

The architecture of Web-based products



The architecture of Web-based products

- Internet
- IP address, domain, DNS
- URI / URL
- Client-server applications
- HyperText Transfer Protocol (HTTP)
- HTML (HyperText Markup Language)
- CSS
- Document object model (DOM)
- CMS
- Databases



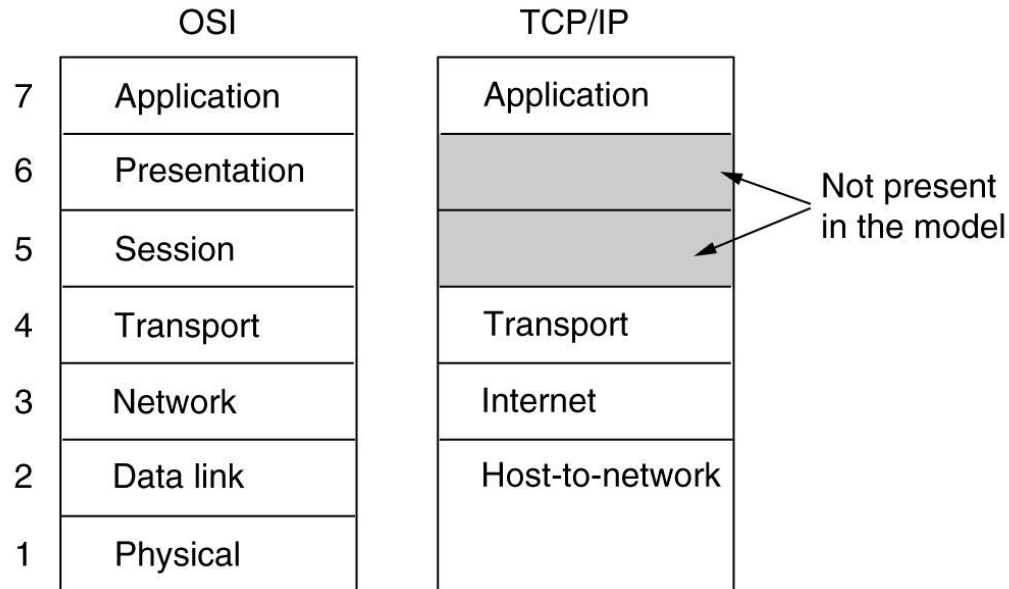
SQL



HTML & CSS

Architecture and principles of the Internet

Internet - is a set of computers (hosts) and various networks, unified network based on TCP / IP communications protocols. Computers connected to the Internet can be of any hardware and software platforms, but they have to maintain a stack of protocols (protocol family) TCP / IP connection.



Architecture and principles of the Internet

Domain name - a sequence of two or more words separated by dots. Word, or it is also called - the *domain* can consist of any combination of letters of the English alphabet, numbers and punctuation "-" (the "minus" or "dash"). Other characters in the *domain* name can not be used.



Architecture and principles of the Internet

How are interconnected IP-addresses and domain names?

Communication between IP-addresses and domain names provides an international distributed database based on the use of so-called of DNS-servers.

On the **DNS-server** in special registers and files stored line - what domain name matches any **IP-address** (forward zone) and reverse line **IP-address** - the domain name (reverse zone) for all that are in the domain managed by the owner of domain names.

Any change of correspondence should always be written on the **DNS-server**, only after that it "shall enter into force." For example, you can on your computer to register a domain name you want, but know this will be for him only you. If such an entry appears on the **DNS-server**, after some time (a maximum - a few hours) it knows the whole world.



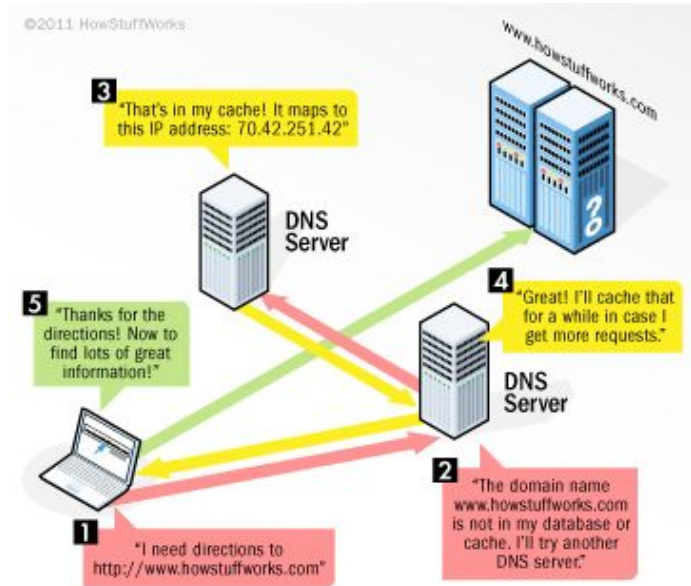
Architecture and principles of the Internet

How are interconnected IP-addresses and domain names?

Communication between IP-addresses and domain names provides an international distributed database based on the use of so-called DNS-servers.

The principle of operation of any DNS-server is quite simple.

Any client or server, which you want to determine whether the domain name - IP-address or IP-address - the domain name refers to DNS-server request. And if the DNS-server "knows" the correspondence of its stored zones, it reports it immediately. But if the **DNS-server** does not know anything about the requested domain name or **IP-address**, it refers to other **DNS-servers** "for help." Get answers to your questions from others of **DNS-servers**, the server informs the searched information to the Client.

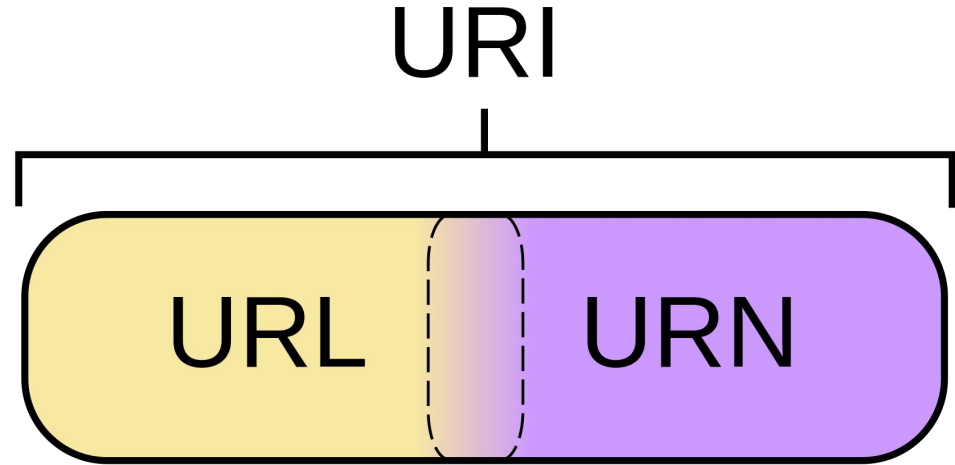


Architecture and principles of the Internet

Uniform Resource Identifier (URI)

Uniform Resource Identifier (URI) is a string of characters used to identify a resource. Such identification enables interaction with representations of the resource over a network, typically the World Wide Web, using specific protocols.

Schemes specifying a concrete syntax and associated protocols define each **URI**. The most common form of **URI** is the **Uniform Resource Locator (URL)**, frequently referred to informally as a web address. More rarely seen in usage is the **Uniform Resource Name (URN)**, which was designed to complement **URLs** by providing a mechanism for the identification of resources in particular namespaces.



Architecture and principles of the Internet

Uniform Resource Identifier (URI)

A **URL** is a **URI** that, in addition to identifying a web resource, specifies the means of acting upon or obtaining the representation of it, i.e. specifying both its primary access mechanism and network location.

For example, the URL:

http://example.org/wiki/Main_Page

refers to a resource identified as **[/wiki/Main_Page](#)** whose representation, in the form of **HTML** and related code, is obtainable via **HyperText Transfer Protocol (HTTP)** from a network host whose domain name is example.org.

<a href=
http://example.com/page>
(absolute)

VS.

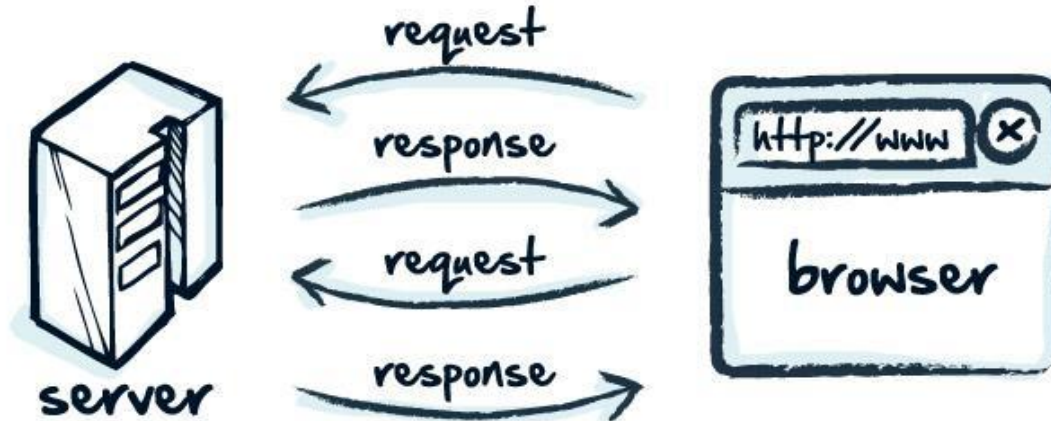
<a href=
/page>
(relative)

The architecture of web-based products

Web-based applications are a special type of software, built on the architecture "client-server".

The **client** part of implemented user interface (GUI) generates requests to the server and processes the responses from him.

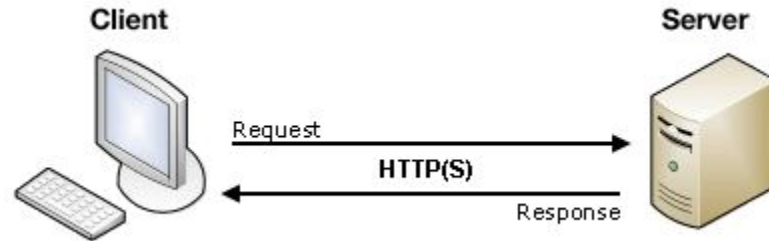
The **server** part receives a request from a client, performs the calculations, then creates a web page and sends it to the client over a network using HTTP protocol.



The architecture of web-based products

Web-based applications are a special type of software, built on the architecture "client-server".

Displaying the query results and receiving data from the **client** and their transfer to the server is usually handled by a special application - **browser** (Chrome, Internet Explorer, Mozilla, Opera, etc.). It is known that one of the functions of a **browser** is to display data received from **the Internet**, in the form of pages, described in the HTML language, therefore, the result is transmitted to the client server, to be submitted in this language.



In the **Web-based application** server side runs special software (Web-server), which receives the client requests, processes them, generates a response in the form of a page described in the HTML language, and transmits it to the client.

“Client-server” architecture of Web-based products

*By having executable part, **Web-application** capable of performing substantially the same operations as the conventional Windows-application, with the only limitation that the code is executed on the **server** as a system interface performs the **browser**, and as a medium through which the exchange takes place data - the **Internet**.*

The most common operations performed by **Web-based applications** include:

- receiving data from the user and stores them in the **server**;
- perform various actions on the user's request: retrieving data from the database (DB), add, delete, modify data in the database, conducting complex calculations;
- authenticate the user interface and display system according to the user;
- display constantly changing operational information.



Hypertext information exchange HTTP protocol

*HTTP - it is an application layer **protocol** developed for the exchange of hypertext information on the **Internet**. The **protocol** used by one of the most popular web systems - Word Wide Web - since 1990.*

Provides client communication (usually browser) and server (web-server) via the customer's requests, and server responses.

HTTP supports next methods (query types) from the client:

- GET - obtain content from the server;
- HEAD - get the title without the response body;
- POST - sending data to the server.
- And other (PUT, DELETE, TRACE, OPTIONS, CONNECT, PATCH).

HTTP request =

Request URI + HOST + `[[headers]]` + `<empty_line>` + body

```
POST /index.html HTTP/1.1
HOST: example.org:8080
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml
...
```

`param1=value1¶m2=value2`

HTTP response=

Response code + `headers` + `<empty_line>` + body

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 3123
...
```

`<html>...`

Hypertext information exchange HTTP protocol

Response Codes

- 1xx – information
- 101** – Switching Protocols
- 2xx – success
- 200** – OK
- 201** – created (new resource)
- 206** – partial content
- 3xx – redirection
- 301** – moved permanently
- 304** – not modified
- 4xx – client-side error
- 401** – unauthorized
- 404** – not found
- 5xx – server-side error
- 503** – server unavailable



301



404



500

Hypertext information exchange HTTP protocol

Requests (parameters) and examples

Examples

```
GET /hello.htm HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

GET parameters:
`http://server.name/path?param1=value1
¶m2=value`

```
POST /cgi-bin/process.cgi HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive

licenseID=string&content=string&/paramsXML=string
```

POST parameters:
POST /path HTTP/1.1
`param1=value1¶m2=value2`

Hypertext information exchange HTTP protocol

Cookies

Cookies - not a large amount of data sent by the server to browser and stored on disk. Maximum size - 4K.

Needed to support **stateful-services** (internet shopping cart store, etc.) or for storing the session key.

Cookies have a lifetime and are sent to the server with each request.

Examples



Hypertext Markup Language (HTML)

HTML is the standard markup language used to create web pages. Along with CSS, and JavaScript, HTML is a cornerstone technology, used by most websites to create visually engaging web pages, user interfaces for web applications, and user interfaces for many mobile applications.

The basic element of **HTML** is **TAG**. The tag is always concluded between the brackets **<>** and has the following form:

<TAG attribute 1 = VALUE attribute ... N = VALUE>

An **HTML** document begins opening tag **<HTML>** and ends with the closing tag **</HTML>**. Between this pair of container tags are two other main parts of an **HTML** document: header enclosed in a container **<HEAD> ... </HEAD>** and the body of the document in the container **<BODY> ... </BODY>**. Thus, a simple structure of an **HTML** document looks like this:

<HTML> - html file already begins from this tag

<HEAD>

Document Header

</HEAD>

<BODY>

Document Body

</BODY>

</HTML> - End of html document

Hypertext Markup Language (HTML)

Header is an optional part of the HTML document and used to determine the service information and the name of the document.

In the case of the container in the document header `<HEAD> ... </ HEAD>` it only required element is a container `<TITLE> ... </ TITLE>`, which specifies the name of the document. This is the name the user sees in the browser window title when viewing Web-pages on the Internet.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html lang="en-US" xml:lang="en-US">
<head>
<title>World Wide Web Online</title>
<link rel="shortcut icon" href="http://www.w3.org/Icons/stock-48x32.png" type="image/x-icon">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="Keywords" content="World Wide Web Online">
<meta name="Description" content="World Wide Web Online">
<link rel="stylesheet" type="text/css" href="http://www.w3.org/Icons/stock-48x32.png">
</head>
<body>
<table width="890" border="0">
<tr>
<td width="180"><img alt="World Wide Web Online logo" data-bbox="180 180 300 300">
<td align="left" valign="top">
<a href="http://www.w3.org/Icons/stock-48x32.png">
</td>
</tr>
</table>
```

All other **header** elements are not displayed by the browser and are used to determine various properties of the document, its relationship with other Web-pages and service information for external programs.

The **body** of an **HTML** document is located after the header and limited container `<BODY> ... </ BODY>`.

The contents of the body of the document is displayed in the

Web-browser window and consists of labeled tags to format text, tables, graphics and other multimedia elements, links to other resources.

Hypertext Markup Language (HTML)

Main HTML tags

Tag	Description
<html> ... </html>	Declares the Web page to be written in HTML
<head> ... </head>	Delimits the page's head
<title> ... </title>	Defines the title (not displayed on the page)
<body> ... </body>	Delimits the page's body
<h <i>n</i> > ... </h <i>n</i> >	Delimits a level <i>n</i> heading
 ... 	Set ... in boldface
<i> ... </i>	Set ... in italics
<center> ... </center>	Center ... on the page horizontally
 ... 	Brackets an unordered (bulleted) list
 ... 	Brackets a numbered list
 ... 	Brackets an item in an ordered or numbered list
 	Forces a line break here
<p>	Starts a paragraph
<hr>	Inserts a horizontal rule
	Displays an image here
 ... 	Defines a hyperlink

All list of HTML tags is available over the following link: <http://www.w3schools.com/tags>

Cascading Style Sheets (CSS)

CSS - a language that contains a set of properties to determine the appearance of the document. CSS Specification defines the properties and descriptive language to communicate with the HTML-elements.

According to the methods of adding style to the document, there are three kinds of styles.

Internal styles are defined by the style attribute specific tags. Interior style is valid only for certain tags such elements. This method is a little different from the traditional HTML.

External (related) styles are defined in a separate file with the extension .css; external styles allow all pages of the site to look the same way. To link to a file that describes the style, use the tag `<link>`, located in the container `<head> ... </ head>`. This tag must be assigned two attributes: **rel = "stylesheet"** and **href**, defines the address of the file styles.

Connection styles

The selector is located on the left side of the rule, determines the element(s) for which the rule is set. Further, braces style declaration lists separated by a semicolon.

CSS Reference is available over the following link:

<http://www.w3schools.com/cssref/default.asp>

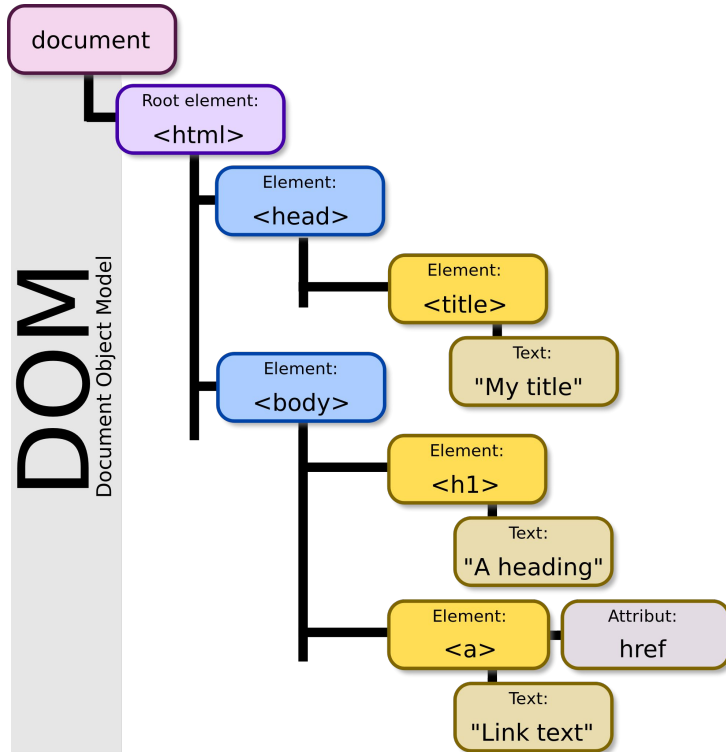
```
<p style="color:blue">Blue paragraph</p>
<p style="color:red">Red Paragraph</p>
```

```
<head>
.....
<style type="text/css">
  p {color:#808080;}
</style>
.....
</head>
```

```
<head>
.....
<link rel="stylesheet"
href="style.css">
.....
</head>
```

Document Object Model (DOM)

It is a cross-platform and language-independent convention for representing and interacting with objects in HTML, XHTML, and XML documents.



The nodes of every document are organized in a tree structure, called the *DOM* tree. Objects in the *DOM* tree may be addressed and manipulated by using methods on the objects. The public interface of a *DOM* is specified in its application programming interface (API).

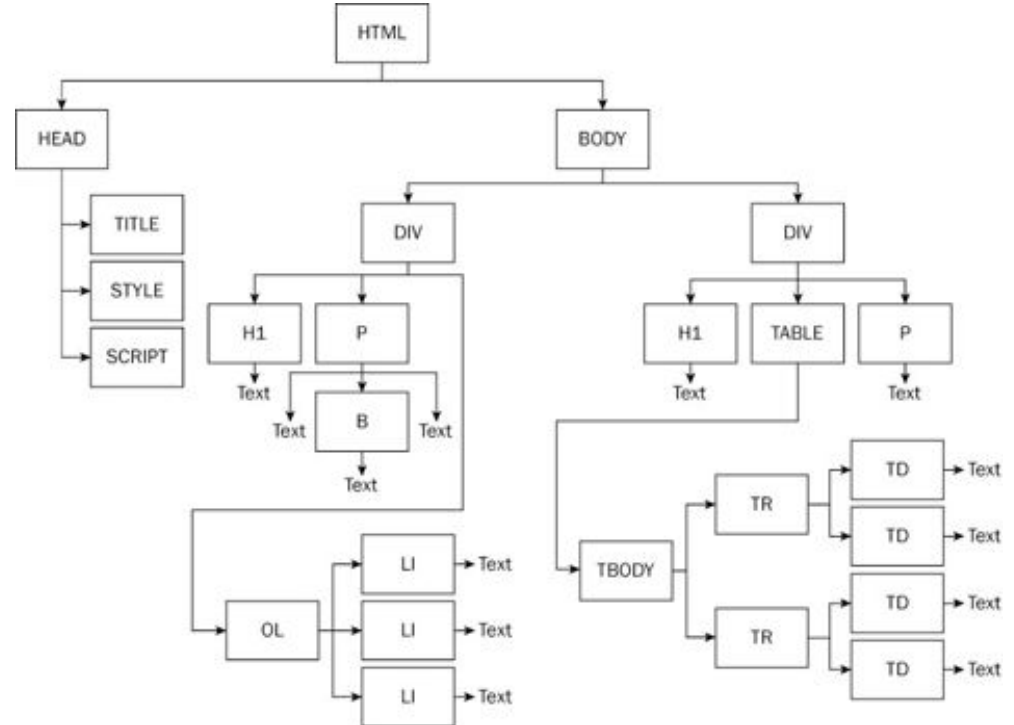
Document Object Model (DOM)

When an **HTML** page is rendered in browsers, the browser downloads the **HTML** into local memory and automatically parses it to display the page on screen. The **DOM** is also the way JavaScript transmits the state of the browser in **HTML** pages.

When a web page is loaded, the browser creates a **Document Object Model** of the page.

With the object model, **JavaScript** is fully enabled to create dynamic **HTML**:

- JavaScript can add, change, and remove all the **HTML** elements and attributes in the page
- JavaScript can change all the **CSS** styles in the page
- JavaScript can react to all existing events in the page
- JavaScript can create new events in the page



Content management system (CMS)

It's an application that allows publishing, editing, modifying, organizing, deleting, and maintaining content from a central interface. Such systems of content management provide procedures to manage workflow in a collaborative environment.

CMSs are often used to run websites containing blogs, news, and shopping. Many corporate and marketing websites use CMSs. CMSs typically aim to avoid the need for hand coding, but may help it for specific elements or entire pages.



Content management system (CMS)

Ecommerce CMS includes the software for businesses in which products sale and purchase are done. The transaction of products are usually in the form of digitally enabled, i.e. via credit card processing or any payment gateway or third party tools. In the field of mobile applications or websites or any ecommerce application, **CMSs** are usually used because they have built-in plugins with desired functionalities as demanded by customers.



The most commonly used tools are:

- WordPress
- Joomla
- Magento
- OsCommerce
- Opencart

Databases

Database (DB) - a shared set of logically related data (and a description), designed to meet the information needs of the organization.

DBMS (database management system) - the software that allows users to define, create and maintain a database and to obtain controlled access to it.

SQL language - the first and only standard language for database work, which has received fairly widespread. Virtually all of the major database developers now create their own products using the SQL language.

Main queries:

Database creation: **create database db_name;**

Usage of created DB: **use db_name;**

To create table in DB: **create table table_name (first_column_name type, .. , last_column_name type);**

To show all available databases: **show databases;**

To show a list of tables in the current database: **show tables;**

To show description specified table columns: **describe table_name;**

To delete table: **drop table table_name;**

To delete database: **drop database имя_базы данных;**

Databases

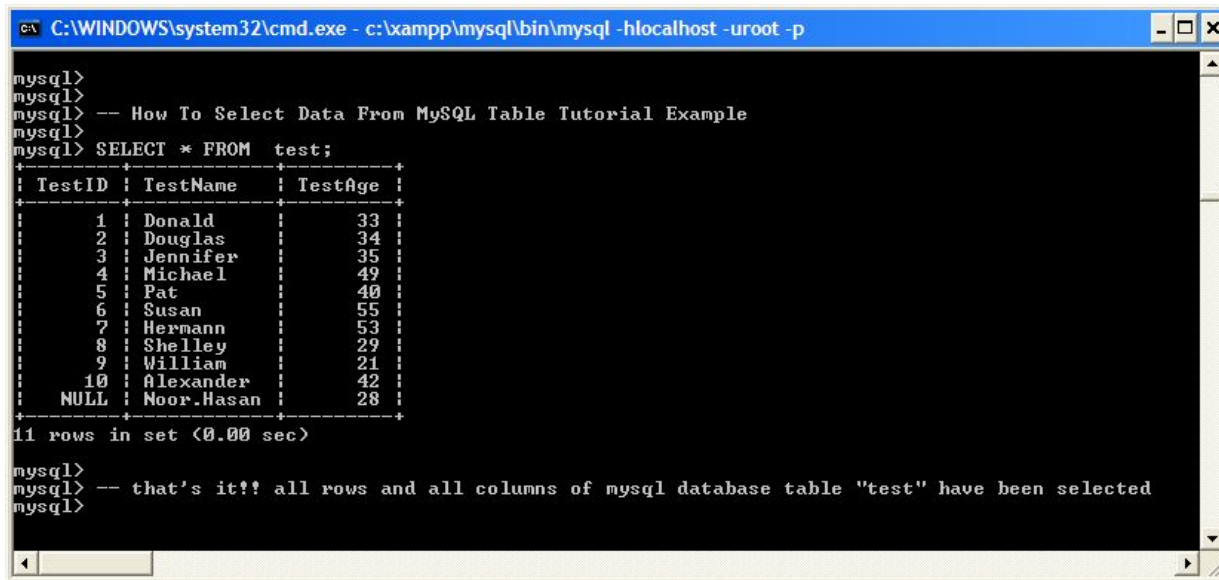
Select Data From a MySQL Database

The **SELECT** statement is used to select data from one or more tables

SELECT column_name(s) **FROM** table_name

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

SELECT * **FROM** table_name



```
C:\WINDOWS\system32\cmd.exe - c:\xampp\mysql\bin\mysql -hlocalhost -uroot -p

mysql>
mysql>
mysql> -- How To Select Data From MySQL Table Tutorial Example
mysql>
mysql> SELECT * FROM test;
+----+-----+-----+
| TestID | TestName | TestAge |
+----+-----+-----+
| 1 | Donald | 33 |
| 2 | Douglas | 34 |
| 3 | Jennifer | 35 |
| 4 | Michael | 49 |
| 5 | Pat | 40 |
| 6 | Susan | 55 |
| 7 | Hermann | 53 |
| 8 | Shelley | 29 |
| 9 | William | 21 |
| 10 | Alexander | 42 |
| NULL | Noor.Hasan | 28 |
+----+-----+-----+
11 rows in set (0.00 sec)

mysql>
mysql> -- that's it!! all rows and all columns of mysql database table "test" have been selected
mysql>
```

Databases

Different SQL queries

- Update Data In a MySQL Table:

UPDATE table_name **SET** column1=value, column2=value2,... **WHERE** some_column=some_value;

- Limit Data Selections From a MySQL Database:

SELECT * **FROM** Orders **LIMIT** 30;

- Add column to table:

ALTER TABLE table_name **ADD COLUMN** column_name type;

To combine rows from two or more tables (JOIN). The most common type of join is: SQL INNER JOIN (simple join). An SQL INNER JOIN returns all rows from multiple tables where the join condition is met. Example:

SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate

FROM Orders

INNER JOIN Customers

ON Orders.CustomerID=Customers.CustomerID;

Types of JOIN queries:

INNER JOIN: Returns all rows when there is at least one match in BOTH tables

LEFT JOIN: Return all rows from the left table, and the matched rows from the right table

RIGHT JOIN: Return all rows from the right table, and the matched rows from the left table

FULL JOIN: Return all rows when there is a match in ONE of the tables