

# iranphp articles

استفاده از mysql بخش نخست  
هوتن القاسپور  
.....  
.....

عنوان مقاله :  
نگارنده :  
آدرس پست الکترونیک :  
تاریخ نگارش :

## استفاده از mysql بخش نخست:

در این سلسله مقالات طریقه کار با بانک اطلاعاتی mysql را php بحث می کنم فرض من بر این است که شما اصول بانک های اطلاعاتی و دستور زبان SQL و همچنین برنامه نویسی با php را بلدید یکی از خصوصیات ممتاز php را میتوان توانایی کار با اکثر بانک های اطلاعاتی مثل sql server و یا mysql و بسیاری از دیتابیس های دیگر ذکر کرد php. برای بانک اطلاعاتی mysql توابع خاصی را در نظر گرفته است که در اینجا به معرفی متداول ترین و کاربردی ترین آنها با ذکر مثال و کد نمونه می پردازم برای شرح تعدادی از توابع را معرفی و پس از تمام معرفی توابع در ادامه تکنیک های خاص استفاده از آنها را ذکر خواهیم کرد. در طول معرفی توابع از تعدادی مثال استفاده خواهد شد اما برای اینکه واقعا چیزی یاد بگیرید تنها باید از فکر خودتان استفاده کنید و بنا به کاربرد هایی که برای توابع ذکر میشود برنامه هایی را نوشته و اجرا کنید .

حتما توجه کنید که این لیست فقط مانند یک فهرست عمل می کند و برای آگاهی بیشتر از توابع ذکر شده باید به مستندات php مراجعه کنید . البته در ادامه پر کاربرد ترین و عمومی ترین آن ها را توضیح خواهیم داد. این مطالب را می توانید در مستندات php به زبان اصلی بخوانید هدف از این مطالب آشنایی شما با برخی توابع کاربردی است که در قسمت های آتی تکنیک های استفاده از آن ها را به شما یاد خواهیم داد و در انتها با هم یک پروژه را انجام میدهم. برخی از مطالب و توابعی را هم که در اینجا ذکر نکرده ام را (مثل حلقه foreach) در ادامه و در طی کار به شما خواهیم گفت .

### توابع مورد کاربرد در php 3.4.3

**\*=** توابعی که در این بخش راجع به آن ها صحبت خواهیم کرد

**\*\*=** قسمت هایی که در صورت ترجمه به اصل مطلب صدمه می خورد

**\*mysql\_affected\_rows** - تعداد رکورد های تحت تاثیر قرار گرفته در آخرین پرس و جوی انجام داده را می گیرد

**mysql\_change\_user** - کاربر فعال را در ارتباط جاری با بانک اطلاعاتی را عوض میکند عوض می کند

**mysql\_client\_encoding** - تنظیمات کاراکتری پیش فرض را از ارتباط جاری را باز میگرداند

**\*mysql\_close** - ارتباط جاری با پایگاه داده را قطع میکند

**\*mysql\_connect** - یک ارتباط جدید با پایگاه داده برقرار می کند

**mysql\_create\_db** - یک بانک اطلاعاتی میسازد

**mysql\_data\_seek** - نشانگر (پوینتر) داخلی مجموعه جواب را حرکت می دهد

**mysql\_db\_name** - اطلاعات جواب پرس و جو را می گیرد

**mysql\_db\_query** - پرس و جو را ارسال می کند

**mysql\_drop\_db** - بانک اطلاعاتی را حذف میکند

**mysql\_errno** - شماره خطای ایجاد شده در آخرین عملیات را باز می گرداند

**mysql\_error** - متن خطای ایجاد شده در آخرین عملیات را باز میگرداند

**mysql\_escape\_string** - یک رشته را برای کار با بانک اطلاعاتی تطبیق می کند

**\*mysql\_fetch\_array** - یک مجموعه جواب بازگشتی را در آرایه ای انجمنی یا اندیسی یا هر دو می ریزد

- \*`mysql_fetch_assoc` - یک مجموعه جواب بازگشتی را در آرایه ای انجمنی می ریزد
- `mysql_fetch_field` - اطلاعات یک ستون را از یک مجموعه جواب میگیرد و یک شی را باز میگرداند
- `mysql_fetch_lengths` - طول هر کدام از جواب های خروجی را باز می گرداند
- \*`mysql_fetch_object` - یک مجموعه جواب را در یک شی می ریزد
- \*`mysql_fetch_row` - یک مجموعه جواب را به صورت یک آرایه شمارشی در می آورد
- `mysql_field_flags` - نمایه فیلد معرفی شده در مجموعه جواب را باز می گرداند
- `mysql_field_len` - طول فیلد مشخص شده را باز میگرداند
- `mysql_field_name` - نام فیلد مشخص شده در مجموعه جواب را باز می گرداند
- `mysql_field_seek` - نشانه گر جواب را در فیلد مبدا مشخص شده قرار می دهد
- `mysql_field_table` - نام جدولی را که فیلد در آن قرار دارد را باز می گرداند
- `mysql_field_type` - نوع فیلد موجود در مجموعه جواب را باز می گرداند
- `mysql_free_result` - حافظه را از مجموعه جواب خالی می کند
- `mysql_get_client_info` - اطلاعات خدمات گیرنده پایگاه داده را بدست می آورد
- `mysql_get_host_info` - اطلاعات میزبان پایگاه داده را بدست می آورد
- `mysql_get_proto_info` - پروتکل (قوانین) مورد استفاده در پایگاه داده را میگیرد نسخه
- `mysql_get_server_info` - اطلاعات خدمات دهنده پایگاه داده را بدست می آورد
- `mysql_info` - اطلاعاتی در مورد جدیدترین پرسش و پاسخ را باز می گرداند
- \*`mysql_insert_id` - شناسه آخرین فیلد اضافه شده را باز می گرداند
- \*`mysql_list_dbs` - بانک های اطلاعاتی موجود را نشان می دهد
- \*`mysql_list_fields` - ستون های جدول را نشان می دهد
- `mysql_list_processes` - پردازش های انجام شده را نشان می دهد
- \*`mysql_list_tables` - لیست جدول های یک بانک اطلاعاتی را نشان می دهد
- \*`mysql_num_fields` - تعداد فیلد های یک مجموعه جواب را باز می گرداند
- \*`mysql_num_rows` - تعداد سطرهای یک مجموعه جواب را باز می گرداند

\*`mysql_pconnect` - یک اتصال دائمی با پایگاه داده برقرار می کند

\*`mysql_ping` - یک ارتباط را پینگ می کند و در صورت برقرار نبودن ارتباط آنرا برقرار میسازد

\*`mysql_query` - یک پرس و جو را ارسال میکند

`mysql_real_escape_string` کاراکترهای ویژه را در یک رشته برای استفاده از پایگاه داده بهینه میکند و در این عمل از تنظیمات کاراکتری موجود در پایگاه داده استفاده می کند.

\*`mysql_result` - اطلاعات مجموعه جواب را باز میگرداند

\*`mysql_select_db` - یک بانک اطلاعاتی را انتخاب می کند

`mysql_stat` - وضعیت سیستم جاری را باز می گرداند

`mysql_tablename` - نام جدول فیلد را باز میگرداند

\*\*`mysql_thread_id` - شناسه thread جاری را باز می گرداند

`mysql_unbuffered_query` - یک پرس و جو را بدون واسطه به پایگاه داده ارسال می کند

در نسخه پنجم php که هنوز در مرحله بتا است توابع بسیار قدرتمند دیگری نیز اضافه شده است که معمولاً به صورت `*Mysqli` می باشند. در کل 5 php خیلی به سمت شی گرای رفته است بنابراین توابع مورد استفاده در آن نیز تا حد امکان سعی در اضافه نمودن هرچه بیشتر امکانات شی گرای دارند. پس از پایان این درس توابع نسخه پنجم را نیز بررسی خواهیم کرد که البته بعلا تغییرات ناچیز یادگیری و استفاده از آنها چندان مشکل نخواهد بود. توجه کنید لازم نیست شما تک تک این توابع را حفظ کنید بلکه با دانلود کردن مرجع کامل php از سایت `php.net` می توانید هر زمان که لازم داشتید توضیحات مربوط به آن را بخوانید من توصیه می کنم که فایل `chm` را دانلود کنید زیرا استفاده از آن سریع تر و راحت تر است

\*تابع `mysql_ping` توجه کنید این تابع تنها کاری که می کند بررسی اتصال به موتور `mysql` است و معمولاً در اسکریپت های تست که برای تست کردن آمادگی سرور بکار می رود استفاده می شود. این تابع یک اتصال به `mysql` را `ping` می کند و در صورت موفقیت و بازگشت جواب `true` و در غیر اینصورت `false` را باز می گرداند. در صورت قطع بودن ارتباط سعی میکند آنرا برقرار سازد.

```
mysql_ping()  
Ping a server connection or reconnect if there is no connection (PHP 4 >= 4.3.0)  
bool mysql_ping ( [resource link_identifier ] )  
Example:
```

```
<?php  
$conn=mysql_connect($db_host, $db_user, $db_password);  
$db=mysql_select_db($db_name);  
if(mysql_ping($conn)){  
echo "your database is online",  
} else {  
echo "your database is offline",  
}  
mysql_close($conn);  
?>
```

\*تابع `mysql_connect` که وظیفه برقراری ارتباط با بانک اطلاعاتی را برعهده دارد و معمولاً اولین خط اسکریپت کار با `mysql` را تشکیل میدهد زیرا تا زمانی که این تابع مقدار `true` را بازنگرداند شما اصلاً به موتور `mysql` متصل نیستید تا بتوانید کاری را انجام دهید. توجه کنید پس از پایان اسکریپت یا فراخوانی تابع `mysql_close` این اتصال قطع خواهد شد.

```
mysql_connect()  
Open a connection to a MySQL Server (PHP 3, PHP 4 )  
resource mysql_connect ( [string server [, string username [, string password [, bool new_link [, int client_flags ]]]]] )  
Example:  
  
<?php  
$conn = mysql_connect("localhost","mysql_username","mysql_password") or die(Could not connect to  
database daemon:.mysql_error());  
$db = mysql_select_db("your database name",$conn) or die(Could not select database:.mysql_error());  
$query = mysql_query ("select * from tablename");  
mysql_close ($conn);  
?>
```

\*تابع `mysql_pconnect` که وظیفه برقراری ارتباط دائمی با بانک اطلاعاتی را داراست و بعد از پایان اسکریپت و یا فراخوانی تابع `mysql_close` ارتباط قطع نمی‌شود. کاربرد این تابع در مواقعی است که شما می‌خواهید در زمان صرفه جویی کنید زیرا این تابع باعث می‌شود در فراخوانی‌های متعدد از همین ارتباط برای تبادل داده استفاده شود و ارتباط جدیدی برقرار نگردد. معمولاً این تابع زمانی استفاده می‌شود که چندین صفحه اتصال هابی با آرگومان‌های مشابه را نیاز داشته باشند.

توجه کنید برای استفاده از این تابع شما حتماً باید `php` را به عنوان ماژول نصب کنید نه `CGI` این یکی از راه‌های سریع کردن و بهینه کردن کد است اما توجه کنید که این امر می‌تواند باعث مشکل نیز بشود زیرا سبب ایجاد اتصالات بی‌هوده و بی‌کار که اصطلاحاً آنها را `idle` می‌نامند می‌گردد توجه کنید این اتصالات به هر حال منابع سرور شما را مصرف خواهند کرد بنابراین کاربرد غلط آن می‌تواند موجب مشکلاتی شود.

```
mysql_pconnect()  
Open a persistent connection to a MySQL server (PHP 3, PHP 4 )  
resource mysql_pconnect ( [string server [, string username [, string password [, int client_flags]]]] )  
Example:  
  
<?php  
$conn=mysql_pconnect("localhost","mysql_username","mysql_password") or die(Could not connect to  
database daemon:.mysql_error());  
$db = mysql_select_db ("your database name",$conn) or die(Could not select  
database:.mysql_error());  
$Query = mysql_query ("select * from tablename");  
mysql_close ($conn);  
?>  
P = persistent = دائمی
```

\*تابع `mysql_close` که باعث بسته شدن اتصال غیر دائمی به پایگاه داده می‌شود.

```
mysql_close()  
Close MySQL connection (PHP 3, PHP 4 )  
bool mysql_close ( [resource link_identifier ] )  
Example:  
  
<?php  
$conn = mysql_connect ("localhost","mysql_username","mysql_password") or die(Could not connect to  
database daemon:.mysql_error());  
$db = mysql_select_db ("your database name",$conn) or die(Could not select  
database:.mysql_error());
```

```
$Query = mysql_query("select * from tablename");  
mysql_close ($conn);  
?>
```

\*تابع mysql\_select\_db که بوسیله این تابع از اتصال جاری یک پایگاه داده را انتخاب می کنیم تا دستورات بعدی ما رو آن پایگاه داده اعمال شود .

```
mysql_select_db()  
Select a MySQL database (PHP 3, PHP 4 )  
bool mysql_select_db ( string database_name [, resource link_identifier ] )  
Example:
```

```
<?php  
$conn = mysql_connect("localhost", "mysql_user", "mysql_password") or die(Could not  
connect to database daemon: .mysql_error());  
$db = mysql_select_db("your database name", $conn) or die(Could not select  
database: .mysql_error());  
$query = ("select * from tablename");  
$result = mysql_query ($query);  
mysql_close ($conn);  
?>
```

تابع mysql\_query که جزو مهمترین توابع است زیرا این تابع دستورات sql شما را اجرا می کند این تابع معمولاً با توابع mysql\_fetch\_\* همراه حلقه ها کار اصلی را در رابطه با پایگاه داده انجام میدهند

```
mysql_query()  
Send a MySQL query (PHP 3, PHP 4 )  
resource mysql_query ( string query [, resource link_identifier] )  
Example:
```

```
<?php  
$conn = mysql_connect("localhost", "mysql_user", "mysql_password") or die(Could not connect to  
database daemon: .mysql_error());  
$db = mysql_select_db("your database name", $conn) or die(Could not select database: .mysql_error());  
$query = ("insert into tablename values(1,2)");  
$result = mysql_query($query);  
mysql_close($conn);  
?>
```

تابع mysql\_result که برای یک رکورد از پردازش انجام شده توسط mysql\_query() کاربرد دارد همان طور که قبلاً گفتیم برای خواندن تمامی رکورد ها باید از توابع mysql\_fetch\_\* به همراه حلقه ها استفاده کرد. آرگومان اول مجموعه جواب اجرا شده بوسیله تابع mysql\_query و آرگومان دوم شماره رکوردی که می خواهید بخوانید و آرگومان سوم فیلدی است که می خواهید بخوانید که در صورت معرفی نکردن آرگومان سوم اولین فیلد خوانده خواهد شد .

```
mysql_result()  
Get result data (PHP 3, PHP 4 )  
mixed mysql_result ( resource result, int row [, mixed field ] )  
Example:
```

```
<?php  
$conn=mysql_connect("localhost", "mysql_user", "mysql_password");  
$db=mysql_select_db("test");  
$query="select * from test";
```

```
$result=mysql_query($query);
if($result) /*for testing result answer*/{
    $record1=mysql_result($result,0,'name');
    echo "Rcord 1 is:",$record1;
    echo "<p>";
    $record2=mysql_result($result,1,'name');
    echo "Rcord 2 is:",$record2;
}
?>
```

تابع `mysql_affected_rows` که تعداد سطرهایی را که توسط آخرین دستور `delete` یا `replace` یا `insert` یا `update` تحت تاثیر قرار گرفته اند را باز می گرداند

```
mysql_affected_rows()
Get number of affected rows in previous MySQL operation (PHP 3, PHP 4 )
int mysql_affected_rows ( [resource link_identifier ] )
Example:
```

```
<?php
/* connect to database */
mysql_pconnect("localhost", "mysql_user", "mysql_password") or
die("Could not connect: " . mysql_error());
mysql_select_db("mydb");
/* this should return the correct numbers of deleted records */
mysql_query("DELETE FROM mytable WHERE id < 10");
printf ("Records deleted: %d\n", mysql_affected_rows());
/* without a where clause in a delete statement, it should return 0 */
mysql_query("DELETE FROM mytable");
printf ("Records deleted: %d\n", mysql_affected_rows());
?>
```

تابع `mysql_num_rows` این تابع عملکردی مشابه تابع `mysql_affected_rows` دارد با این تفاوت که این تابع تعداد رکورد های مجموعه جواب بازگردانده شده از یک جستجوی `select` را بدست می آورد. از این تابع می توان برای بدست آوردن رکوردهای یک جدول استفاده کرد .

```
mysql_num_rows()
Get number of rows in result (PHP 3, PHP 4 )
int mysql_num_rows ( resource result )
Example:
```

```
<?php
$conn = mysql_connect("localhost", "mysql_user", "mysql_password");
mysql_select_db("database", $conn);
$result = mysql_query("SELECT * FROM table", $link);
$num_rows = mysql_num_rows($result);
echo "$num_rows Rows\n";
?>
```

تابع `mysql_free_result` که وظیفه آزاد سازی حافظه مصرف شده بوسیله نتایج بدست آمده از `query` را داراست. این تابع در اسکریپت های طولانی که اعمال زیادی را انجام می دهند و حافظه زیادی را مصرف می کنند کاربرد دارد. ولی در اسکریپت های عادی به آن نیازی نیست زیرا در php پس از پایان اسکریپت به طور خورکار حافظه اختصاص داده شده به `query` آزاد می شود .

```
mysql_free_result()
Free result memory (PHP 3, PHP 4 )
bool mysql_free_result ( resource result )
Example:
```



```
<?php
$conn = mysql_connect("localhost", "mysql username", "mysql _password") or die(Could not connect to
database daemon: .mysql_error());
$db = mysql_select_db("your database name", $conn) or die(Could not select database: .mysql_error());
$query = mysql_query ("select * from tablename");
mysql_free_result ($query);
mysql_close ($conn);
?>
```

تابع `mysql_fetch_array` همان طور قبلا بارها گفتم برای خواندن تمامی سطرهای یک جدول از توابع `mysql_fetch_*` به همراه حلقه ها استفاده میشود این توابع وظیفه دارند حاصل یک `query` را در یک آرایه یا مجموعه از اشیا بریزند تا بتوان به کمک حلقه ها تک تک آن ها را خواند. تابع `mysql_fetch_array` مجموعه جواب یک `query` را در یک آرایه می ریزد که بوسیله آرگومان دوم می توان طریقه ذخیره و دسترسی رادر آرایه(نوع آرایه (معین کرد. در اینجا لغات کلیدی را که در مثال های آتی استفاده می کنم را توضیح میدهم آنها به عنوان ثابت در `php` تعریف شده اند.

ثابت

توضیحات

```
MYSQL_ASSOC
Columns are returned into the array having the fieldname as the array index.
```

بوسیله این ثابت می تونید از نام فیلدهای خود برای دسترسی به اطلاعات آن ها استفاده کنید

```
MYSQL_NUM
Columns are returned into the array having a numerical index to the fields. This index starts with
0, the first field in the result.
```

بوسیله این ثابت می تونید از شماره اندیس آرایه برای دسترسی به اطلاعات فیلدها استفاده کنید

```
MYSQL_BOTH
Columns are returned into the array having both a numerical index and the fieldname as
the array index.
```

بوسیله این ثابت می تونید از نام فیلد با اندیس آن در آرایه برای دسترسی به اطلاعات آن استفاده کنید

```
mysql_fetch_array()
Fetch a result row as an associative array, a numeric array, or both. (PHP 3, PHP 4 )
array mysql_fetch_array ( resource result [, int result_type ] )
1)Example for MYSQL_NUM:

<?php
mysql_connect("localhost", "mysql_user", "mysql_password") or die("Could not connect: " .
mysql_error());
mysql_select_db("mydb");
$result = mysql_query("SELECT id, name FROM mytable");
while ($row = mysql_fetch_array($result, MYSQL_NUM)) {
    printf ("ID: %s Name: %s", $row[0], $row[1]);
}
mysql_free_result($result); ?>

2)Example for MYSQL_ASSOC:
```



```
<?php
mysql_connect("localhost", "mysql_user", "mysql_password") or die("Could not connect: " .
mysql_error());
mysql_select_db("mydb");
$result = mysql_query("SELECT id, name FROM mytable");
while ($row = mysql_fetch_array($result, MYSQL_ASSOC)) {
    printf ("ID: %s Name: %s", $row["id"], $row["name"]);
}
mysql_free_result($result);
?>
```

3) Example for MYSQL\_BOTH:

```
<?php
mysql_connect("localhost", "mysql_user", "mysql_password") or die("Could not connect: " .
mysql_error());
mysql_select_db("mydb");
$result = mysql_query("SELECT id, name FROM mytable");
while ($row = mysql_fetch_array($result, MYSQL_BOTH)) {
    printf ("ID: %s Name: %s", $row[0], $row["name"]);
}
mysql_free_result($result);
?>
```

توجه کنید که در صورت معرفی نکردن ثابت به طور پیش فرض از MYSQL\_NUM استفاده میشود .  
در ادامه تنها دو تابع از سری توابع \*mysql\_fetch را معرفی میکنم زیرا تابع mysql\_fetch\_array تقریباً تمام نیازهای شما را برطرف می کند اما در قسمت مباحث پیشرفته توابع mysql\_fetch\_lengths و mysql\_fetch\_field را نیز معرفی خواهیم کرد . توجه کنید شما هم اکنون هم میتوانید بدون مشکل یک برنامه کامل با php & mysql بنویسید .

تابع mysql\_fetch\_row که عملکردی کاملاً مشابه mysql\_fetch\_array به همراه ثابت MYSQL\_NUM دارد .  
تابع mysql\_fetch\_assoc که عملکردی کاملاً مشابه mysql\_fetch\_array به همراه ثابت MYSQL\_ASSOC دارد .  
تابع mysql\_fetch\_object :

این تابع رکورد های بازگردانده شده بوسیله یک دستور select را به صورت مجموعه ای از اشیاء باز می گرداند که در پایان پس از اتمام رکورد ها بوسیله بازگرداندن false اتمام رکورد ها را اعلام می کند. در صورتی که به برنامه نویسی شی گرا علاقه دارید این تابع را دوست خواهید داشت .

```
mysql_fetch_object()
Fetch a result row as an object (PHP 3, PHP 4 )
object mysql_fetch_object ( resource result )
```

```
<?php
mysql_connect("hostname", "user", "password");
mysql_select_db("mydb");
$result = mysql_query("select * from mytable");
while ($row = mysql_fetch_object($result)) {
    echo $row->user_id;
    echo $row->fullname;
}
?>
```

و آخرین تابع که یک تابع کاربردی است به نام mysql\_insert\_id

این تابع باعث بازگشت ID آخرین فیلد ایجاد شده بوسیله دستور insert می گردد .

\*در جدول هایی که به طور ذاتی نمی توان برای آنها یک کلید اصلی (Primary key) تعریف کرد معمول است که یک فیلد به نام ID که دارای خاصیت AUTO\_INCREMENT است تعریف می کنند که یک فیلد عددی است و برای فهمیدن تعداد رکوردهای جدول نیز کاربرد دارد .

```
mysql_insert_id()
```

Get the ID generated from the previous INSERT operation (PHP 3, PHP 4 )

```
int mysql_insert_id ( [resource link_identifier ] )
```

Example:

```
<?php
mysql_connect("localhost", "mysql_user", "mysql_password") or
    die("Could not connect: " . mysql_error());
mysql_select_db("mydb");
mysql_query("INSERT INTO mytable (name) values ('houtan')");
printf ("Last inserted record has id %d\n", mysql_insert_id());
?>
```

خب توابعی که فکر می‌کردم برای شروع کار با mysql لازم میشود را گفتم اما همان طور که قبلا گفتم توابع زیادی در php برای کار با mysql تعبیه شده است در قسمت مباحث پیشرفته توابعی را به شما خواهیم گفت که توانایی‌های جالب و بعضا قدرتمندی دارند اما شما هم اکنون هم می‌توانید یک برنامه کامل و حرفه‌ای را با استفاده از این توابع بنویسید که در ادامه یک برنامه برای تمرین با هم خواهیم نوشت اما به یاد داشته باشید تا زمانی که خودتان برنامه نویسی و به مشکل برخوردید چیز زیادی در عمل یاد نخواهید گرفت در ضمن آخرین نکته این‌که اگر دارید تمرین میکنید و هیچ مشکلی در برنامه‌هایی که میخواهید بنویسید ندارید و می‌بینید همه چیز را بلدید فکر کنم باید در نحوه تمرین کردن خود تجدید نظر کنید.