به نام خدا

برنامه نویسی مبتنی بر وب

MySQL Database

SQL چیست؟

- SQL stands for Structured Query Language
- SQL lets you access and manipulate databases
- SQL became a standard of the American National Standards Institute (ANSI) and of the International Organization for Standardization (ISO)

What Can SQL do?

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

نکته

Note: Most of the SQL database programs also have their own proprietary extensions in addition to the SQL standard!

Using SQL in Your Web Site

- To build a web site that shows data from a database, you will need:
  - ✓ An RDBMS* database program (i.e. MS Access, SQL Server, MySQL)
  - ✓ To use a server-side scripting language, like PHP or ASP
  - ✓ To use SQL to get the data you want
  - ✓ To use HTML / CSS to style the page

*Relational Database Management System

واژگان SQL

- Tables
- Columns
- fields
- record

Most Important SQL Commands

- **SELECT** - extracts data from a database
- **UPDATE** - updates data in a database
- **DELETE** - deletes data from a database
- **INSERT INTO** - inserts new data into a database
- **CREATE DATABASE** - creates a new database
- **ALTER DATABASE** - modifies a database
- **CREATE TABLE** - creates a new table
- **ALTER TABLE** - modifies a table
- **DROP TABLE** - deletes a table
- **CREATE INDEX** - creates an index (search key)
- **DROP INDEX** - deletes an index

Select دستور

SELECT Syntax

```
SELECT column1, column2, ...
FROM table_name;
```

```
SELECT * FROM table_name;
```

استفاده از Where

WHERE Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

Note: The WHERE clause is not only used in SELECT statement, it is also used in UPDATE, DELETE statement, etc.!

Example

```
SELECT * FROM Customers
WHERE Country='Mexico';
```

Example

```
SELECT * FROM Customers
WHERE CustomerID=1;
```

استفاده از and و or

AND Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 AND condition2 AND condition3 ...;
```

OR Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 OR condition2 OR condition3 ...;
```

INSERT INTO دستور

INSERT INTO Syntax

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

```
INSERT INTO table_name
VALUES (value1, value2, value3, ...);
```

Example

```
INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)
VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway');
```

MySQL چیست؟

- پایگاه داده MySQL، روی وب استفاده می شود.
- پایگاه داده MySQL، روی سرور اجرا می شود.
- پایگاه داده MySQL، خیلی سریع، قابل اطمینان و استفاده از آن آسان است.
- پایگاه داده MySQL، از استاندرد SQLاستفاده می کند.
- قابلیت حمل بر روی سیستم عامل های مختلف از قبیل Linuxو windowsو غیره را دارد.
- استفاده و دانلود MySqlرایگان است.
- پایگاه داده MySQL، توسط شرکت Oracleتوسعه و پشتیبانی می شود.

PHP + MySQL Database System

- PHPبه همراه MySQl قابلیت cross-platform را داراست. یعنی پروژه یا سایت تان را می توانید روی Windowsپیاده کنید و روی Unixبه کاربران ارائه دهید.

Queries

- کوئری را می توان پرسیدن یا درخواست کردن معنی کرد. با کوئری ها ما می توانیم یک درخواست از پایگاه داده برای بدست آوردن اطلاعاتی خاص داشته باشیم.

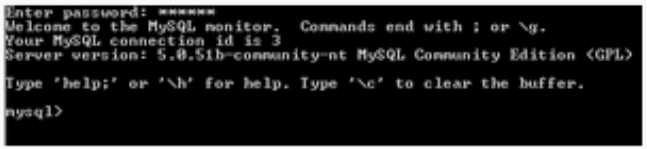
A company may have a database with the following tables:

- Employees
- Products
- Customers
- Orders

```
SELECT LastName FROM Employees
```

کوئری بالا همه ی داده های ستون LastName از جدول Customer را انتخاب می کند و جدول زیر را نمایش می دهد

MySQL



Screenshot of the default MySQL command-line banner and prompt

Original author(s)	MySQL AB
Developer(s)	Oracle Corporation
Initial release	23 May 1995; 22 years ago
Stable release	8.0.11[1] / 19 April 2018; 23 days ago
Repository	https://github.com/mysql/mysql-server git://anongit.gentoo.org/proj/mysql-extras.git
Development status	Active
Written in	C, C++[2]
Operating system	Windows, Linux, Solaris, macOS, FreeBSD[3]
Available in	English
Type	RDBMS
License	GPL (version 2) or proprietary[4]
Website	www.mysql.com

MySQL Customers

MEDIA & ENTERTAINMENT

- YouTube
- Netflix
- Spotify
- MORE

TELECOM

- Italtel
- Verizon Wireless
- Nokia
- MORE

WEB: SOCIAL NETWORKS

- YouTube
- Facebook
- Twitter
- MORE

WEB: ECOMMERCE

- Taobao
- Uber
- Airbnb
- MORE

https://www.mysql.com/customers/

ارتباط به پایگاه داده mysqlدر PHP

- در PHP نسخه ۵ به بالا، برای کار با پایگاه داده MySQL می توانید یکی از روش های زیر را استفاده نمایید:

  - ✓ افزونه MySQLi(کاراکتر i مخفف improved به معنی بهبود یافته است).
  - ✓ PDO که سرنام واژگان PHP Data Objects است.

- در نسخه های قبلی PHPاز افزونه MySQL استفاده می شده، اما استفاده از این افزونه، از ۲۰۱۲ دیگر توصیه نمی شود.

MySQLi or PDO?

- هر کدام از گزینه های MySQLi و PDO، مزیت های خودشان را دارند.
- گزینه PDO با ۱۲ پایگاه داده مختلف کار می کند، اما MySQLi تنها با پایگاه داده MySQL کار خواهد کرد.
- بنابراین اگر بخواهید زمانی به یک پایگاه داده دیگر سوییچ کنید، گزینه PDO کار را آسان تر خواهد کرد و فقط کافی است که ارتباط یا connection به پایگاه داده را تغییر دهید و احتمالاً چند تغییر کوچک در کوئری ها را خواهید داشت.
- هر دو گزینه شیء گرا هستند، اما MySQLi رویه های API را نیز ارائه می دهد.
- هر دو گزینه، دستورات آماده برای مقابله با تزریقات SQL یا SQL injection را پشتیبانی می کنند، که البته این موضوع، برای حفظ امنیت برنامه های کاربردی وب بسیار حیاتی است.

ارتباط یا Connection به پایگاه داده MySQL

- Open a Connection to MySQL
- قبل از اینکه به داده های پایگاه داده دسترسی داشته باشید، باید یک ارتباط یا Connection تعریف کنید.

Example (MySQLi Object-Oriented)

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

نکته

- توجه داشته باشید که در مثال شیء گرای بالا، $connect_error تا PHP 5.2.9و PHP ۵.۳.۰کار نمی کند، بنابراین اگر می خواهید که کدتان با نسخه های قدیمی PHP نیز سازگار باشد، بجای آن از کد زیر استفاده نمایید.

```
// Check connection
if (mysqli_connect_error()) {
    die("Database connection failed: " . mysqli_connect_error());
}
```

ایجاد ارتباط به روش رویه ای

Example (MySQLi Procedural)

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = mysqli_connect($servername, $username, $password);

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
?>
```

ایجاد ارتباط به روش PDO

Example (PDO)

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";

try {
    $conn = new PDO("mysql:host=$servername;dbname=myDB", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    echo "Connected successfully";
    }
catch(PDOException $e)
    {
    echo "Connection failed: " . $e->getMessage();
    }
?>
```

نکته

توجه داشته باشید که در مثال PDO بالا، با تنظیم گزینه dbname با مقدار "myDB" نام پایگاه داده را نیز مشخص کرده ایم. اگر PDO موفق به اتصال به پایگاه داده نشود یک استثنا یا exception پرتاب (thrown) می شود.

نکته: یکی از بزرگترین مزیت های PDO وجود کلاس exception برای مدیریت خطاها است. همان طور که می دانید این خطاها ممکن است در کوئری های پایگاه داده رخ دهد و اگر یک استثنا یا exception از داخل بلاک try{} پرتاب (thrown) شود، اجرای اسکریپت متوقف شده و جریان کار به اولین بلاک catch{} منتقل می شود. در واقع استثنای که در قسمت try رخ داده در قسمت catch به دام می افتد.

Close the Connection **بستن ارتباط**

- ارتباط یا Connection به صورت اتوماتیک وقتی که اسکریپت پایان یافت، بسته می شود، اما اگر مایل باشید می توانید این کار را قبل از پایان اسکریپت انجام دهید.

Example (MySQLi Object-Oriented)

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

Example (MySQLi Procedural)

```
mysqli_close($conn);
```

Example (PDO)

```
$conn = null;
```

ایجاد پایگاه داده MySQL

- برای ایجاد یا حذف پایگاه داده، باید دسترسی لازم را داشته باشید. دستور CREATE DATABASE برای ایجاد یک پایگاه داده در MySQL استفاده می شود.

مثال

Example (MySQLi Object-oriented)

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Create database
$sql = "CREATE DATABASE myDB";
if ($conn->query($sql) === TRUE) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . $conn->error;
}

$conn->close();
?>
```

Code 4.txt

نکته

- زمانی که می خواهید یک پایگاه داده جدید ایجاد کنید، در شیء mysqli تنها سه آرگومان اول را باید مشخص نمایید.
- (servername, username and password)

- اگر برای اتصال به پایگاه داده باید از یک Port مشخص استفاده نمایید، برای تنظیم این آرگومان در شیء mysqli باید آرگومان چهارم یعنی database-name را با یک رشته خالی تنظیم نمایید و سپس آرگومان پنجم را برای تنظیم Port استفاده نمایید.
- new mysqli("localhost", "username", "password", "", port)

Example (MySQLi Procedural)

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = mysqli_connect($servername, $username, $password);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// Create database
$sql = "CREATE DATABASE myDB";
if (mysqli_query($conn, $sql)) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

روش رویه ای

Code 5.txt

روش PDO

Example (PDO)

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";

try {
    $conn = new PDO("mysql:host=$servername;dbname=myDB", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "CREATE DATABASE myDBPDO";
    // use exec() because no results are returned
    $conn->exec($sql);
    echo "Database created successfully<br>";
    }
catch(PDOException $e)
    {
    echo $sql . "<br>" . $e->getMessage();
    }

$conn = null;
?>
```

Code 6.txt

نکته

- یکی از بزرگترین مزیت های PDO وجود کلاس exception برای مدیریت خطاها است. همان طور که می دانید این خطاها ممکن است در کوئری های پایگاه داده رخ دهد و اگر یک استثنا یا exception از داخل بلاک try { } پرتاب (thrown) شود، اجرای اسکریپت متوقف شده و جریان کار به اولین بلاک catch { } منتقل می شود. در واقع استثنای که در قسمت try رخ داده در قسمت catch به دام می افتد. در بلاک catch، با استفاده از دستور echo، دستور SQL و متن خطای تولید شده را چاپ می کنیم.

ایجاد جدول Create MySQL Tables

- یک جدول در پایگاه داده شامل یک نام منحصر بفرد می باشد و شامل تعدادی ستون و ردیف است. دستور CREATE TABLE برای ایجاد یک جدول در MySQL استفاده می شود.

We will create a table named "MyGuests", with five columns: "id", "firstname", "lastname", "email" and "reg_date":

```
CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP
)
```

نکاتی درباره جدول اسلاید قبل

- نوع داده مشخص می کند که چه داده هایی در فیلدها نگهداری شوند. در مثال قبل، موقعی که نوع فیلدها را varchar در نظر می گیرید باید حداکثر طول آنرا در پرانتز مشخص کنید(تا ۲۵۵ کاراکتر)، اگر به اندازه ای بزرگتر از ۲۵۵ نیاز دارید از نوع text استفاده کنید(تا ۶۵,۵۳۵ کاراکتر).

نکاتی درباره جدول اسلاید قبل

- **بعد از مشخص کردن نوع داده یا Data Type، گزینه های اختیاری دیگری نیز برای هر ستون وجود دارد:**
  - ✓ NOT NULL: محدودیت NOT NULL یک ستون را مجبور می کند که مقدار خالی را قبول نکند.
  - ✓ مقدار DEFAULT: محدودیت DEFAULT برای وارد کردن مقداری به صورت پیش فرض در یک ستون استفاده می شود.
  - ✓ UNSIGNED: این محدودیت تنها برای نوع عددی استفاده می شود و زمان اضافه کردن رکورد تنها اجازه وارد کردن صفر و اعداد مثبت وجود دارد.
  - ✓ AUTO INCREMENT: مقدار فیلد بصورت اتوماتیک به ازای هر رکورد جدید ۱ واحد اضافه می شود. (خود افزا)
  - ✓ PRIMARY KEY: هر جدول می تواند یک کلید اصلی (PRIMARY KEY) داشته باشد، کلید اصلی برای منحصر به فرد کردن ردیفهای یک جدول استفاده می شود، بنابراین مقدار فیلدی را که کلید در نظر می گیرد در کل نباید تکراری باشد. کلید اصلی معمولاً از نوع عددی و خودافزا است، همچنین باید با مقدار NOT NULL تنظیم شود.

Example (MySQLi Object-oriented)

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// sql to create table
$sql = "CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP
)";

if ($conn->query($sql) === TRUE) {
    echo "Table MyGuests created successfully";
} else {
    echo "Error creating table: " . $conn->error;
}

$conn->close();
?>
```

مثال

Code 7.txt

روش رویه ای و روش PDO برای ایجاد جدول

Code 8.txt

روش رویه ای

Code 9.txt

روش PDO

درج داده در MySQL

- بعد از اینکه پایگاه داده و جداول را ایجاد کردیم، حالا نوبت به درج اطلاعات می رسد. دستور INSERT INTO برای درج کردن یک رکورد جدید در جدول استفاده می شود.
- در اینجا چند قانون ساختاری وجود دارد که باید پیروی کنید:

✓ کوئری های SQLدر PHPباید با علامت کوتیشن محصور شوند.

✓ مقادیر رشته ای استفاده شده در کوئری ها باید با علامت کوتیشن محصور شوند.

✓ اعداد را نباید با کوتیشن محصور کنید.

✓ کلمه NULLنباید با کوتیشن محصور شود.

درج داده در MySQL

The INSERT INTO statement is used to add new records to a MySQL table:

```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
```

توجه: اگر هنگام تعریف جدول، برای یک ستون، گزینه AUTO_INCREMENT را تنظیم کرده باشیم (مانند فیلد id) و یا فیلد مورد نظر از نوع TIMESTAMP باشد (مانند فیلد reg_date) هنگام درج اطلاعات نیازی به مشخص کردن مقدار نیست و MySQL بصورت اتوماتیک آنها را پر خواهد کرد.

مثال

- می خواهیم به جدول MyGuests زیر که ایجاد کرده ایم، مقادیری را اضافه نمائیم.

```
CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP
)
```

Example (MySQLi Object-oriented)

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";

if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>
```

ادامه مثال

Code 10.txt

Example (PDO)

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "INSERT INTO MyGuests (firstname, lastname, email)
    VALUES ('John', 'Doe', 'john@example.com')";
    // use exec() because no results are returned
    $conn->exec($sql);
    echo "New record created successfully";
    }
catch(PDOException $e)
    {
    echo $sql . "<br>" . $e->getMessage();
    }

$conn = null;
?>
```

Code 11.txt

دستور select

- دستور SELECT برای انتخاب داده از یک جدول استفاده می شود. به این صورت که می توانیم با استفاده از نام ستون ها، تعداد محدودی از ستون های یک جدول را انتخاب کنیم. یا می توانیم با استفاده از کاراکتر * تمام ستون های یک جدول را انتخاب کنیم

```
SELECT column_name(s) FROM table_name
```

or we can use the * character to select ALL columns from a table:

```
SELECT * FROM table_name
```

انتخاب داده ها با استفاده از MySQLi

- در مثال زیر، ستون های id و firstname و lastname از جدول MyGuests در صفحه نمایش داده خواهند شد.

```php
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"].
"<br>";
    }
} else {
    echo "0 results";
}
```

Code 12.txt

توضیحات مثال قبل

توضیح مثال:

1. ابتدا یک کوئری SQL را تنظیم کردیم که ستون های id و firstname و lastname از جدول MyGuests را انتخاب می کند. نتیجه اجرای کوئری توسط تابع query() در متغیر $result ذخیره می شود.
2. حالا نتیجه کوئری در متغیر $result است و می خواهیم که آنها را در خروجی نمایش دهیم، اما قبل از آن باید مطمئن شویم که کوئری ما نتیجه ای را در بر داشته است، بنابراین با استفاده از تابع num_rows() چک می کنیم که ردیف ها بزرگتر از صفر باشد.
3. با استفاده از تابع fetch_assoc() ردیف اول داده ها برگردانده می شود، در یک اسکریپت با هر بار فراخوانی این تابع ردیف های بعدی بر می گردد.
4. با استفاده از حلقه while و با هر بار فراخوانی تابع fetch_assoc() نتیجه در متغیر $row ذخیره می شود و این کار تا آخرین رکورد ادامه می یابد و اطلاعات ستون ها چاپ می شود.

Example (MySQLi Object-oriented)

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    echo "<table><tr><th>ID</th><th>Name</th></tr>";
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "<tr><td>".$row["id"]."</td><td>".$row["firstname"]." ".$row["lastname"]."</td></tr>";
    }
    echo "</table>";
} else {
    echo "0 results";
}
$conn->close();
?>
```

Code 13.txt