySQL ו-MySQLi](#)  
[יצירת חיבור עם בסיס הנתונים](#)  
[ביצוע שאילתות לבסיס הנתונים](#)

## ההרחבות MySQL ו-MySQLi

התמיכה של PHP בעבודה עם בסיס הנתונים MySQL קיימת עוד מגירסה PHP 2.0. שפת התיכנות PHP ובסיס הנתונים MySQL קשורים זה בזה באופן שמאפשר כתיבת תכניות ב-PHP אשר מתקשרות עם בסיס הנתונים בקלות רבה.

עם הופעת PHP 5.0 הופיעה הרחבה חדשה בשם MySQLi (ראשי תיבות של MySQL Improved). הרחבה זו מאפשרת להשתמש בחלק גדול מהיכולות שהופיעו ב-MYSQL בהלך השנים האחרונות, לרבות עבודה עם Prepared Statements והאפשרות לנהל Transactions.

עד להופעת PHP 5.0 תכניות PHP שהתממשקו עם בסיס הנתונים MySQL עשו שימוש בהרחבה MySQL.

פרט ליכולות החדשות שההרחבה MySQLi מציגה, הבדל בסיסי נוסף בין שתי ההרחבות הוא שההרחבה MySQLi מספקת מימשק שמאפשר לעבוד עימה באופן המקובל בתיכנות מונחה עצמים.

בין שתי ההרחבות (MySQL ו-MySQLi) קיים דימיון רב ומסיבה זו המעבר מעבודה עם MySQL לעבודה עם MySQLi חלק באופן יחסי.

כדי לאפשר את העבודה עם ההרחבה MySQLi יש לעדכן את הקובץ `php.ini` בהתאם. אם מערכת ההפעלה היא windows אז יש לערוך את הקובץ `php.ini` ולהוסיף לתוכו את השורה:

```
extensions=php_mysql.dll
```

## יצירת חיבור עם בסיס הנתונים

כדי שניתן יהיה לבצע פעולות מול בסיס הנתונים שעימו עובדים תחילה יש לייצור את החיבור עימו. כאשר מסיימים את העבודה מול בסיס הנתונים יש לסגור את החיבור שעימו עבדנו.

אובייקט מהמחלקה `mysqli` מייצג חיבור עם בסיס הנתונים. כאשר קוראים ל-`constructor` של `mysqli` ניתן לשלוח אליו את הערכים המתאימים ובדרך זו לקבל אובייקט מסוג `mysqli` שמייצג חיבור עם בסיס הנתונים שלנו.

```
$mysqli = new mysqli("server_name","username","password","db_name");
```

לחילופין, ניתן לייצור אובייקט מהמחלקה `mysqli` באמצעות הפעלת ה-`constructor` של המחלקה וללא שליחה של הערכים שמאפיינים את החיבור הנדרש (שם השרת, שם משתמש, סיסמא ושם בסיס הנתונים) ולאחר מכן לקבוע אותם באובייקט שנוצר באמצעות הפעלת המתודה `.connect`.

```
$mysqli = new mysqli();
```

```
$mysqli -> connect("server_name","username","password","db_name");
```

בהנחה שכבר יש ברשותנו אובייקט מטיפוס `mysqli` אשר מייצג חיבור לבסיס נתונים וברצוננו להחליף את בסיס הנתונים בבסיס נתונים אחר (שגם הוא מנוהל על ידי אותה התקנה של MySQL שעימה עובדים) כל שיש לעשות הוא להפעיל על אובייקט ה-`mysqli` את המתודה `select_db` ולשלוח אליה את שמו של בסיס הנתונים האחר שעימו רוצים לעבוד.

```
$mysqli -> select_db("other_db_name");
```

לאחר שמקבלים אובייקט `mysqli` אשר מייצג חיבור עם בסיס הנתונים שעימו רוצים לעבוד ניתן לבצע באמצעותו את השאלות השונות שבהן אנו מעוניינים.

לאחר שמסיימים לעבוד עם בסיס נתונים יש לשחרר את ה-`resources` ששימשו את אובייקט ה-`mysqli` שעימו עבדנו באמצעות הפעלת המתודה `.close`.

```
$mysqli -> close();
```

## ביצוע שאילתות לבסיס הנתונים

ביצוע שאילתה לבסיס הנתונים מתבצע באמצעות הפעלת המתודה `query` על אובייקט ה-`mysqli` שעומו עובדים. כאשר קוראים להפעלתה ניתן לשלוט באופן שבו התשובה לשאילתה שלנו תוחזר אלינו. אם שולחים לפרמטר `resultmode` של המתודה `query` את הערך `MYSQLI_STORE_RESULT` אז מקבלים את התשובה לשאילתה בבת אחת אל תוך בפר שעומו עובדים וניתן באופן מיידי לקבל את כל אחת מהשורות שהתשובה כוללת. אם שולחים לפרמטר `resultmode` של המתודה `query` את הערך `MYSQLI_USE_RESULT` אז התשובה לשאילתה תגיע בחלקים (על פי הצורך). הגישה לכל אחת מהשורות של התשובה איטית יותר. לעומת זאת, כאשר התשובה כוללת מספר גדול במיוחד של שורות אנו נהנה מביצועים יותר טובים.

הדוגמה הבאה מציגה שימוש בבסיס הנתונים MySQL.

```
<?php
mysqli = new
mysqli("127.0.0.1","iuser","ipassword","store_db");
$query = "SELECT product, id FROM products ORDER BY product";
$result = $mysqli->query($query,MYSQLI_STORE_RESULT);
while(list($name,$id) = $result->fetch_row())
{
    printf("%s,%s<br>",$name,$id);
}
$mysqli->close();
?>
```

הפלט שנקבל הוא:

```
chair,101
table,100
```