PHP - MySQL Fonksiyonları

- mysql_affected_rows → Bir önceki MySQL işleminden etkilenen satırların sayısını getirir.

Kullanımı

```
int mysql_affected_rows(int [link_identifier] );
```

mysql_affected_rows(), belirlenen link tanımlayıcı ile ilişkili olarak, sunucudaki son INSERT, UPDATE veya DELETE sorgularının etkilediği satırların sayısını döndürür. Eğer link tanımlayıcı belirlenmemişse, son açılan link kabul edilir.

Son sorgu, WHERE ' i olmayan bir DELETE sorgusu ise, tüm kayıtlar tablodan silinecektir. Fakat bu fonksiyon sıfır değeri döndürecektir.

Bu komut SELECT işlemleri için etkili değildir. Sadece kayıtları değiştiren işlemler üzerinde etkilidir. Bir SELECT' ten dönen satırların sayısını döndürmek için mysql_num_rows() kullanılır.

- mysql_affected_rows → Bir önceki MySQL işleminden etkilenen satırların sayısını getirir.

Kullanımı

```
int mysql_affected_rows(int [link_identifier] );
```

mysql_affected_rows(), belirlenen link tanımlayıcı ile ilişkili olarak, sunucudaki son INSERT, UPDATE veya DELETE sorgularının etkilediği satırların sayısını döndürür. Eğer link tanımlayıcı belirlenmemişse, son açılan link kabul edilir.

Son sorgu, WHERE ' i olmayan bir DELETE sorgusu ise, tüm kayıtlar tablodan silinecektir. Fakat bu fonksiyon sıfır değeri döndürecektir.

Bu komut SELECT işlemleri için etkili değildir. Sadece kayıtları değiştiren işlemler üzerinde etkilidir. Bir SELECT' ten dönen satırların sayısını döndürmek için mysql_num_rows() kullanılır.

- **mysql_change_user → Aktif bağlantı üzerindeki oturum açmış kullanıcıyı değiştirir.**

Kullanımı

```
int mysql_change_user(string user, string password, string
[database] , int [link_identifier] );
```

mysql_change_user(), aktif bağlantı üzerindeki oturum açmış kullanıcıyı değiştirir. Bir veritabanı belirlenmişse, bu, kullanıcı değiştikten sonra default veya yürürlükteki veritabanıdır. Eğer yeni user/password kombinasyonu yetkilendirilmede başarısız olursa, yürürlükteki bağlı kullanıcı aktif olarak kalır.

Not: Bu fonksiyon PHP 3.0.13 ile birlikte ortaya çıktı ve MySQL 3.23.3 veya daha yüksek versiyonları gerektirir.

- mysql_close → MySQL bağlantısını kapatır.

Kullanımı

`int mysql_close(int [link_identifier] );` → Başarı durumunda true, hata durumunda false döner.

mysql_close(), belirlenen link tanımlayıcı ile ilişkide bulunan bir MySQL veritabanına olan linki kapatır. Eğer link tanımlayıcı belirlenmemişse, son açılan link kabul edilir.

Not: Sürekli olmayan açık linkler, betimin çalışması bittiğinde otomatik olarak kapandığı için, bu genellikle gerekmez.

mysql_close(), mysql_pconnect() tarafından oluşturulan sürekli linkleri kapatmayacaktır.

Örnek: MySQL kapatma örneği

```
 2 <?php

 3      $link = mysql_connect ("kraemer", "marliesle", "secret")
{

 4          or die ("Could not connect");

 5     }

 6     print ("Connected successfully");

 7     mysql_close ($link);

 8 ?>
```

Bkz: mysql_connect() ve mysql_pconnect().

- **mysql_connect → Bir MySQL sunucuya bir bağlantı açar.**

Kullanımı

```
int mysql_connect(string [hostname [:port] [:/path/to/socket] ] ,
string [username] , string [password] );
```

Başarı durumunda pozitif bir MySQL link tanımlayıcı, başarısızlık durumunda bir hata mesajı döndürür.

mysql_connect(), bir MySQL sunucuya bir bağlantı kurar. Değişkenlerin tümü seçimlidir ve bunlar kaybolursa, varsayılanlar kabul edilir. ('localhost', sunucu process' e sahip olan kullanıcının kullanıcı ismi, boş password).

Hostname string' i bir port numarası da içerebilir. ör. "hostname:port" veya bir sokete bir path. ör. localhost için ":/path/to/socket" .

Not: ":port" için destek PHP 3.0B4'de eklendi.

":/path/to/socket" için destek PHP 3.0.10 ile eklendi.

Başarısızlık durumunda, fonksiyon ismine ön bekleme '@' ile hata mesajı durdurulabilir.

İkinci bir çağrı, aynı değişkenlerle mysql connect()' e yapılır. Hiçbir yeni link

kurulmayacaktır.Fakat bunun yerine halen açık olan linkin link tanımlayıcısı geri dönecektir.

Betimin yürütmesi sonlanır sonlanmaz, mysql_close() çağrılarak dışsal bir şekilde kapatılmadıkça, sunucuya olan link kapatılacaktır.

Örnek: MySQL bağlanma örneği

```
<?php
    $link = mysql_connect ("kraemer", "marliesle", "secret")
{
        or die ("Could not connect");
    }
    print ("Connected successfully");
    mysql_close ($link);
?>
```

Bkz. mysql_pconnect() ve mysql_close().

- **mysql_create_db → Bir MySQL veritabanı yaratır.**

Kullanımı

```
int mysql_create_db(string database name, int [link_identifier]
);
```

mysql_create_db(), belirlenen link tanımlayıcı ile ilişkide bulunarak, sunucuda yeni bir veritabanı yaratma girişiminde bulunur.

Örnek: MySQL veritabanı yaratma örneği

```
2 <?php
3    $link = mysql_pconnect ("kron", "jutta", "geheim") {
4       or die ("Could not connect");
5    }
6    if (mysql_create_db ("my_db")) {
7       print ("Database created successfully\n");
8    } else {
9       printf ("Error creating database: %s\n", mysql_error ());
10    }
11 ?>
```

- mysql_data_seek → İçsel sonuç pointer' ını taşır.

Kullanımı

```
int mysql_data_seek(int result_identifier, int row_number);
```

Başarı durumunda true, başarısızlık durumunda false döner.

mysql_data_seek(), belirlenen satır numarasını işaret eden belirli sonuç tanımlayıcı ile ilişkili olarak, MySQL sonucunun içsel satır pointer' ını taşır. mysql_fetch_row()' a bir sonraki çağrım bu satıra dönecektir.

Row_number 0'dan başlar.

Örnek: MySQL veri arama örneği

```
2 <?php
3    $link = mysql_pconnect ("kron", "jutta", "geheim") {
4       or die ("Could not connect");
5    }
7    mysql_select_db ("samp_db") {
8       or die ("Could not select database");
9    }
11    $query = "SELECT last_name, first_name FROM friends";
12    $result = mysql_query ($query) {
13       or die ("Query failed");
14    }
16    # fetch rows in reverse order
18    for ($i = mysql_num_rows ($result) - 1; $i >=0; $i--) {
19       if (!mysql_data_seek ($result, $i)) {
20          printf ("Cannot seek to row %d\n", $i);
21          continue;
22       }
24       if(!($row = mysql_fetch_object ($result)))
25          continue;
26
27       printf ("%s %s<BR>\n", $row->last_name, $row->first_name);
28    }
30    mysql_free_result ($result);
31 ?>
```

- mysql_db_query → Bir MySQL sorgusunu MySQL' e gönderir.

Kullanımı

```
int mysql_db_query(string database, string query, int
[link_identifier] );
```

Pozitif bir MySQL sonuç tanımlayıcı, veya hata durumunda false sorgu sonucu ile döner.

mysql_db_query(), bir veritabanı seçer ve bunun üzerinde bir sorguyu çalıştırır.Eğer seçimli link tanımlayıcı belirlenmemişse, fonksiyon MySQL sunucuya açık bir link bulmayı deneyecektir ve böyle bir link bulamazsa, değişkensiz olarak mysql_connect() çağrılmış gibi, bir tane yaratmayı deneyecektir.

Bkz. mysql_connect().

Aşağı doğru uygunluk için mysql() kullanılabilir.

- **mysql_drop_db → Bir MySQL veritabanını silme**

Kullanımı

```
int mysql_drop_db(string database_name, int [link_identifier] );
```

Başarı durumunda true, başarısızlık durumunda false döner.

mysql_drop_db(), belirlenen link tanımlayıcı ile ilişkili olarak, tüm veritabanını sunucudan silmeye çalışır.

Bkz. mysql_create_db(). Aşağı doğru uygunluk için mysql_dropdb() kullanılabilir.

- **mysql_errno → Önceki MySQL işlemindeki hata mesajının numarasını döndürür.**

Kullanımı

```
int mysql_errno(int [link_identifier] );
```

MySQL veritabanından dönen hatalar artık uyarı yayınlamazlar. Bunun yerine, bu fonksiyonlar hata numarasını getirmede kullanılır.

```
2 <?php
3 mysql_connect("marliesle");
4 echo mysql_errno().": ".mysql_error()."<BR>";
5 mysql_select_db("nonexistentdb");
6 echo mysql_errno().": ".mysql_error()."<BR>";
7 $conn = mysql_query("SELECT * FROM nonexistenttable");
8 echo mysql_errno().": ".mysql_error()."<BR>";
9 ?>
```

Bkz. mysql_error()

- **mysql_error → Önceki MySQL işleminin hata mesajının metnini geri döndürür.**

Kullanımı

```
string mysql_error(int [link_identifier] );
```

MySQL veritabanından dönen hatalar artık uyarı yayınlamazlar. Bunun yerine bu fonksiyonlar hata string' ini getirmede kullanılırlar.

```
2 <?php
3 mysql_connect("marliesle");
4 echo mysql_errno().": ".mysql_error()."<BR>";
5 mysql_select_db("nonexistentdb");
6 echo mysql_errno().": ".mysql_error()."<BR>";
7 $conn = mysql_query("SELECT * FROM nonexistenttable");
8 echo mysql_errno().": ".mysql_error()."<BR>";
9 ?>
```

Bkz. mysql_errno()

- mysql_fetch_array → Bir associative dizi gibi bir sonuç satırını getirir.

Kullanımı

```
array mysql_fetch_array(int result, int [result_type] );
```

Getirilen satıra uyan bir dizi veya daha fazla satır yoksa false geri döner.

mysql_fetch_array(), mysql_fetch_row()' un genişletilmiş bir versiyonudur. Sonuç dizinin sayısal göstergelerinde saklanan veriye ek olarak, saha isimlerini anahtar olarak kullanarak, ilişkili göstergelerde veriyi de saklar.

Eğer sonucun bir veya daha fazla sütunu aynı saha isimlerine sahipse, son sütun önceliği alır. Aynı ismin diğer sütunlarına erişmek için, sütunun sayılsal indeks' i mevcut olmalı veya sütun için bir alias tanımlanmalıdır.

```
select t1.f1 as foo t2.f1 as bar from t1, t2
```

mysql_fetch_array() kullanmak, mysql_fetch_row() kullanmaktan önemli düzeyde yavaş değildir. Hatta önemli bir ek değer sunar.

mysql_fetch_array()' deki seçimli ikinci değişken olan result_type bir sabittir ve şu değerleri alabilir: MYSQL_ASSOC, MYSQL_NUM, ve MYSQL_BOTH. (Bu özellik PHP 3.0.7 ile eklenmiştir)

Daha ayrıntılı bilgi için bkz. mysql_fetch_row().

Örnek: mysql fetch array

```
2 <?php
3 mysql_connect($host,$user,$password);
4 $result = mysql_db_query("database","select * from table");
5 while($row = mysql_fetch_array($result)) {
6     echo $row["user_id"];
7     echo $row["fullname"];
8 }
9 mysql_free_result($result);
10 ?>
```

- mysql_fetch_field → Bir sonuçtan sütun bilgisini getirir ve bir nesne şeklinde geri döndürür.

Kullanımı

```
object mysql_fetch_field(int result, int [field_offset] );
```

Saha bilgisini içeren bir nesne geri döner.

mysql_fetch_field(), belirli bir sorgu sonucundaki sahalar hakkında bilgi edinmek için kullanılabilir. Saha ofset değeri belirlenmemişse, mysql_fetch_field() tarafından henüz getirilmemiş olan bir sonraki saha getirilir.

Nesnenin özellikleri şunlardır:

name - sütun ismi

table - sütunun ait olduğu tablonun ismi

max_length - sütunun maksimum uzunluğu

not_null - eğer sütun null olamazsa bu değer 1' dir

primary_key - eğer sütun bir primary key ise bu değer 1' dir.

unique_key - eğer sütun bir unique key ise bu değer 1' dir

multiple_key - eğer sütun bir non-unique key ise bu değer 1' dir

numeric - eğer sütun sayısalsa bu değer 1' dir

blob - eğer sütun bir BLOB ise bu değer 1' dir

type - sütunun tipi

unsigned - eğer sütun işaretsizse bu değer 1' dir

zerofill - eğer sütun zero-filled ise bu değer 1' dir

Bkz. mysql_field_seek()

- mysql_fetch_lengths → Bir sonuçtaki her çıktının uzunluğunu getirir.

Kullanımı

```
array mysql_fetch_lengths(int result);
```

mysql_fetch_row()' in getirdiği son satırdaki her sahanın uzunluğunu içeren bir dizi döndürür. Hata durumunda false döner.

mysql_fetch_lengths(), bir dizideki mysql_fetch_row(), mysql_fetch_array(), ve mysql_fetch_object() tarafından döndürülen son satırdaki her sonuç sütunun uzunluklarını saklar. Ve ofset 0' dan başlar.

Bkz. mysql_fetch_row().

- **mysql_fetch_object → Nesne formunda bir sonuç satırı getirir.**

Kullanımı

```
object mysql_fetch_object(int result, int [result_typ] );
```

Getirilen satırla ilişkili olan özellikleri ile bir nesne veya daha fazla satır yoksa false döndürür.

mysql_fetch_object(), mysql_fetch_array()' e çok benzer. Tek farkı, bir dizi yerine tek bir nesnenin geri dönmesidir. Dolaylı olarak, bu, veriye onların ofsetleri (sayılar yasal olmayan özellik isimleridir) ile değil, sadece saha isimleriyle erişilebileceği anlamına gelir.

Seçimli değişken olan result_typ bir sabittir ve şu değerleri alabilir: MYSQL_ASSOC, MYSQL_NUM, ve MYSQL_BOTH.

Fonksiyonun hızı, mysql_fetch_array() ile aynıdır ve hemen hemen mysql_fetch_row() kadar hızlıdır. (aradaki fark önemsiz derecede azdır).

Örnek: mysql fetch object

```
2 <?php
3 mysql_connect($host,$user,$password);
4 $result = mysql_db_query("database","select * from table");
5 while($row = mysql_fetch_object($result)) {
6     echo $row->user_id;
7     echo $row->fullname;
8 }
9 mysql_free_result($result);
10 ?>
```

Bkz. mysql_fetch_array() ve mysql_fetch_row().

- mysql_fetch_row → Bir sonuç satırını, bir enumerated dizi formunda getirir.

Kullanımı

```
array mysql_fetch_row(int result);
```

Getirilen satırla ilişkili bir dizi veya daha fazla satır yoksa false geri döner.

mysql_fetch_row(), belirlenen sonuç tanımlayıcıya bağlı olarak, sonuçtan bir satır veri getirir. Satır, bir dizi formunda döner. Her sonuç sütunu, ofset 0' dan başlayacak şekilde, bir dizi ofsette tutulur.

mysql_fetch_row()' e yapılan sonraki çağrı, sonuç kümesindeki sonraki satırı veya daha fazla satır yoksa false değerinin döndürecektir.

Bkz. mysql_fetch_array(), mysql_fetch_object(), mysql_data_seek(), mysql_fetch_lengths(), ve mysql_result().

- mysql_field_name → Bir sonuçtaki belirlenen bir sahanın ismini getirir.

Kullanımı

```
string mysql_field_name(int result, int field_index);
```

mysql_field_name() belirlenen sahanın ismini döndürür. Fonksiyonun değişkenleri, sonuç tanımlayıcı ve saha indeksidir, ör. mysql_field_name($result,2);

Sonuç tanımlayıcıya bağlı olarak, sonuçtaki ikinci saha ismi dönecektir.

Aşağı doğru uygunluk için mysql_fieldname() de kullanılabilir.

- mysql_field_seek → Belirli bir saha ofsetini sonuç pointer' a atar.

Kullanımı

```
int mysql_field_seek(int result, int field_offset);
```

Belirli bir saha ofsetini araştırır. Eğer mysql_fetch_field()' a bir sonraki çağrı bir saha ofseti içermeyecekse, bu saha geri döner.

Bkz. mysql_fetch_field().

- **mysql_field_table → Belirtilen sahayı içinde bulunduran tablonun ismini getirir.**

Kullanımı

```
string mysql_field_table(int result, int field_offset);
```

Saha için tablo ismini getirir. Aşağı doğru uygunluk için mysql_fieldtable() da kullanılabilir.

- **mysql_field_type → Bir sonuçtaki belirtilen bir sahanın tipini getirir.**

Kullanımı

```
string mysql_field_type(int result, int field_offset);
```

mysql_field_type(), mysql_field_name() fonksiyonuna benzer. Değişkenler aynıdır. Fakat saha tipi geri döner. Bu, "int", "real", "string", "blob", veya diğerlerinden biri olacaktır.

Örnek: mysql field types

```
2 <?php
3 mysql_connect("localhost:3306");
4 mysql_select_db("wisconsin");
5 $result = mysql_query("SELECT * FROM onek");
6 $fields = mysql_num_fields($result);
7 $rows   = mysql_num_rows($result);
8 $i = 0;
9 $table = mysql_field_table($result, $i);
10 echo "Your '".$table."' table has ".$fields." fields and ".$rows." records <BR>";
11 echo "The table has the following fields <BR>";
12 while ($i < $fields) {
13     $type  = mysql_field_type  ($result, $i);
14     $name  = mysql_field_name  ($result, $i);
15     $len   = mysql_field_len   ($result, $i);
16     $flags = mysql_field_flags ($result, $i);
17     echo $type." ".$name." ".$len." ".$flags."<BR>";
18     $i++;
19 }
20 mysql_close();
21 ?>
```

Aşağı doğru uygunluk için mysql_fieldtype() da kullanılabilir.

- mysql_field_flags → Bir sonuçtaki belirtilen saha ile ilişkili bayrakları getirir.

Kullanımı

```
string mysql_field_flags(int result, int field_offset);
```

mysql_field_flags(), belirtilen sahanın, saha bayraklarını döndürür. Bayraklar tek bir word şeklinde raporlanır ve her bayrak bir boşlukla birbirinden ayrılmamıştır. Bu nedenle, explode() kullanılarak dönüş değeri bölünebilir.

Eğer mevcut MySQL versiyonu bunları destekleyecek kadar yeterliyse, şu bayraklar raporlanır: "not_null", "primary_key", "unique_key", "multiple_key", "blob", "unsigned", "zerofill", "binary", "enum", "auto_increment", "timestamp".

Aşağı doğru uygunluk için mysql_fieldflags() de kullanılabilir.

- **mysql_field_len → Belirtilen sahanın uzunluğunu döndürür.**

Kullanımı

```
int mysql_field_len(int result, int field_offset);
```

mysql_field_len() belirtilen sahanın uzunluğunu döndürür. Aşağı doğru uygunluk için mysql_fieldlen() de kullanılabilir.

- **mysql_free_result → Sonuç belleği boşaltır.**

Kullanımı

```
int mysql_free_result(int result);
```

Betimin çalışmasından dolayı bellek çok dolduysa, sadece mysql_free_result()' ın çağırılması yeterlidir. Belirtilen sonuç tanımlayıcı için ilgili tüm sonuç belleği otomatik olarak boşaltılacaktır.

Aşağı doğru uygunluk için mysql_freeresult() da kullanılabilir.

- mysql_insert_id → Bir önceki INSERT işleminden oluşan id' yi getirir.

Kullanımı

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

mysql_insert_id(), bir AUTO_INCREMENTED sahası için üretilen ID' yi döndürür. Verilen link_identifier kullanan son INSERT sorgusu tarafından getirilen auto-generated ID' i döndürecektir. Eğer link_identifier belirtilmemişse, son açık link kabul edilir.

- mysql_list_fields → MySQL sonuç sahalarını listeler

Kullanımı

```
int mysql_list_fields(string database_name, string table_name,
int [link_identifier] );
```

mysql_list_fields(), verilen bir tablename hakkında bilgi getirir. Değişkenler, veritabanı ismi ve tablo ismidir. mysql_field_flags(), mysql_field_len(), mysql_field_name(), ve mysql_field_type() tarafından kullanılabilecek bir sonuç pointer' ı döndürür.

Bir sonuç tanımlayıcı, pozitif bir integer' dır. Bir hata oluştuğunda, fonksiyon -1 döndürür. Hatayı tanımlayan bir string Sphperrmsg' a konacaktır ve fonksiyon @mysql() olarak çağırılmazsa bu hata string' i aynı zamanda yazdırılacaktır.

Aşağı doğru uygunluk için mysql_listfields() da kullanılabilir.

- **mysql_list_dbs → MySQL sunucuda uygun olan veritabanlarını listeler**

Kullanımı

```
int mysql_list_dbs(int [link_identifier] );
```

mysql_list_dbs(), o anki mysql daemon' da uygun olan veritabanlarını içeren bir sonuç pointer' ı döndürecektir. Bu sonuç pointer' ını dolaşmak için mysql_tablename() fonksiyonu kullanılır.

Aşağı doğru uygunluk için mysql_listdbs() de kullanılabilir.

- **mysql_list_tables → Bir MySQL veritabanındaki tabloları listeler**

Kullanımı

```
int mysql_list_tables(string database, int [link_identifier] );
```

mysql_list_tables(), bir veritabanı ismi alır ve mysql_db_query() fonksiyonundakine çok benzer olarak bir sonuç pointer' ı döndürür. mysql_tablename() fonksiyonu, sonuç pointer' daki gerçek tablo isimlerini açmak için kullanılmalıdır.

Aşağı doğru uygunluk için mysql_listtables() da kullanılabilir.

- **mysql_num_fields → Sonuçtaki sahaların sayısını verir.**

Kullanımı

```
int mysql_num_fields(int result);
```

mysql_num_fields(), bir sonuç kümesindeki sahaların sayısını döndürür.

Bkz. mysql_db_query(), mysql_query(), mysql_fetch_field(), mysql_num_rows().

Aşağı doğru uygunluk için mysql_numfields() da kullanılabilir.

- **mysql_num_rows → Sonuçtaki satırların sayısını listeler.**

Kullanımı

```
int mysql_num_rows(int result);
```

mysql_num_rows(), bir sonuç kümesindeki satırların sayısını döndürür.

Bkz. mysql_db_query(), mysql_query() ve mysql_fetch_row().

Aşağı doğru uygunluk için mysql_numrows() da kullanılabilir.

- **mysql_pconnect → Bir MySQL sunucuya sürekli bir bağlantı açar.**

Kullanımı

```
int mysql_pconnect(string [hostname [:port] [:/path/to/socket] ]
, string [username] , string [password] );
```

Başarı durumunda pozitif bir MySQL link tanımlayıcı veya hata durumunda false değeri döndürür.

mysql_pconnect(), bir MySQL sunucuya bir bağlantı kurar. Tüm değişkenler seçimlidir ve eğer bunlar olmazsa, varsayılanları kabul edilir ('localhost', sunucu process' in sahibi olan kullanıcının kullanıcı ismi, boş password).

Hostname string' i, bir port numarası da içerebilir. Ör. "hostname:port" veya bir sokete bir path Ör. localhost için ":/path/to/socket"

Not: ":port" için destek 3.0B4'de eklendi. ":/path/to/socket" için destek 3.0.10'de eklendi.

mysql_pconnect(), mysql_connect()' e çok benzer. Ancak iki temel farkı vardır.

İlk olarak, bağlanıldığında, fonksiyon aynı host, username ve password ile açık bir link arayacaktır. Eğer bir tane bulunursa, onun için, yeni bir bağlantı açmak yerine, bir tanımlayıcı dönecektir.

Diğer fark; SQL sunucuya olan bağlantı, betimin çalışması bittiğinde kapatılmayacaktır. Bunun yerine, link gelecekteki kullanımlar için açık kalacaktır. (mysql_close(), mysql_pconnect() tarafından kurulan linki kapatmayacaktır).

Bu nedenle bu tür linkler 'persistent (sürekli)' olarak geçer.

- mysql_query → MySQL' e bir SQL sorgusu gönderir

Kullanımı

```
int mysql_query(string query, int [link_identifier] );
```

mysql_query(), sunucudaki o an aktif olan ve belirtilen link tanımlayıcı ile ilişkili olan veritabanına bir sorgu gönderir. Eğer link_identifier belirtilmemişse, son açılan link kabul edilir. Hiçbir link açık değilse, fonksiyon, mysql_connect() değişkensiz çağırılmış gibi bir link kurmaya çalışır ve onu kullanır.

Sorgu string' i bir noktalı virgül ile bitmemelidir.

mysql_query(), sorgunun başarılı olup olmamasına göre TRUE (non-zero) veya FALSE döndürür. TRUE dönüş değeri sorgunun yasal olduğunu ve sunucu tarafından çalıştırılabileceğini gösterir.Etkilenen ya da geri dönen satır sayısı hakkında hiçbir şey içermez. Bir sorgu, hiçbir satırı etkilemediği veya hiçbir satır döndürmediği halde başarılı olabilir.

Aşağıdaki sorgu syntactic olarak geçersizdir. Bu nedenle mysql_query() başarısız olur ve FALSE döndürür:

Örnek: mysql_query()

```
2 <?php
3 $result = mysql_query ("SELECT * WHERE 1=1")
4     or die ("Invalid query");
5 ?>
```

Aşağıdaki sorgu, eğer my_col my_tbl tablosunda bir sütun değilse, semantik olarak geçersizdir. Bu nedenle mysql_query() geçersizdir ve FALSE döndürür:

Örnek: mysql_query()

```
2 <?php
3 $result = mysql_query ("SELECT my_col FROM my_tbl")
4     or die ("Invalid query");
5 ?>
```

Eğer sorgunun ilişkili olduğu tablolara erişim izniniz yoksa, mysql_query() başarısız olacaktır ve FALSE döndürecektir.

Sorgunun başarılı olduğu varsayılırsa, kaç tane satırın etkilendiğini bulmak için mysql_affected_rows() çağırılabilir. (DELETE, INSERT, REPLACE, veya UPDATE işlemleri için). SELECT işlemleri için, mysql_query(), mysql_result()' a geçirebileceğiniz yeni bir sonuç tanımlayıcı döndürür. Sonuç kümesiyle yaptığınızda, mysql_free_result()' un çağırılmasıyla ilişkilendirilmiş kaynakları boşaltabilirsiniz.

Bkz. mysql_affected_rows(), mysql_db_query(), mysql_free_result(), mysql_result(), mysql_select_db(), ve mysql_connect().

- mysql_result → Sonuç veriyi getirir.

Kullanımı

```
int mysql_result(int result, int row, mixed [field] );
```

mysql_result(), bir MySQL sonuç kümesinden bir hücrenin içeriğini döndürür. Saha değişkeni, sahanın ofseti, sahanın ismi veya sahanın tablosu nokta sahanın ismi olabilir (SahaIsmi.TabloIsmi). Eğer sütun ismi alias' lanmışsa ('select foo as bar from...'), sütun ismi yerine alias kullanılır.

Büyük sonuç kümeleri üzerinde çalışıldığında, tüm satırı getirecek fonksiyonlardan biri kullanılmalıdır (aşağıda belirtildiği gibi). Bir fonksiyon çağrımında bu fonksiyonlar birden çok hücrenin içeriğini döndüreceğinden, bunlar, mysql_result()' tan çok daha hızlı olacaktır. Ayrıca, saha değişkeni için sayısal bir değer belirlemenin, bir saha ismi veya SahaIsmi.TabloIsmi değişkeni belirlemeden daha hızlı olduğu unutulmamalıdır.

mysql_result()' ı çağırma, sonuç kümesiyle ilgilenen diğer fonksiyonların çağırılmasıyla karıştırılmamalıdır.

Tavsiye edilen yüksek performanslı alternatifler: mysql_fetch_row(), mysql_fetch_array(), ve mysql_fetch_object().

- **mysql_select_db → MySQL veritabanı seçer**

Kullanımı

```
int mysql_select_db(string database_name, int [link_identifier]
);
```

Başarı durumunda true, hata durumunda false döner.

mysql_select_db(), sunucudaki belirtilen link tanımlayıcı ile ilişkili olan o anki aktif veritabanını set eder. Eğer hiçbir link tanımlayıcı belirlenmemişse, son açılan link kabul edilir. Hiçbir link açık değilse, fonksiyon mysql_connect() çağırılmış gibi bir link kurmaya ve bunu kullanmaya çalışacaktır.

mysql_query()'e sonradan gelen her çağrım, aktif veritabanı üzerinde yapılacaktır.

Bkz. mysql_connect(), mysql_pconnect(), ve mysql_query().

Aşağı doğru uygunluk için mysql_selectdb() de kullanılabilir.

- **mysql_tablename → Sahanın tablo ismini getirir**

Kullanımı

```
string mysql_tablename(int result, int i);
```

mysql_tablename(), hem mysql_list_tables() tarafından döndürülen bir sonuç pointer' ını hem de bir integer indeksi alır ve bir tablo ismi döndürür. mysql_num_rows() fonksiyonu, sonuç pointer' daki tablo sayısını belirlemede kullanılabilir.

Örnek: Mysql_tablename() örneği

```
2 <?php
3 mysql_connect ("localhost:3306");
4 $result = mysql_list_tables ("wisconsin");
5 $i = 0;
6 while ($i < mysql_num_rows ($result)) {
7     $tb_names[$i] = mysql_tablename ($result, $i);
8     echo $tb_names[$i] . "<BR>";
9     $i++;
10 }
11 ?>
```