

Core Module-2

(320 hours)

Hyper Text Markup Language (HTML)

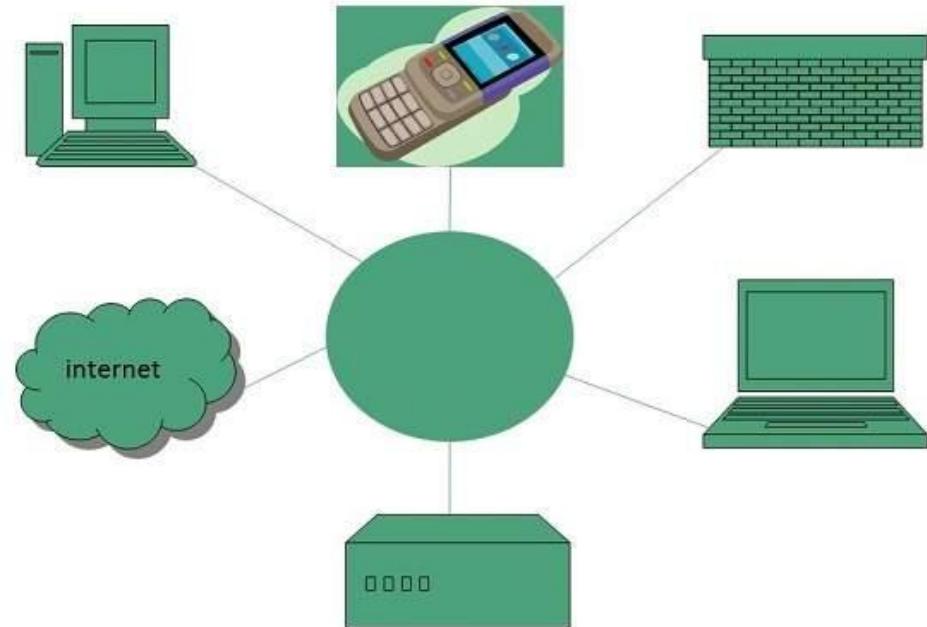
In this section, we will discuss:

- Introduction to internet, browsing, and emailing.
- Introduction to HTML.
- Different editors used for webpage development
- Applications of HTML

Introduction to Internet, Browsing, and Emailing

Overview of Internet

- Internet can be defined as an interconnected network of computers.
- The concept of Internet originated in year 1969.



Introduction to Internet, Browsing, and Emailing

Overview of Internet (continued)

- A special computer DNS (Domain Name Server) is used to give name to the IP Address so that user can locate a computer by a name.
- Internet is accessible to every user all over the world.

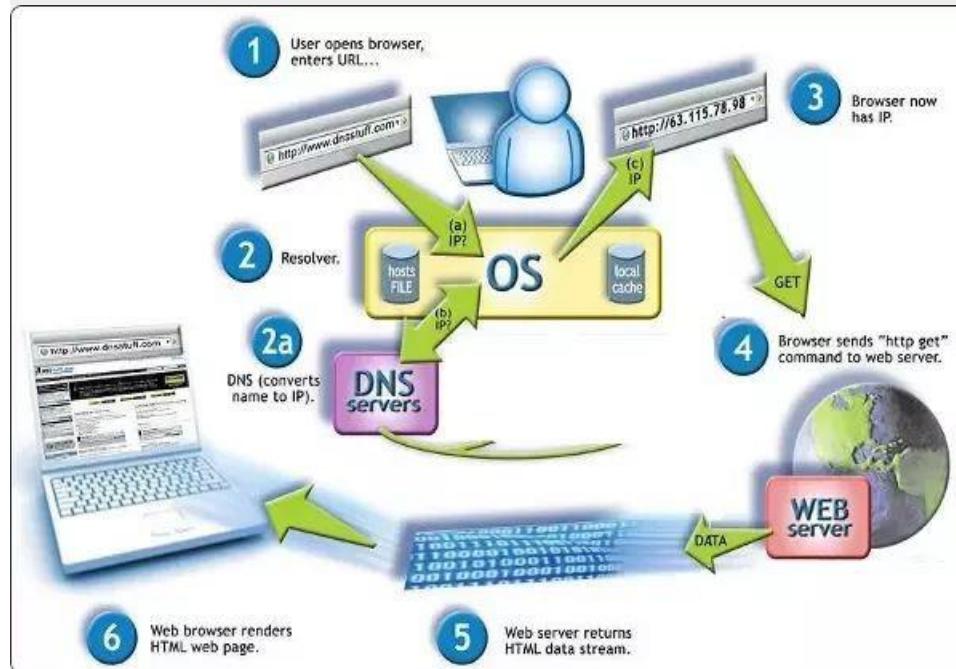


Introduction to Internet, Browsing, and Emailing

Overview of Internet (continued)

Example:

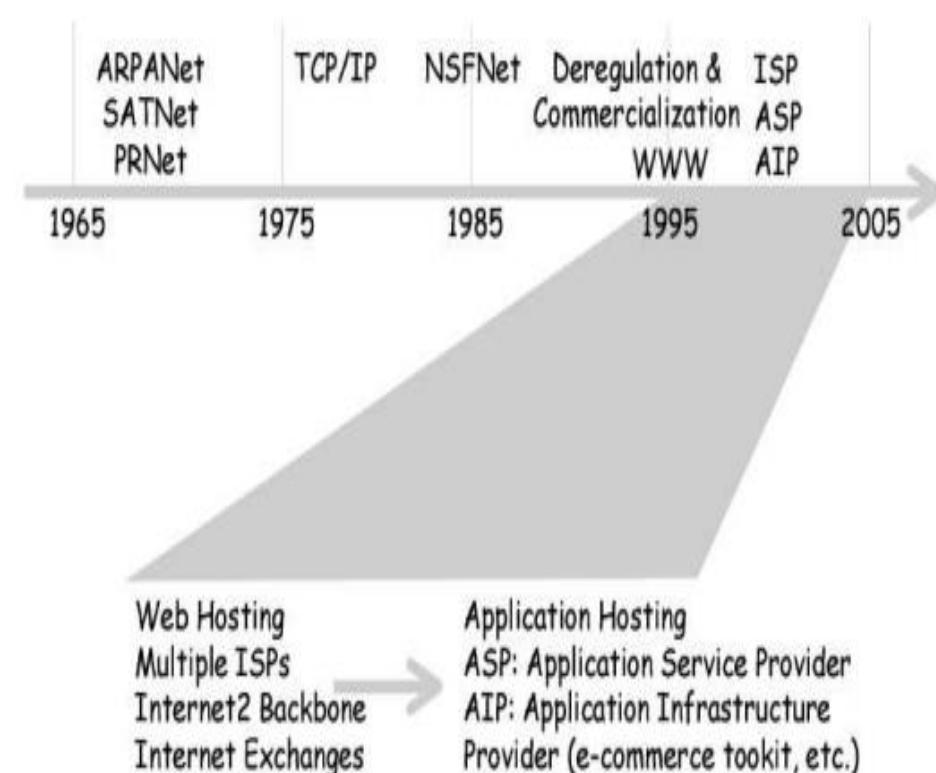
- DNS server will resolve a name `http://www.edunetworld.com` to a particular IP address to uniquely identify the computer on which this website is hosted.



Introduction to Internet, Browsing, and Emailing

Basics of Internet Architecture

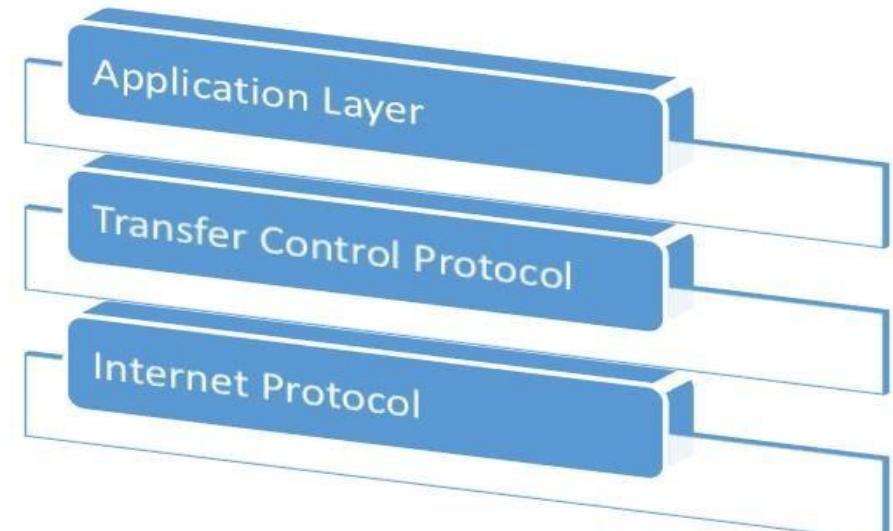
- Internet architecture is a meta-network, which refers to a congregation of thousands of distinct networks interacting with a common protocol.
- Protocol used is TCP/IP.
- This protocol connects any two networks that differ in hardware, software and design.



Introduction to Internet, Browsing, and Emailing

Basics of Internet Architecture

- Internet Protocol
- Transmission Control Protocol
- Application Protocol

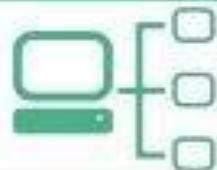


Introduction to Internet, Browsing, and Emailing

Internet Protocol

- The Internet Protocol (IP) is the method or protocol by which data is sent from one computer to another on the Internet.
- Internet Protocol, or IP, is the method that governs how computers share data across the Internet.

Internet Protocols



HTTP



Introduction to Internet, Browsing, and Emailing

Transmission Control Protocol

- It provides end to end transmission of data.
- It is a very complex protocol as it supports recovery of lost packets.

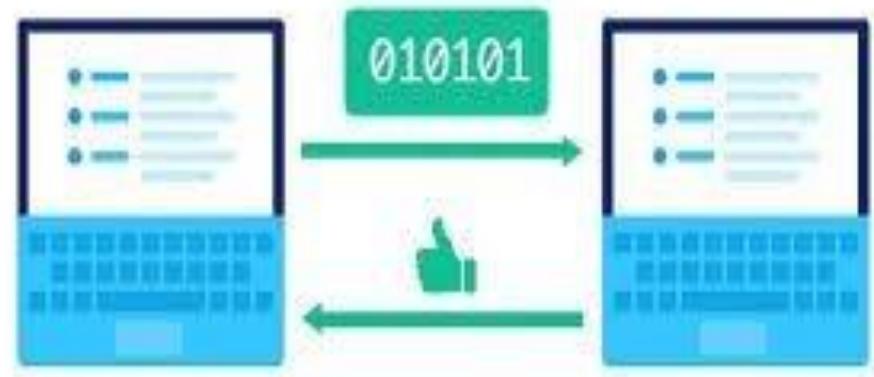


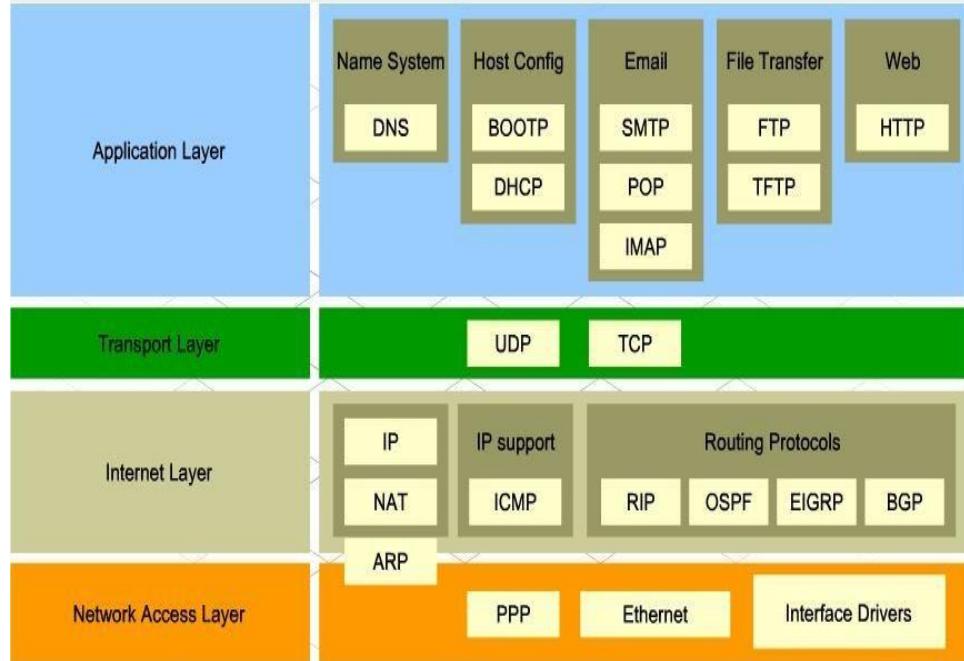
Image Source:

https://cdn.kastatic.org/googleusercontent/DVpBV1WYaAsfuAxR7GQncLGPIt4rbzCh0EmgvCtt8RVtqKPU_yc15ZA1auln8li_2nR2e8d8YW9tOv96Du7sqcvU

Introduction to Internet, Browsing, and Emailing

Application Protocol

- Third layer in internet architecture is the application layer which has different protocols on which the internet services are built.
- Some of the examples of internet services include email (SMTP facilitates email feature), file transfer (FTP facilitates file transfer feature), etc.



Introduction to Internet, Browsing, and Emailing

Services on Internet

- World Wide Web
- Websites



InterNet and World Wide Web

Introduction to Internet, Browsing, and Emailing

World Wide Web

- Web documents can be linked together, and are called "Hypertext".
- It is a way of exchanging information between computers on the Internet, tying them together into a vast collection of interactive multimedia resources.
- Tim Berners Lee invented the WWW in 1989.



Introduction to Internet, Browsing, and Emailing

World Wide Web (continued)

- To support hypertext documents, web uses a protocol called "Hypertext Transfer Protocol" (HTTP).
- HTTP and Links are foundation for WWW.



Introduction to Internet, Browsing, and Emailing

Websites

- A collection of associated web pages is called "Website".
- Websites are housed on the web servers.
- Copying a page onto a server is called "publishing" the page, which is also called "posting or uploading".



Introduction to Internet, Browsing, and Emailing

Accessing Web Browser

- A web browser is a software application which enables a user to display and interact with text, images, videos, music, and other information that could be on a website.
- "World Wide Web" or simple "Web" is the name given to all the resources of internet.



Introduction to Internet, Browsing, and Emailing

Accessing Web Browser (continued)

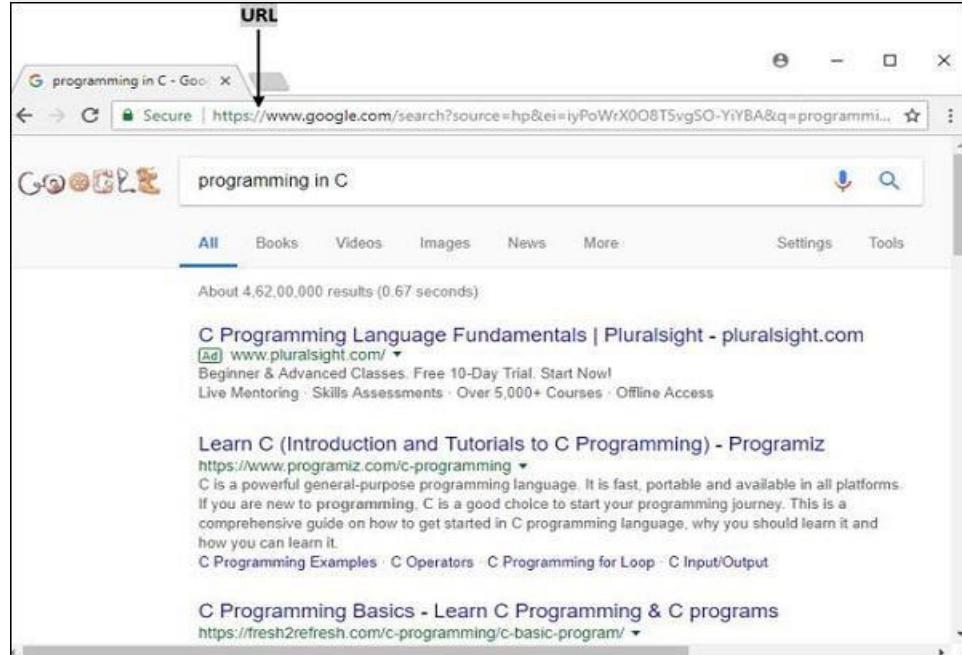
- Installation
- Launching a Web Browser
- Opening a webpage
- Popular Web browsing software



Accessing Web Browser

Installation

- Download the browser installer. Click on the blue “Download Chrome” button to begin downloading the browser. ...
- Run the installer. The downloaded file should then appear in the bottom of your browser. ...
- Install Google Chrome. ...
- Finalize the installation.



Accessing Web Browser

Launching a Web Browser

- Web browser is an application that is located on a computer's disk. Once you have an internet connection, you can launch a web browser using the following methods
- Step 1 – Go to "Start Menu".
- Step 2 – From the menu opened, click on the web browser Chrome

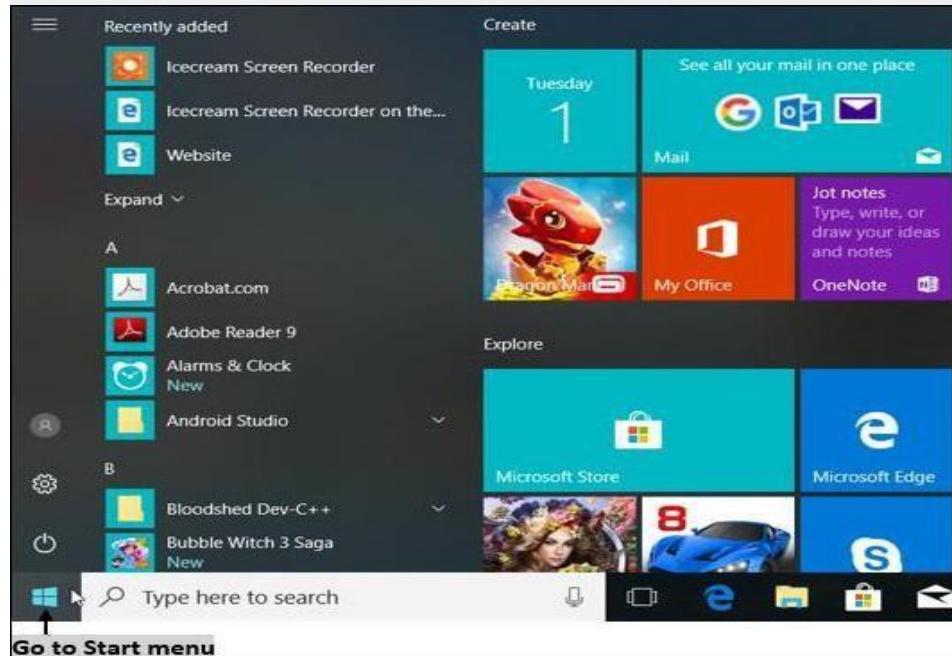


Image Source :

https://www.tutorialspoint.com/computer_concepts/computer_concepts_accessing_web_browser.htm

Accessing Web Browser

Launching a Web Browser (continued)

- Method 2 – Alternate way is to click the shortcut icon on the taskbar or desktop



Image Source :

https://www.tutorialspoint.com/computer_concepts/computer_concepts_accessing_web_browser.htm

Accessing Web Browser

Opening a webpage

- There are several ways to access a web page like using URLs, hyperlinks, using navigating tools, search engine, etc.



Image Source :

https://www.tutorialspoint.com/computer_concepts/computer_concepts_accessing_web_browser.htm

Accessing Web Browser

Popular Web Browsing Software

- Google Chrome
- Mozilla Firefox
- Opera
- Internet Explorer



Introduction to Internet, Browsing, and Emailing

Services Available on the Internet

- Data Transfer
- Internet banking
- E-commerce
- E-Learning
- E-Governance
- Browsing and Chatting
- E-Mail



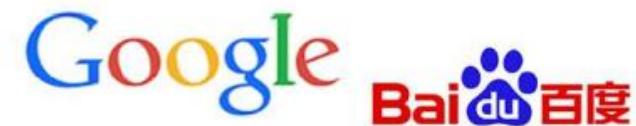
Introduction to Internet, Browsing, and Emailing

Search Engines

- Search Engine is an application that allows you to search for content on the web.

Most popular search engines

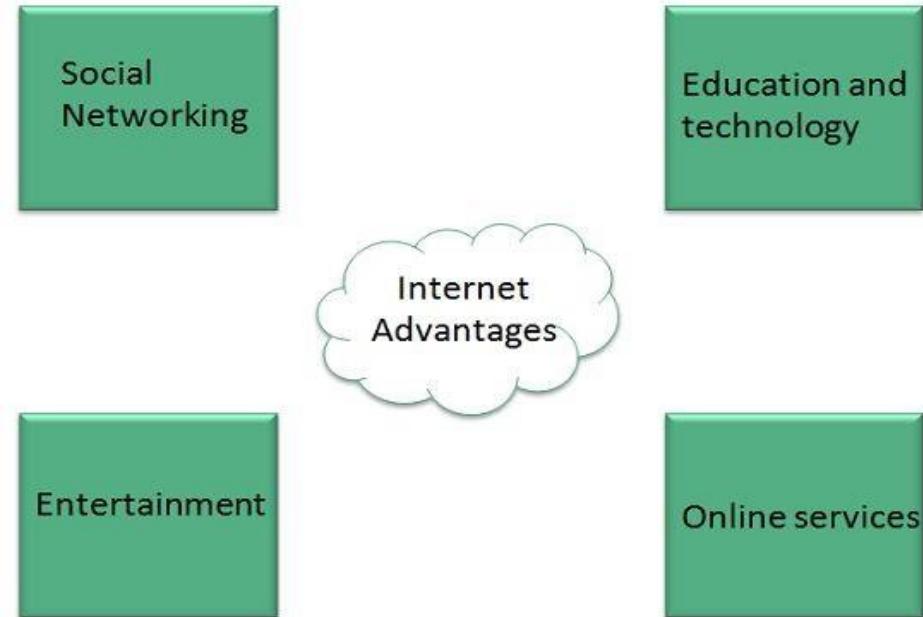
- Google
- Bing
- Yahoo
- Ask
- AOL



Introduction to Internet, Browsing, and Emailing

Advantages of Internet

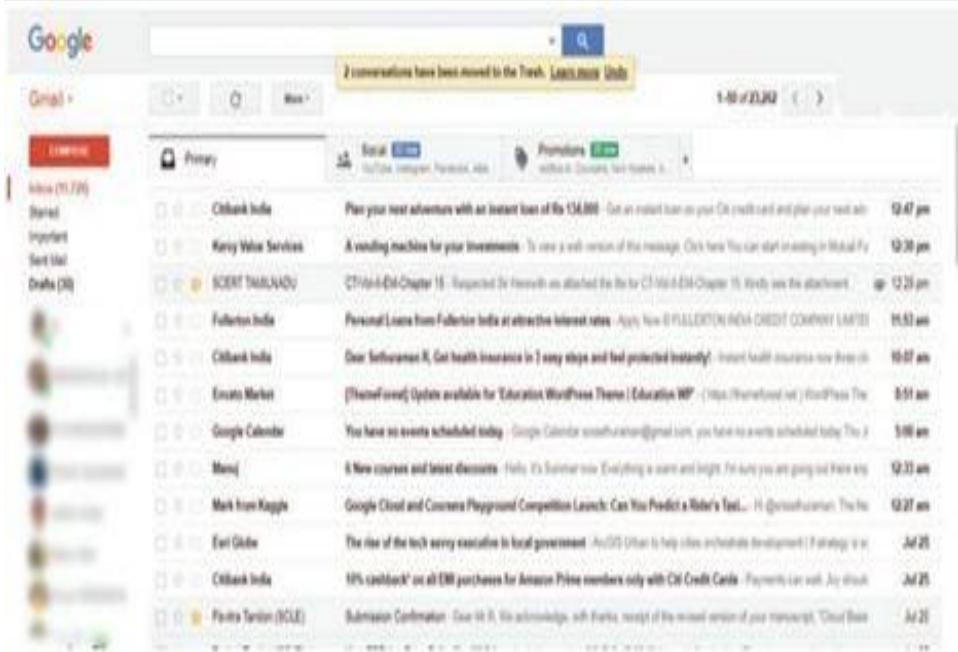
- Information, knowledge, and learning
- Connectivity, communication, and sharing
- Address, mapping, and contact information
- Banking, bills, and shopping
- Selling and making money
- Entertainment



Introduction to Internet, Browsing, and Emailing

Structure and Working of E-Mail

- Electronic Mail (email or e-mail) is a method of exchanging messages between people using electronic devices.
- Email operates across computer networks, which is primarily called as Internet.
- The structure of the E-mail address is `username@domain name`



Introduction to Internet, Browsing, and Emailing

Structure and Working of E-Mail (continued)

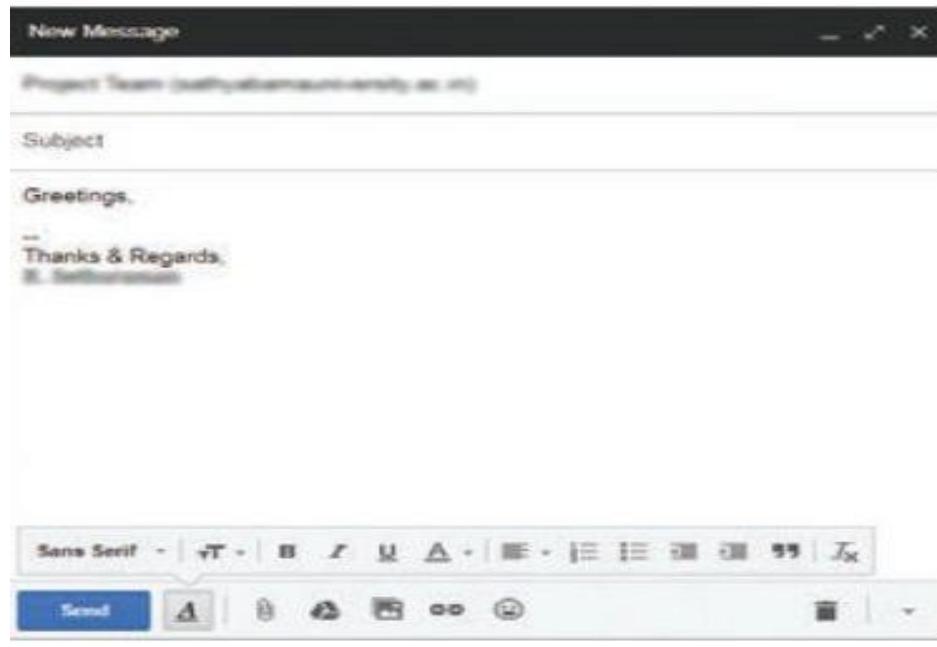
- An example of E-mail address is raman@gmail.com
- An E-mail address consists of two parts separated by @ symbol. The first part Raman is the user name that identifies the address and the second part gmail.com is the domain name of the E-mail server.



Introduction to Internet, Browsing, and Emailing

How Email works on the Internet ?

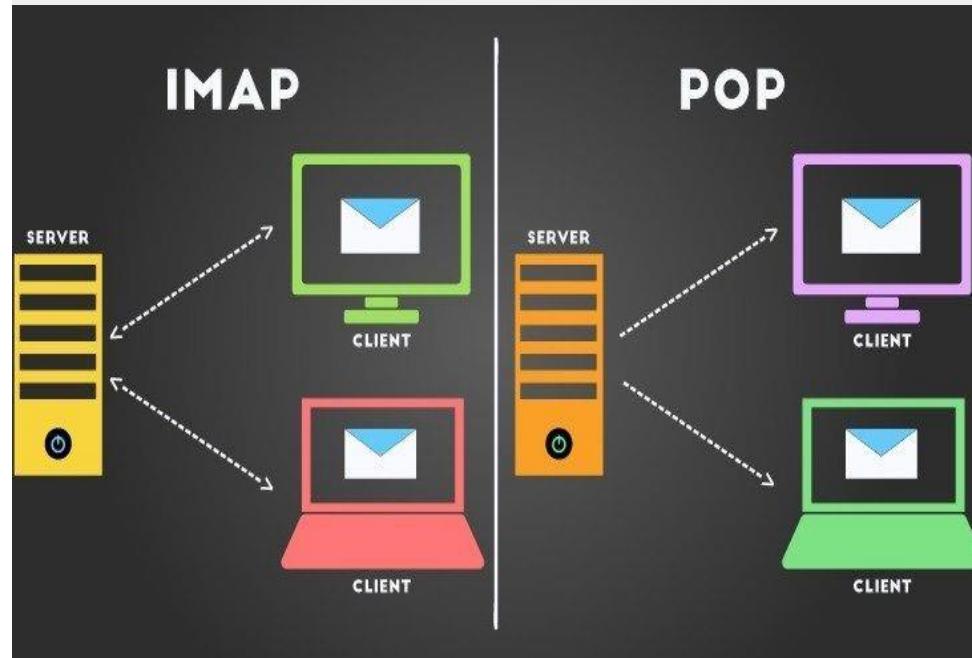
- To send Internet e-mail, requires an Internet connection and access to a mail server. The standard protocol used for sending Internet e-mail is called
- SMTP (Simple Mail Transfer Protocol).
- The SMTP protocol is used to both send and receive email messages over
- the Internet.



Introduction to Internet, Browsing, and Emailing

How Email works on the Internet ? (continued)

- When a message is sent, the email client sends the message to the SMTP server. If the recipient of the email is local the message is kept on the server for accessing by the POP, IMAP or other mail services for later retrieval.



Structure and Working of E-Mail

Structure of an Email message

- To
- CC
- BCC
- Subject
- Attachment
- Body
- Signature

How EMAIL REALLY WORKS



Structure and Working of E-Mail

Advantages

- Reliable
- Speed
- Inexpensive
- Waste Reduction



Structure and Working of E-Mail

Disadvantages

- Forgery
- Overload
- Junk

Advantages

- Easy to use
- Can be sent to multiple users at the same time

Disadvantages

- Lacks the personal touch
- Needs internet connection
- Can be hacked



Buzzle.com

Introduction to Internet, Browsing, and Emailing

Internet Applications

- Sending and receiving email.
- Searching and browsing information
- Copying files between computers.
- Conducting financial transactions.
- Navigating
- Playing interactive games.
- Video and music streaming.
- Chat or voice communication (direct messaging, video conferencing)



Introduction to HTML



Introduction to HTML

What is HTML?

- HTML stands for Hyper Text Markup Language
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements are represented by tags



Introduction to HTML

What is HTML?(continued)

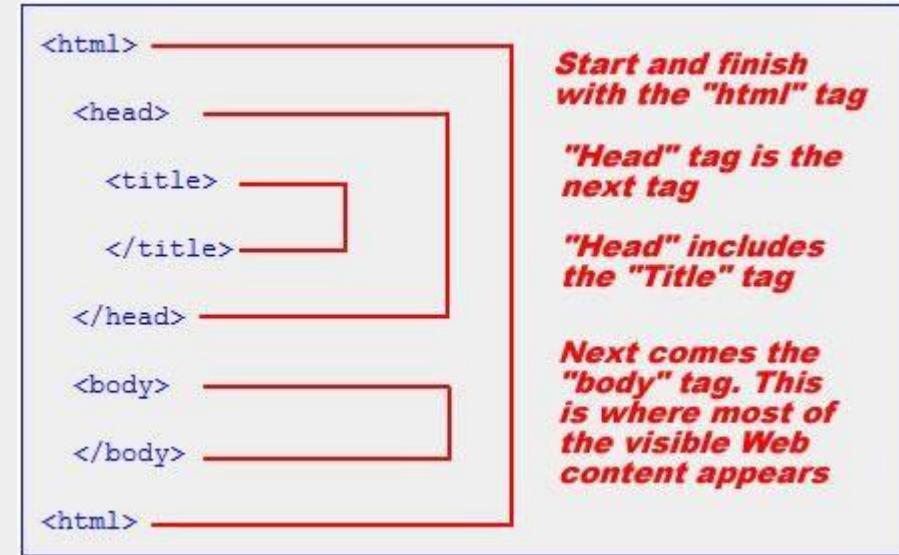
- "**Hypertext**" refers to the hyperlinks that an HTML page may contain.
- "**Markup language**" refers to the way tags are used to **define** the page layout and elements within the page.
- HTML page extension always will be .html



Introduction to HTML

Structure of an HTML document

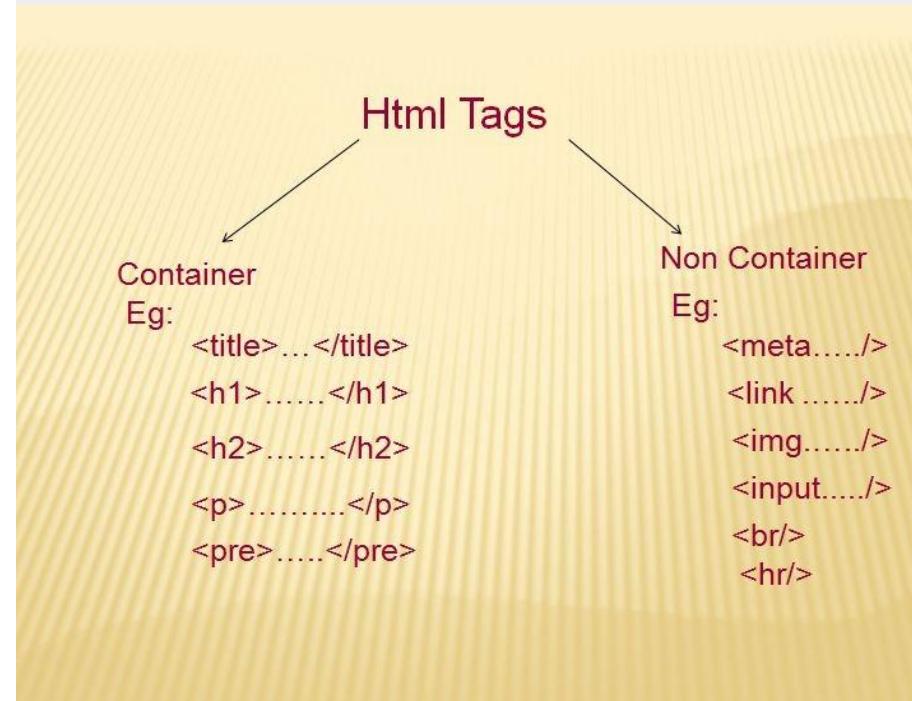
- The `<!DOCTYPE html>` declaration defines this document to be HTML5
- The `<html>` element is the root element of an HTML page
- The `<head>` element contains meta information about the document
- The `<title>` element specifies a title for the document
- The `<body>` element contains the visible page content



Introduction to HTML

HTML Tags and Attributes

- HTML Elements
- Heading
- Formatting
- Colors
- Texts and Images
- Linking
- Table
- Div and Span
- Lists
- Audio and Videos
- HTML Forms and Input



HTML Tags and Attributes

HTML Elements

- Elements are the fundamentals of Hypertext Markup Language (HTML).
- Each HTML document is made of elements that are specified using tags.

Types of HTML Elements

- empty elements
- Block-level elements
- Inline elements

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title of the document</title>
  </head>
  <body>
    <h1>Title of the document</h1>
    <p>The first paragraph</p>
    <p>The second paragraph, <br/> where line break is
       inserted </p>
    <footer> <p>Author: W3docs team</p> <p><a
       href="info@w3docs.com">Send</a> message to the
       author</a>.</p> </footer>
  </body>
</html>
```

HTML Tags and Attributes

Heading

- Headings are defined with the `<h1>` to `<h6>` tags.
- `<h1>` defines the most important heading.
- `<h6>` defines the least important heading.

```
<!DOCTYPE html>
<html>
<body>
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
</body>
</html>
```

HTML Tags and Attributes

Formatting

- HTML also defines special **elements** for defining text with a special **meaning**.

The formatting tags are divided into two categories:

- physical tags**, used for text styling (visual appearance of the text)
- logical or semantic** tags used to add semantic value to the parts of the text.

HTML TEXT FORMATTING

Tag	Purpose
 and 	To Bold Text
<i> and 	To italic Text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<small>	Defines smaller text
<big>	Defines bigger text
<ins>	Defines inserted text
	Defines deleted text
<mark>	Defines marked/highlighted text

HTML Tags and Attributes

Formatting

- **** - Bold text
- **** - Important text
- **<i>** - Italic text
- **** - Emphasized text
- **<mark>** - Marked text
- **<small>** - Small text
- **** - Deleted text
- **<ins>** - Inserted text
- **<sub>** - Subscript text
- **<sup>** - Superscript text

```
<!DOCTYPE html>
<html>
<body>
<b>This text is bold</b>
<strong>This text is strong</strong>
<i>This text is italic</i>
<em>This text is emphasized</em>
<p>The interest rate is only 10%*</p>
    <small>* per day</small>
<p>She likes <del>violets</del>
<ins>snowdrops</ins>.</p>
The formula of water is H<sub>2</sub>O
<p>E = mc<sup>2</sup>
</body>
</html>
```

HTML Tags and Attributes

Colors

There are three ways of how you can
change the color of the text in HTML:

- Hex color codes
- HTML color names
- RGB values.

```
<p style="color:red;"> This is a text in green</p>
```

```
<p style="color:#8ebf42;"> This is a text in green</p>
```

```
<p style="color:rgba(255,0,0,0.5);">Red paragraph  
text</p>
```

HTML Tags and Attributes

Images

- Images can improve the design and the appearance of a web page.
- images are defined with the `` tag.
- It is an empty tag, means it does not have end tag.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Placing Images in HTML Documents</title>
</head>
<body>
    
</body>
</html>
```

HTML Tags and Attributes

Links

- HTML links are hyperlinks.
- When you move the mouse over a link, the mouse arrow will turn into a little hand.

Internal Links

- An “**Internal Link**” is a **link** in your site that navigates the visitor to another page in your website.

Examples

- Internal link:
 - `my resume`
- Local Link:
 - `Go to page 2`
- Global link:
 - `Yahoo!`
- Link to email:
 - `E-mail at HTML.net`



HTML Tags and Attributes

Links

External Links

- The “**External Link**” navigates the visitor away from your site to another website in the internet (like <http://google.com>).

External Links (aka backlinks or inbound links)	Internal Links
Difficult to Control	Easy, fast and free to create
Pass SEO authority from other sites to your site, increasing your “domain authority”	Pass SEO authority between pages on your site, increasing the “page authority” of specific pages
Appear within the body text, in content	Appear in website navigation, as well as in the content

HTML Tags and Attributes

Tables

- The HTML tables allow displaying the data (e.g. image, text, link) in columns and rows of cells.
- The **<table>** tag defines an **HTML table**.
- An **HTML table** consists of the **<table>** element and one or more **<tr>**, **<th>**, and **<td>** elements.

First_Name	Last_Name	Marks
Sonoo	Jaiswal	60
James	William	80
Swati	Sironi	82
Chetna	Singh	72

HTML Tags and Attributes

Tables

- <tr> element defines a **table** row
- <th> element defines a **table** header
- <td> element defines a **table** cell.

```
<table style="width:80%;">
  <tr>
    <th>Month</th>
    <th>Date</th>
  </tr>
  <tr>
    <td>January</td>
    <td>10.01.2014</td>
  </tr>
  <tr>
    <td>February</td>
    <td>10.01.2014</td>
  </tr>
</table>
```

HTML Tags and Attributes

Div and Span

- <div> tag is used as a block part of the webpage.
- tag is used as an inline part of the webpage.

```
<div>A Computer Science Portal  
<span>Geeks<span></div>
```

HTML Tags and Attributes

Lists

- A list is a record of short pieces of information.

For example:

- A shopping list
- To-do list

```
<ul style="list-style-type:disc;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

```
<ol type="A">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

HTML Tags and Attributes

Type of Lists

- Unordered HTML List
- Ordered HTML List
- HTML Description Lists
- Nested HTML Lists

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

HTML Tags and Attributes

Audio and Video Tags

```
<audio controls>
<source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>
```

```
<video width="320" height="240" controls>
<source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

HTML Tags and Attributes

HTML Forms

- HTML form contains **form elements**.
- Form elements are different types of input elements, like: text fields, checkboxes, radio buttons, submit buttons and more.

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"
  value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname"
  value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

HTML Tags and Attributes

HTML Input

- The `<input>` element is displayed in several ways, depending on the type attribute.

Type	Description
<code><input type="text"></code>	Defines a single-line text input field
<code><input type="radio"></code>	Defines a radio button (for selecting one of many choices)
<code><input type="submit"></code>	Defines a submit button (for submitting the form)

Introduction to HTML

Markup Validation Service

- An **HTML validator** is a quality assurance program used to check Hypertext Markup Language (**HTML**) markup elements for syntax errors.
- A **validator** can be a useful tool for an **HTML** user who receives data electronically from a variety of input sources.



Introduction to HTML

HTML5

- HTML5 is the latest standard for browsers to display and interact with web pages.

Features:

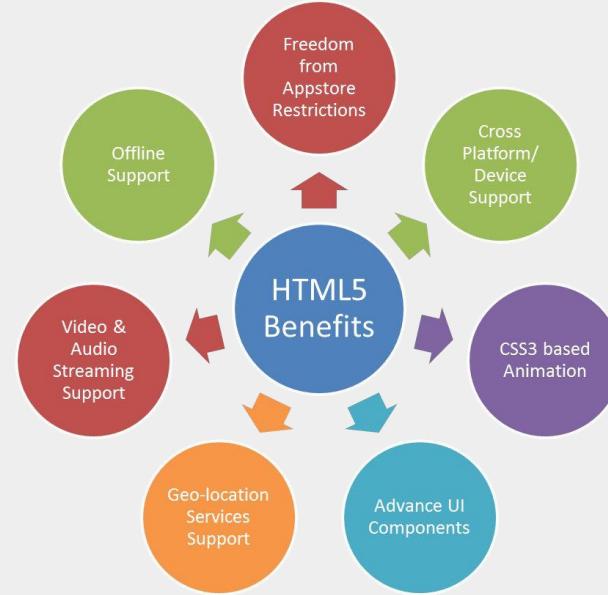
- semantic elements
- graphic elements
- attributes of form elements
- multimedia elements



Introduction to HTML

HTML5

- Page Layout - Semantic Elements
- Page Layout
- HTML5 Web Forms
- Scalable Vector Graphics
- HTML5 Media



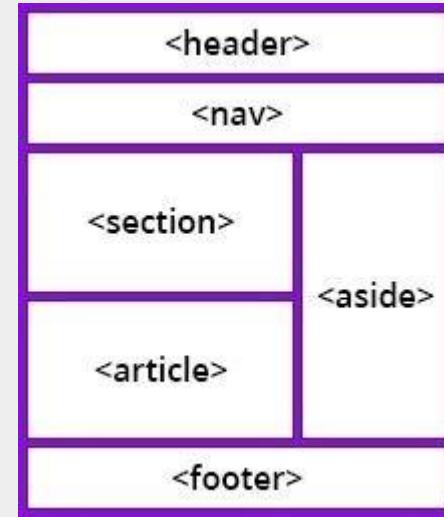
HTML5

Page Layout-Semantic Elements

- A **semantic element** clearly describes its meaning to both the browser and the developer.

Semantic Elements:

- **<header>**
- **<nav>**
- **<article>**



HTML5

Page Layout-Semantic Elements(continued)

- <section>
- <aside>
- <footer>
- <address>
- <main>

```
<header>
  <h1>Welcome to Web123.com</h1>
  <nav>
    <ul>
      <li>Home |</li>
      <li>About us |</li>
      <li>Contact us</li>
    </ul>
  </nav>
</header>
<footer>
  <p>© Copyright 2019. All rights reserved. </p>
</footer>
```

HTML5

Page Layout

- Page layout is the part of graphic design that deals with the arrangement of visual elements on a page.



The image shows a screenshot of the Yoko's Kitchen website, which is a template for a Japanese cooking classes page. The layout includes a header with the logo and navigation links (home, classes, catering, about, contact), a main content area with two course cards, and a sidebar with a 'Popular Recipes' section and a 'Contact' section.

YOKO'S KITCHEN
JAPANESE COOKING CLASSES

home **classes** **catering** **about** **contact**

Japanese Vegetarian
Five week course in London

A five week introduction to traditional Japanese vegetarian meals, teaching you a selection of rice and noodle dishes.

Bok Choi

Sauces Masterclass
One day workshop

An intensive one-day course looking at how to create the most delicious sauces for use in a range of Japanese cookery.

Teriyaki Sauce

Popular Recipes

- Yakitori (grilled chicken)
- Tsukune (minced chicken patties)
- Okonomiyaki (savory pancakes)
- Mizutaki (chicken stew)

Contact

Yoko's Kitchen
27 Redchurch Street
Shoreditch
London E2 7DP

HTML5

HTML5 Web Forms

- HTML5 Web Forms are used to design the web forms
- HTML5 introduces several attributes, input types and elements for markup tool kit.

```
<form action="/subscribe" method="post">
  <fieldset>
    <legend>Subscribe to the Newsletter</legend>
    <input type="email" name="email">
    <button>Ok</button>
  </fieldset>
</form>
```

HTML5

SCALABLE VECTOR GRAPHICS

- SVG defines vector-based graphics in XML format.
- SVG stands for Scalable Vector Graphics.
- Every element and every attribute in SVG files can be animated

```
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-
width="4" fill="yellow" />
</svg>
```

HTML5

HTML5 Media

HTML Plug-ins

- Plug-ins can be added to web pages with the `<object>` tag or the `<embed>` tag.
- Plug-ins can be used for many purposes: display maps, scan for viruses, verify your bank id, etc.

```
<object width="100%" height="500px" data="snippet.html"></object>
```

```
<embed width="400" height="50" src="bookmark.swf">
```

HTML5

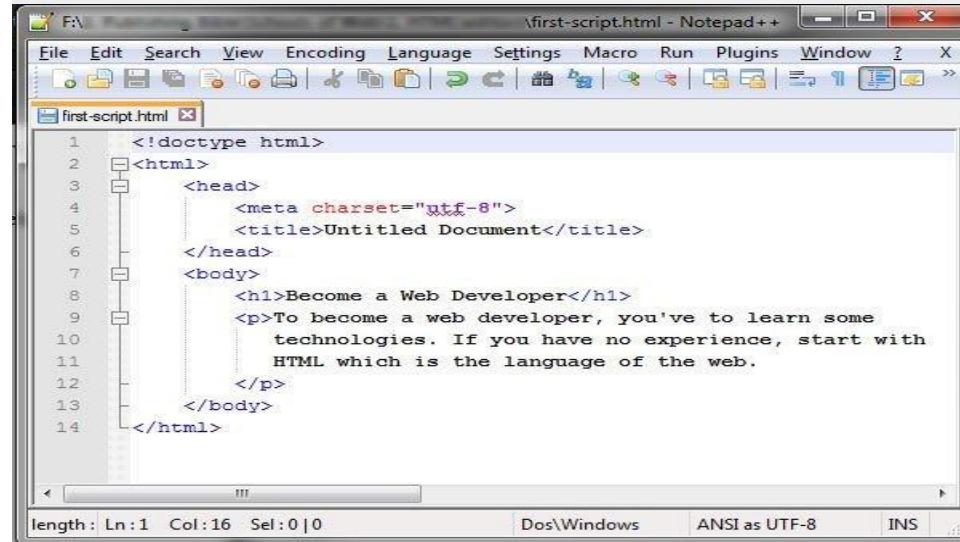
HTML5 Media

YouTube

- The easiest way to play videos in HTML, is to use YouTube.

```
<iframe width="420" height="315"
src="https://www.youtube.com/embed/tgbNymZ7vqY?
autoplay=1">
</iframe>
```

Different editors used for Webpage Development



The screenshot shows a Notepad++ window with the file 'first-script.html' open. The code in the editor is as follows:

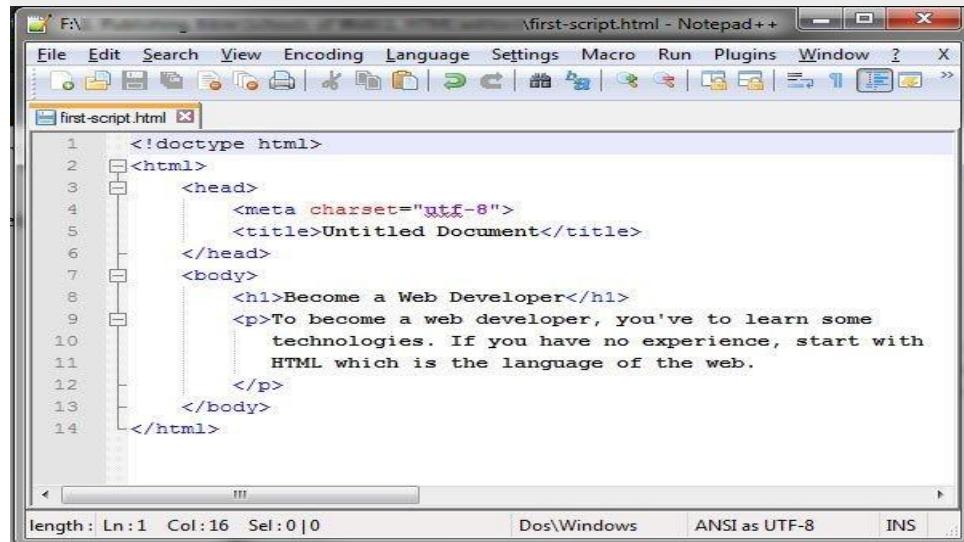
```
1  <!doctype html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Untitled Document</title>
6      </head>
7      <body>
8          <h1>Become a Web Developer</h1>
9          <p>To become a web developer, you've to learn some
10             technologies. If you have no experience, start with
11             HTML which is the language of the web.
12         </p>
13     </body>
14 </html>
```

The status bar at the bottom of the Notepad++ window displays: Length: Ln:1 Col:16 Sel:0|0 Dos\Windows ANSI as UTF-8 INS.

Different editors used for Webpage Development

HTML Editors

- HTML text editors are used to create and modify web pages.
- HTML codes can be written in any text editors including the **notepad**.



The screenshot shows the Notepad++ application window with the title bar 'first-script.html - Notepad++'. The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Macro, Run, Plugins, Window, and Help. The toolbar has various icons for file operations. The code editor window displays the following HTML code:

```
1  <!doctype html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Untitled Document</title>
6      </head>
7      <body>
8          <h1>Become a Web Developer</h1>
9          <p>To become a web developer, you've to learn some
10             technologies. If you have no experience, start with
11             HTML which is the language of the web.
12         </p>
13     </body>
14 </html>
```

The status bar at the bottom shows 'Length: Ln:1 Col:16 Sel:0|0' and 'Dos\Windows ANSI as UTF-8 INS'.

Different editors used for Webpage Development

Common features of HTML Code Editors

- Auto-completion.
- Adding library for HTML entities.
- With the help of Site Explorer, you can view the files in a hierarchical pattern.
- Some editors have built-in FTP to upload the files faster.
- Advance HTML editors provide support for other languages like CSS and JavaScript.
- highlighting syntax errors

Key Features of an HTML Editor
Interactive Text, HTML & Source Code Editor
Cleadding of Messy Code
Word to HTML Conversion
Find and Replace Tools for Texts Replacements
Table to DIV Conversions

Different editors used for Webpage Development

Different editors used for Web Page

Some of the Popular Html Editors

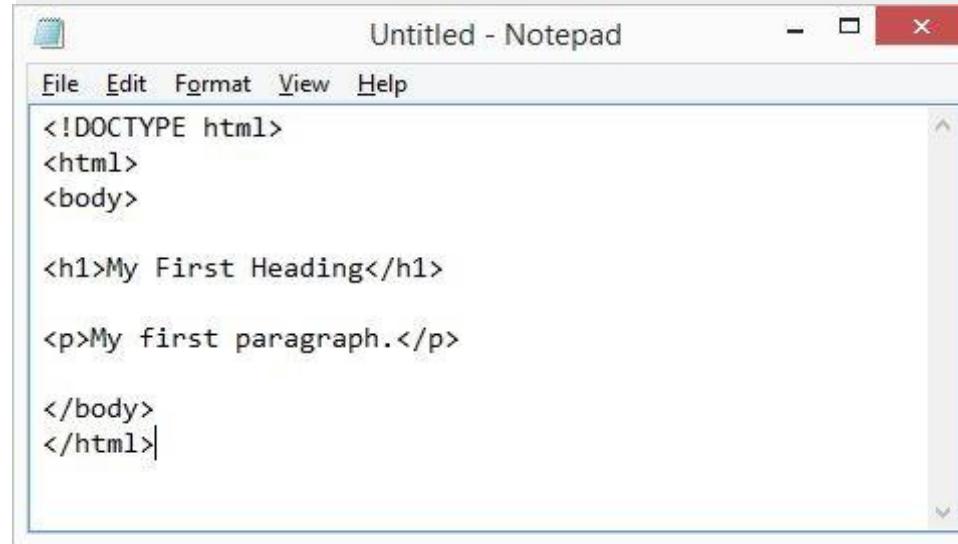
- Phase 5 HTML Editor
- Notepad ++
- Sublime Text
- jEdit HTML Editor
- AdobeBrackets
- SynWrite Editor
- Visualcode Editor



Different editors used for Webpage Development

Different editors used for Web Page(continued)

- **Notepad:** Notepad is a simple text editor. It is an inbuilt desktop application available in Windows OS.



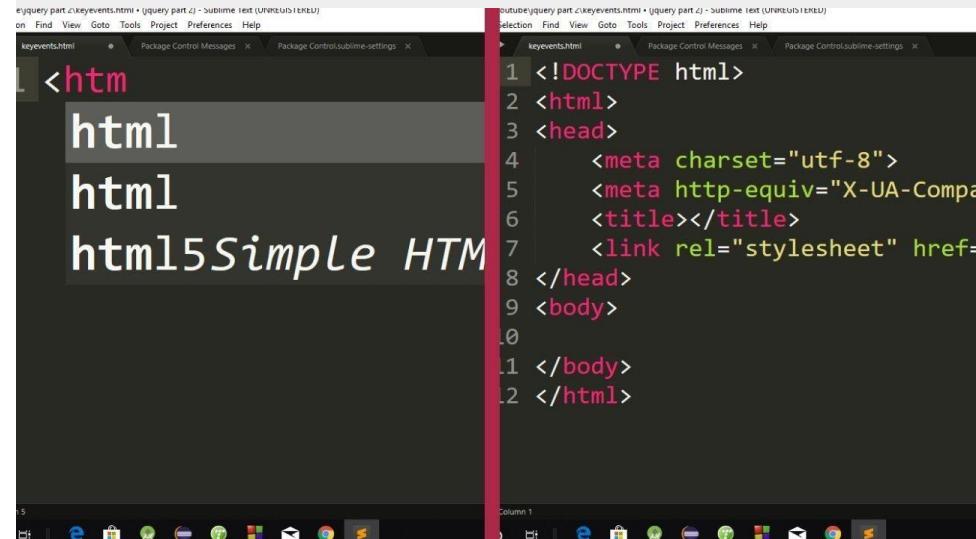
The image shows a screenshot of the Windows Notepad application. The window title is "Untitled - Notepad". The menu bar includes "File", "Edit", "Format", "View", and "Help". The main content area contains the following HTML code:

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

Different editors used for Webpage Development

Different editors used for Web Page(continued)

- **Sublime Text 3:** Sublime is a cross platform code editor tool. It supports all markup languages.



The screenshot shows two Sublime Text 3 windows side-by-side. Both windows have the title 'keyevents.html' and are using the 'Package Control' theme. The left window shows a partial HTML file with the following code:

```
<htm
html
html
html5Simple HTM
```

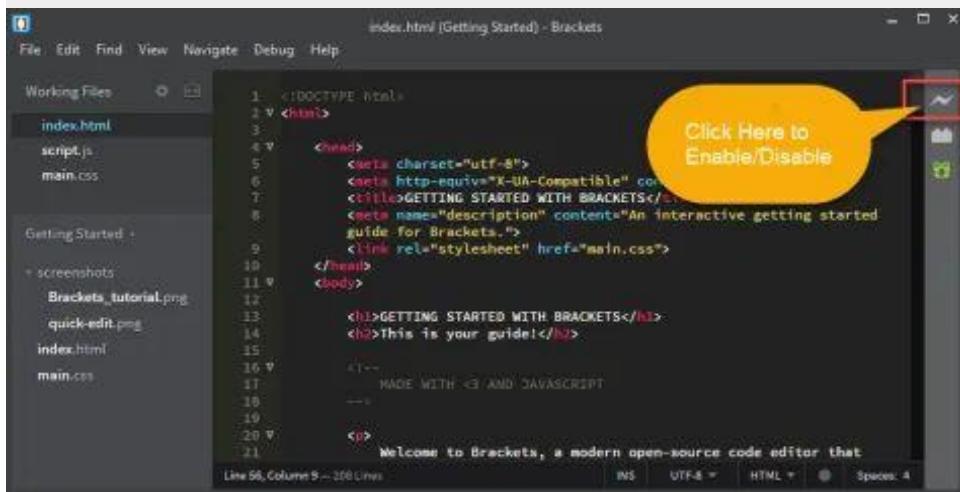
The word 'html' is being typed, and the code completion dropdown is open, showing 'html', 'html', and 'html5Simple HTM' as suggestions. The right window shows the full HTML code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <title></title>
7   <link rel="stylesheet" href="style.css" type="text/css" media="screen" />
8 </head>
9 <body>
10
11 </body>
12 </html>
```

Different editors used for Webpage Development

Different editors used for Web Page(continued)

- **Bracket:** Bracket is an open-source software primarily used for Web development. It provides live HTML, CSS, JavaScript editing functionality.



The screenshot shows the Brackets code editor interface. The title bar reads "index.html (Getting Started) - Brackets". The menu bar includes File, Edit, Find, View, Navigate, Debug, and Help. The left sidebar shows "Working Files" with "index.html" selected, along with "script.js" and "main.css". Below that is a "Getting Started" section with "screencasts", "Brackets_tutorial.png", "quick-edit.png", "index.html", and "main.css". The main editor area displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3
4   <head>
5     <meta charset="utf-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <title>GETTING STARTED WITH BRACKETS</title>
8     <meta name="description" content="An interactive getting started guide for Brackets.">
9     <link rel="stylesheet" href="main.css">
10
11   </head>
12
13   <body>
14     <h1>GETTING STARTED WITH BRACKETS</h1>
15     <h2>This is your guide!</h2>
16
17     <p>MADE WITH <code>HTML</code> AND <code>JAVASCRIPT</code></p>
18
19
20
21 </body>

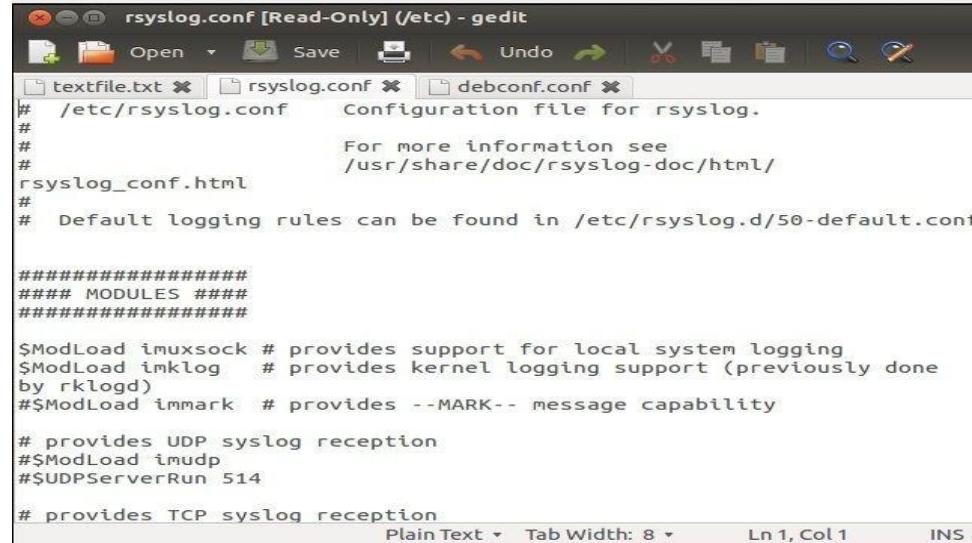
```

A yellow speech bubble with the text "Click Here to Enable/Disable" points to a gear icon in the top right corner of the editor window. The status bar at the bottom shows "Line 56, Column 5 — 200 Lines", "INS", "UTF-8", "HTML", and "Spaces: 4".

Different editors used for Webpage Development

Different editors used for Web Page(continued)

- **gedit** provides a simple interface from which you have access to a full **text editor** with programming functions and is compatible with most languages



The screenshot shows the gedit text editor window with the file `/etc/rsyslog.conf` open. The window title is `rsyslog.conf [Read-Only] (/etc) - gedit`. The text in the editor is a configuration file for rsyslog, starting with a license notice and a section for modules. The code is as follows:

```
rsyslog.conf [Read-Only] (/etc) - gedit
textfile.txt ✘ rsyslog.conf ✘ debconf.conf ✘
/etc/rsyslog.conf    Configuration file for rsyslog.
#
# For more information see
#          /usr/share/doc/rsyslog-doc/html/
rsyslog_conf.html
#
# Default logging rules can be found in /etc/rsyslog.d/50-default.conf

#####
#### MODULES #####
#####

$ModLoad imuxsock # provides support for local system logging
$ModLoad imklog   # provides kernel logging support (previously done
by rklogd)
#$ModLoad immark  # provides --MARK-- message capability

# provides UDP syslog reception
#$ModLoad imudp
#$UDPServerRun 514

# provides TCP syslog reception
```

At the bottom of the editor, there are buttons for **PlainText**, **Tab Width: 8**, **Ln 1, Col 1**, and **INS**.

Different editors used for Webpage Development

SublimeText

Installation

- **Step 1** – Download the **.exe** package from the official website as shown below
<https://www.sublimetext.com/3>
- **Step 2** – Now, run the executable file. This defines the environment variables. When you run the executable file, you can observe the following window on your screen. Click **Next**.



Different editors used for Webpage Development

SublimeText

- **Step 3** –Now, choose a destination location to install Sublime Text3 and click **Next**.
- **Step 4** –Verify the destination folder and click **Install**.
- **Step 5** –Now, click **Finish** to complete the installation.
- **Step 6** –Upon a successful installation, your editor will appear as shown below



Different editors used for Webpage Development

SublimeText

Features

- Syntax Highlight
- Auto Indentation
- File Type Recognition
- Sidebar with files of mentioned directory
- Macros
- Plug-in and Packages

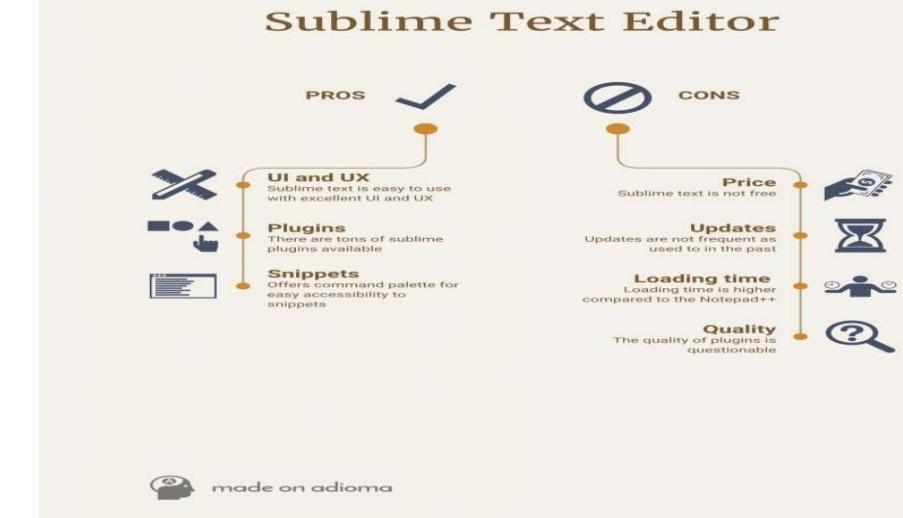


Different editors used for Webpage Development

SublimeText

Pros

- Performance: all the operations like opening, closing, searching is fast
- Package: lots of packages and themes
- Customize: looks nice, also has many theme to choose from, and able to configure using JSON file.
- Reliable: no need to do anything else once everything is installed and set up.

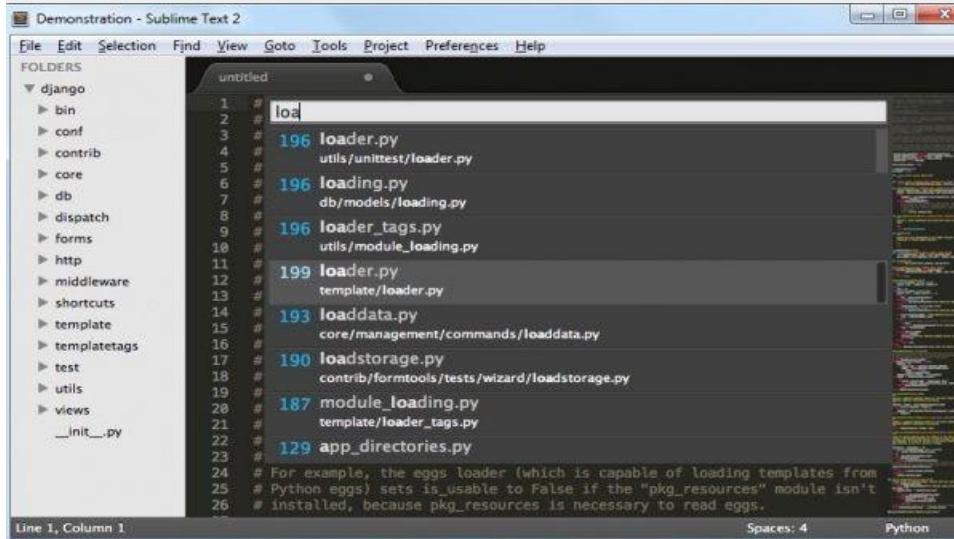


Different editors used for Webpage Development

SublimeText

Cons

- Cost: free download, but require \$70 license fee, otherwise frequent “buy license” pop-up.
- Package: package control is not installed ahead, so some searching needed; powerful only with plugins.
- Suggestions: only based on snippet, no context or language based suggestions.



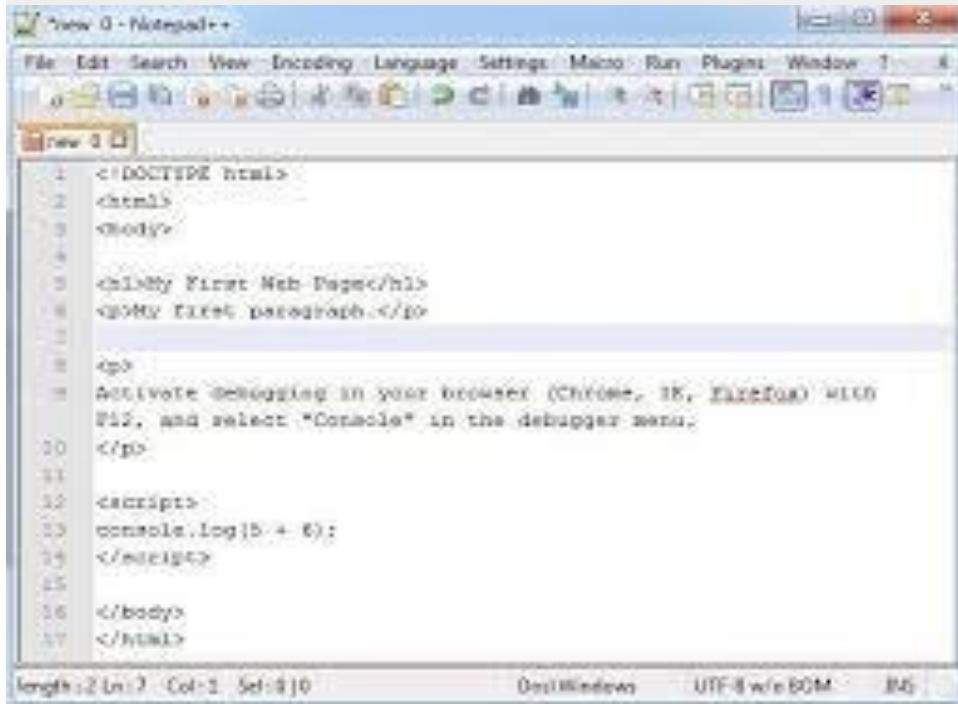
```
1 # load
2 # 196 loader.py
3 # utils/unittest/loader.py
4 # 196 loading.py
5 # db/models/loading.py
6 # 196 loader_tags.py
7 # utils/module_loading.py
8 #
9 #
10 #
11 #
12 #
13 #
14 #
15 #
16 #
17 #
18 #
19 #
20 #
21 #
22 #
23 #
24 # For example, the eggs loader (which is capable of loading templates from
25 # Python eggs) sets is_usable to False if the "pkg_resources" module isn't
26 # installed, because pkg_resources is necessary to read eggs.
```

Different editors used for Webpage Development

Notepad++

Features

- AutoSaved.
- Finding and replacing strings of text with regular expressions.
- Guided indentation.
- Line bookmarking.
- Macros.
- Split screen **editing** and synchronized scrolling.



The screenshot shows the Notepad++ interface with an open file named "New 3.html". The code in the editor is:

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h1>My First Web Page</h1>
6 <p>My First paragraph.</p>
7
8 <p>Activate debugging in your browser (Chrome, IE, Firefox) with
9 F12, and select "Console" in the debugger menu.
10 </p>
11
12 <script>
13 console.log(5 + 6);
14 </script>
15
16 </body>
17 </html>
```

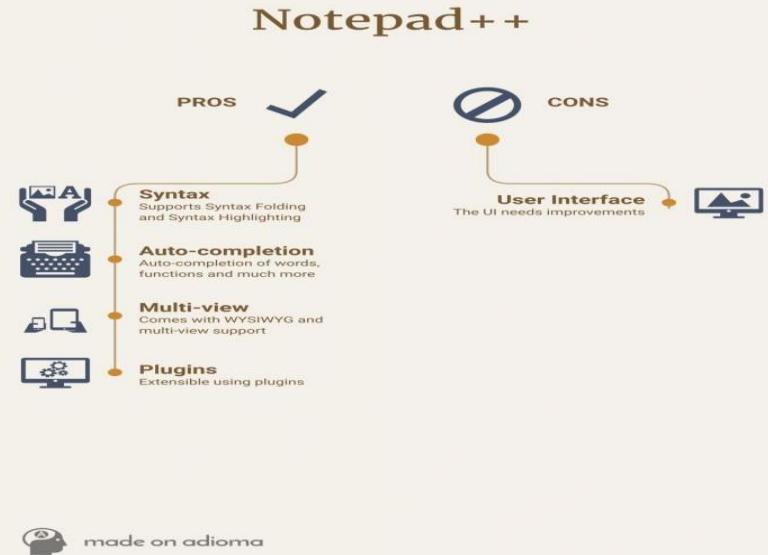
The status bar at the bottom shows: length:2 ln:7 Col:1 Sel:8|0

Different editors used for Webpage Development

Notepad++

Pros

- Light and fast
- Portable
- Free
- Collaborative editing

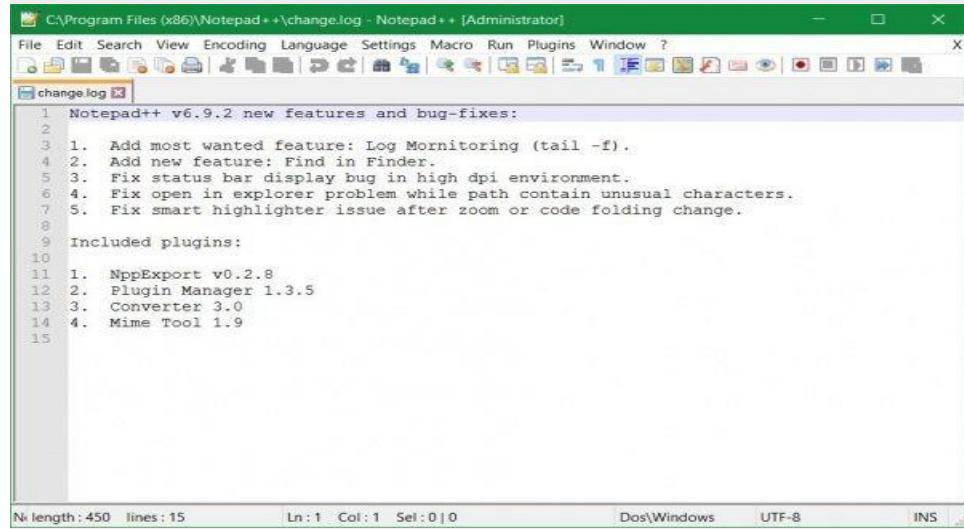


Different editors used for Webpage Developing

Notepad++

Cons

- Single platform support – Despite the software being as good as it can to be offered for free, it has limited support. Notepad++ is only available on Windows leaving out macOS and Linux.



A screenshot of the Notepad++ application window. The title bar reads "C:\Program Files (x86)\Notepad++\change.log - Notepad++ [Administrator]". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Macro, Run, Plugins, Window, and Help. The toolbar below the menu bar has various icons for file operations. The main text area displays a change log for Notepad++ v6.9.2. The log lists new features and bug-fixes, including Log Monitoring (tail -f), Find in Finder, and fixes for status bar display and explorer problems. It also lists included plugins: NppExport v0.2.8, Plugin Manager 1.3.5, Converter 3.0, and Mime Tool 1.9. The status bar at the bottom shows "length: 450 lines: 15 Ln:1 Col:1 Sel:0|0 Dos\Windows UTF-8 INS".

```
1 Notepad++ v6.9.2 new features and bug-fixes:
2
3 1. Add most wanted feature: Log Mornitoring (tail -f).
4 2. Add new feature: Find in Finder.
5 3. Fix status bar display bug in high dpi environment.
6 4. Fix open in explorer problem while path contain unusual characters.
7 5. Fix smart highlighter issue after zoom or code folding change.
8
9 Included plugins:
10
11 1. NppExport v0.2.8
12 2. Plugin Manager 1.3.5
13 3. Converter 3.0
14 4. Mime Tool 1.9
15
```

Different editors used for Webpage Developing

Different editors used for Web Page
Comparison



	CoffeeCup HTML Editor	ActiveState Komodo Edit	NetBeans IDE	Notepad++	Visual Studio Code
Free	✓	✓	✓	✓	✓
Open Source		✓	✓		
Number of Users	1 user	1 user	1 user	1 user	1 user
Upgrade Cost	\$29 once	\$7 monthly	Free	Free	Not available
Support	Online	Online	Online	Forum	Online
Text Editor	✓	✓	✓	✓	✓
WYSIWYG Editor	✓		With plugin		
Training	Documentation	Documentation	Documentation	Documentation	Documentation
Deployment	Installed (Windows)	Installed (Mac & Windows)	Cloud, SaaS, Web	Installed (Windows)	Installed (Mac & Windows)



Application of HTML



Image Source

<https://support.modernretail.com/hc/en-us/articles/201127998-W3C-Markup-Valid>

Application of HTML

How its industry using HTML?

- The **World Wide Web Consortium (W3C)** is an international community where Member organizations, full-time staff and the general public work together to develop Web standards.

The best known and widely used Web standards are:

- HTML (HyperText Markup Language)
- XML (eXtensible Markup Language)
- CSS (Cascading Style Sheets)



Image Source

<https://support.modernretail.com/hc/en-us/articles/201127998-W3C-Markup-Validation-Service>

Application of HTML

How its industry using HTML?

- Web Design and Applications
- Web of Devices
- Web Architecture
- Semantic Web
- XML Technology
- Web Services
- Browsers and Authoring Tools

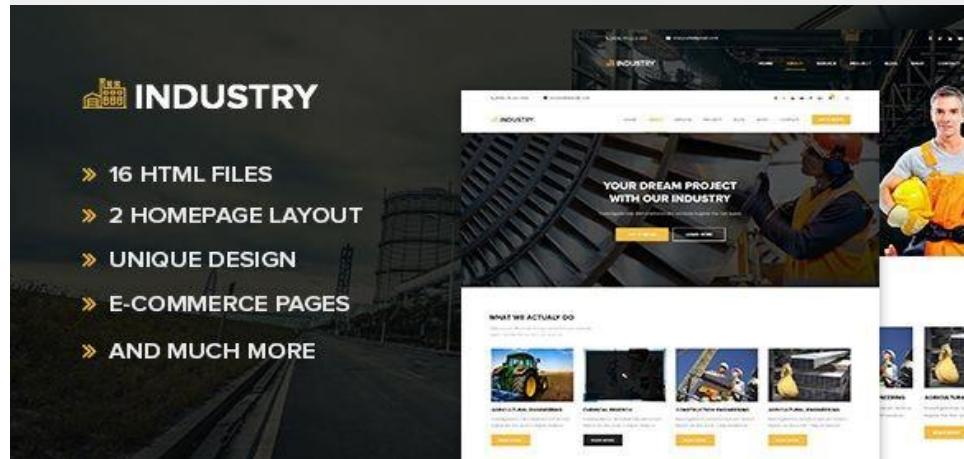
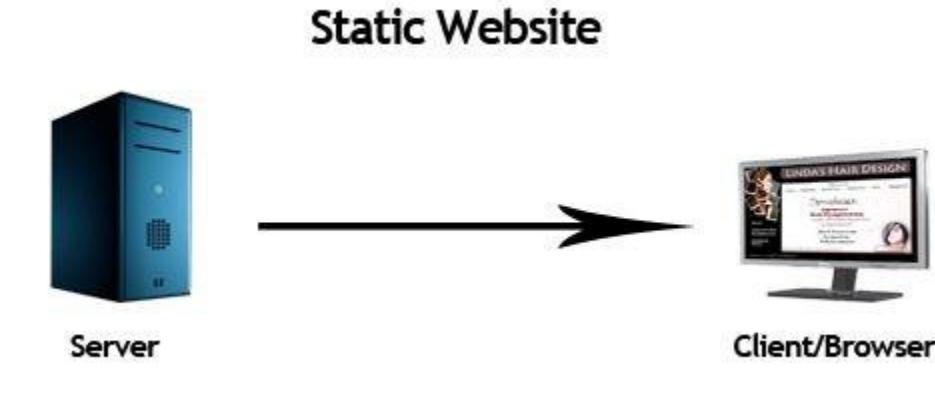


Image Source: <https://themeforest.net/item/industry-minimal-factory-industry-html-template/20142043>

Application of HTML

What is Static Website?

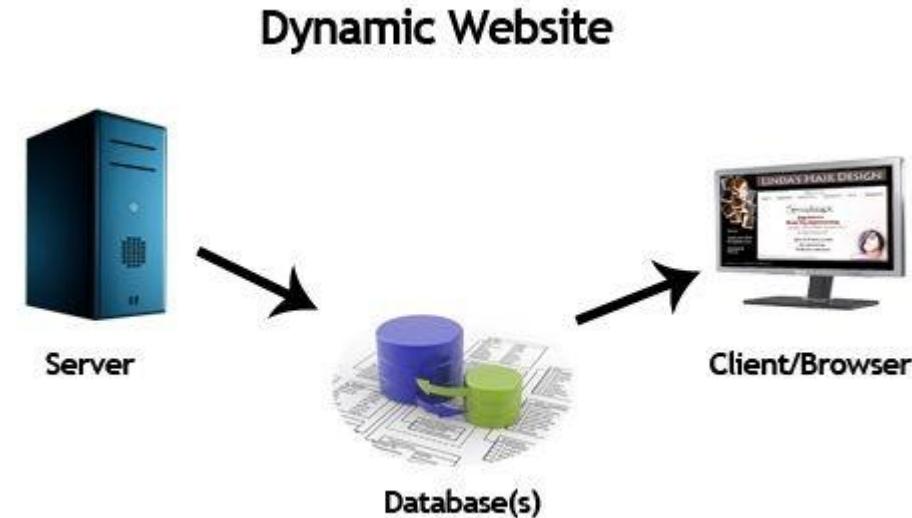
- Website is a collection of related web pages that may contain text, images, audio and video.
- A **static website** contains **Web** pages with fixed content
- A **static** site can be built by simply creating a few HTML pages and publishing them to a **Web** server.



Application of HTML

What is Dynamic Website?

- Dynamic website is a collection of dynamic web pages whose content changes dynamically.
- It accesses content from a database or Content Management System (CMS).
- Dynamic website uses client-side scripting or server-side scripting, or both to generate dynamic content.



Application of HTML

Practical Application

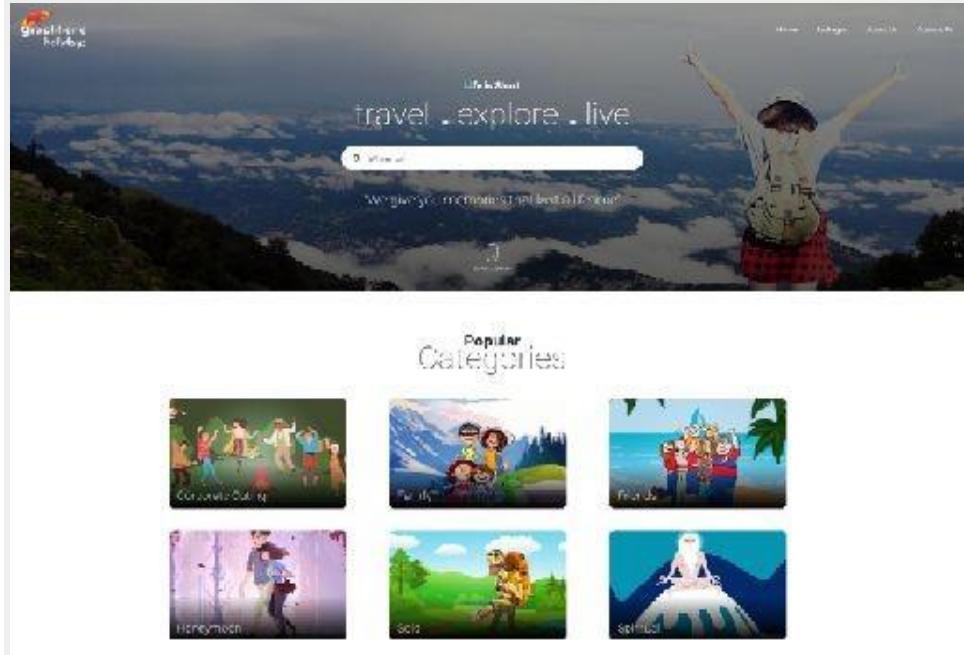
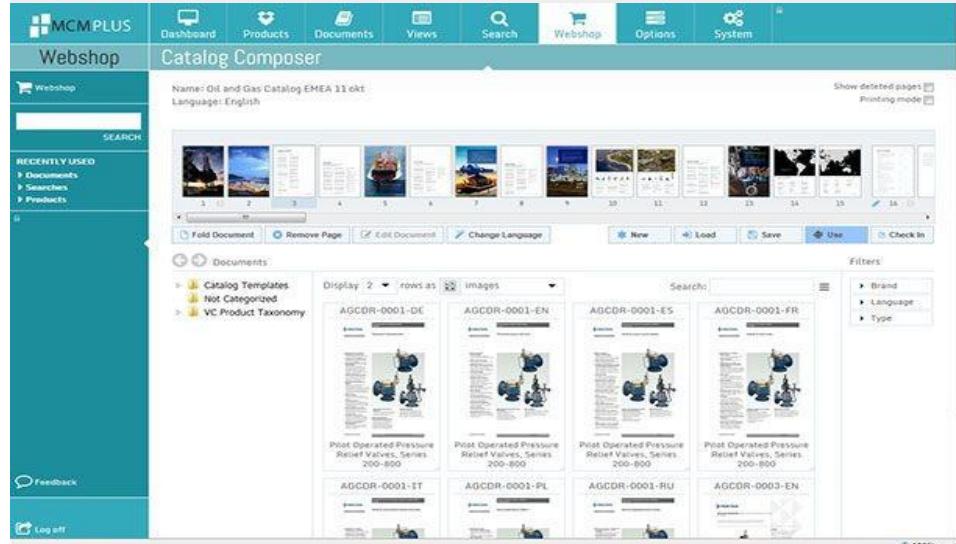


Image Source: <https://www.graphheneinfotech.com/blog/static-website-vs-dynamic-website/>

Application of HTML

Top 10 Uses of HTML

- Web pages development
- Web document creation
- Internet navigation
- Cutting edge feature
- Responsive images on web pages



Application of HTML

Top 10 Uses of HTML

- Client-side storage
 - Offline capabilities usage
 - Data Entry support with HTML
 - Game development usage
 - Native APIs usage to enrich website



Able to Create Styles of web pages using CSS

(25 hours)

In this section, we will discuss:

- Introduction to CSS
- Limitations of CSS
- Advantages of CSS
- CSS Syntax
- Three ways to integrate CSS
- Merits and demerits of -external Style Sheets,, Embedded Style Sheets

Introduction to CSS

CSS

- CSS stands for Cascading Style Sheets.
- It is the language for describing the presentation of Web pages, including colours, layout, and fonts, thus making our web pages presentable to the users.



Introduction to CSS

History of CSS

- CSS was first proposed by Hakon Wium Lie on October 10, 1994
- CSS was proposed in 1994 as a web styling language, to solve some of the problems of HTML 4

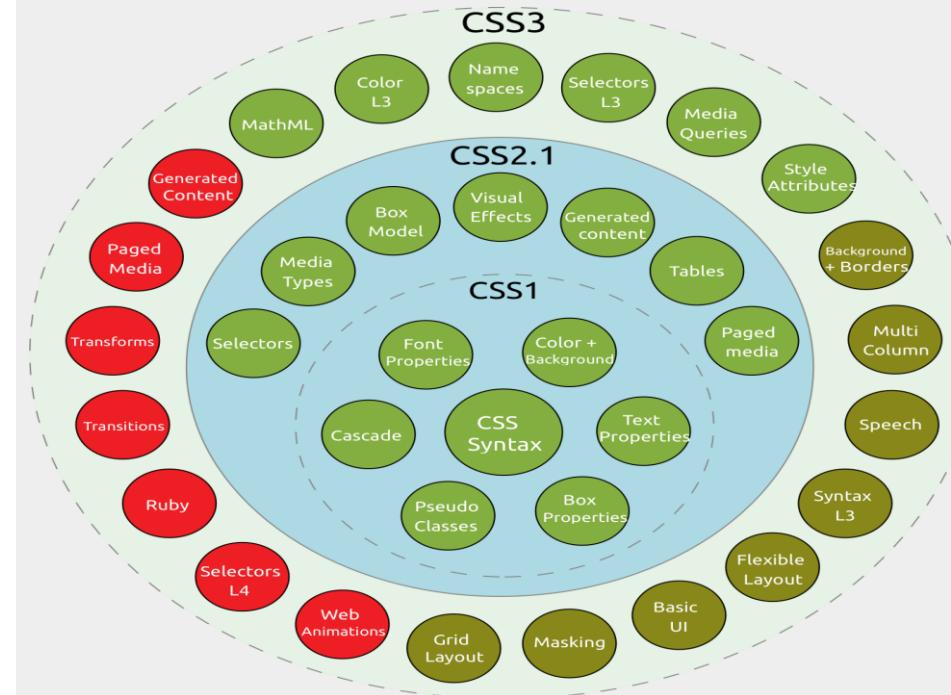


Image Source::

https://miro.medium.com/max/2768/1*7V_zawxy3_kZbHs2d6NT9w.png

Introduction to CSS

Why CSS?

- CSS saves time
- Easy Maintenance
- Search Engines
- Superior styles to HTML
- Offline Browsing

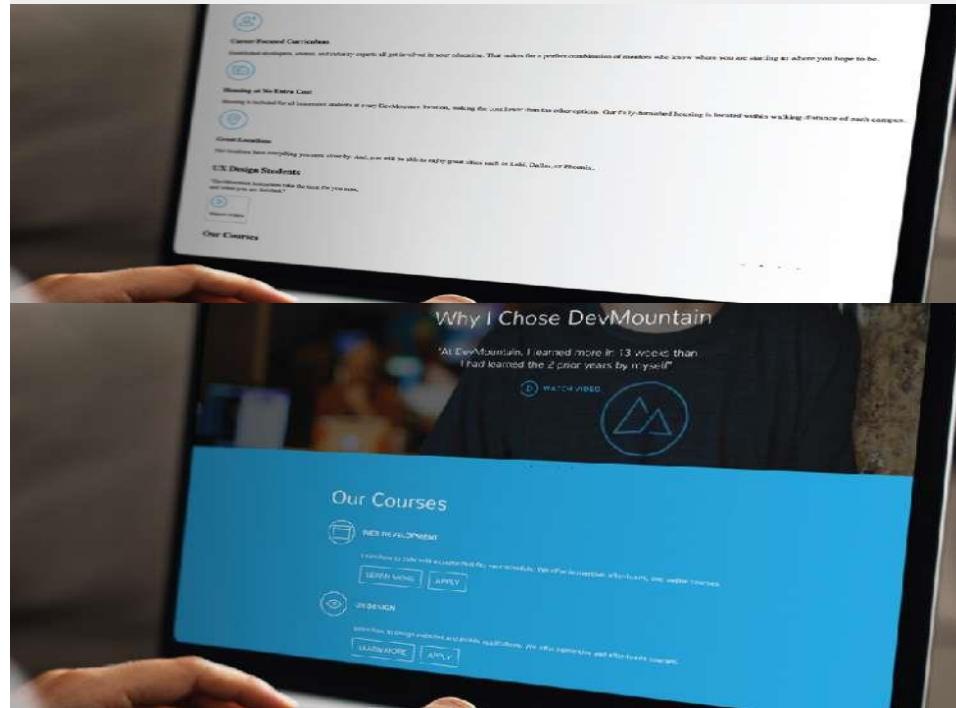


Image Source:

<https://blog.devmountain.com/what-is-css-and-why-use-it/>

Limitations of CSS

Limitations

- Confusion due to many CSS Versions
- Cross-Browser Issues
- Security Issues
- Extra Work for Developers

- **Ascending by selectors is not possible**
- **Limitations of vertical control**
- **No expressions**
- **No column declaration**
- **Pseudo-class not controlled by dynamic behavior**
- **Rules, styles, targeting specific text not possible**

Advantages of CSS

Advantages

- CSS saves time
- Pages load faster
- Superior styles to HTML
- Multiple Device Compatibility
- Global web standards
- Offline Browsing
- Platform Independence

- Better website speed
- Easier to maintain
- Consistent design
- Time-saving
- Better device compatibility
- Flexible positioning of design elements

Image Source:

https://d8it4huxumps7.cloudfront.net/bites/wp-content/banners/2021/10/616fee11ce1e_advantages_and_disadvantages_of_css.png

CSS Syntax

Syntax

3 Elements to a CSS Statement

- Selector
- Property
- Value

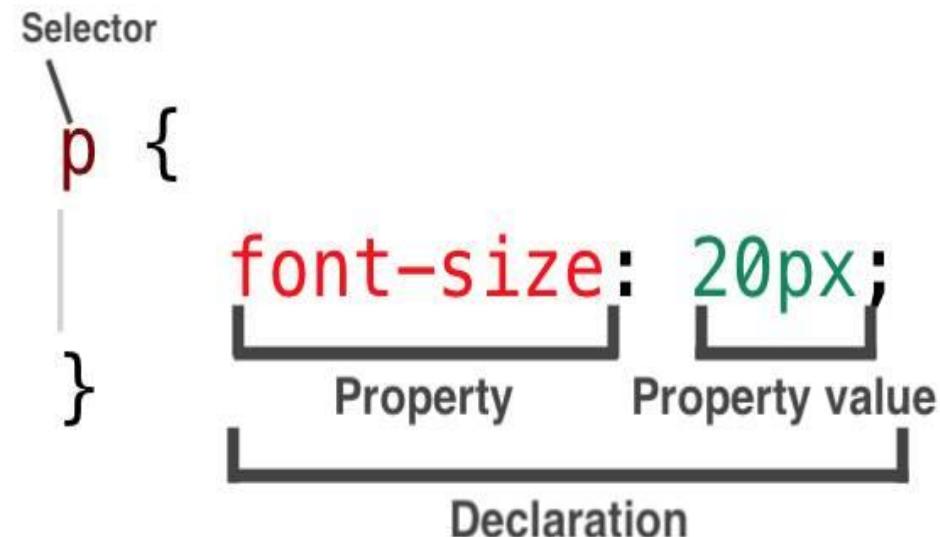


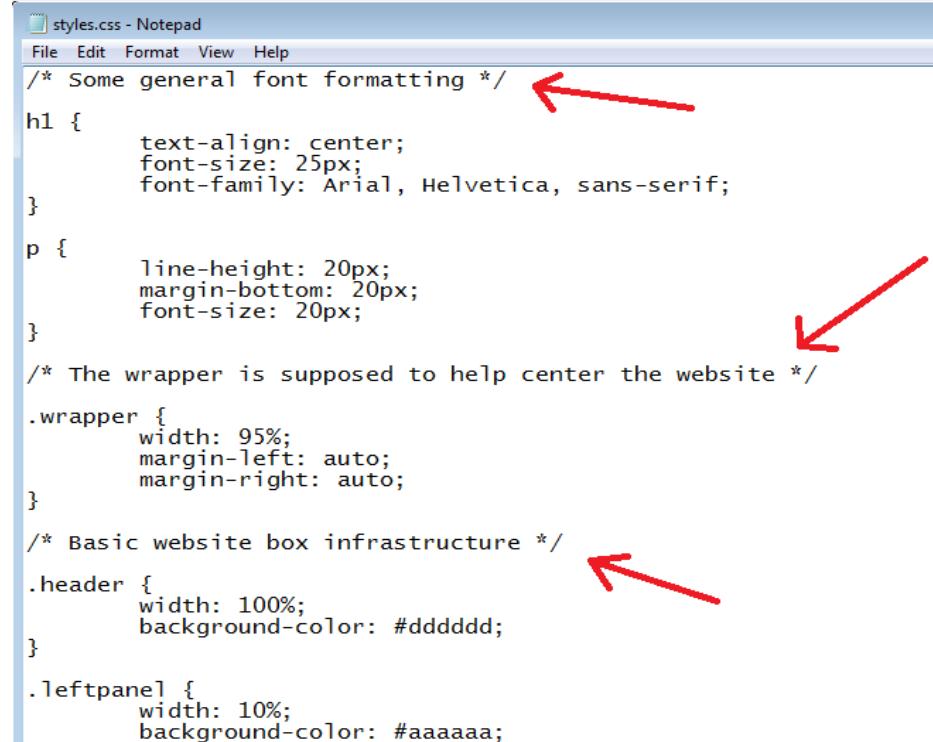
Image Source:

<https://codebrainer.azureedge.net/images/what-is-css-declaration.jpg>

CSS Syntax

CSS Comments

- Comments are used to explain the code, and may help when you edit the source code at a later date.
- Comments are ignored by browsers.



```
styles.css - Notepad
File Edit Format View Help
/* Some general font formatting */
h1 {
    text-align: center;
    font-size: 25px;
    font-family: Arial, Helvetica, sans-serif;
}

p {
    line-height: 20px;
    margin-bottom: 20px;
    font-size: 20px;
}

/* The wrapper is supposed to help center the website */
.wrapper {
    width: 95%;
    margin-left: auto;
    margin-right: auto;
}

/* Basic website box infrastructure */
.header {
    width: 100%;
    background-color: #dddddd;
}

.leftpanel {
    width: 10%;
    background-color: #aaaaaa;
```

Image Source:

https://www.frezenet.ca/wp-content/uploads/2019/03/CSS_9_1.png

CSS Syntax

White Spaces in CSS

- White spaces are special characters that can be an actual space, tab, or newline (carriage return).
- These whitespaces are used to construct your stylesheets extra readable.

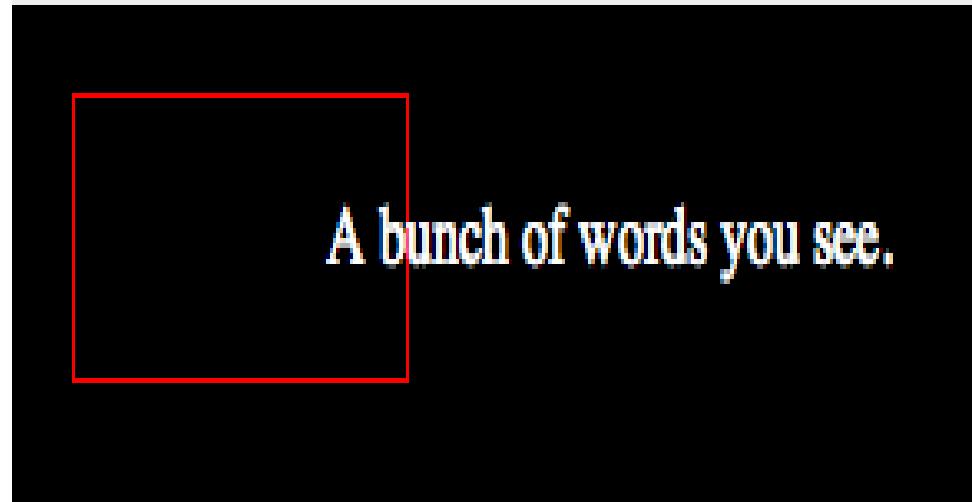


Image Source:

<https://i0.wp.com/css-tricks.com/wp-content/uploads/2011/09/pre.png?resize=290%2C97>

Three ways to integrate CSS

Types of CSS

- **Inline style sheet**
- **Internal style sheet**
- **External style sheet**

Inline CSS

```
<p style="color: blue;">This is a paragraph.</p>
```

Internal CSS

```
<head>
  <style type = text/css>
    body {background-color: blue;}
    p { color: yellow;}
  </style>
</head>
```

External CSS

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

Types of CSS

Inline Style

- Inline styles are placed within an HTML element in the code.
- Inline styles do not have selectors because its written inside the html element.

```
<p style = "color:#009900; font-size:50px;  
text-align:center;">
```

Inline

```
</p>
```

Types of CSS

Internal Style

- An internal CSS is used to define a style for a single HTML page.
- An internal CSS is defined in the `<head>` section of an HTML page, within a `<style>` element.

```
<head>
<style>
p { background-color: Red;
  font-family: serif;
  font-color: White; }
</style>
</head>

<body>
<p>Embedded content</p>
</body>
```

Types of CSS

External Style

- External Styles can be reused to apply on more than one page by only linking the style sheet to the web page.

```
example.css
body {
    background-color: lightblue;
}

h1 {
    color: navy;
    margin-left: 20px;
}

<head>
<link rel="stylesheet" href="example.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
```

Merits and demerits of - external Style Sheets, Embedded Style Sheets

External Style Sheet

Merits

- one change to the style sheet will change all linked pages
- consistent look and feel across multiple web pages

Demerits

- To render the document, the external style sheet should be loaded.

Advantages:

- Provides consistency between pages
- “Single” point of maintenance
- Easier to share with other sites

Disadvantages:

- Slower page load time

Merits and demerits of - external Style Sheets, Embedded Style Sheets

Embedded Style Sheets

Merits

- Multiple tag types can be created in a single document.
- Extra download is unnecessary.

Demerits

- Multiple documents cannot be controlled.

Disadvantages of Internal CSS

- It only effects one page
- It will increase the load time for the page

CSS Values and Units

- CSS has several different units for expressing a length. Many CSS properties take "length" values, such as width, margin, padding, font-size, etc.
- Length is a number followed by a length unit, such as 10px, 2em, etc.

Example :

```
h1 {  
    font-size: 60px;  
}
```

```
p {  
    font-size: 25px;  
    line-height: 50px;  
}
```

Absolute Lengths

- The absolute length units are fixed and a length expressed in any of these will appear as exactly that size.
- Absolute length units are not recommended for use on screen, because screen sizes vary so much. However, they can be used if the output medium is known, such as for print layout.

Unit	Description
cm	centimeters
mm	millimeters
in	inches (1in = 96px = 2.54cm)
px*	pixels (1px = 1/96th of 1in)
pt	points (1pt = 1/72 of 1in)
pc	picas (1pc = 12 pt)

Relative Lengths

Relative length units specify a length relative to another length property. Relative length units scale better between different rendering medium.

Unit	Description
em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to the width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimension
vmax	Relative to 1% of viewport's* larger dimension
%	Relative to the parent element

CSS Styling Text

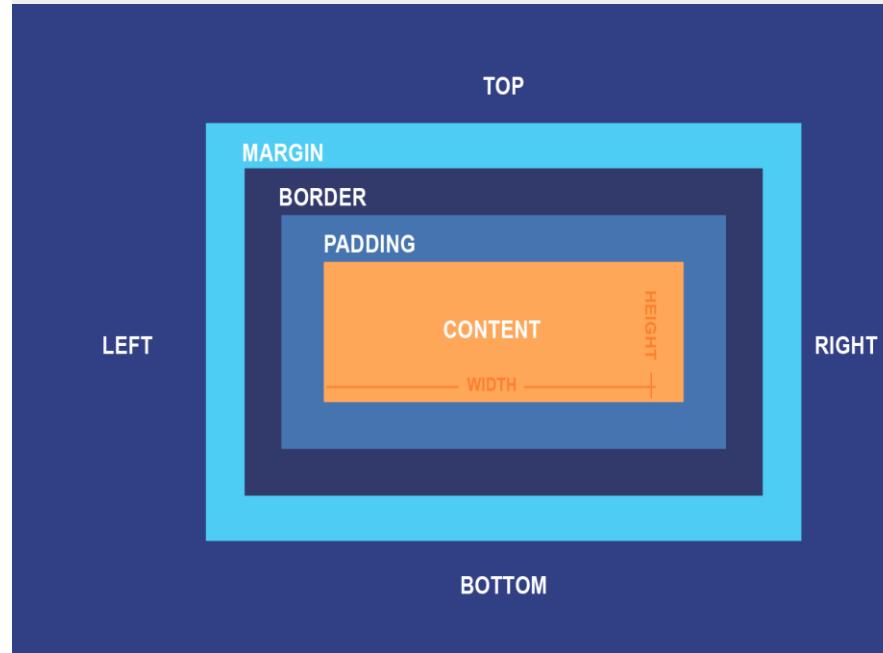
Text Formatting

This text is styled with some of the text formatting properties. The heading uses the text-align, text-transform, and color properties. The paragraph is indented, aligned, and the space between characters is specified.

- Text Color
- Text Color and Background Color
- CSS Text Alignment and Text Direction
- Text Alignment
- Text Direction
- Text Transformation
- CSS Text Indentation,
- Letter Spacing,
- Line Height,
- Word Spacing, and White Space
- Text Shadow

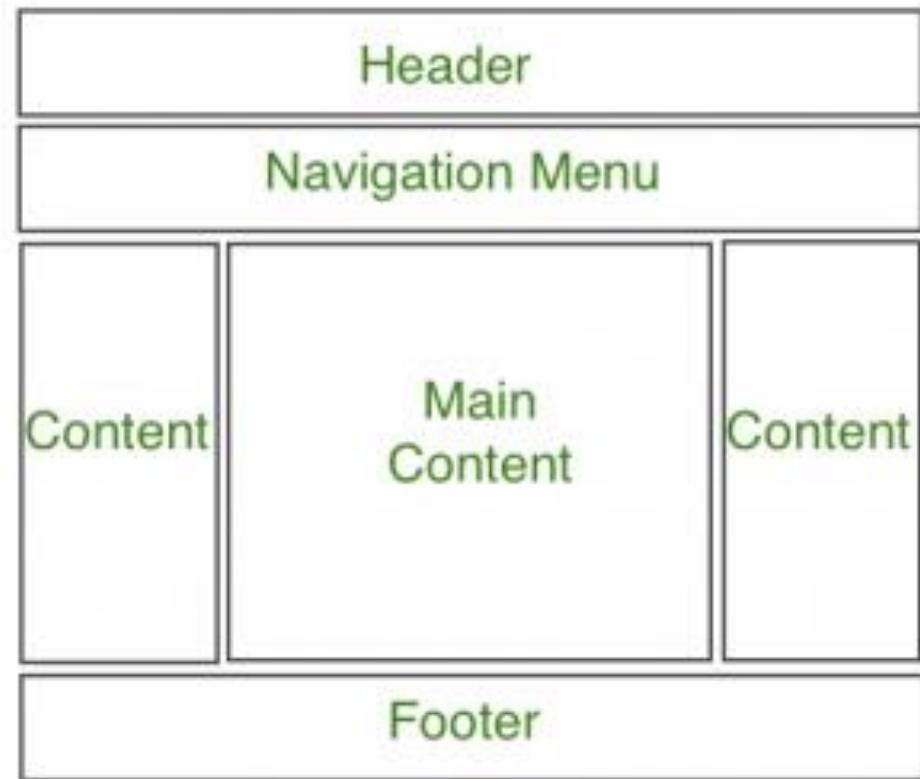
CSS Box Model

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.



Website Layout

A website can be divided into various sections comprising of header, menus, content and footer based on which there are many different layout designs available for developer. Different layouts can be created by using div tag and use CSS property to style it.



JavaScript

(30 hours)

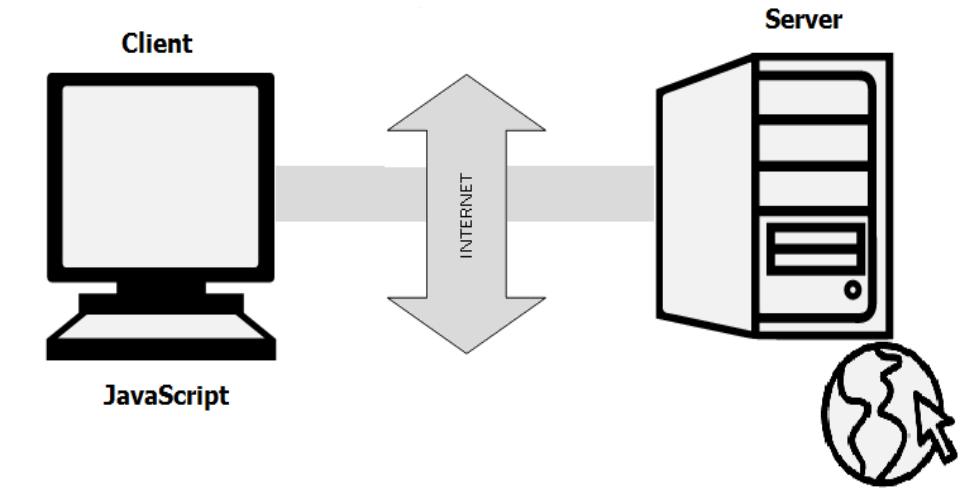
In this section, we will discuss:

- Introduction to JavaScript
- Javascript Datatype
- Javascript Condition
- JavaScript Array
- JavaScript Function

Introduction to JavaScript

What is JavaScript

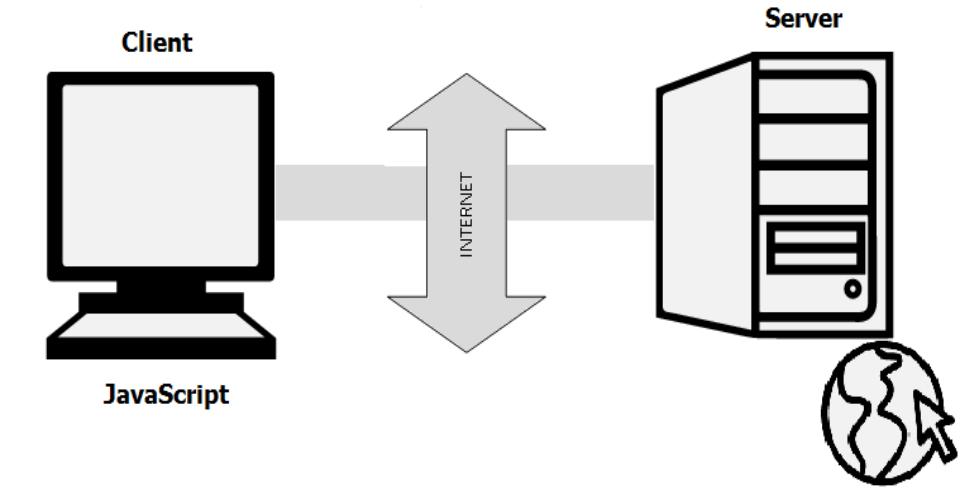
- JavaScript is a very powerful client-side scripting language.
- JavaScript is used mainly for enhancing the interaction of a user with the webpage.



Introduction to JavaScript

Why JavaScript

- JavaScript offers lots of flexibility.
- Mobile app development, desktop app development, and game development.
- With javascript you can find tons of frameworks and libraries already developed, which can be used directly in web development.



Introduction to JavaScript

What is JavaScript Used For

JavaScript is used in various fields from the web to servers:

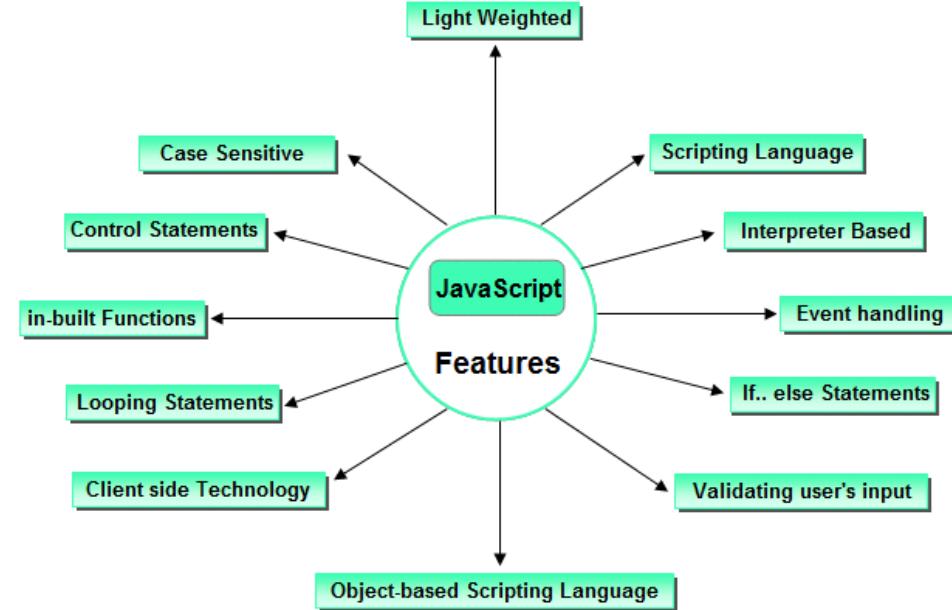
- Web Applications
- Mobile Applications
- Web-based Games
- Back-end Web Development



Introduction to JavaScript

Features of JavaScript

There are various features of javascript:



Introduction to JavaScript

Application of JavaScript

JavaScript is used to create interactive websites. It is mainly used for:

- Client-side validation,
- Dynamic drop-down menus,
- Displaying date and time,
- Displaying pop-up windows and dialog boxes.
- Displaying clocks etc.



Introduction to JavaScript

JavaScript Example

```
<script type="text/javascript">  
document.write("JavaScript is a simple  
language for javatpoint learners");  
</script>
```

JavaScript provides 3 places to put the
JavaScript code:

- within body tag,
- within head tag and
- external JavaScript file.

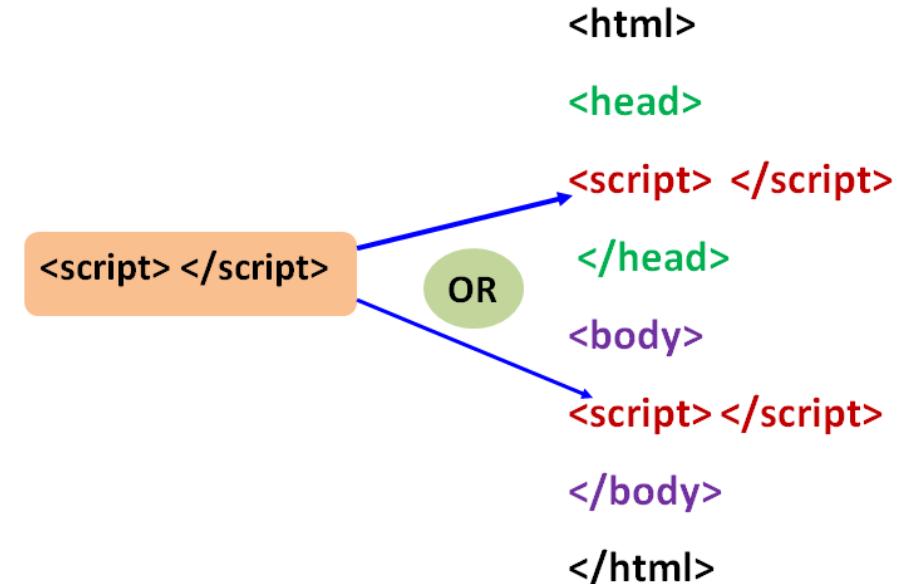


Introduction to JavaScript

JavaScript Syntax

JavaScript provides 3 places to put the JavaScript code:

- within body tag,
- within head tag and
- external JavaScript file.



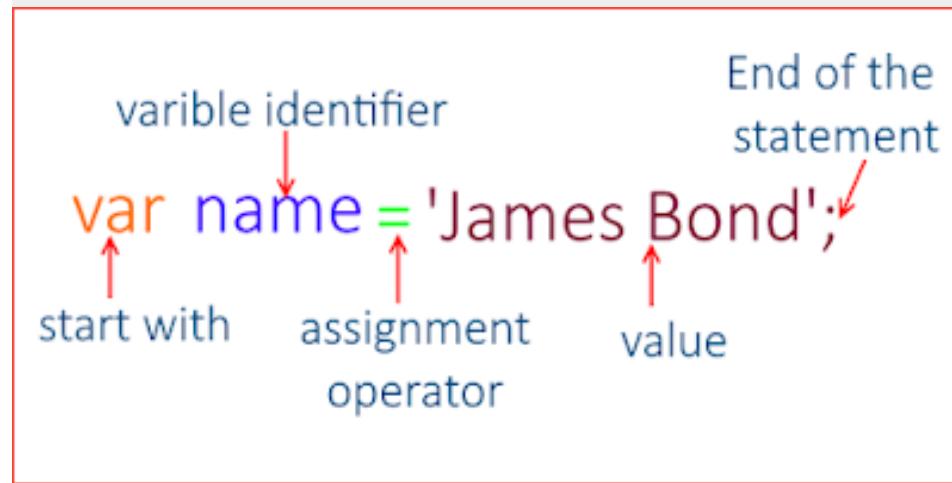
JavaScript Variables

What is Variable?

Variables are used to store data, like string of text, numbers, etc. The data or value stored in the variables can be set, updated, and retrieved whenever needed.

Example:

```
var name = "Peter Parker";
var age = 21;
var isMarried = false;
```



JavaScript Variables

Declaring Multiple Variables at Once

These are the following rules for naming a JavaScript variable:

- A variable name must start with a letter, underscore (_), or dollar sign (\$).
 - A variable name cannot start with a number.



JavaScript Variables

Declaring Multiple Variables at Once

- A variable name can only contain alphanumeric characters (A-z, 0-9) and underscores.
 - A variable name cannot contain spaces.
 - A variable name cannot be a JavaScript keyword or a JavaScript reserved word.



Javascript Datatype

Definition

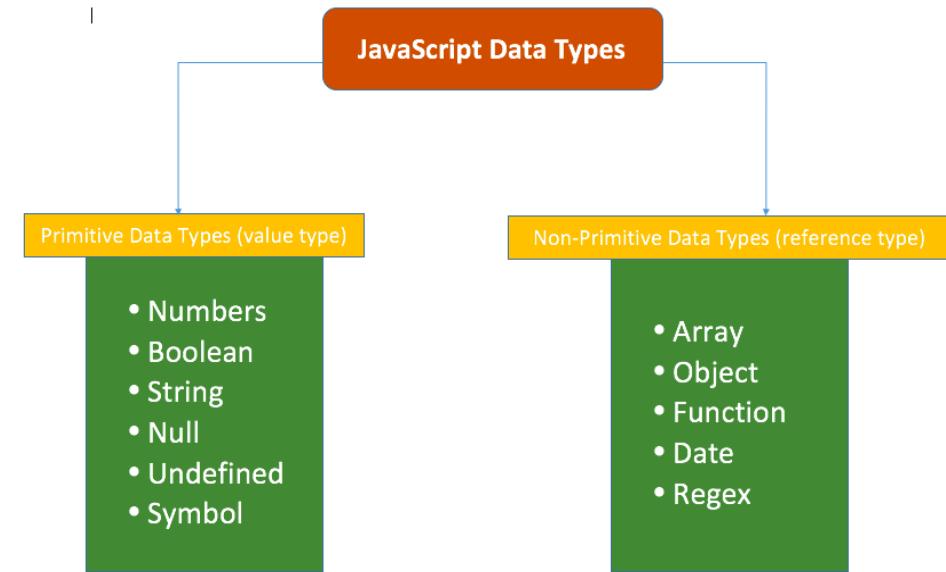
- JavaScript is dynamic and loosely typed language. It means you don't require to specify a type of a variable. A variable in JavaScript can be assigned any type of value, as shown in the following



Javascript Datatype

Types of Datatype

- Primitive Data Types
- Non-Primitive Data Types



Types of Datatype

Primitive Datatypes

- The primitive data types are the lowest level of the data value in JavaScript. The `typeof` operator can be used with primitive data types to know the type of a value.

Data Type	Description
String	String is a textual content wrapped inside '' or " " or ` ` (tick sign). Example: 'Hello World!', "This is a string", etc.
Number	Number is a numeric value. Example: 100, 4521983, etc.
BigInt	BigInt is a numeric value in the arbitrary precision format. Example: 453889879865131n, 200n, etc.
Boolean	Boolean is a logical data type that has only two values, true or false.
Null	A null value denotes an absence of value. Example: var str = null;
Undefined	undefined is the default value of a variable that has not been assigned any value. Example: In the variable declaration, var str;, there is no value assigned to str. So, the type of str can be check using <code>typeof(str)</code> which will return undefined.

Types of Datatype

Non-Primitive Datatypes

- The non-primitive data types contain some kind of structure with primitive data.

Data Type	Description
Object	An object holds multiple values in terms of properties and methods.
	Example: <code>var person = { firstName: "James", lastName: "Bond", age: 15 };</code>
Date	Date object represents date & time including days, months, years, hours, minutes, seconds and milliseconds.
	Example: <code>var today = new Date("25 July 2021");</code>
Array	An array stores multiple values using special syntax.
	Example: <code>var nums = [1, 2, 3, 4];</code>

Types of Datatype

Non-Primitive Datatypes

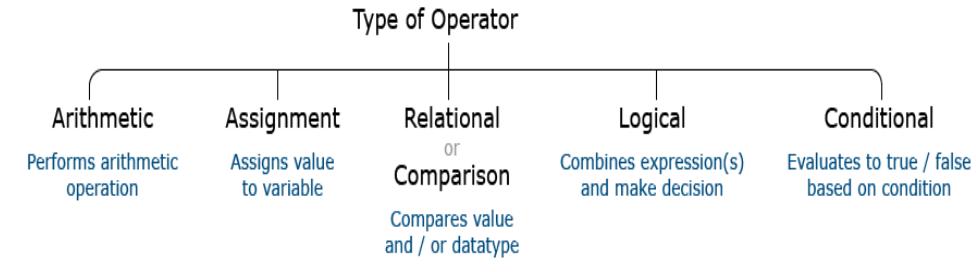
- The non-primitive data types contain some kind of structure with primitive data.

Data Type	Description
Object	An object holds multiple values in terms of properties and methods. Example: <code>var person = { firstName: "James", lastName: "Bond", age: 15 };</code>
Date	Date object represents date & time including days, months, years, hours, minutes, seconds and milliseconds. Example: <code>var today = new Date("25 July 2021");</code>
Array	An array stores multiple values using special syntax. Example: <code>var nums = [1, 2, 3, 4];</code>

Javascript Operators

What is JavaScript Operator

- An operator performs some operation on single or multiple operands (data value) and produces a result.
- Syntax:
<Left operand> operator <right operand>
<Left operand> operator



Types of Javascript Operators

Arithmetic Operators

- Arithmetic operators are used to perform arithmetic operations on the operands.

Operator	Description	Example
+	Addition	$10+20 = 30$
-	Subtraction	$20-10 = 10$
*	Multiplication	$10*20 = 200$
/	Division	$20/10 = 2$
%	Modulus (Remainder)	$20\%10 = 0$
++	Increment	<code>var a=10; a++; Now a = 11</code>
--	Decrement	<code>var a=10; a--; Now a = 9</code>

Types of Javascript Operators

Comparison Operators

- The JavaScript comparison operator compares the two operands.

Operator	Description	Example
<code>==</code>	Is equal to	<code>10==20 = false</code>
<code>==</code>	Identical (equal and of same type)	<code>10==20 = false</code>
<code>!=</code>	Not equal to	<code>10!=20 = true</code>
<code>!=</code>	Not Identical	<code>20!=20 = false</code>
<code>></code>	Greater than	<code>20>10 = true</code>
<code>>=</code>	Greater than or equal to	<code>20>=10 = true</code>
<code><</code>	Less than	<code>20<10 = false</code>
<code><=</code>	Less than or equal to	<code>20<=10 = false</code>

Types of Javascript Operators

Bitwise Operators

- The bitwise operators perform bitwise operations on operands.

Operator	Description	Example
&	Bitwise AND	$(10==20 \& 20==33) = \text{false}$
	Bitwise OR	$(10==20 20==33) = \text{false}$
^	Bitwise XOR	$(10==20 ^ 20==33) = \text{false}$
~	Bitwise NOT	$(\sim 10) = -10$
<<	Bitwise Left Shift	$(10<<2) = 40$
>>	Bitwise Right Shift	$(10>>2) = 2$
>>>	Bitwise Right Shift with Zero	$(10>>>2) = 2$

Types of Javascript Operators

Logical Operators

- The following operators are known as JavaScript logical operators.

Operator	Description	Example
<code>&&</code>	Logical AND	<code>(10==20 && 20==33) = false</code>
<code> </code>	Logical OR	<code>(10==20 20==33) = false</code>
<code>!</code>	Logical Not	<code>!(10==20) = true</code>

Types of Javascript Operators

Assignment Operators

- The following operators are known as JavaScript assignment operators.

Operator	Description	Example
=	Assign	$10+10 = 20$
+=	Add and assign	<code>var a=10; a+=20; Now a = 30</code>
-=	Subtract and assign	<code>var a=20; a-=10; Now a = 10</code>
=	Multiply and assign	<code>var a=10; a=20; Now a = 200</code>
/=	Divide and assign	<code>var a=10; a/=2; Now a = 5</code>
%=	Modulus and assign	<code>var a=10; a%=2; Now a = 0</code>

Types of Javascript Operators

Special Operators

- The following operators are known as JavaScript special operators.

Operator	Description
<code>(?:)</code>	Conditional Operator returns value based on the condition. It is like if-else.
<code>,</code>	Comma Operator allows multiple expressions to be evaluated as single statement.
<code>delete</code>	Delete Operator deletes a property from the object.
<code>in</code>	In Operator checks if object has the given property
<code>instanceof</code>	checks if the object is an instance of given type
<code>new</code>	creates an instance (object)
<code>typeof</code>	checks the type of object.
<code>void</code>	it discards the expression's return value.
<code>yield</code>	checks what is returned in a generator by the generator's iterator.

Javascript Condition

JavaScript If-else

- The JavaScript if-else statement is used to execute the code whether condition is true or false. There are three forms of if statement in JavaScript.
 - i. If Statement
 - ii. If else statement
 - iii. if else if statement

Operator	Description
<code>(?:)</code>	Conditional Operator returns value based on the condition. It is like if-else.
<code>,</code>	Comma Operator allows multiple expressions to be evaluated as single statement.
<code>delete</code>	Delete Operator deletes a property from the object.
<code>in</code>	In Operator checks if object has the given property
<code>instanceof</code>	checks if the object is an instance of given type
<code>new</code>	creates an instance (object)
<code>typeof</code>	checks the type of object.
<code>void</code>	it discards the expression's return value.
<code>yield</code>	checks what is returned in a generator by the generator's iterator.

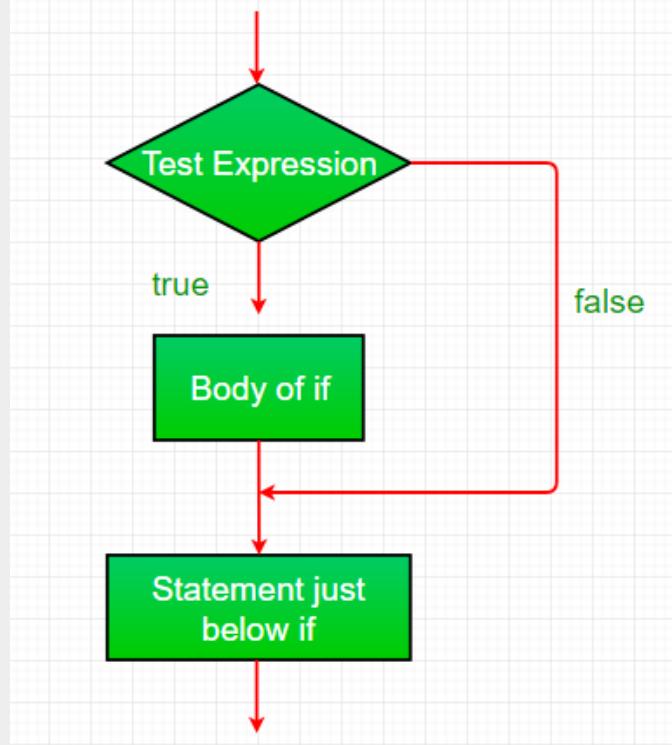
JavaScript If-else

If statement

It evaluates the content only if expression is true.

Syntax:

```
if(expression)
{
  //content to be evaluated
}
```



JavaScript If-else

If statement

```
<script>  
var a=20;  
if(a>10){  
document.write("value of a is greater than  
10");  
}  
</script>
```

Output :

value of a is greater than 10

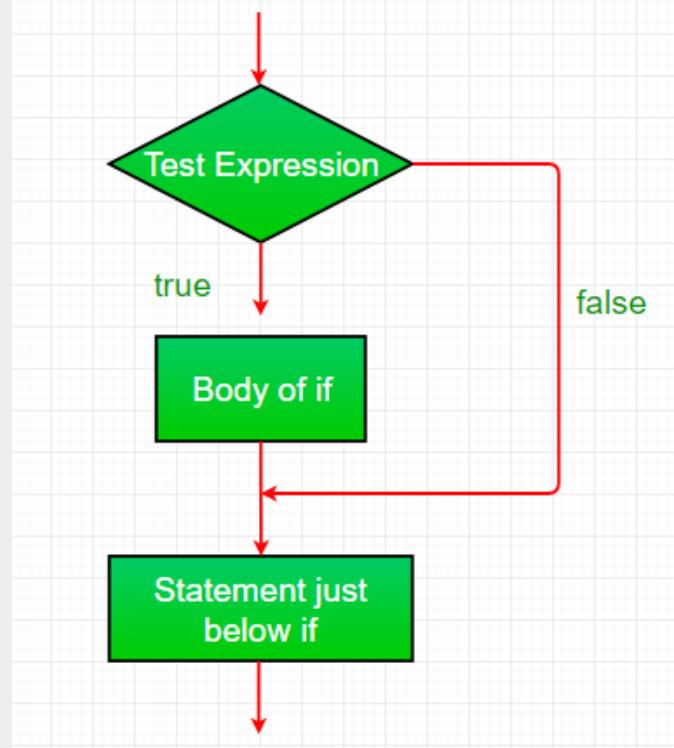


Image Source: <https://media.geeksforgeeks.org/wp-content/uploads/if.png>

JavaScript If-else

If...else Statement

It evaluates the content whether condition is true or false.

Syntax

```
if(expression){  
//content to be evaluated if condition is true  
}  
else{  
//content to be evaluated if condition is false  
}
```

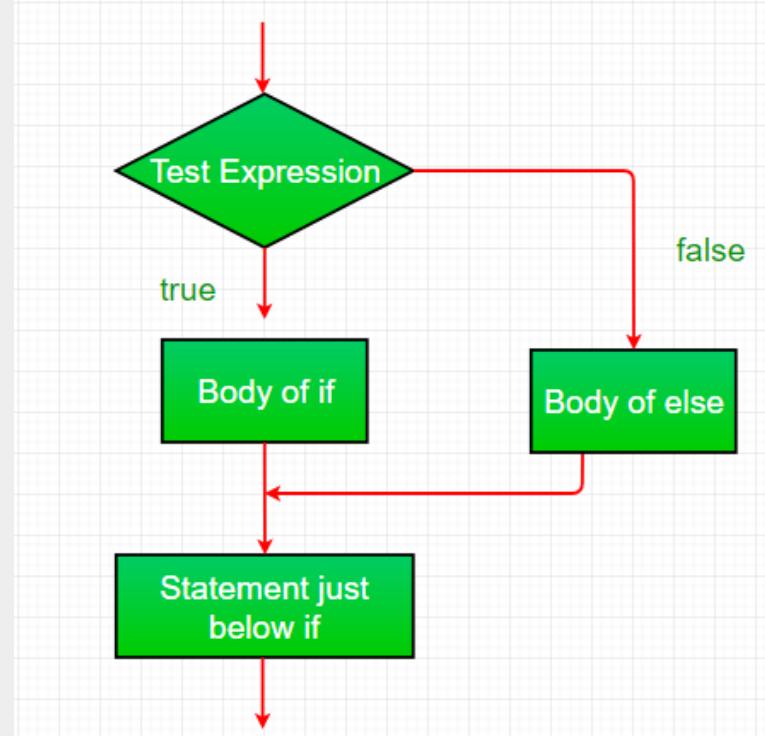


Image Source: <https://media.geeksforgeeks.org/wp-content/uploads/if-else.png>

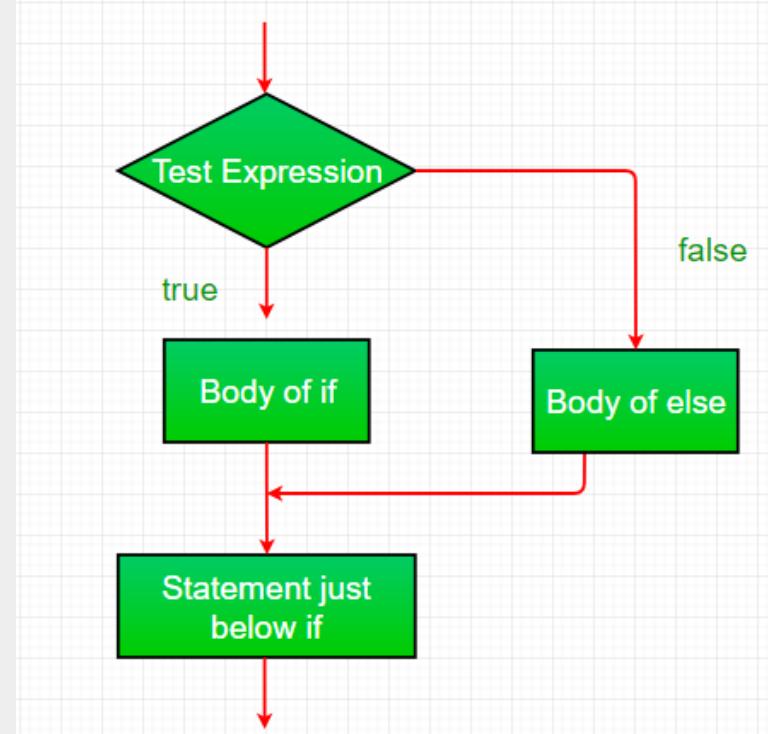
JavaScript If-else

If...else Statement

```
<script>
var a=20;
if(a%2==0){
document.write("a is even number"); }
else{
document.write("a is odd number"); }
</script>
```

Output :

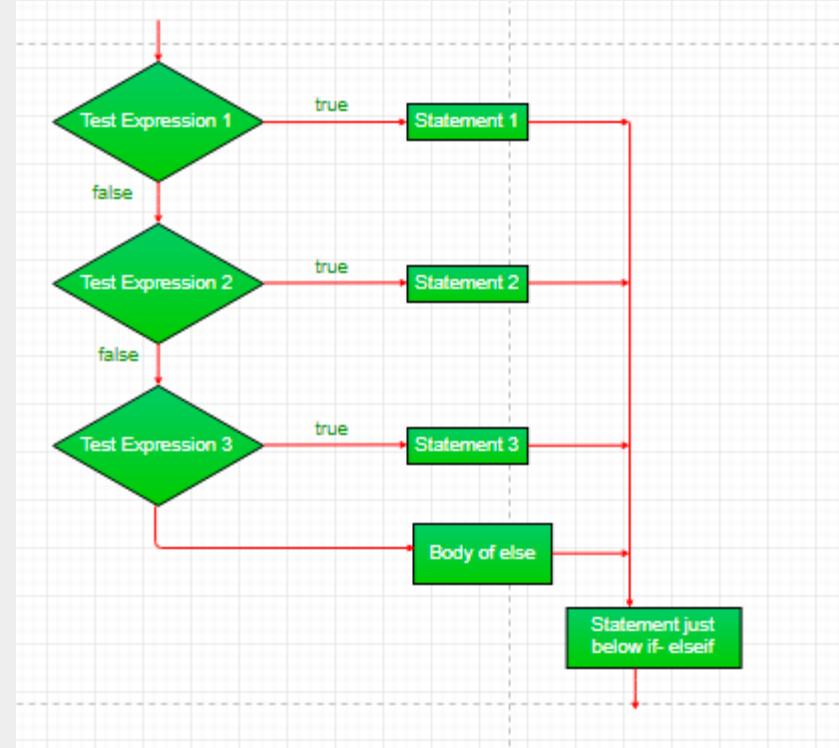
a is even number



JavaScript If-else

If...else if statement

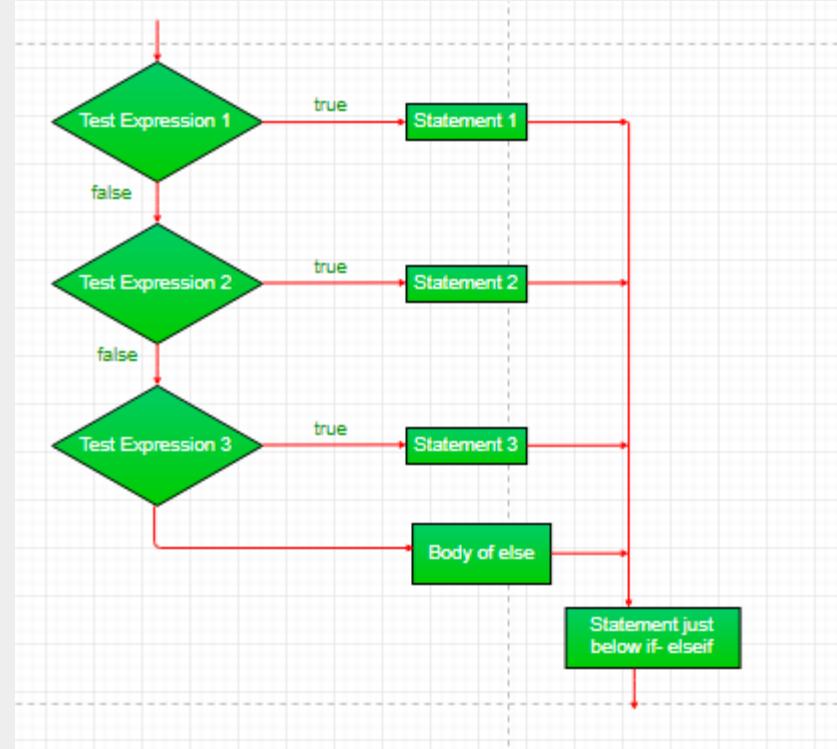
It evaluates the content only if expression is true from several expressions.



JavaScript If-else

If...else if statement

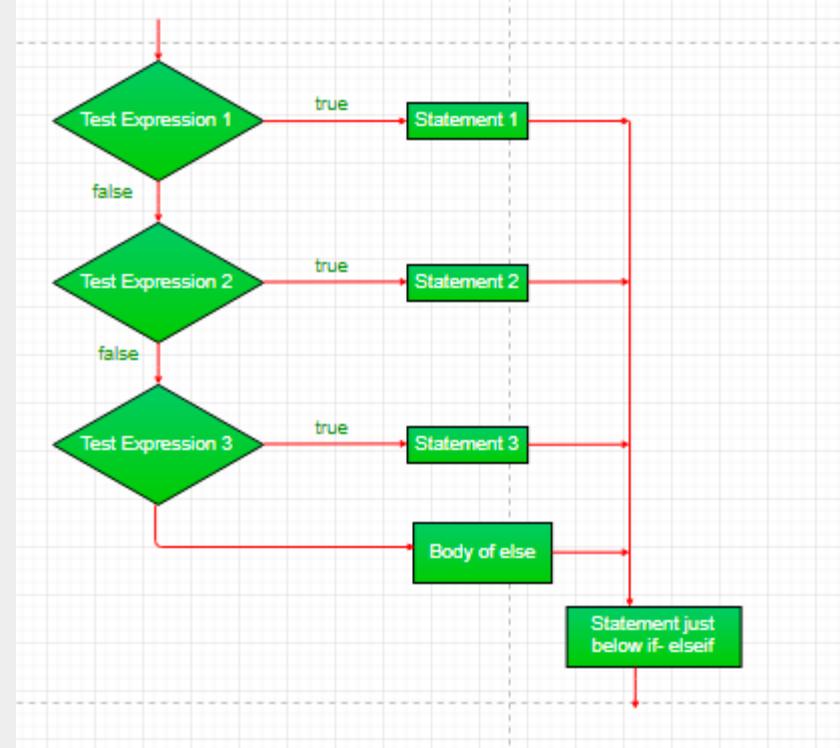
```
if(expression1){  
//content to be evaluated if expression1 is  
true }  
else if(expression2)  
{ //content to be evaluated if expression2 is  
true }  
else if(expression3)  
{ //content to be evaluated if expression3 is  
true }  
else{ //content to be evaluated if no  
expression is true }
```



JavaScript If-else

If...else if statement

```
<script>  
var a=20;  
if(a==10){  
document.write("a is equal to 10");  
}  
else if(a==15){  
document.write("a is equal to 15");  
}
```

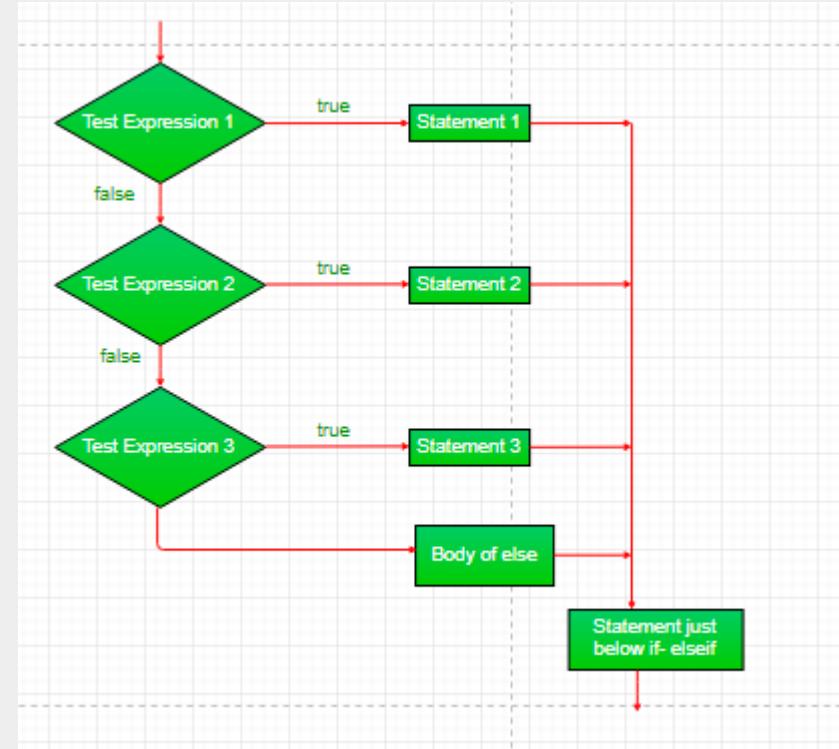


JavaScript If-else

If...else if statement

```
else if(a==20){  
    document.write("a is equal to 20");  
}  
else{  
    document.write("a is not equal to 10, 15 or  
    20");  
}  
</script>
```

Output of the above example
a is equal to 20



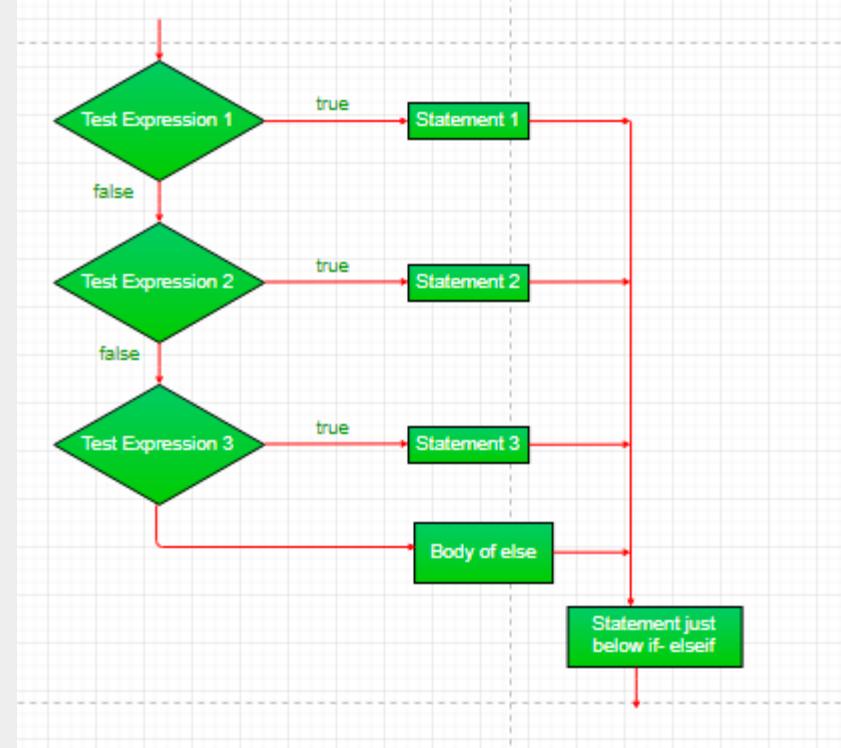
JavaScript If-else

If...else if statement

```
else if(a==20){  
document.write("a is equal to 20");  
}  
else{  
document.write("a is not equal to 10, 15 or  
20");  
}  
</script>
```

Output :

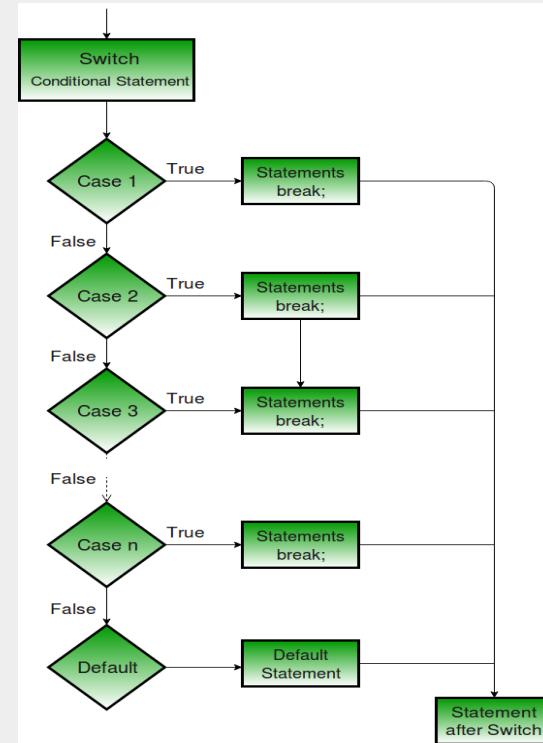
a is equal to 20



Javascript Condition

JavaScript Switch

- The JavaScript switch statement is used to execute one code from multiple expressions.
- It is just like else if statement that we have learned in previous page. But it is convenient than if..else..if because it can be used with numbers, characters etc.

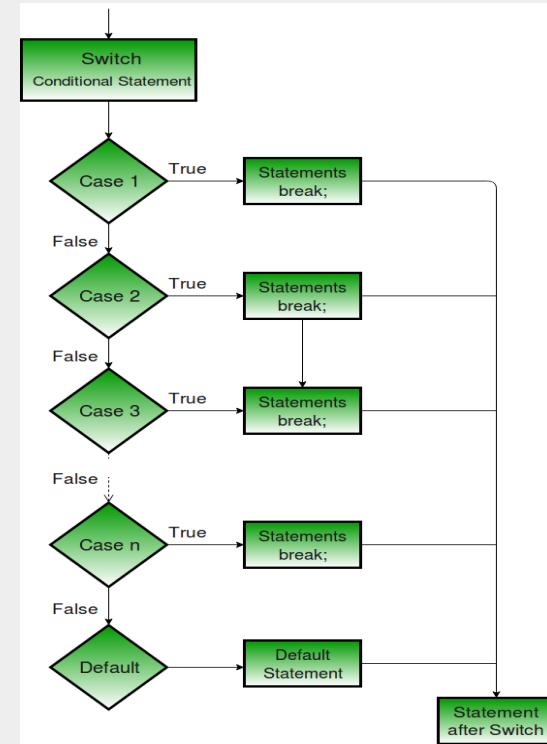


Javascript Condition

JavaScript Switch

Syntax:

```
switch(expression){  
  case value1:  
    code to be executed;  
    break;  
  case value2:  
    code to be executed;  
    break;  
  .....  
  default:
```

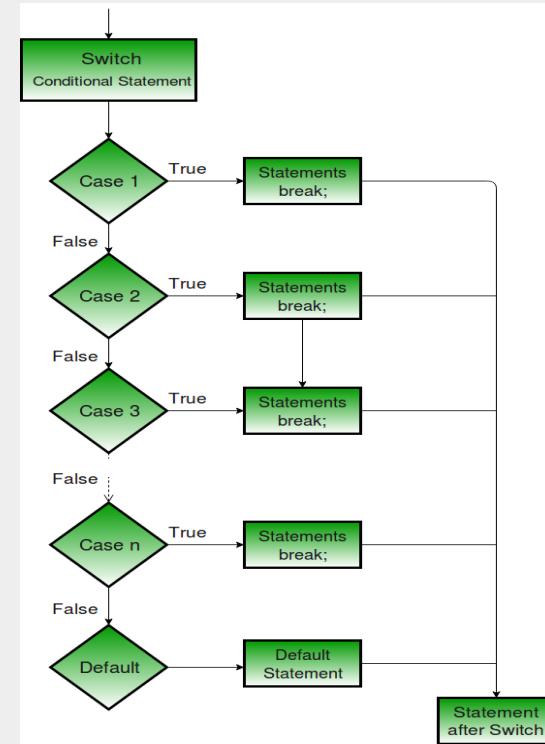


Javascript Condition

JavaScript Switch

Syntax:

```
switch(expression){  
  case value1:  
    code to be executed;  
    break;  
  case value2:  
    code to be executed;  
    break;  
  .....  
  default:
```

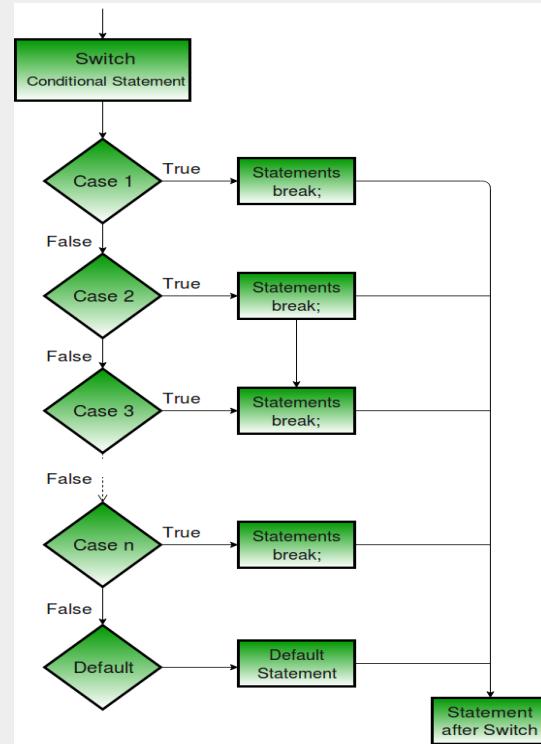


Javascript Condition

JavaScript Switch

Example:

```
<script>
var grade='B';
var result;
switch(grade){
case 'A':
result+=" A Grade";
case 'B':
result+=" B Grade";
case 'C':
```



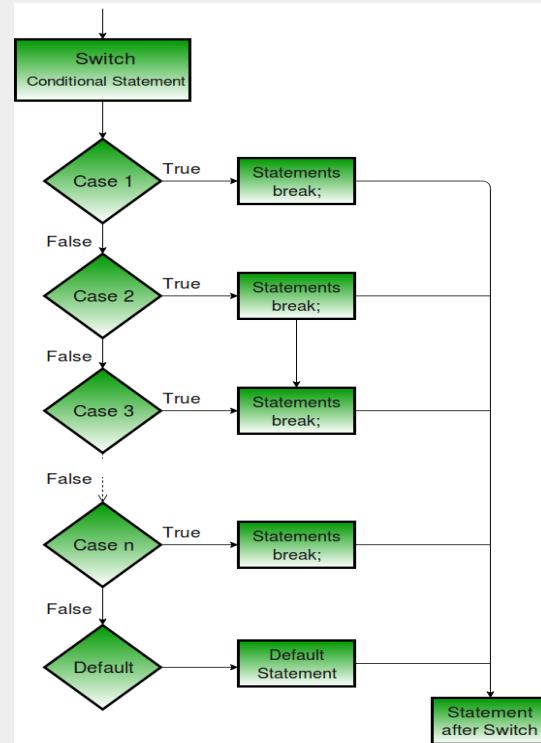
Javascript Condition

JavaScript Switch

```
result+=" C Grade";  
default:  
result+=" No Grade";  
}  
document.write(result);  
</script>
```

Output:

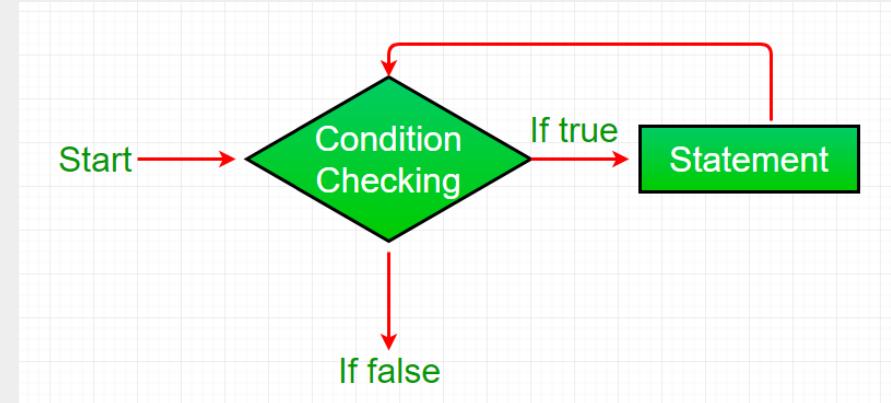
undefined B Grade C Grade No Grade



Loop Control statement

JavaScript Types of Loop

- The JavaScript loops are used to iterate the piece of code using for, while, do while or for-in loops.
- There are four types of loops in JavaScript.
 - i. for loop
 - ii. while loop
 - iii. do-while loop
 - iv. for-in loop



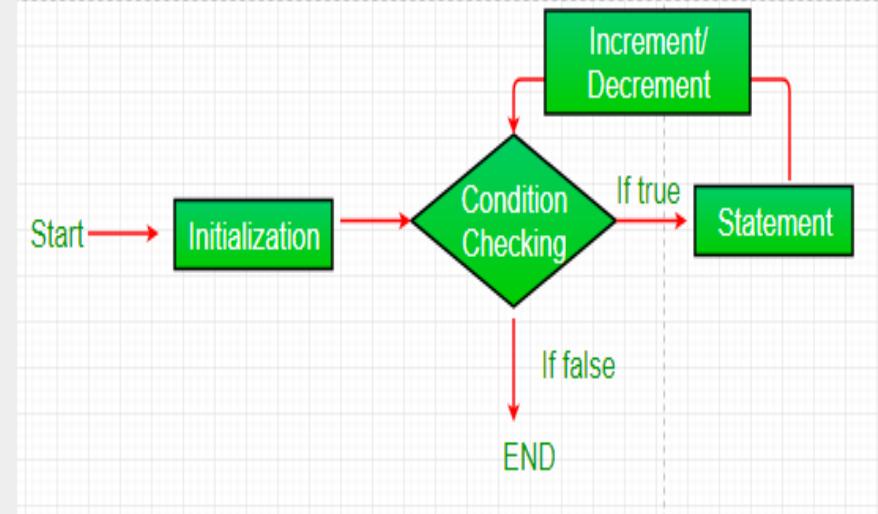
Loop Control statement

For loop

The JavaScript for loop iterates the elements for the fixed number of times. It should be used if number of iterations is known.

Syntax:

```
for (initialization; condition; increment)  
{  
    code to be executed  
}
```



Loop Control statement

While loop

The JavaScript while loop iterates the elements for the infinite number of times. It should be used if number of iteration is not known.

Syntax:

```
while (condition)
{
    code to be executed
}
```

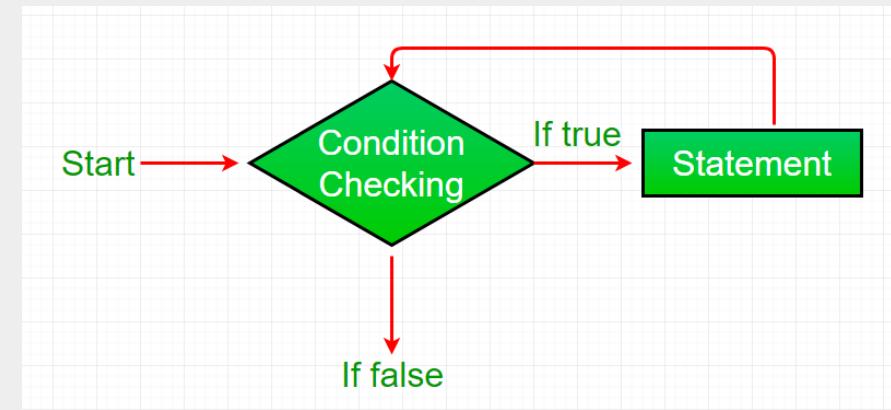


Image Source: <https://media.geeksforgeeks.org/wp-content/uploads/Loop1.png>

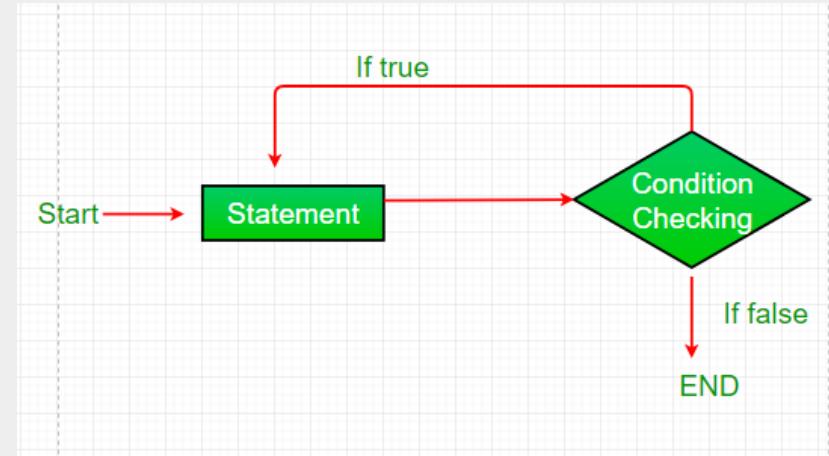
Loop Control statement

do while loop

The JavaScript do while loop iterates the elements for the infinite number of times like while loop. but, code is executed at least once whether condition is true or false.

Syntax:

```
do{  
    code to be executed  
}while (condition);
```



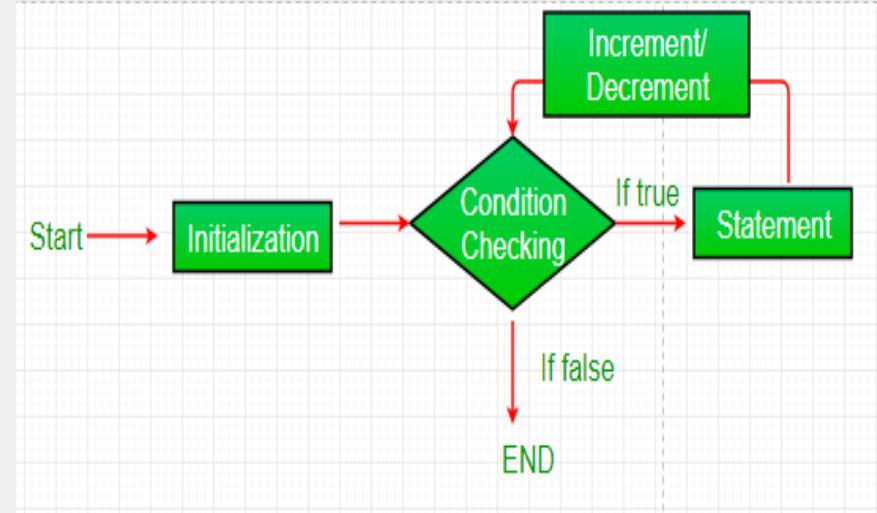
Loop Control statement

for-in loop

The for..in loop provides a simpler way to iterate through the properties of an object.

Syntax:

```
for (variableName in Object)
{
    statement(s)
}(condition);
```



JavaScript Array

What is Arrays

JavaScript array is an object that represents a collection of similar type of elements.

There are 3 ways to construct array in JavaScript

- i. By array literal
- ii. By creating instance of Array directly (using new keyword)
- iii. By using an Array constructor (using new keyword)

Arrays in JavaScript



Image Source: <https://cdn.educba.com/academy/wp-content/uploads/2019/09/Arrays-in-JavaScript.png>

JavaScript Array

JavaScript array literal

Syntax:

```
var arrayname=[value1,value2.....valueN];
```

Arrays in JavaScript



Image Source: <https://cdn.educba.com/academy/wp-content/uploads/2019/09/Arrays-in-JavaScript.png>

JavaScript Array

JavaScript Array directly

Syntax:

```
var arrayname=new Array();
```

Arrays in JavaScript



Image Source: <https://cdn.educba.com/academy/wp-content/uploads/2019/09/Arrays-in-JavaScript.png>

JavaScript Array

JavaScript array constructor

Here, you need to create instance of array by passing arguments in constructor so that we don't have to provide value explicitly.

Example:

```
<script>  
var emp=new Array("Jai","Vijay","Smith");  
for (i=0;i<emp.length;i++){  
document.write(emp[i] + "<br>");  
}  
</script>
```

Arrays in JavaScript

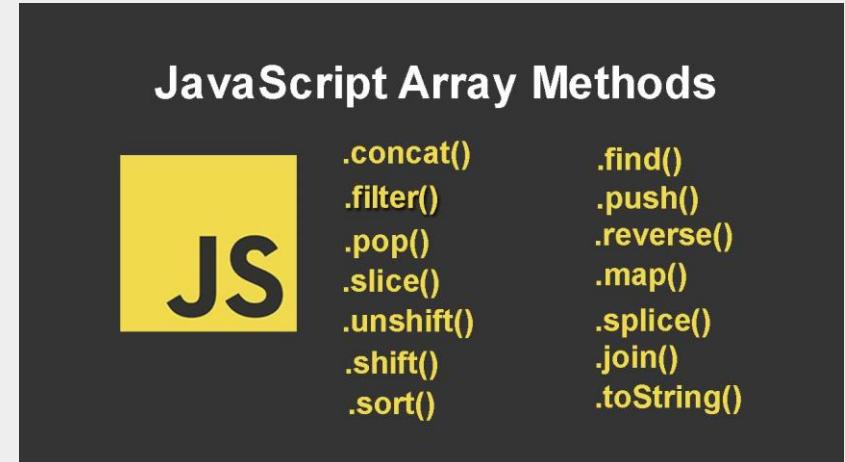


Image Source: <https://cdn.educba.com/academy/wp-content/uploads/2019/09/Arrays-in-JavaScript.png>

JavaScript Array

JavaScript Array Methods

Let's see the list of JavaScript array methods.



Javascript Object

What is Javascript Object

- A JavaScript object is an entity having state and behavior. For example: car, pen, bike, chair, glass, keyboard, monitor etc.
- JavaScript is an object-based language. Everything is an object in JavaScript.
- JavaScript is template based not class based.



Image Source: <https://cdn.educba.com/academy/wp-content/uploads/2020/05/Object-in-JavaScript.jpg>

Javascript Object

Creating Objects in JavaScript

There are 3 ways to create objects.

- i. By object literal
- ii. By creating instance of Object directly
- iii. By using an object constructor



Javascript Object

Defining method in JavaScript Object

- We can define method in JavaScript object. But before defining method, we need to add property in the function with same name as method.



JavaScript Function

What is Javascript Function

- A function is a block of code that performs a specific task.
- Suppose you need to create a program to create a circle and color it. You can create two functions to solve this problem:
 - a function to draw the circle
 - a function to color the circle

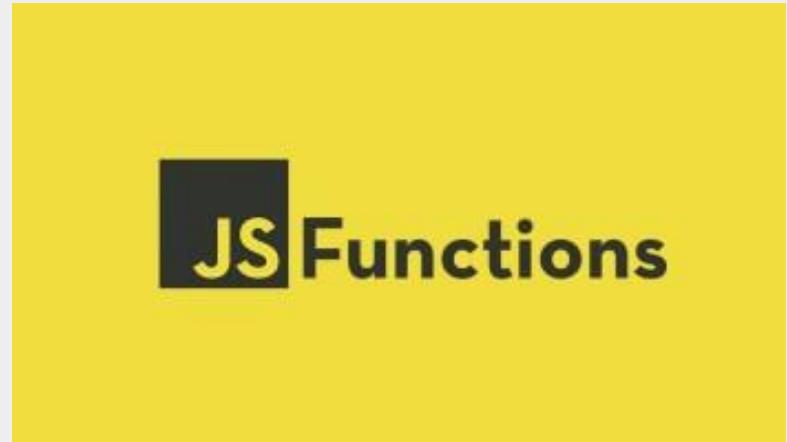


Image Source: <https://i0.wp.com/blog.alexdevero.com/wp-content/uploads/2020/01/javascript-functions-all-you-need-to-know-pt.1.jpg?fit=1024%2C635&ssl=1&resize=350%2C200>

JavaScript Function

Declaring a Function

- A function is declared using the function keyword.
- The basic rules of naming a function are similar to naming a variable.
- The body of function is written within {}.

Syntax:

```
function nameOfFunction () {  
    // function body  
}
```



Image Source: <https://raddevon.com/wp-content/uploads/2019/07/function-declaration-diagram-1024x282.png>

JavaScript Function

Calling a Function

- A function is declared using the function keyword.
- The basic rules of naming a function are similar to naming a variable.
- The body of function is written within {}.

Syntax:

```
function nameOfFunction () {  
    // function body  
}
```

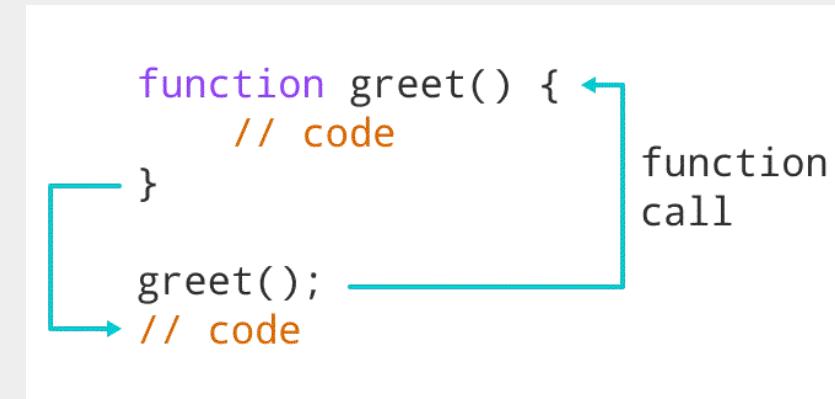


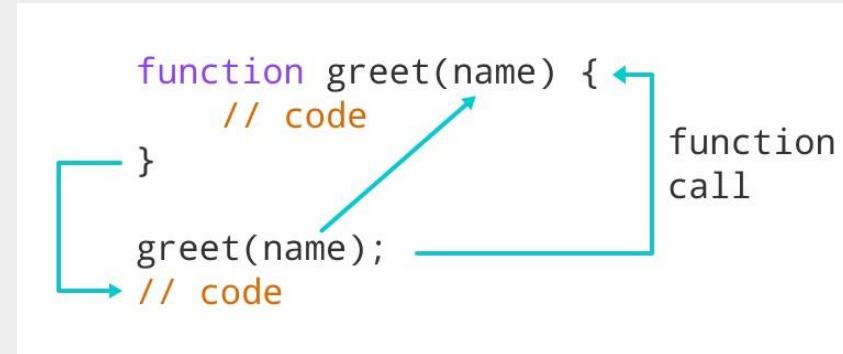
Image Source:

<https://cdn.programiz.com/cdn/farfuture/NdxxeWIRfoHMPgdWPKeVy1wN9MwAgqoYqZkFQDMFQ/mtime:1591592059/sites/tutorial2program/files/javascript-function-example1.png>

JavaScript Function

Function Parameters

- A function can also be declared with parameters. A parameter is a value that is passed when declaring a function.



JavaScript Function

Function Return

- The return statement can be used to return the value to a function call.
- The return statement denotes that the function has ended. Any code after return is not executed.
- If nothing is returned, the function returns an undefined value.

```
function add(num1, num2) {  
    // code  
    return result;  
}  
  
let x = add(a, b);  
// code
```

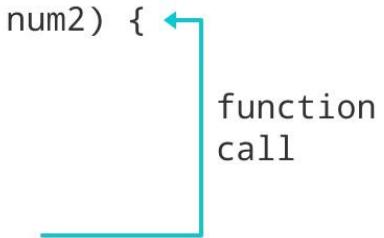


Image Source: <https://cdn.programiz.com/cdn/farfuture/b4h4Zo5ZYxj-EyfQyaoJ5TqbKEefFgqqusPGLWPFS0/mtime:1591786573/sites/tutorial2program/files/javascript-return-statement.png>

JavaScript Function

Benefits of Using a Function

- Function makes the code reusable. You can declare it once and use it multiple times.
- Function makes the program easier as each small task is divided into a function.
- Function increases readability.

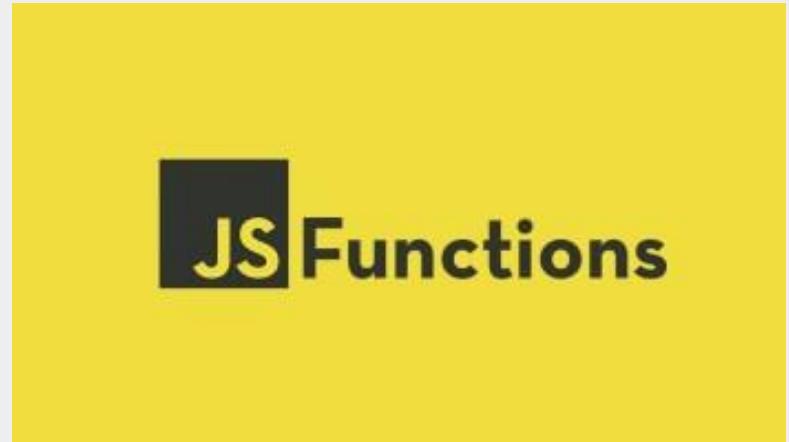


Image Source: <https://i0.wp.com/blog.alexdevero.com/wp-content/uploads/2020/01/javascript-functions-all-you-need-to-know-pt.1.jpg?fit=1024%2C635&ssl=1&resize=350%2C200>

JavaScript Function

Function Expressions

In Javascript, functions can also be defined as expressions.

```
// program to find the square of a number
let x = function (num) { return num * num };
console.log(x(4));
// can be used as variable value for other
variables
let y = x(3);
console.log(y);
```

Output

16

9

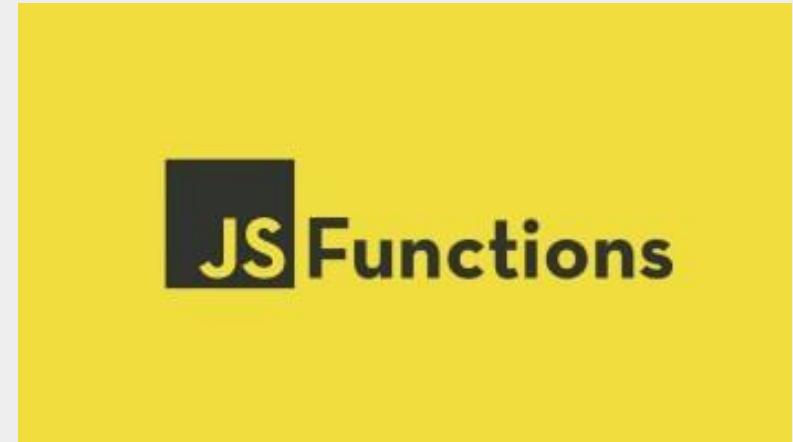


Image Source: <https://i0.wp.com/blog.alexdevero.com/wp-content/uploads/2020/01/javascript-functions-all-you-need-to-know-pt.1.jpg?fit=1024%2C635&ssl=1&resize=350%2C200>

Advance JavaScript

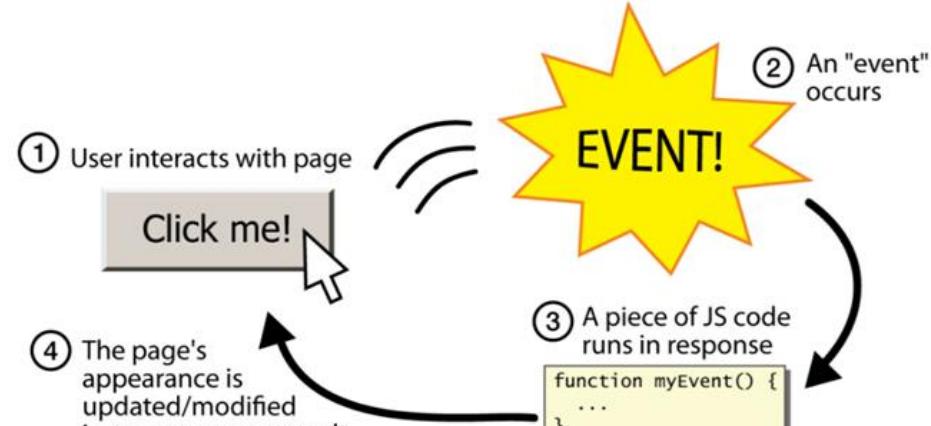
In this section, we will discuss:

- Introduction to events.
- Introduction to exceptional handling.
- BOM property and methods
- setTimeout(), setInterval() method
- Date methods
- Introduction to Ajax
- Intdoduction to JSON , JSON methods

Introduction to Events, Events Handling

Events

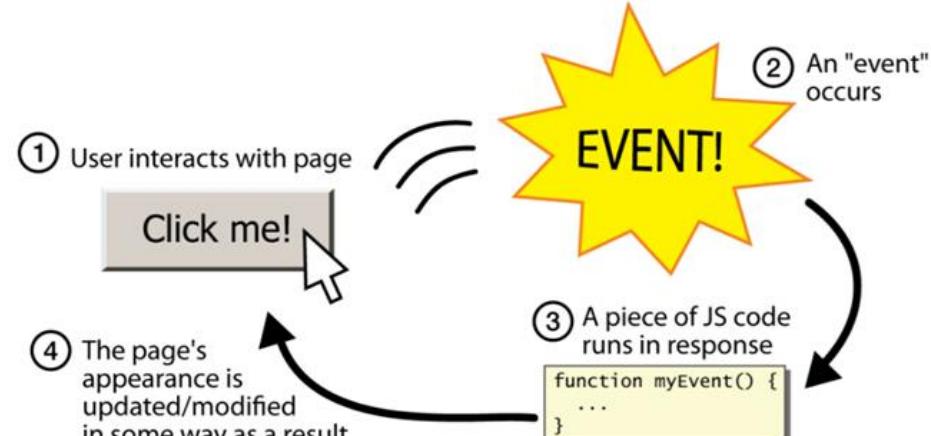
- JavaScript's interaction with HTML is handled through events that occur when the user or the browser manipulates a page.
- When the page loads, it is called an event.



Introduction to Events, Events Handling

Events continued..

- Event-driven programming is when parts of the programming are executed in an unpredictable sequence in response to specific events.
- Events are objects in JavaScript with case-sensitive names, all of which are lower-case.



Introduction to Events, Events Handling

Event Handling

- Event Handling is the mechanism that controls the event and decides what should happen if an event occurs.
- This mechanism has the code which is known as event handler that is executed when an event occurs.

Introduction to Events, Events Handling

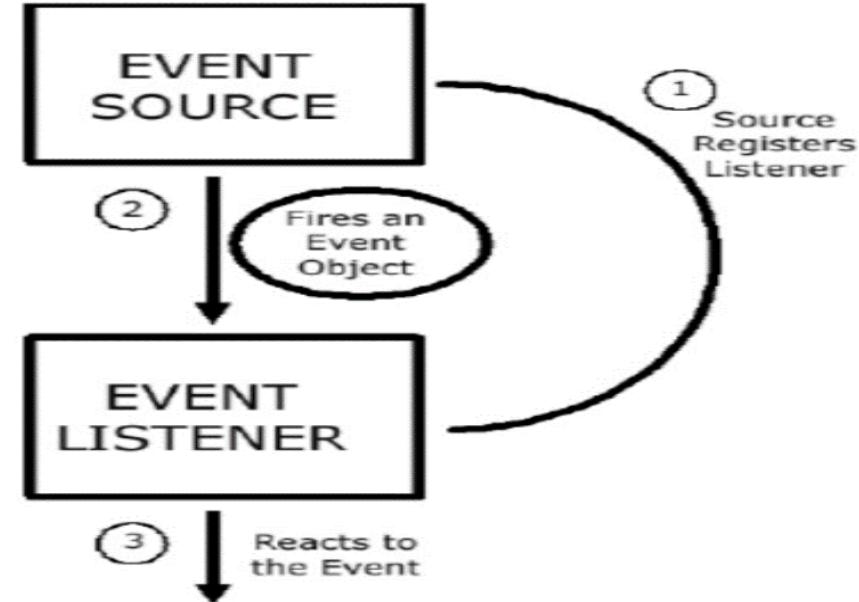
Event Handling

- Source - The source is an object on which event occurs. Source is responsible for providing information of the occurred event to its handler. Java provides classes for source object.
- Listener - It is also known as event handler. An event handler is a script that is implicitly executed in response to an event happening.

Introduction to Events, Events Handling

Functions of Event Handling

- Event Handling identifies where an event should be forwarded.
- It makes the forward event.
- It receives the forwarded event.
- It takes some kind of appropriate action in response, such as writing to a log, sending an error or recovery routine or sending a message.
- The event handler may ultimately forward the event to an event consumer.



Introduction to Events, Events Handling

Javascript Events

- onclick - Occurs when the user clicks on a link or form element
- onload - Occurs when a page is loaded into the browser (i.e., opened)
- onsubmit - Occurs when a form's Submit button is clicked
- onfocus - Occurs when a user gives input or focus to a form element

```
<!DOCTYPE html>
<!-- load.html
|An example to illustrate the load event
A document for load.js -->
<html lang = "en">
<head>
<title> onLoad event handler </title>
<meta charset = "utf-8" />
<script type = "text/javascript" src = "load.js" > </script>
</head>
<body onload = "loadGreeting();">
<p />
</body>
</html>

// load.js
// An example to illustrate the load event
// The onload event handler
function loadGreeting() {
  alert("You are visiting the home page of \n" + "Pete's Pickled
Peppers\n" + "WELCOME!!!!");
}
```

Introduction to Exceptional Handling

Exceptional handling, Errors

- Exception handling is one of the powerful JavaScript features to handle errors and maintain a regular JavaScript code/program flow.
- Errors occur due to mistakes made by developers, wrong input, or unforeseeable things.

Introduction to Exceptional Handling

Reasons why exception occurs

- Dividing a number by zero: This results in infinity, thus throwing an exception.
- When a requested file does not exist in the system.
- When the user provides the wrong input.
- When the network drops during communication.

Introduction to Exceptional Handling

Error types

- Syntax errors
- Runtime errors
- Logical errors

Introduction to Exceptional Handling

Error types

- Syntax errors - These are errors that cannot be interpreted by the computer. These errors stop the program from working. These errors are:
- Spelling errors (wrong spelling such as fiction instead of function).
- The omission of important characters, such as not using a semicolon to end a statement.
- Use of the wrong indentation.

Introduction to Exceptional Handling

Error types

- Runtime errors - These errors take place during execution.
- The errors get detected when your program runs.
- It crashes or raises an exception. Thus, exception handlers handle exception errors.

Introduction to Exceptional Handling

Error types

- Runtime errors - These errors are often caused by:
- The program not being able to find data because it does not exist.
- The data being an invalid type of data.

Introduction to Exceptional Handling

Error types

- Logical Errors - These types of errors do not throw an error or an exception at all.
- This is because they result from the code not doing what the developer intends it to.
- It's challenging to find logical errors.
- They can only be found through thorough testing.

Introduction to Exceptional Handling

Error Objects

- When a runtime error occurs, it stops the code and raises an error object.
- The error object has two properties:
- Name: It gives the error name.
- Message: It sets or returns the error message in the form of a string.

Introduction to Exceptional Handling

Six types of Error Objects

- **EvalError:** The `EvalError` function indicates the error that occurred in the `eval()` function.
- It's a global function that evaluates the JavaScript string.
- **RangeError:** `RangeError` exceptions occur when a numeric value is outside the specified range.

Introduction to Exceptional Handling

Six types of Error Objects

- **ReferenceError:** A ReferenceError exception occurs when undeclared variables are used.
- These exceptions commonly occur due to spelling errors on variables.
- **Syntax Error:** A Syntax Error exception occurs when JavaScript language rules get broken.

Introduction to Exceptional Handling

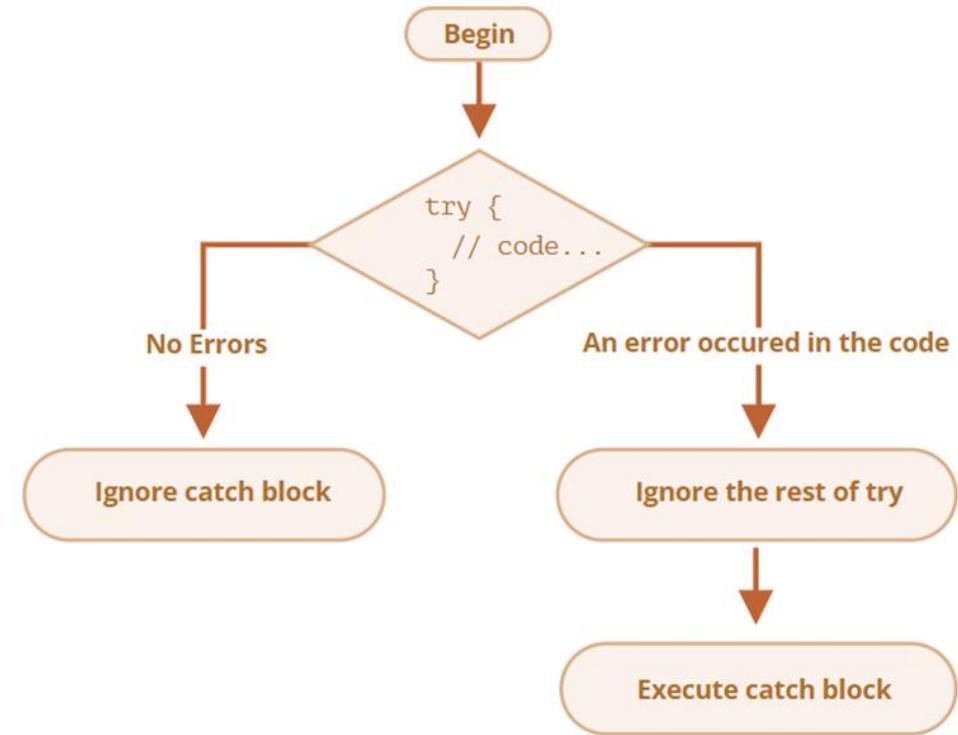
Six types of Error Objects

- **TypeError:** A `TypeError` exception occurs when a value is different from the one expected.
- **URIError:** A `URIError` exception is raised by `encodeURI()` and `decodeURI()` methods.

Introduction to Exceptional Handling

Exceptional Handling

- A try-catch-finally statement is a code or program that handles exceptions.
- The try clause runs the code that generates exceptions.
- The catch clause catches exceptions that are thrown.
- A finally clause always gets executed.
- The throw statement generates exceptions.



Introduction to Exceptional Handling

Exceptional Handling

- Throw statements - The throw statement is to raise your built-in exceptions.

```
function myFunction() {  
    const x = 50;  
    const y = 0;  
    try {  
        if (y === 0) {  
            throw ("This is division by zero error");  
        } else {  
            const z = x / y;  
        }  
    } catch (error) {  
        alert("Error: " + error);  
    }  
}
```

Introduction to Exceptional Handling

Exceptional Handling

- Try catch statements - The try clause has the main code that may generate exceptions.
- If an exception is raised, the catch clause gets executed.

```
function myFunction() {  
    const j = 70;  
    try {  
        alert ("The value of j is: " + j);  
    } catch (error) {  
        alert("Error: " + error.message);  
    }  
}
```

Introduction to Exceptional Handling

Exceptional Handling

- Try catch finally statements - The finally statement is the last block to be executed.
- It executes after try and catch clauses.

```
function myFunction() {  
    const j = 70;  
    try {  
        alert("The value of j is : " + j);  
    } catch (error) {  
        alert("Error: " + error.message);  
    } finally {  
        alert("Finally: Finally gets executed")  
    }  
}
```

Introduction to Browser Object Model

Browser Object Model

- The browser object model (BOM) is a hierarchy of browser objects that are used to manipulate methods and properties associated with the Web browser itself.

Introduction to Browser Object Model

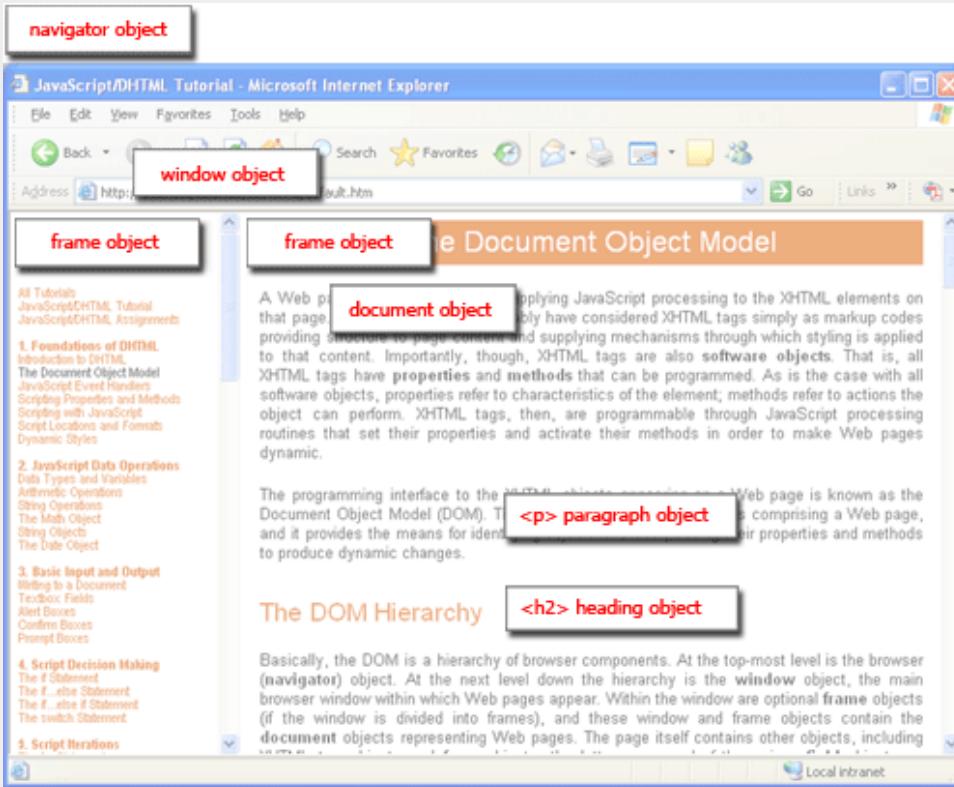
Browser Object Model

- The Document Object consists of objects that are used to manipulate methods and properties of the document or Web page loaded in the browser window.
- The document object represents the Web page currently loaded in the browser window.

Introduction to Browser Object Model

The BOM Hierarchy

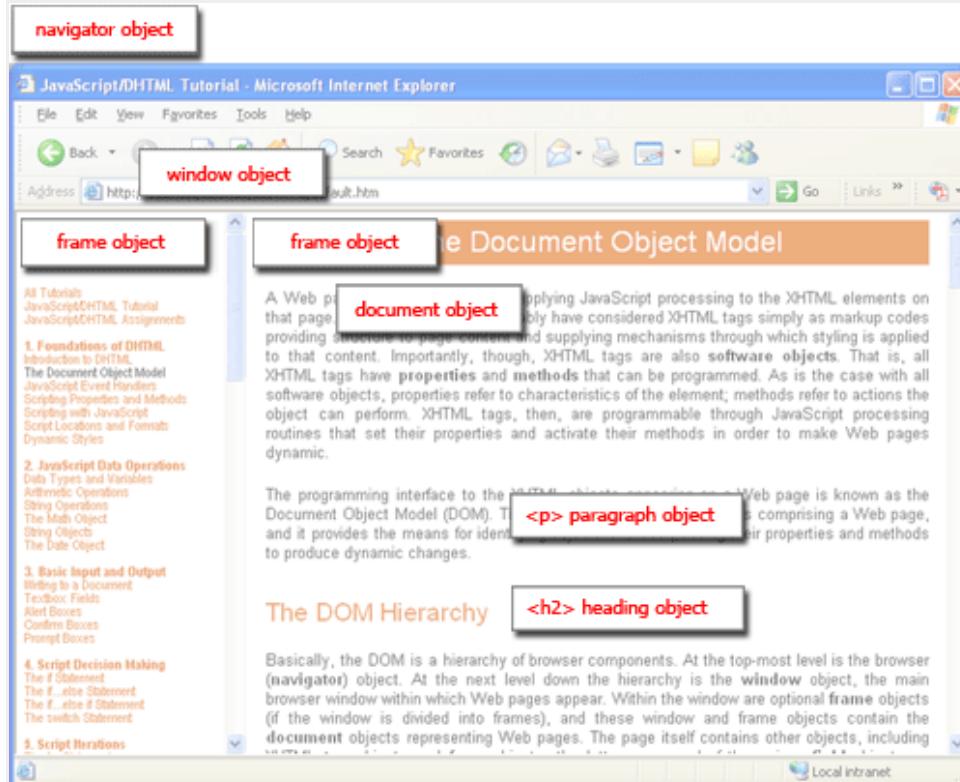
- The top-level object in the BOM is the window object.
- The window object represents the browser window.
- The window object includes a number of properties and methods that can be used to control the Web browser.



Introduction to Browser Object Model

The BOM Hierarchy

- The screen object, that contains information about the visitor's screen.
- The history object, that is part of the window object and contains the URLs that have been visited by the user.
- The location object that contains information about the current URL.



Introduction to Browser Object Model

Window Object Method

- `window.alert()` - Creates dialog box with message and an OK button
- `window.blur()` - Remove focus from window
- `window.close()` - Closes a browser window
- `window.confirm()` - Creates dialog box with message, an OK button and a cancel button

```
<script type="text/javascript">
function msg(){
var v= confirm("Are u sure?");
if(v==true){
alert("ok");  }
else{
alert("cancel");
}
} </script>
<input type="button" value="delete record" onclick="msg()"/>
```

Introduction to Browser Object Model

Window Object Method

- `window.prompt()` - Creates dialog box for retrieving user input
- `window.setInterval()` - Do something repeatedly at specified intervals
- `window.setTimeout()` - Do something after a specified amount of time

```
<script type="text/javascript">
function msg(){
setTimeout(
function(){
alert("Display JavaScript after 2 seconds")
},2000);
}
</script>
<input type="button" value="click" onclick="msg()"/>
```

Introduction to Browser Object Model

Window Object Property

- `window.closed` - Whether the window has been closed
- `window.length` - Number of `<iframe>` elements in window
- `window.name` - Gets or sets the name of the window
- `window.innerHeight` - Height of window
- `window.innerWidth` - Width of window

Introduction to Browser Object Model

Window Object Property

- `window.screen` – X - coordinate of pointer, relative to top left corner of screen
- `window.screen` – Y - coordinate of pointer, relative to top left corner of screen
- `window.location` - Current URL of window object (or local file path)
- `window.history` - Reference to history object for browser window or tab.

Introduction to Browser Object Model

Window Object Property

- `window.screen` - Reference to screen object
- `window.pageXOffset` - Distance document has been scrolled horizontally
- `window.pageYOffset` - Distance document has been scrolled vertically

Introduction to Browser Object Model

Navigator Object

- The JavaScript navigator object is used for browser detection.
- The navigator object is the window property, so it can be accessed by: `window.navigator` Or, `navigator`

Introduction to Browser Object Model

Navigator Object Method

- `javaEnabled()` - checks if java is enabled.
- `taintEnabled()` - checks if taint is enabled. It is deprecated since JavaScript 1.2.

```
<script>
document.writeln("<br/>navigator.appCodeName:
"+navigator.appCodeName);
document.writeln("<br/>navigator.appName: "+navigator.appName);
document.writeln("<br/>navigator.appVersion:
"+navigator.appVersion);
document.writeln("<br/>navigator.cookieEnabled:
"+navigator.cookieEnabled);
document.writeln("<br/>navigator.language: "+navigator.language);
document.writeln("<br/>navigator.userAgent: "+navigator.userAgent);
document.writeln("<br/>navigator.platform: "+navigator.platform);
document.writeln("<br/>navigator.onLine: "+navigator.onLine);
</script>
```

Introduction to Browser Object Model

Navigator Object Property

- appName - returns the name
- appVersion - returns the version
- appCodeName - returns the code name
- cookieEnabled - returns true if cookie is enabled otherwise false
- userAgent - returns the user agent
- Language - returns the language. It is supported in Netscape and Firefox only.

Introduction to Browser Object Model

Screen Object

- The Screen object provides the attributes of the screen on which the current window is being rendered.
- The Screen object is typically used by the web analytic software like Google Analytics to collect information of the client device on which the web browsers are running.

Introduction to Browser Object Model

Screen Object Property

- Width - returns the width of the screen
- Height - returns the height of the screen
- availWidth - returns the available width
- availHeight - returns the available height
- colorDepth - returns the color depth
- pixelDepth - returns the pixel depth.

```
<script>  
document.writeln("<br/>screen.width: "+screen.width);  
document.writeln("<br/>screen.height: "+screen.height);  
document.writeln("<br/>screen.availWidth: "+screen.availWidth);  
document.writeln("<br/>screen.availHeight: "+screen.availHeight);  
document.writeln("<br/>screen.colorDepth: "+screen.colorDepth);  
document.writeln("<br/>screen.pixelDepth: "+screen.pixelDepth);  
</script>
```

Introduction to Browser Object Model

setTimeout() method

- The `setTimeout()` method in JavaScript is used to execute a function after waiting for the specified time interval.
- This method returns a numeric value that represents the ID value of the timer.
- The `setTimeout()` method executes the function only once. This method can be written with or without the `window` prefix.

Introduction to Browser Object Model

setTimeout() method

- Syntax- `window.setTimeout(function, milliseconds);`
- **function:** It is the function containing the block of code that will be executed.
- **milliseconds:** This parameter represents the time-interval after which the execution of the function takes place.

```
<html>
<head>
<title> setTimeout() method </title>
</head>
<body>
<h1> Hello World :) :) </h1>
<h3> This is an example of using the setTimeout() method </h3>
<p> Click the following button before 2 seconds to see the effect.
</p>
<button onclick = "stop()"> Stop </button>
<script>
var a = setTimeout(fun1, 2000);
function fun1() {
var win1 = window.open();
win1.document.write(" <h2> Learn Javascript </h2>");
setTimeout(function(){win1.close()}, 2000);
}
function stop() {
clearTimeout(a);
}
</script>
</body>
</html>
```

Introduction to Browser Object Model

setInterval() method

- The setInterval() method in JavaScript is used to repeat a specified function at every given time-interval.
- The setInterval() method in JavaScript is used to repeat a specified function at every given time-interval.

Introduction to Browser Object Model

setInterval() method

- Syntax - `window.setInterval(function, milliseconds);`
- **function:** It is the function containing the block of code that will be executed.
- **milliseconds:** This parameter represents the length of the time interval between each execution

```
<html>
<head>
<title> setInterval() method </title>
</head>
<body>
<h1> Hello World :) :) </h1>
<h3> This is an example of using the setInterval() method </h3>
<p> Here, the background color changes on every 200 milliseconds.
</p>
<button onclick = "stop()"> Stop </button>

<script>
var var1 = setInterval(color, 200);
function color() {
var var2 = document.body;
var2.style.backgroundColor = var2.style.backgroundColor ==
"lightblue" ? "lightgreen": "lightblue";
}
function stop() {
clearInterval(var1);
}
</script>
</body>
</html>
```

Introduction to Browser Object Model

setInterval() method

- Syntax - `window.setInterval(function, milliseconds);`
- **function:** It is the function containing the block of code that will be executed.
- **milliseconds:** This parameter represents the length of the time interval between each execution

```
<html>
<head>
<title> setInterval() method </title>
</head>
<body>
<h1> Hello World :) :) </h1>
<h3> This is an example of using the setInterval() method </h3>
<p> Here, the background color changes on every 200 milliseconds.
</p>
<button onclick = "stop()"> Stop </button>

<script>
var var1 = setInterval(color, 200);
function color() {
var var2 = document.body;
var2.style.backgroundColor = var2.style.backgroundColor ==
"lightblue" ? "lightgreen": "lightblue";
}
function stop() {
clearInterval(var1);
}
</script>
</body>
</html>
```

JavaScript Date

Date

- The JavaScript date object can be used to get year, month and day.
- You can display a timer on the webpage by the help of JavaScript date object.

Syntax:

```
new Date();
new Date(value);
new Date(dateAsString);
new Date(year, month[, day[, hour[, minute[, second[, millisecond]]]]]);
```

JavaScript Date

Date Methods

- `getDate()` - It returns the integer value between 1 and 31 that represents the day for the specified date on the basis of local time.
- `getDay()` - It returns the integer value between 0 and 6 that represents the day of the week on the basis of local time.
- `setDate()` - It sets the day value for the specified date on the basis of local time.

```
Current Date and Time: <span id="txt"></span>
<script>
var today=new Date();
document.getElementById('txt').innerHTML=today;
</script>
```

JavaScript Date

Date Methods

- `toJSON()` - It returns a string representing the Date object. It also serializes the Date object during JSON serialization.
- `toString()` - It returns the date in the form of string.
- `toTimeString()` - It returns the time portion of a Date object.
- `valueOf()` - It returns the primitive value of a Date object.

Example - Convert to JSON

```
var date1 = new Date();
date1.toJSON();
```

Output - "2022-02-23T15:49:08.596Z"

Introduction to Ajax

Ajax

- Ajax stands for Asynchronous JavaScript And Xml.
- Ajax is just a means of loading data from the server and selectively updating parts of a web page without reloading the whole page.

Introduction to Ajax

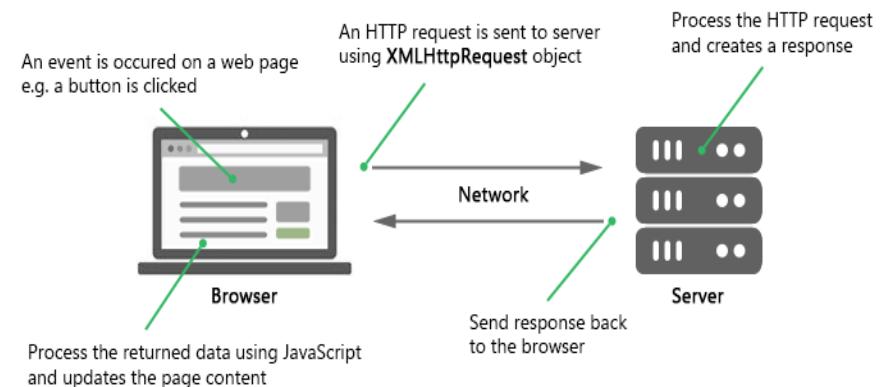
Ajax Applications Example

- Example of some large-scale Ajax-driven online applications are: Gmail, Google Maps, Google Docs, YouTube, Facebook, Flickr, and so many other applications.

Introduction to Ajax

How Ajax Works

- To perform Ajax communication JavaScript uses a special object built into the browser—an XMLHttpRequest (XHR) object—to make HTTP requests to the server and receive data in response.



Introduction to Ajax

Sending Request and Retrieving the Response

- The first thing you must do is to instantiate an XMLHttpRequest object, as shown below:
- `var request = new XMLHttpRequest();`

Introduction to Ajax

Sending Request and Retrieving the Response

- The next step in sending the request to the server is to instantiating the newly-created request object using the open() method of the XMLHttpRequest object.
- `request.open("GET", "info.txt");` -Or-
`request.open("POST", "add-user.php");`

Introduction to Ajax

Sending Request and Retrieving the Response

- Finally send the request to the server using the send() method of the XMLHttpRequest object.
- `request.send();` -Or- `request.send(body);`

Introduction to Ajax

Performing an Ajax GET Request

- The GET request is typically used to get or retrieve some kind of information from the server that doesn't require any manipulation or change in database

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>JavaScript Ajax GET Demo</title>
<script>
function displayFullName() {
    // Creating the XMLHttpRequest object
    var request = new XMLHttpRequest();

    // Instantiating the request object
    request.open("GET", "greet.php?fname=John&lname=Clark");

    // Defining event listener for readystatechange event
    request.onreadystatechange = function() {
        // Check if the request is completed and was successful
        if(this.readyState === 4 && this.status === 200) {
            // Inserting the response from server into an HTML element
            document.getElementById("result").innerHTML = this.responseText;
        }
    };

    // Sending the request to the server
    request.send();
}
</script>
</head>
<body>
    <div id="result">
        <p>Content of the result DIV box will be replaced by the server response</p>
    </div>
    <button type="button" onclick="displayFullName()">Display Full Name</button>
</body>
</html>
```

Introduction to Ajax

Performing an Ajax POST Request

- The POST method is mainly used to submit a form data to the web server.

```
<script>
function postComment() {
    // Creating the XMLHttpRequest object
    var request = new XMLHttpRequest();

    // Instantiating the request object
    request.open("POST", "confirmation.php");

    // Defining event listener for readystatechange event
    request.onreadystatechange = function() {
        // Check if the request is compete and was successful
        if(this.readyState === 4 && this.status === 200) {
            // Inserting the response from server into an HTML element
            document.getElementById("result").innerHTML = this.responseText;
        }
    };

    // Retrieving the form data
    var myForm = document.getElementById("myForm");
    var formData = new FormData(myForm);

    // Sending the request to the server
    request.send(formData);
}
</script>
```

Introduction to Ajax

Performing an Ajax POST Request

- Explicitly set the request header using `setRequestHeader()` method
- `request.setRequestHeader("Content-type", "application/x-www-form-urlencoded");`
- The `setRequestHeader()` method, must be called after calling `open()`, but before calling `send()`.

```
<?php
if($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = htmlspecialchars(trim($_POST["name"]));
    $comment = htmlspecialchars(trim($_POST["comment"]));

    // Check if form fields values are empty
    if(!empty($name) && !empty($comment)) {
        echo "<p>Hi, <b>$name</b>. Your comment has been received
        successfully.<p>";
        echo "<p>Here's the comment that you've entered: <b>
        $comment</b></p>";
    } else {
        echo "<p>Please fill all the fields in the form!</p>";
    }
} else {
    echo "<p>Something went wrong. Please try again.</p>";
}
?>
```

Introduction to JSON

JSON

- JSON stands for JavaScript Object Notation.
- JSON is extremely lightweight data-interchange format for data exchange between server and client.

Example

```
{  
  "book": {  
    "name": "Harry Potter and the Goblet of Fire",  
    "author": "J. K. Rowling",  
    "year": 2000,  
    "genre": "Fantasy Fiction",  
    "bestseller": true  
  }  
}
```

Introduction to JSON

JSON

- JSON is based on two basic structures:
- Object: This is defined as an unordered collection of key/value pairs (i.e. key:value).
- Each object begins with a left curly bracket { and ends with a right curly bracket }.
- Multiple key/value pairs are separated by a comma ,.

Example

```
{  
  "book": {  
    "name": "Harry Potter and the Goblet of Fire",  
    "author": "J. K. Rowling",  
    "year": 2000,  
    "genre": "Fantasy Fiction",  
    "bestseller": true  
  }  
}
```

Introduction to JSON

JSON

- Array: This is defined as an ordered list of values.
- An array begins with a left bracket [and ends with a right bracket].
- Values are separated by a comma ,.

Example

```
{  
  "book": {  
    "name": "Harry Potter and the Goblet of Fire",  
    "author": "J. K. Rowling",  
    "year": 2000,  
    "genre": "Fantasy Fiction",  
    "bestseller": true  
  }  
}
```

Introduction to JSON

JSON Methods

- **JSON.parse()** - This method takes a JSON string and transforms it into a JavaScript object.
- **JSON.stringify()** - This method converts a JavaScript value (JSON object) to a JSON string representation.

```
<script>
//JavaScript to illustrate JSON.parse() method.
var j = '{"Name":"Krishna","Email": "XYZ", "CN": "12345"}';
var data = JSON.parse(j);
document.write("Convert string in JSON format using parse() method<br>");
document.write(data.Email); //expected output: XYZ
//JavaScript to illustrate JSON.stringify() method.
var j = {Name:"Krishna",
Email: "XYZ", CN : 12345};
var data = JSON.stringify(j);
document.write("<br>Convert string in JSON format using stringify()
method<br>");
document.write(data); //expected output:
{"Name":"Krishna","Email":"XYZ","CN":12345}
</script>
```

Bootstrap

(6 hours)

In this section, we will discuss:

- Overview of Bootstrap
- Bootstrap Container
- Bootstrap Components
- Advance Bootstrap Components
- Bootstrap 5 Utilities

Overview of Bootstrap

Introduction to Bootstrap

- Bootstrap is a free front-end framework.
- Bootstrap include HTML and CSS based design templates.
- It's also free!
- Bootstrap also gives you the ability to easily create responsive designs.

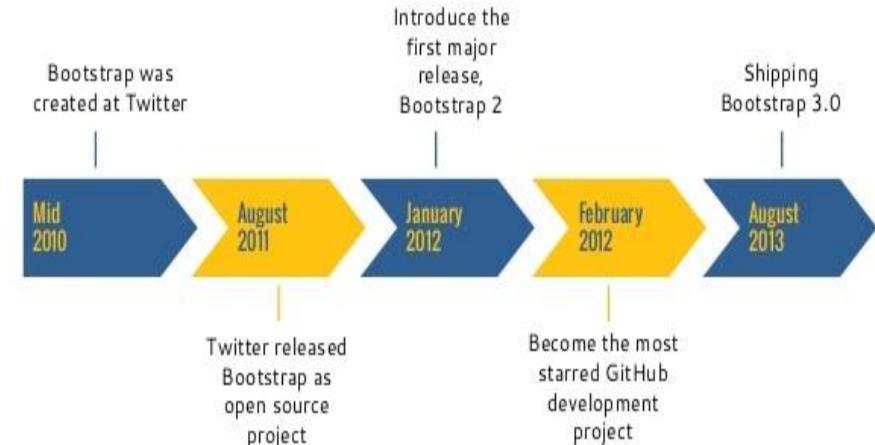


Overview of Bootstrap

History of Bootstrap

- Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter. It was released as an open source product in August 2011 on GitHub.
- In June 2014 Bootstrap was the No.1 project on GitHub.

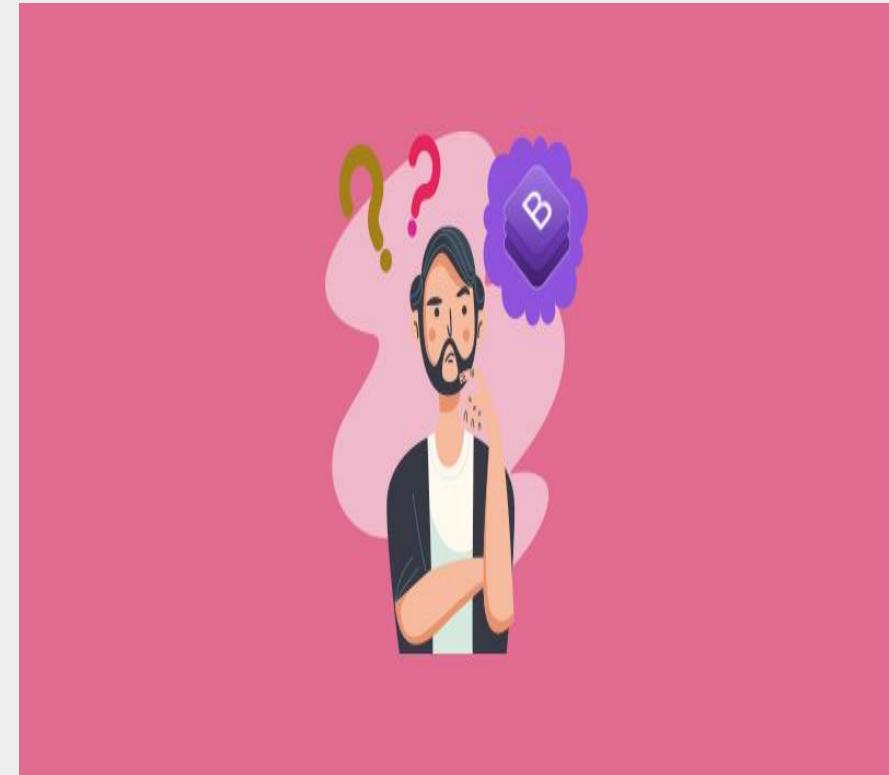
The Bootstrap history



Overview of Bootstrap

Why use Bootstrap

- Mobile first approach
- Browser support
- Easy to get started
- Responsive design



Overview of Bootstrap

Advantages of Bootstrap

- Saving time
- Easy to use
- Responsive Design
- Cross Browser Compatibility
- Open Source
- Customization

The Main Benefits of Bootstrap



Responsiveness



Cross-browser compatibility



Short development time



Flexibility



Feature-rich core

Overview of Bootstrap

Bootstrap Container

- Fixed Container
- Fluid Container

Bootstrap simple container

Normal Size



Bootstrap fluid container

Normal Size



Bootstrap Container

Fixed Container

- The `.container` class provides a responsive fixed width container.

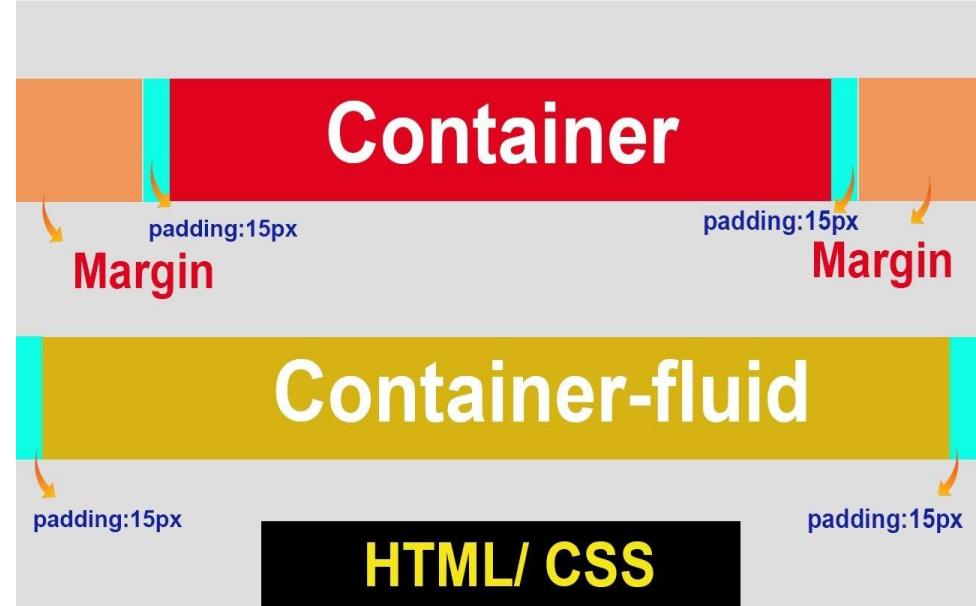


`.container`

Bootstrap Container

Fluid Container

- The `.container-fluid` class provides a full-width container which spans the entire width of the viewport.
- In the below example, the div with class “`container-fluid`” will take-up the complete width of the viewport and will expand or shrink when ever the viewport is resized.



Overview of Bootstrap

How to use Bootstrap

There are two ways to include Bootstrap on the website.

- Include Bootstrap from the CDN link.
- Download Bootstrap from getbootstrap.com and use it.



Overview of Bootstrap

Bootstrap Grid

- Bootstrap's grid system allows up to 12 columns across the page
- You can divide the container in rows and each row in columns with space multiple of the 12



Overview of Bootstrap

Bootstrap Grid

- In this example, we divide the space inside the container in 1 row and this row in 3 columns with the same size.
 $4+4+4 = 12$
- You can use any combination that the sum be equal to 12.



Overview of Bootstrap

Bootstrap Tables

- 4 main classes:
 - .table
 - .table-striped
 - .table-bordered
 - .table-hover
- 5 contextual classes:
 - .active
 - .success
 - .info
 - .warning
 - .danger

Firstname	Lastname	Email
John	Doe	john@example.com
Joseph	Saints	joseph@example.com
Mary Help	Saints	mary@example.com

Overview of Bootstrap

Bootstrap Images

- Three main classes:
 - . img-rounded
 - . img-circle
 - . img-thumbnail



Overview of Bootstrap

Bootstrap Alerts

- Bootstrap provides an easy way to create predefined alert messages
- Alerts are created with the `.alert` class, followed by one of the four contextual classes
 - `.alert-success`
 - `.alert-info`
 - `.alert-warning`
 - `.alert-danger`

The Bootstrap Alert demo

Your account is created successfully!

You will be redirected to another website, Are you sure?

The Bootstrap alerts are created by using `.alert` class.

Your account will be deleted permanently!

Overview of Bootstrap

Bootstrap Buttons

- Bootstrap provides seven styles of buttons:
- To achieve the button styles, Bootstrap has the following contextual classes:
 - .btn-default
 - .btn-primary
 - .btn-success
 - .btn-info
 - .btn-warning
 - .btn-danger
 - .btn-link



Overview of Bootstrap

Bootstrap Labels

- Labels are used to provide information about something.
- Bootstrap create labels with colorful backgrounds to highlight the text inside the label.

This is the **default** label.

This is the **primary** label.

This is the **success** label.

This is the **info** label.

This is the **warning** label.

This is the **danger** label.

Overview of Bootstrap

Bootstrap Labels

- Use the `.label` class, followed by one of the six contextual classes
 - `.label-default`
 - `.label-primary`
 - `.label-success`
 - `.label-info`
 - `.label-warning`
 - `.label-danger`

This is the `default` label.

This is the `primary` label.

This is the `success` label.

This is the `info` label.

This is the `warning` label.

This is the `danger` label.

Overview of Bootstrap

Bootstrap Components

- Bootstrap provides a variety of customizable and reusable components which makes the development faster and easier.
- They are heavily based on the base modifier nomenclature i.e. the base class has many groups of shared properties together while the modifier class has a group of individual styles.

Components of Bootstrap

- | | | |
|----------------|----------------|-----------------|
| 1.Jumbotron | 9.List group | 17.Carousel |
| 2.Alerts | 10.Card | 18.Toast |
| 3.Buttons | 11.Dropdown | 19.Tooltip |
| 4.Button group | 12.Nav | 20.Popovers |
| 5.Badge | 13.Navbar | 21.Collapse |
| 6.Progress Bar | 14.Forms | 22.Modal |
| 7.Spinner | 15.Input group | 23.Pagination |
| 8. Scrollspy | 16.Breadcrumb | 24.Media object |

Bootstrap Components

List of Components

- Jumbotron
- Alerts
- Buttons
- Button group
- Badge
- Progress Bar
- Spinner
- Scrollspy
- List group
- Card
- Dropdown

Components of Bootstrap

- | | | |
|----------------|----------------|-----------------|
| 1.Jumbotron | 9.List group | 17.Carousel |
| 2.Alerts | 10.Card | 18.Toast |
| 3.Buttons | 11.Dropdown | 19.Tooltip |
| 4.Button group | 12.Nav | 20.Popovers |
| 5.Badge | 13.Navbar | 21.Collapse |
| 6.Progress Bar | 14.Forms | 22.Modal |
| 7.Spinner | 15.Input group | 23.Pagination |
| 8. Scrollspy | 16.Breadcrumb | 24.Media object |

Bootstrap Components

List of Components

- Navs
- Navbar
- Forms
- Input groups
- Toast
- Tooltip
- Popovers
- Collapse
- Modal
- Pagination
- Media object

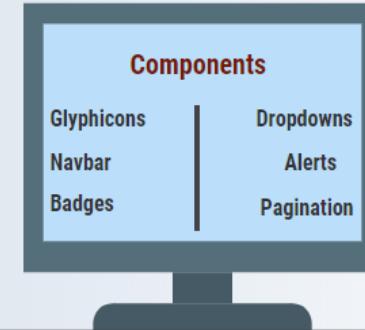
Components of Bootstrap

- | | | |
|----------------|----------------|-----------------|
| 1.Jumbotron | 9.List group | 17.Carousel |
| 2.Alerts | 10.Card | 18.Toast |
| 3.Buttons | 11.Dropdown | 19.Tooltip |
| 4.Button group | 12.Nav | 20.Popovers |
| 5.Badge | 13.Navbar | 21.Collapse |
| 6.Progress Bar | 14.Forms | 22.Modal |
| 7.Spinner | 15.Input group | 23.Pagination |
| 8. Scrollspy | 16.Breadcrumb | 24.Media object |

Bootstrap Components

Advance Bootstrap Components

- Accessibility
- Breadcrumbs
- Calendar
- Carousel
- Checkbox
- Color Picker
- Combobox
- Contact Form
- Datepicker
- Dialogue boxes & alerts
- File upload

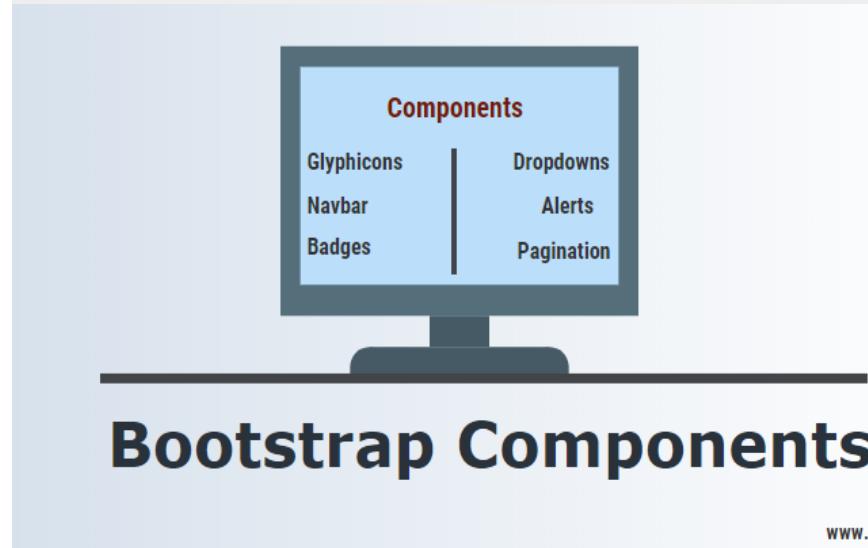


Bootstrap Components

Bootstrap Components

Advance Bootstrap Components

- Form validation
- Image gallery
- In-place editing
- Layout Grid
- Magnify
- Modal windows
- Navigation
- Pagination
- Progress bars
- Ratings
- Social Buttons
- Tabs



Bootstrap Components

www.educba.com

Advance Bootstrap Components

Accessibility

- This plugin adds accessibility mark-up to the default components of Bootstrap. Components include: Alert, Tooltip, Popover, Modal Dialog, Dropdown Menu, Tab Panel, Collapse and Carousel.

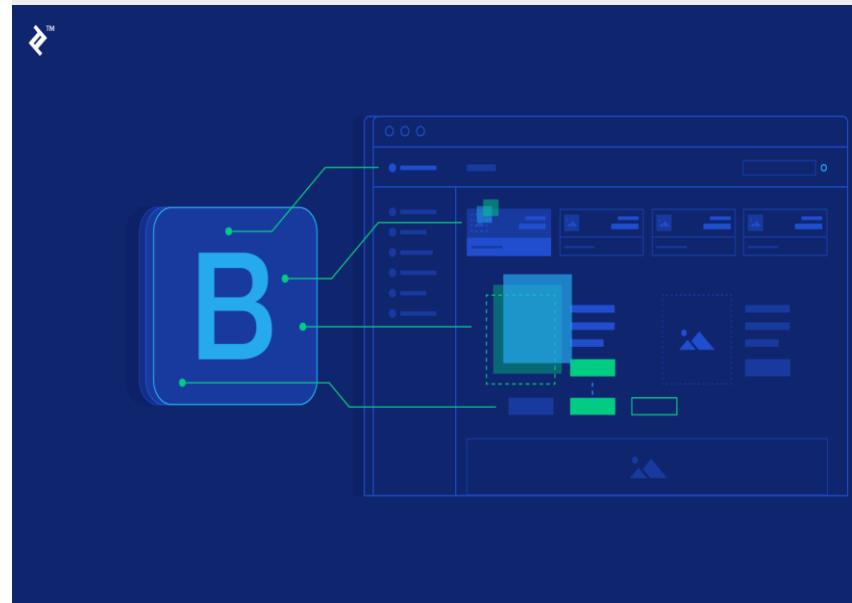


Image: https://bs-uploads.toptal.io/blackfish-uploads/components/seo/content/og_image_file/og_image/906545/REDESIGN-Speeding-up-Application-Development-with-Bootstrap-Luke_Social-3b49f4c2abf94e39b2f311ed9f0c3785.png

Advance Bootstrap Components

Breadcrumbs

- A Bootstrap JavaScript plugin that allows you to programmatically manipulate breadcrumb navigation.

Electronics Home / Smart Phones / Brand Name Product Page

Electronics Home / Smart Phones / Brand Name Product Page

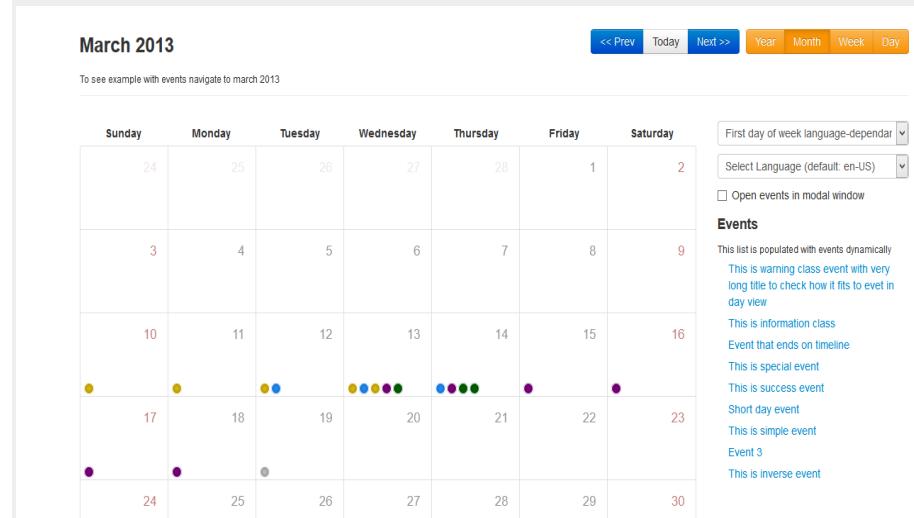
Electronics Home / Smart Phones / Brand Name Product Page

Electronics Home / Smart Phones / Brand Name Product Page

Advance Bootstrap Components

Calendar

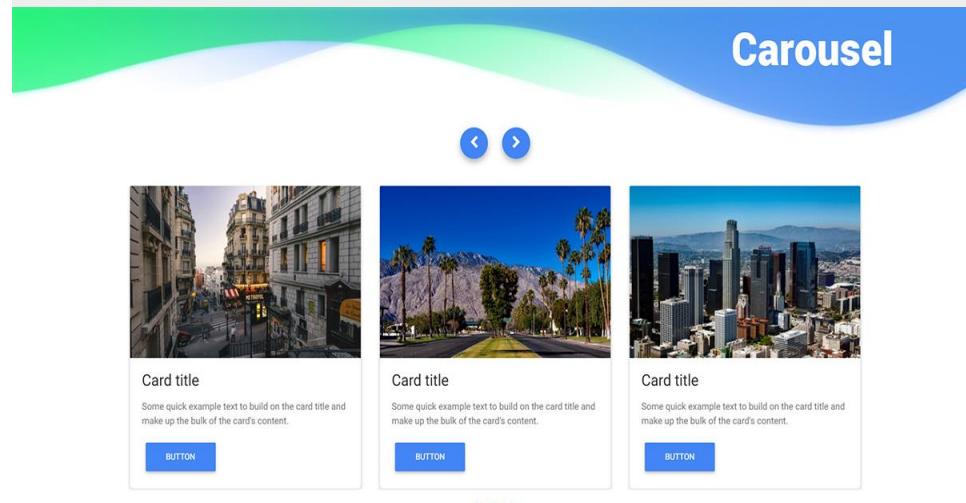
- A Full view calendar based on Bootstrap.



Advance Bootstrap Components

Carousel

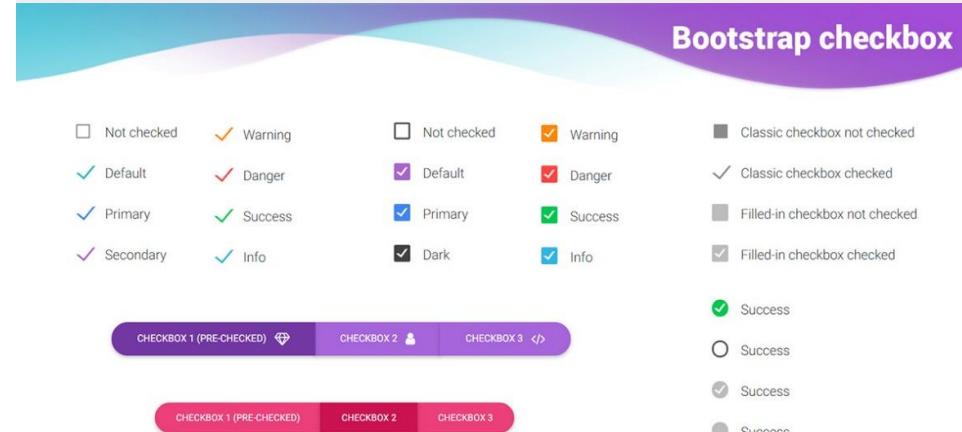
- A collection of plugins for displaying a carousel in full screen modal window.



Advance Bootstrap Components

Checkbox

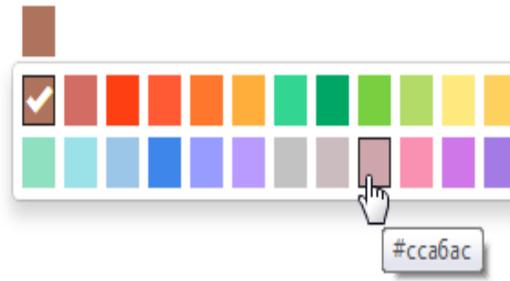
- A jQuery plugin for replacing the default checkboxes and radio inputs.



Advance Bootstrap Components

Color Picker

- A very simple and lightweight (200 lines of JavaScript and 100 lines of CSS) jQuery color picker for Bootstrap.



Advance Bootstrap Components

Combobox

- A combobox plugin that integrates well with Bootstrap.

Bootstrap Combobox

Initial select

Initial select

Advance Bootstrap Components

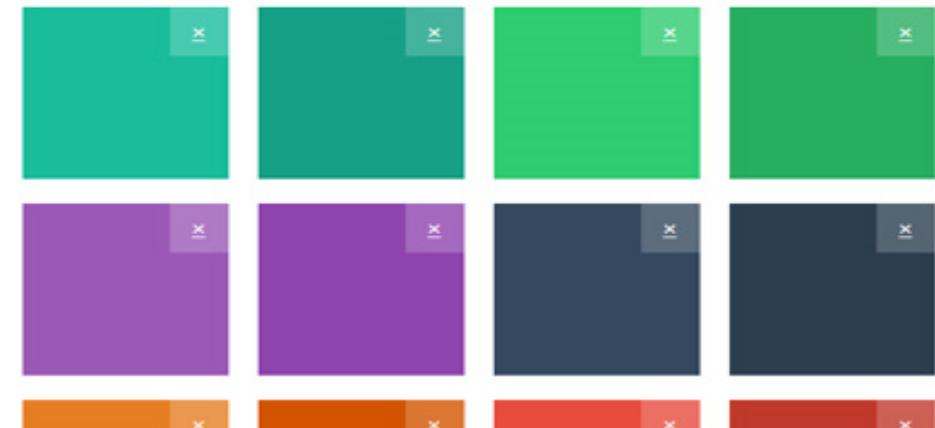
Contact Form

- Bootstrap-Contact – A simple PHP contact form using Bootstrap and the jQuery validation plugin.
- jQuery Gridform – A jQuery plugin for creating complex table-based forms with Bootstrap.

jQuery Gridly

Gridly is a jQuery plugin to enable dragging and dropping as well as resizing on a grids. In the bricks.

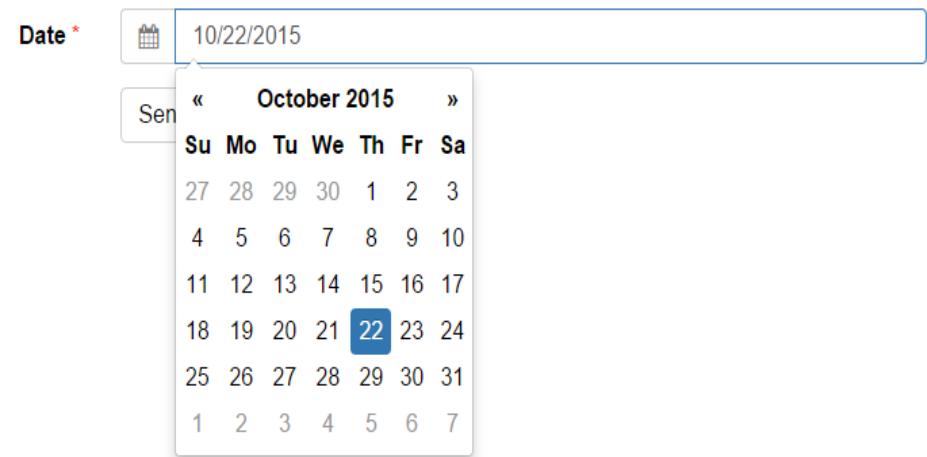
Example



Advance Bootstrap Components

Datepicker

- A plugin for adding a datepicker field to any element.



Advance Bootstrap Components

Date Range Picker

- This date range picker component creates a drop-down from which you can select a range of dates.

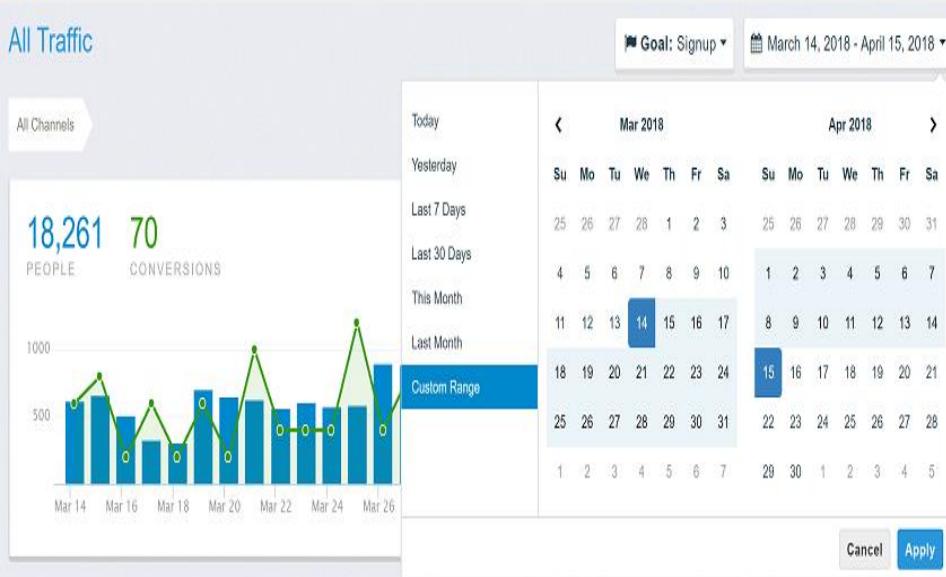


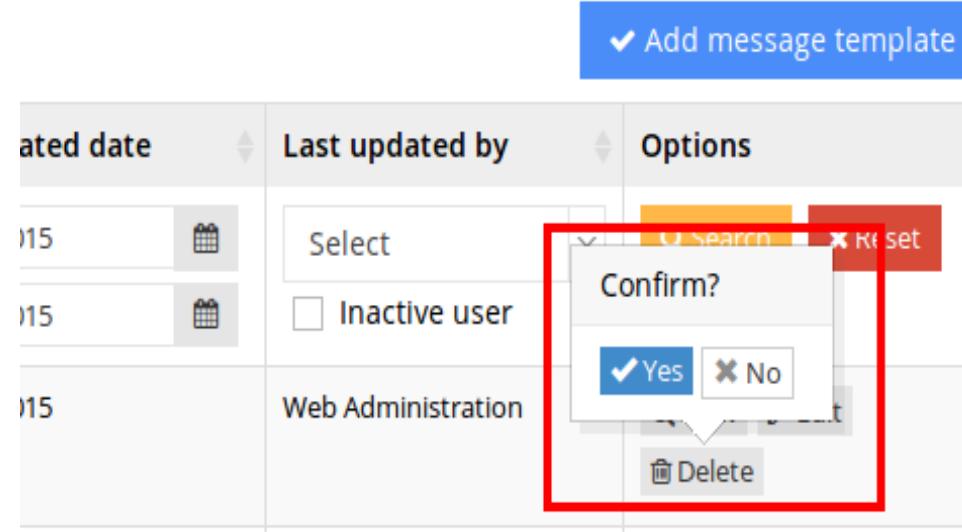
Image:

<https://camo.githubusercontent.com/d7265660ea14a97c52ab3f6ce8684294b1df30892fcfd596f2c961ca8d1bfce2/68747470733a2f2f692e692d6775722e636f6d2f5554526c6161722e706e67>

Advance Bootstrap Components

Dialogue Boxes & alerts

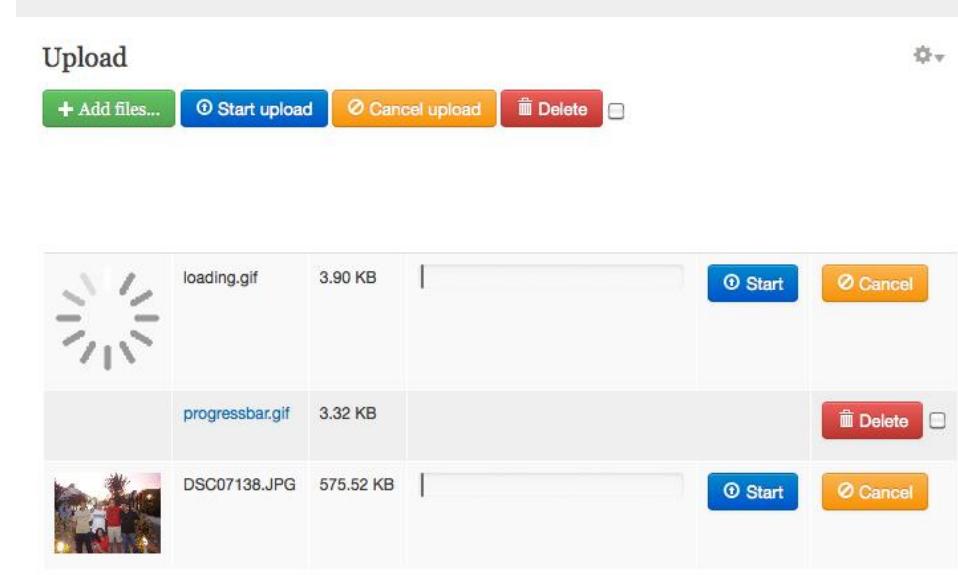
- Bootstrap Confirmation – A plugin that replaces popovers with confirmation dialogs.



Advance Bootstrap Components

File Upload

- jQuery File Upload – A file upload widget which features multiple file selection, drag & drop, progress bars, validation and preview images.



Advance Bootstrap Components

Form Validation

- BootstrapValidator – A jQuery plugin for validating forms within Bootstrap.

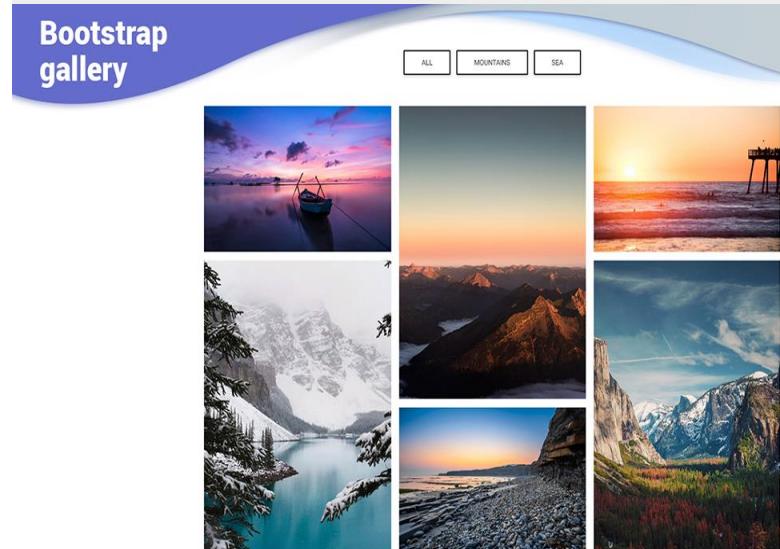
The image shows a screenshot of a Bootstrap form with validation errors. The form fields are labeled 'Email', 'Phone', 'Password', and 'Confirm Password'. The 'Email' field contains '3456@123' and has a red border with a red exclamation mark icon. Below it, the error message 'Bruh, that email address is invalid.' is displayed in red. The 'Phone' field contains 'abc' and has a red border with a red exclamation mark icon. Below it, the error message 'Only accept number.' is displayed in red. The 'Password' and 'Confirm Password' fields both contain '...' and have green borders with green checkmark icons.

Email	Phone	Password	Confirm Password
3456@123	abc

Advance Bootstrap Components

Image Gallery

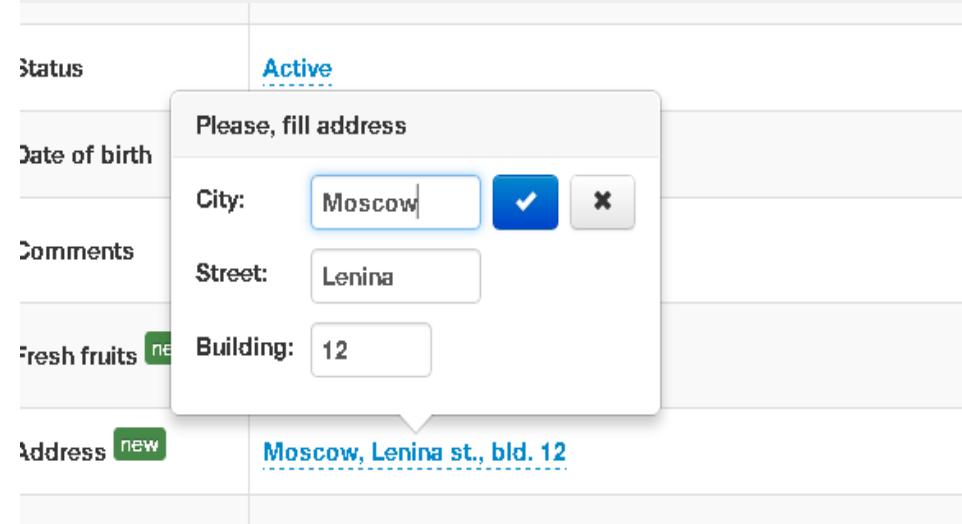
- This plugin shows images and videos in the modal dialog of the Bootstrap.
- It features swipe, mouse & keyboard navigation, transition effects, Fullscreen support and on-demand content loading.



Advance Bootstrap Components

In-Place Editing

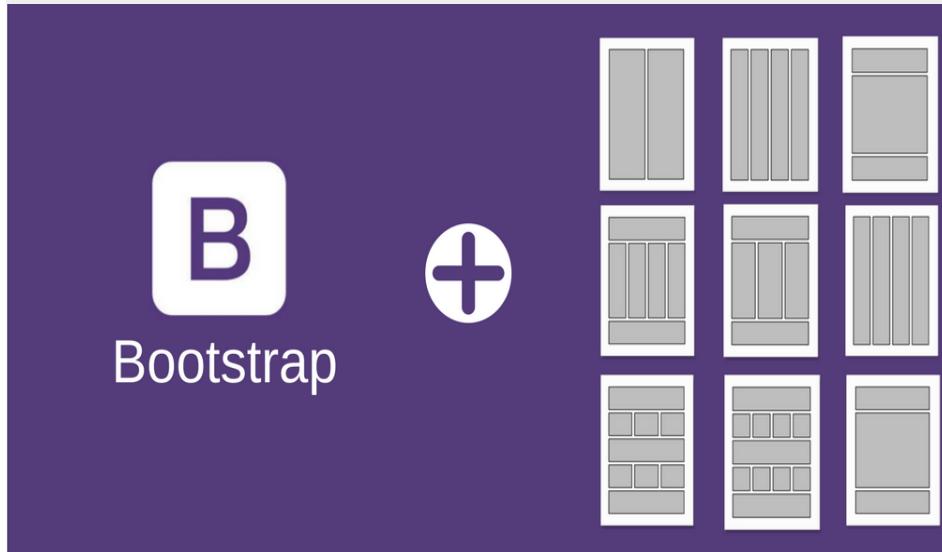
- X-editable – A library that allows you to create editable elements on your Bootstrap page.



Advance Bootstrap Components

Layout Grid

- jQDrawBootstrapGrid – A simple jQuery plugin that draws grid columns to a Bootstrap enabled layout.



Advance Bootstrap Components

Magnify

- A JS plugin for adding a magnifying glass to images on mouseover.



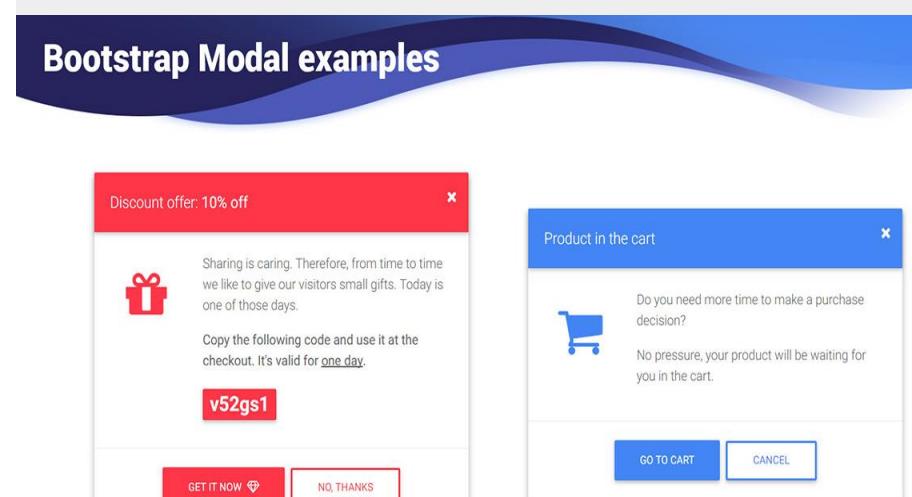
Image:

<https://camo.githubusercontent.com/9dca757e32e3a73c530bae322463bda634071a20d75e4e4e4307181f922cfb57/68747470733a2f2f7261772e6769746875622e636f6d2f6d617263617562652f626f67473747261702d6d61676e6966792f6d61737465722f6578616d706c652f73637265656e73686f742e706e67>

Advance Bootstrap Components

Maodal Window

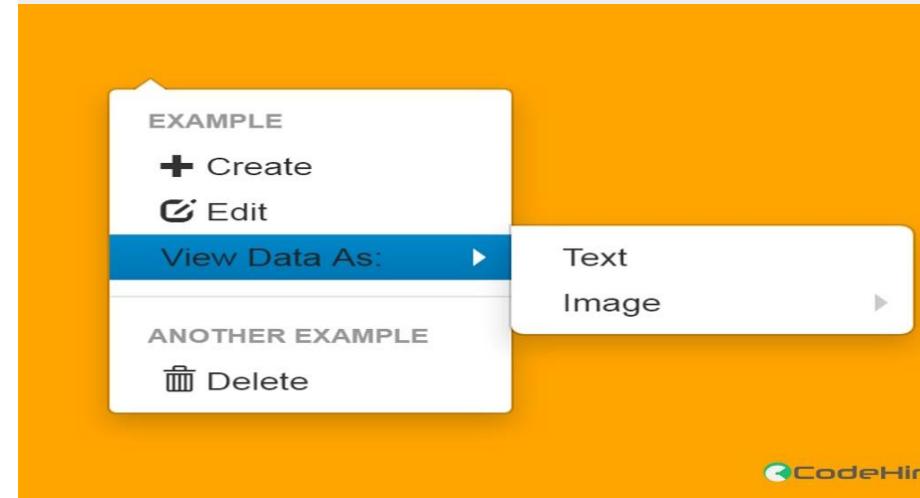
- This plugin extends Bootstrap's native modals to provide additional functionality (responsive, stackable, Ajax...).



Advance Bootstrap Components

Navigation

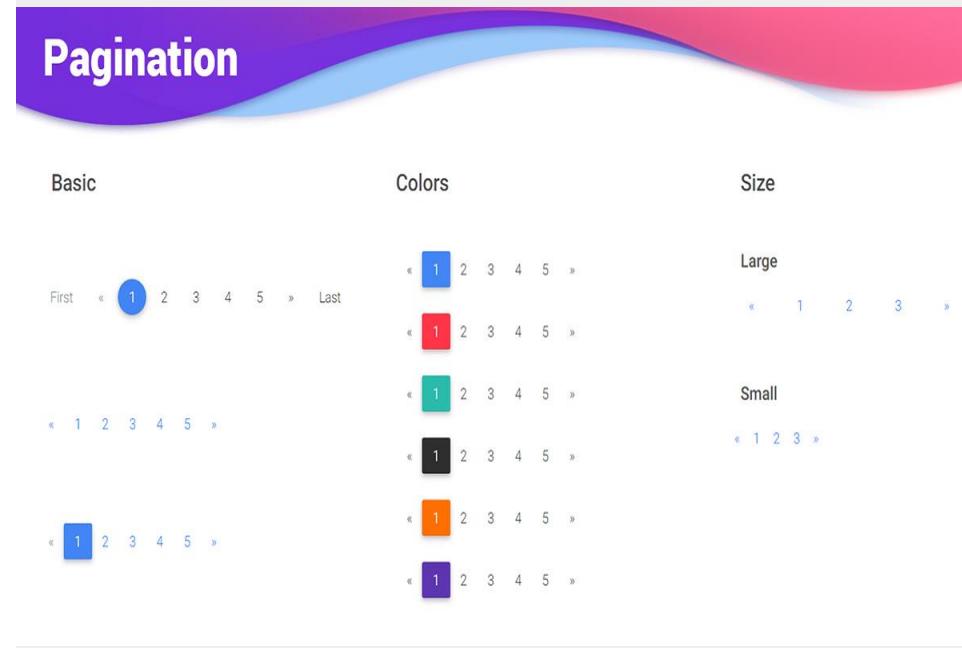
Contextmenu – A context menu plugin for Bootstrap.



Advance Bootstrap Components

Pagination

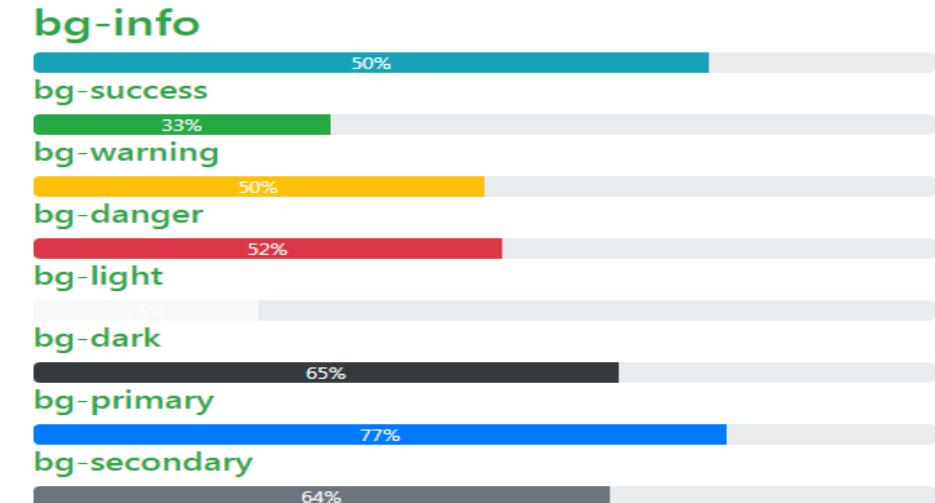
- bootpag – A jQuery plugin helps you create dynamic pagination with Bootstrap.



Advance Bootstrap Components

Progress Bar

- Bootstrap Progressbar – A multi-color progress bar component for Bootstrap.



Advance Bootstrap Components

Ratings

- Bootstrap Star Rating – A jQuery star rating plugin for Bootstrap that supports fractional star fill and RTL input support.

Bootstrap star rating example

Give a rating for Skill:



Give a rating for Knowledge:



Give a rating for PHP:



Advance Bootstrap Components

Social buttons

- Social Buttons for Bootstrap – A pure CSS social sign-in button library.

The Buttons	Available Classes	Different Sizes
 Sign in with App.net	btn-adn btn-bitbucket btn-dropbox btn-facebook btn-flickr btn-foursquare btn-github btn-google btn-instagram btn-linkedin btn-microsoft btn-odnoklassniki btn-openid btn-pinterest btn-reddit btn-soundcloud btn-tumblr btn-twitter	 Sign in with Twitter  Sign in with Twitter  Sign in with Twitter  Sign in with Twitter
 Sign in with Bitbucket		
 Sign in with Dropbox		
 Sign in with Facebook		
 Sign in with Flickr		
 Sign in with Foursquare		
 Sign in with GitHub		
 Sign in with Google		
 Sign in with Instagram		
 Sign in with LinkedIn		
 Sign in with Microsoft		

Check them out in real projects

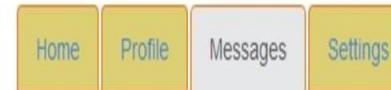
- gae-init
- FOODit

Advance Bootstrap Components

Tabs

- Tabcordion.js – A simple jQuery plugin that transforms a set of Bootstrap tabs into a Bootstrap accordion.

Bootstrap Tabs!



Tab Three

is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Overview of Bootstrap

Bootstrap 5 Utilities

- Background
- Borders
- Color
- Display
- Flex
- Interactions
- Overflow
- Position
- Box shadow
- Sizing
- Text



Image: <https://i.ytimg.com/vi/8scGjrkLkNI/maxresdefault.jpg>

Bootstrap 5 Utilities

Background

- With bootstrap, it's easy to add some background-color CSS rule in an element to convey a specific connotation using its predefined contextual background color classes which follow its built-in theme colors.
- These are composed of a subset of color palettes for generating color schemes.

Class	Description
.bg-primary	Apply background-color: #0D6EFD on an element.
.bg-secondary	Apply background-color: #6C757D on an element.
.bg-success	Apply background-color: #198754 on an element.
.bg-danger	Apply background-color: #DC3545 on an element.
.bg-warning	Apply background-color: #FFC107 on an element.
.bg-info	Apply background-color: #0DCAFO on an element.
.bg-light	Apply background-color: #F8F9FA on an element.
.bg-dark	Apply background-color: #212529 on an element.
.bg-body	Apply Bootstrap's default body background color on an element.
.bg-white	Apply background-color: #FFFFFF on an element.
.bg-transparent	Apply background-color: transparent on an element.

Bootstrap 5 Utilities

Borders

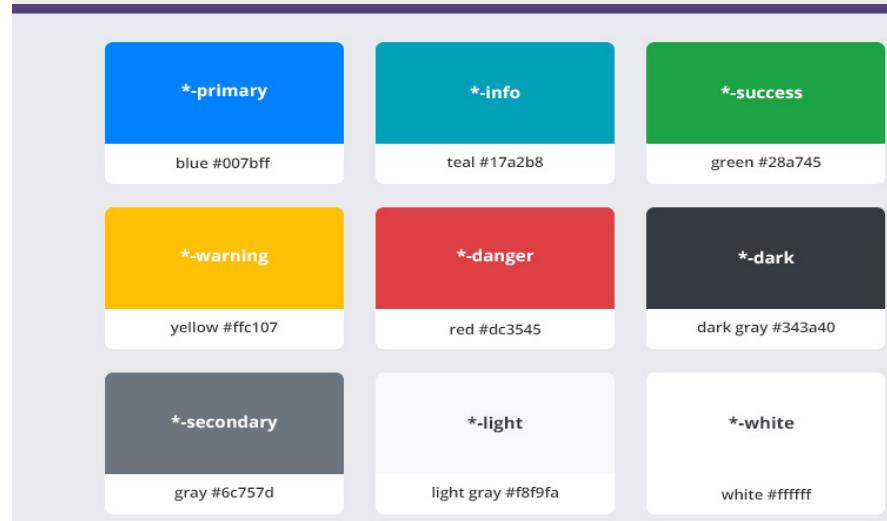
- Another CSS style that is regularly used in any layout design is border. The border properties allow you to define the style, width, and color of an element's border.
- With bootstrap, you can quickly style the border and border-radius of an element by using the predefined border utility classes.

Class	Description
.border	Add a border on all sides of an element.
.border-top	Add a border at the top side of an element.
.border-right	Add a border on the right side of an element.
.border-left	Add a border on the left side of an element.
.border-0	Remove all borders from every side of an element.
.border-top-0	Remove the top side border of an element.
.border-right-0	Remove the right side border of an element.
.border-bottom-0	Remove the bottom side border of an element.
.border-left-0	Remove the left side border of an element.

Bootstrap 5 Utilities

Color

- You can also apply the same contextual colors that we used for the background and border color to every text element through bootstrap text color utility classes.
- These are frequently used for conveying meaning for a particular action or situation on your website or app.



Bootstrap 5 Utilities

Display

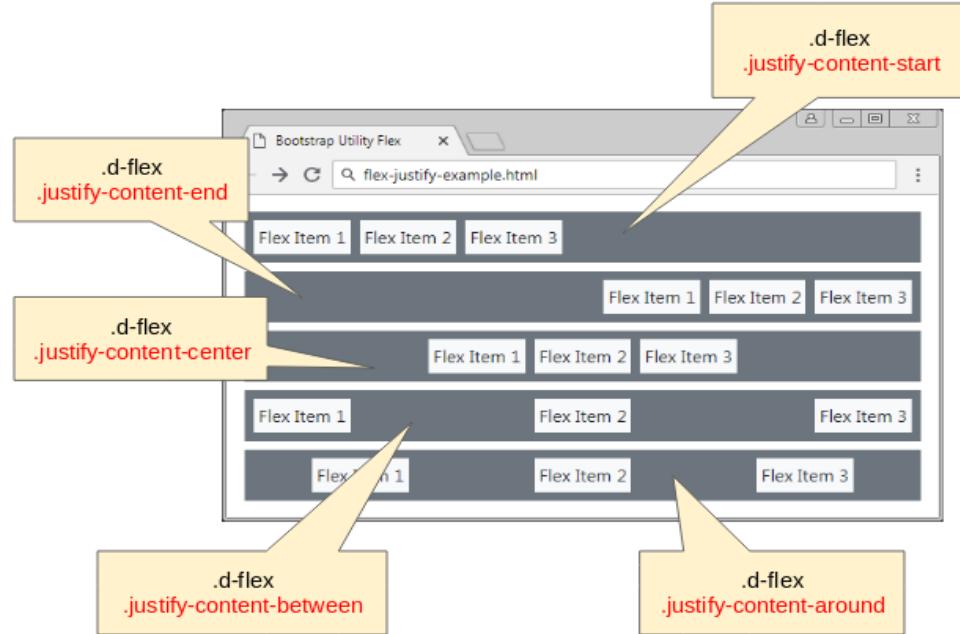
- Another helpful set of bootstrap utility classes that lets you easily and responsively toggle display value of a specific element in a specific breakpoint or viewport are the display utility classes.

Class	Description
.d-none	Hide an element on all screen resolution or viewport size.
.d-inline	Enable an element to create an inline-level block container.
.d-inline-block	Enable an element to create an inline-level block container spanning through the content.
.d-block	Enable an element to create a block-level container.
.d-grid	Enable an element to create a grid-like container.
.d-table	Enable an element to act like a table.
.d-table-row	Enable an element to act like a row of cells in a table.
.d-table-cell	Enable an element to act like a data cell in a table.
.d-flex	Enable an element to act like a block-level flex container.
.d-inline-flex	Enable an element to act like an inline-level flex container.

Bootstrap 5 Utilities

Flex

- Flexbox offers a better way to organize elements in a web page in a predictable manner.
- While it sometimes performs like a float, it offers a lot more than that such as reordering elements and avoiding known issues of float.



Bootstrap 5 Utilities

Interactions

- Bootstrap 5 also provides CSS property controls that allow users to interact with content.
- This determines whether the user can select text or not and if a specific pointer event is active in a text element.
- This doesn't have any effect on content loaded as part of a browser's user interface except in textboxes.

Class	Description
.user-select-all	Enable users to select a text element upon click.
.user-select-auto	Default select behavior on text element.
.user-select-none	Prevent users from selecting a text element including its sub-elements.
.pe-none	Prevents interactions with a pointer (mouse, stylus, touch).
.pe-auto	Default element's pointer behavior.

Bootstrap 5 Utilities

Overflow

- With bootstrap, it's also easy to set your preferred behavior for an element's overflow using the overflow utility classes. For instance, when an element's content is too big to fit in its container context, you can specify whether to clip content in both directions or add a scrollbar on it.

Class	Description
.overflow-auto	If the content is clipped, a scroll-bar will be added to view the rest of the content.
.overflow-hidden	Content is clipped. The rest of the content will be invisible with no scroll-bar provided.
.overflow-visible	Content is not clipped and renders outside the element's box.
.overflow-scroll	Content is clipped. Display scroll-bar whether or not any content is actually clipped to see the rest of the content.

Bootstrap 5 Utilities

Position

- Another useful set of bootstrap utility classes are the position utilities.
- These classes allow you to define the type of positioning method and final location you want an element to behave in a web page.

Class	Description
.position-static	The element is positioned static by default according to the normal flow of the web page.
.position-relative	The element is positioned relative according to the normal flow of the web page which can be affected relatively to itself through the values of top, right, bottom, and left.
.position-absolute	The element is positioned relative to the closest position ancestor with no space generated for the element in the web page. Its position can be affected by the values of top, right, bottom, and left.
.position-fixed	The element is positioned on the same place even if the page is scrolled and is relative to its initial ancestor elements or block. Its final position is determined by the values of top, right, bottom, and left.
.position-sticky	The element is positioned based on the position of the web page scroll. Its position can be affected by the values of top, right, bottom, and left.
.top-*	Use for setting up the vertical top position.
.start-*	Use for setting up the horizontal left position (in LTR).
.bottom-*	Use for setting up the vertical bottom position.
.end-*	Use for setting up the horizontal right position (in LTR).
.[top/start/bottom/end]-0	Set 0 edge position.
.[top/start/bottom/end]-50	Set 50% edge position.
.[top/start/bottom/end]-100	Set 100% edge position.
.translate-middle	Applies the transformations <code>translateX(-50%)</code> and <code>translateY(-50%)</code> to the element which, in combination with the edge positioning utilities, allows you to absolute center an element.

Bootstrap 5 Utilities

Box Shadow

- With the box-shadow CSS property, you can cast shadow effects around an element's frame which is determined by X and Y offsets.
- Bootstrap 5 also has its own out-of-the-box utility classes to quickly add box shadows to your elements.

Class	Description
.shadow-none	Removes box-shadow in an element.
.shadow	Add a basic box-shadow in an element.
.shadow-sm	Add a small amount of box-shadow effect in an element.
.shadow-lg	Add a large amount of box-shadow effect in an element.

Bootstrap 5 Utilities

Sizing

- One of the important factors in web design is the responsive sizes of each element that can span or shrink in size across different screen resolutions or viewport widths.

Class	Description
.w-25	Set the width of an element to 25% to its parent container.
.w-50	Set the width of an element to 50% of its parent container.
.w-75	Set the width of an element to 75% of its parent container.
.w-100	Set the width of an element to 100% of its parent container.
.w-auto	Set the width of an element to auto.
.mw-100	Set the max-width of an element to 100%.
.h-25	Set the height of an element to 25% of its parent container.
.h-50	Set the height of an element to 50% of its parent container.
.h-75	Set the height of an element to 75% of its parent container.
.h-100	Set the height of an element to 100% of its parent container.
.h-auto	Set the height of an element to auto.
.mh-100	Set the max-height of an element to 100%.
.vw-100	Set the width of an element to 100% of the width of the screen resolution or viewport.
.min-vw-100	Set the min-width of an element to 100% of the width of the screen resolution or viewport.
.vh-100	Set the height of an element to 100% of the height of the screen resolution or viewport.
.min-vh-100	Set the min-height of an element to 100% of the height of the screen resolution or viewport.

Bootstrap 5 Utilities

Spacing

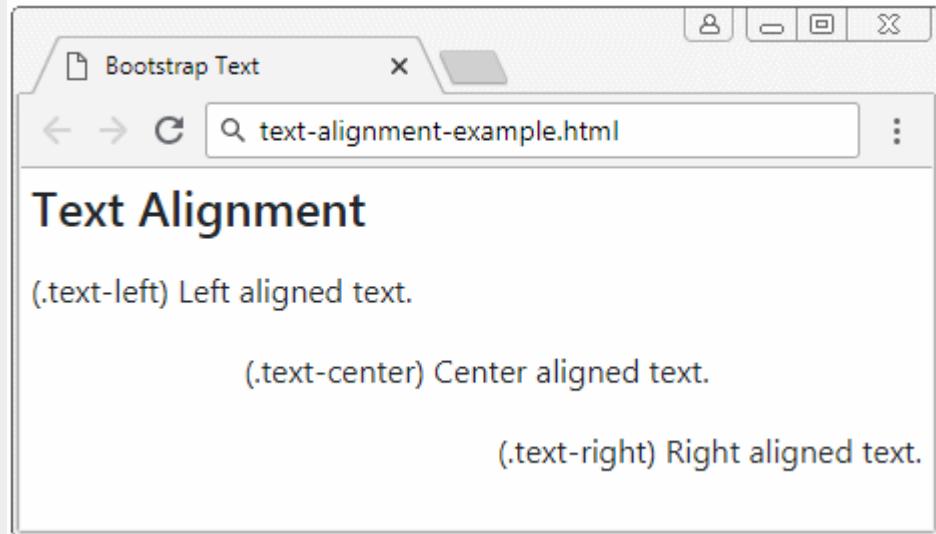
- Bootstrap offers a variety of shorthand responsive margin, padding, and gap utility classes to modify an element position or appearance.
- The standard measurement for each margin, padding and gap classes are ranging from .25rem to 3rem.

Class	Description
.mt-*	Set a top margin on an element.
.mb-*	Set a bottom margin on an element.
.ml-*	Set a left margin on an element.
.mr-*	Set a right margin on an element.
.mx-*	Set a left and right margin on an element.
.my-*	Set a top and bottom margin on an element.
.pt-*	Set a top padding on an element.
.pb-*	Set a bottom padding on an element.
.pl-*	Set a left padding on an element.
.pr-*	Set a right padding on an element.
.px-*	Set a left and right padding on an element.
.py-*	Set a top and bottom padding on an element.
.gap-*	Set a gap on the parent grid container

Bootstrap 5 Utilities

Text

- Bootstrap 5 also added a few new extra utility classes for common text or link control.
- With these classes, you can easily realign text to components, wrap text, modify the font size or weight, transform case and more.



jQuery

(6 hours)

In this section, we will discuss:

- Introduction to jQuery
- How to add jQuery to web pages
- jQuery Syntax
- jQuery Selectors
- jQuery Attributes
- jQuery- Dom Traversing
- jQuery- CSS selectors methods
- jQuery Effect
- jQuery Animations

Introduction to jQuery

What is jQuery

- jQuery is a lightweight, "write less, do more", JavaScript library. The purpose of jQuery is to make it much easier to use JavaScript on your website.
- jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

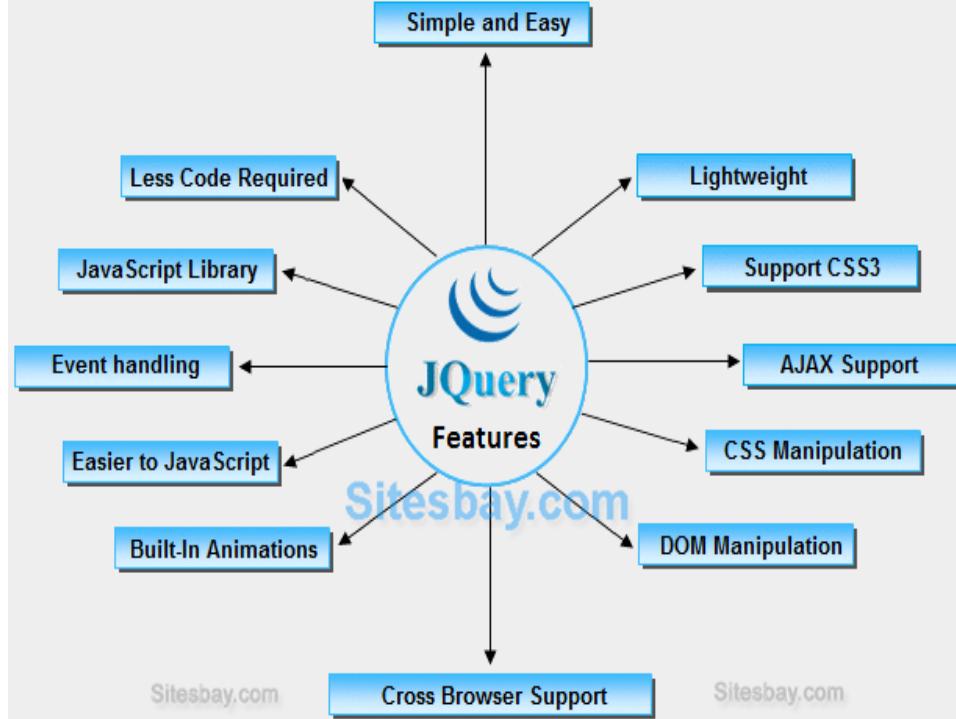


Introduction to jQuery

What is jQuery

The jQuery library contains the following features:

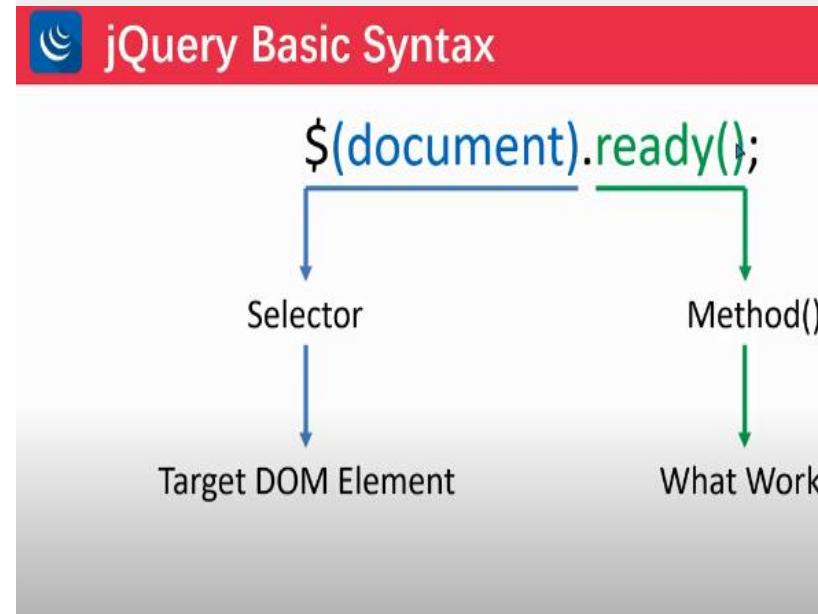
- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- Utilities



Introduction to jQuery

Why jQuery

- A \$ sign is to define/access jQuery
- A (selector) is to “query (or find)” HTML elements in html page
- A jQuery action() is the action to be performed on the selected element(s)



Introduction to jQuery

Advantages of jQuery

- Wide range of plug-ins. jQuery allows developers to create plug-ins on top of the JavaScript library.
- Large development community
- It has a good and comprehensive documentation
- It is a lot more easy to use compared to standard javascript and other javascript libraries.

Advantages of jQuery

- There are lots of other JavaScript frameworks out there, but jQuery seems to be the most popular, and also the most extendable
- Many of the biggest companies on the Web use jQuery, such as:
 - Google
 - Microsoft
 - IBM
 - Netflix

Introduction to jQuery

Disadvantages of jQuery

- While JQuery has an impressive library in terms of quantity, depending on how much customization you require on your website, the functionality may be limited thus using raw javascript may be inevitable in some cases.
- The JQuery javascript file is required to run JQuery commands, while the size of this file is relatively small (25-100KB depending on the server).

jQuery Limitations

Disadvantages include:

- Exceed memory on mobile browsers (See <http://jquerymobile.com/>).
- May not be necessary with modern browsers. (IE9 is catching up).
- Overuse may slow down the client computer.

Introduction to jQuery

How to add jQuery to web pages

- There are several ways to start using jQuery on your web site. You can:
 - Download the jQuery library from [jQuery.com](http://jquery.com)
 - Include jQuery from a CDN, like Google

Adding jQuery to your webpages

2. Alternatively, you can include the jQuery library from a CDN (Content Delivery Network) such as Google or Microsoft.

GOOGLE:

```
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js">
</script>
</head>
```

MICROSOFT:

```
<head>
<script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.11.1.min.js"> </script>
</head>
```

Introduction to jQuery

jQuery Selectors

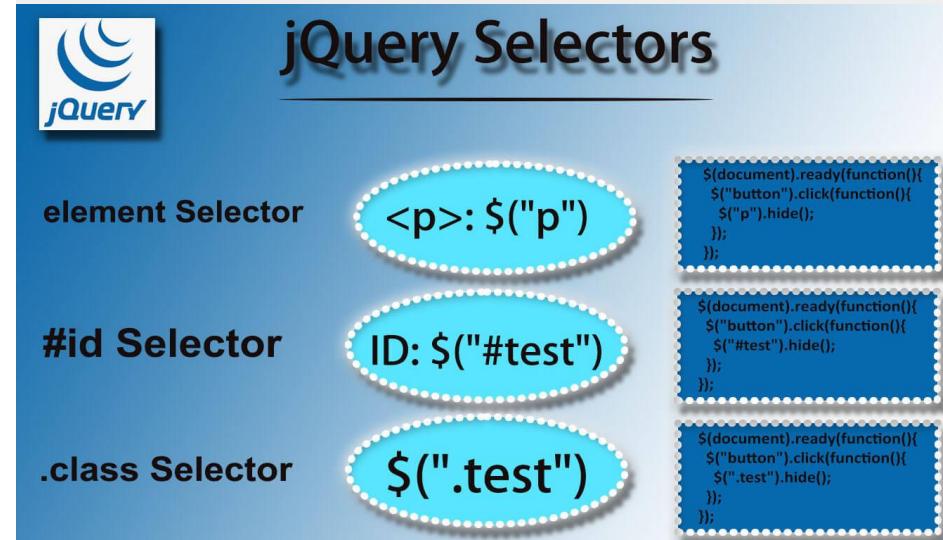
- jQuery selectors allow you to select and manipulate HTML element(s).
- All selectors in jQuery start with the dollar sign and parentheses: `$()`.



jQuery Selectors

Element Selector

- The jQuery element selector selects elements based on the element name.
- Selects all elements with the given tag name.



The diagram illustrates three types of jQuery selectors: Element Selector, #id Selector, and .class Selector. Each selector is shown with its corresponding jQuery code and a visual representation. The code snippets are enclosed in dashed blue boxes.

- element Selector**: `<p>: $("p")`
- #id Selector**: `ID: $("#test")`
- .class Selector**: `$(".test")`

```
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide();
  });
});

$(document).ready(function(){
  $("button").click(function(){
    $("#test").hide();
  });
});

$(document).ready(function(){
  $("button").click(function(){
    $(".test").hide();
  });
});
```

jQuery Selectors

#id Selector

- The jQuery #id selector uses the id attribute of an HTML tag to find the specific element.
- An id should be unique within a page, so you should use the #id selector when you want to find a single, unique element.

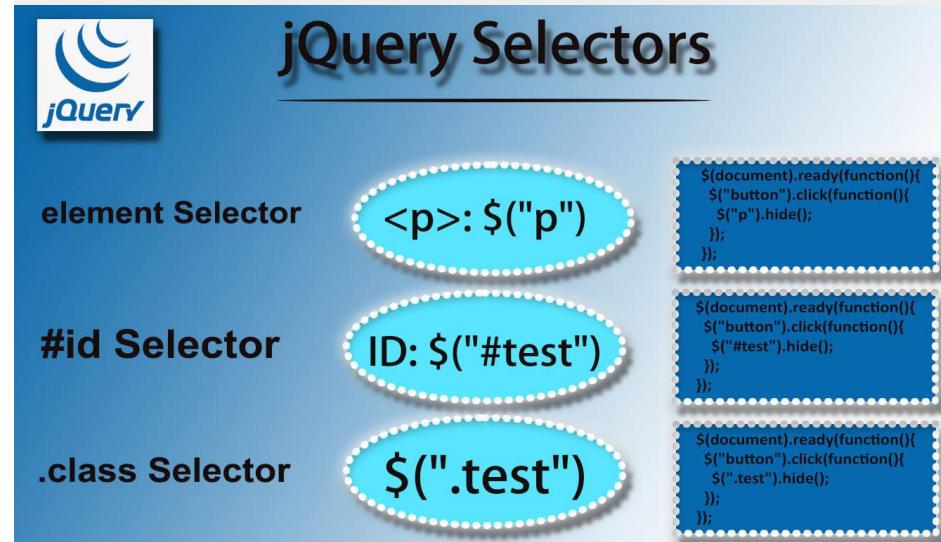
The slide features a blue header with the 'jQuery' logo and the title 'jQuery Selectors'. Below the header, there are three sections: 'element Selector', '#id Selector', and '.class Selector', each associated with a blue oval containing a selector expression. To the right of each selector are three code snippets in a box:

- element Selector**: `<p>: $("p")`
Code snippet: `$(document).ready(function(){ $("button").click(function(){ $("p").hide(); });});`
- #id Selector**: `ID: $("#test")`
Code snippet: `$(document).ready(function(){ $("button").click(function(){ $("#test").hide(); });});`
- .class Selector**: `$(".test")`
Code snippet: `$(document).ready(function(){ $("button").click(function(){ $(".test").hide(); });});`

jQuery Selectors

.class Selector

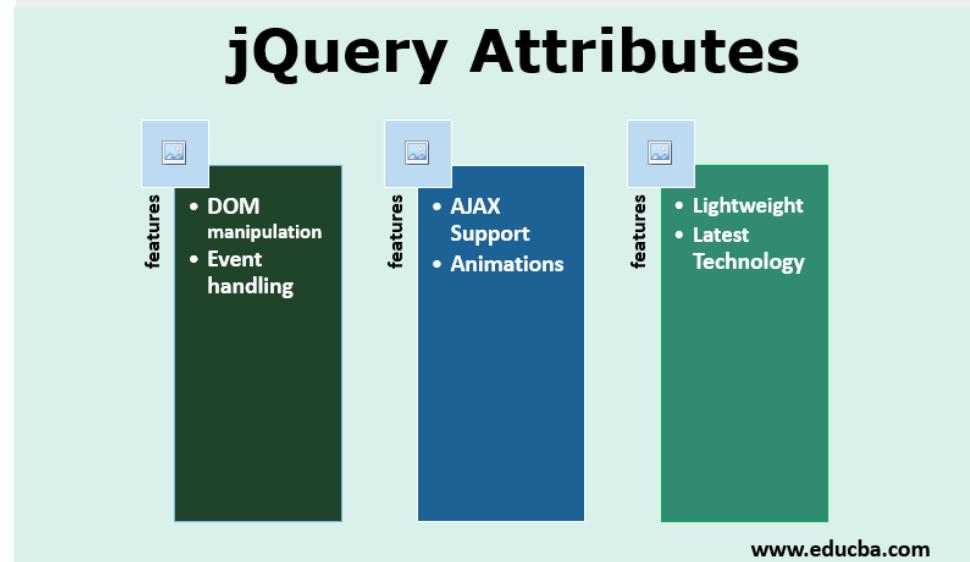
- The jQuery .class selector finds elements with a specific class.
- To find elements with a specific class, write a period character, followed by the name of the class:



Introduction to jQuery

jQuery Attributes

- Some of the most basic components we can manipulate when it comes to DOM elements are the properties and attributes assigned to those elements.

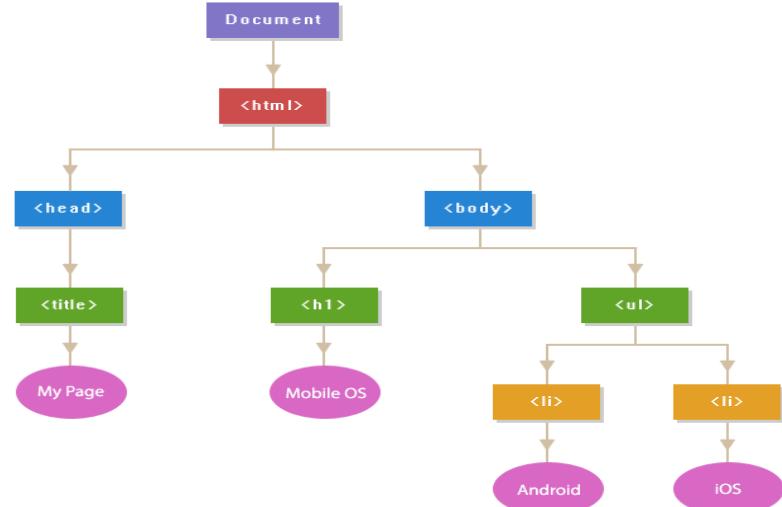


Introduction to jQuery

jQuery Attributes

Most of these attributes are available through JavaScript as DOM node properties. Some of the more common properties are –

- className
- tagName
- id
- href
- title
- rel
- src



jQuery Attributes

Get Attribute Value

- The attr() method can be used to either fetch the value of an attribute from the first element in the matched set or set attribute values onto all matched elements.

```
/* html */
<a data-id="123">link</a>

/* js */
$(this).attr("data-id") // returns string "123"

$(this).data("id") // returns number 123 (jQuery >= 1.4.3 only)
```

jQuery Attributes

Set Attribute Value

- The `attr(name, value)` method can be used to set the named attribute onto all elements in the wrapped set using the passed value.

Setting attributes without value using jQuery

Get div class name

Get paragraph style

This is div.

This is paragraph.

This is div.

jQuery Attributes

Applying Styles

- The addClass(classes) method can be used to apply defined style sheets onto all the matched elements.
- You can specify multiple classes separated by space.

This is first paragraph.

This is second paragraph.

Introduction to jQuery

jQuery- DOM Traversing

- jQuery is a very powerful tool which provides a variety of DOM traversal methods to help us select elements in a document randomly as well as in sequential method.
- Most of the DOM Traversal Methods do not modify the jQuery object and they are used to filter out elements from a document based on given conditions.



Introduction to jQuery

jQuery- DOM Traversing

Most of the DOM Traversal Methods do not modify the jQuery object and they are used to filter out elements from a document based on given conditions.

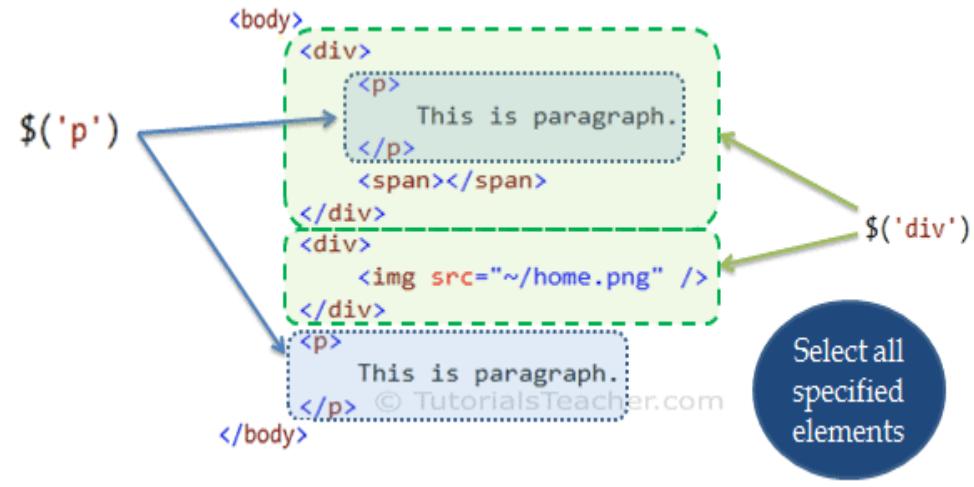
- Find elements by index
- Filtering out element
- Locating descendent element
- jQuery DOM filter method
- jQuery DOM traversing method



Introduction to jQuery

jQuery- CSS Selector

- The jQuery library supports nearly all of the selectors included in Cascading Style Sheet (CSS) specifications 1 through 3, as outlined on the World Wide Web Consortium's site.

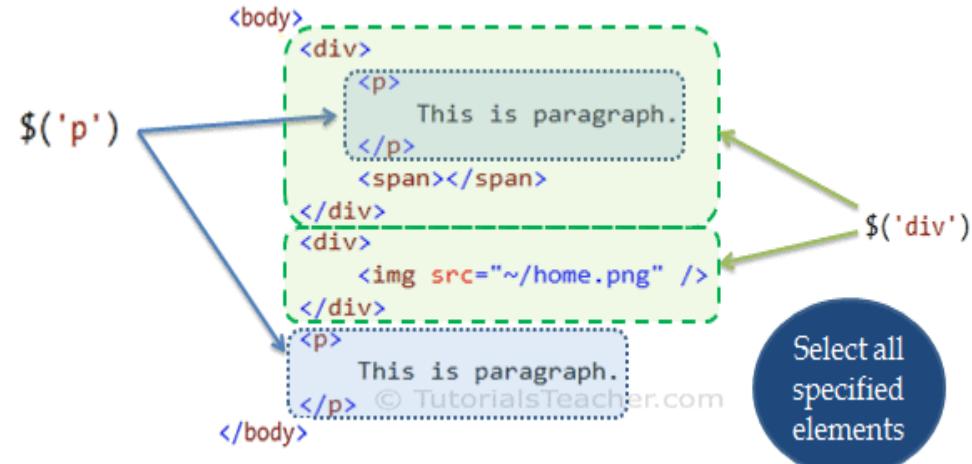


Introduction to jQuery

jQuery- CSS Selector

Most of the JQuery CSS Methods do not modify the content of the jQuery object and they are used to apply CSS properties on DOM elements.

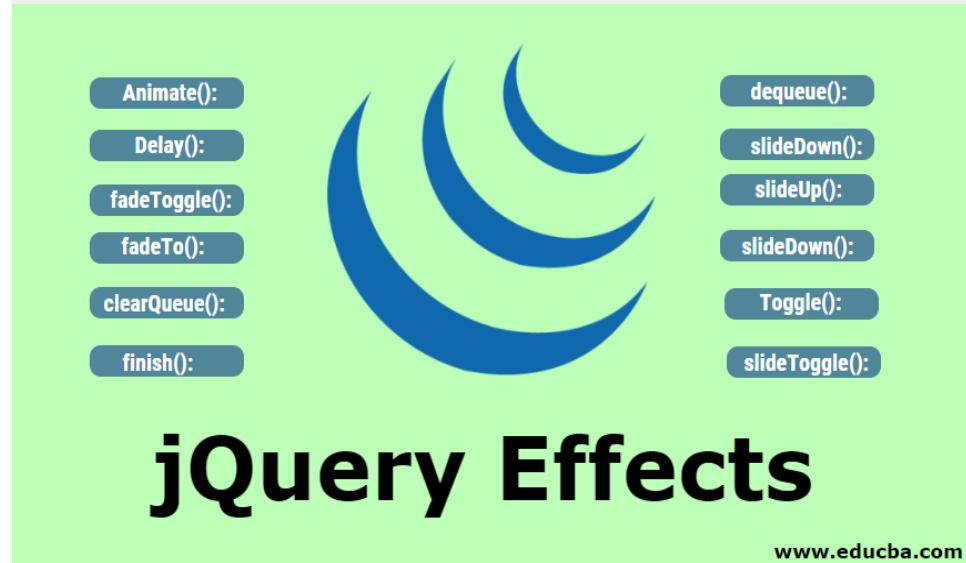
- Apply CSS Properties
- Apply multiple CSS Properties
- Setting Element width & height
- jQuery CSS method



Introduction to jQuery

jQuery- Effect

- jQuery provides a trivially simple interface for doing various kind of amazing effects.
- jQuery methods allow us to quickly apply commonly used effects with a minimum configuration.

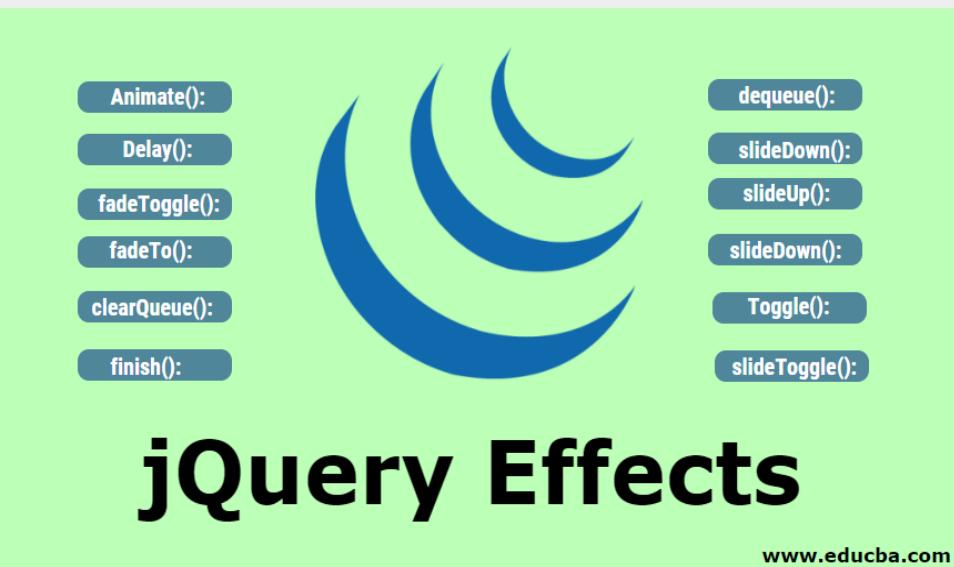


Introduction to jQuery

jQuery- Effect

This content covers all the important jQuery methods to create visual effects.

- jQuery hide() and show()
- jQuery toggle
- jQuery fade
- jQuery slide



Introduction to jQuery

jQuery- Animation

- The jQuery animate() method performs custom animation using element's style properties.
- The animate() method changes existing style properties to the specified properties with motion.



jQuery Animate
Method

.animate()

Basic Concepts of DBMS

In this section, we will discuss:

- Purpose of database systems
- Data abstraction
- Database Users
- Data Independence (Logical & Physical)
- Instance & Schemes
- Three layered Architecture of DBMS
- Different Levels of Abstraction.
- Data Modeling
- E-R Modeling
- Logical Model: Object & Record based – Object oriented model - Entity relationship models
- Entity sets & relationships sets

In this section, we will discuss:

- Concept of attributes and relationships
- Introduction to mapping constraints.
- Basic Concepts of ER Model in DBMS
- Introduction to DBMS
- Structure of DBMS
- Relational Models
- Introduction to Hierarchical Model and Network Model
- Introduction to RDBMS and Relational Models
- Introduction to relational algebra and relational calculus
- Understanding database technologies

In this section, we will discuss:

- Relational Data Structure
- Keys and Relational Data Manipulation
- Relational Algebra
- Relational Algebraic Operations
- Set Operations
- Fundamental Operations
- Relational Calculus
- Data Definition Language
- Operators: select, project, join, rename etc.

Purpose of database systems

What is Database?

What is DBMS and types of DBMS ?

DBMS vs Flat File System

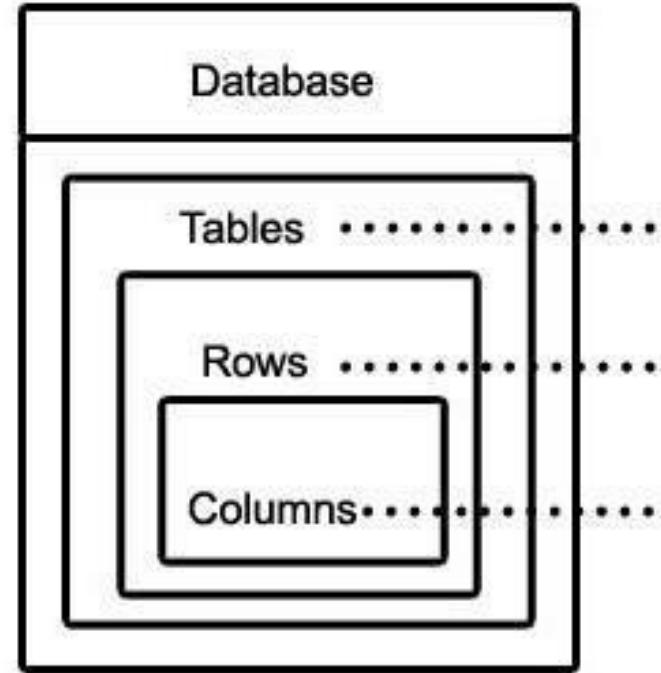


Image Source:

<https://www.google.com/url?sa=i&source=imgres&cd=&cad=rja&uact=8&ved=2ahUKEwjKg9PLw9PoAhXD>

Purpose of database systems

What is Database?

- A database is a data structure that stores organized information.
- Databases contain multiple tables, which may each include several different fields.
- For example, a company database may include tables for products, employees, and financial records. Each of these tables would have different fields that are relevant to the information stored in the table.

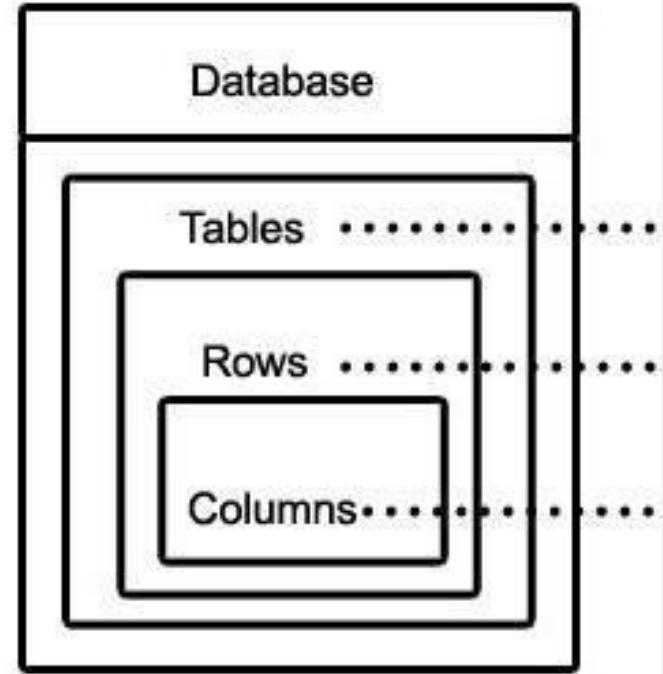


Image Source:

<https://www.google.com/url?sa=i&source=imgres&cd=&cad=rja&uact=8&ved=2ahUKEwjKg9PLw9PoAhXD>

Purpose of database systems

What is DBMS?

- DBMS or Database Management System is a software application used to access, create, and manage databases.
- With the help of DBMS, you can easily create, retrieve and update data in databases.
- A DBMS consists of a group of commands to manipulate the database and acts as an interface between the end-users and the database.

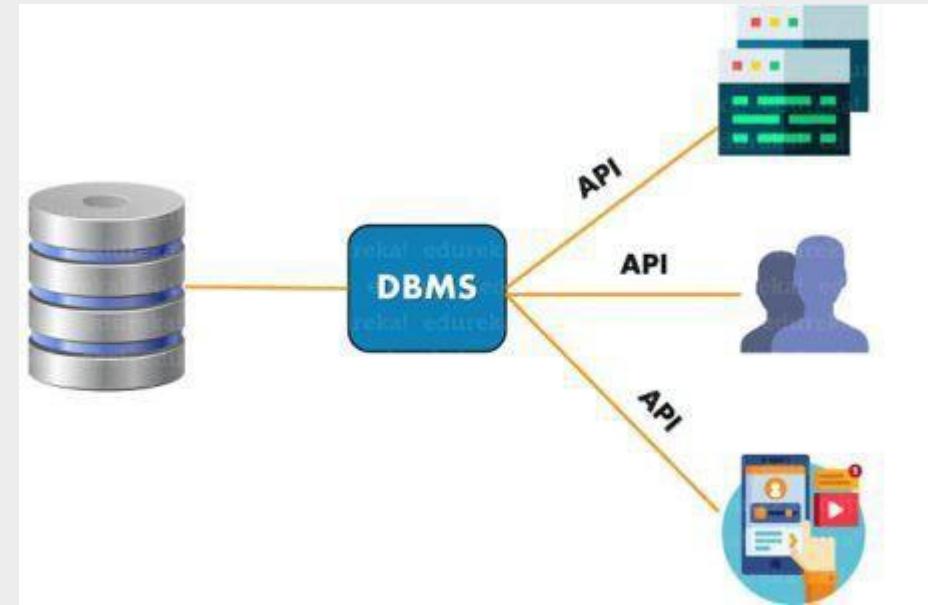


Image Source::

<https://d1jnx9ba8s6j9r.cloudfront.net/blog/wp-content/uploads/2019/10/DATABASES-457x300.png>

Purpose of database systems

Types of DBMS

- File Processing System
- Relational Database Management System (RDBMS)
- Network Database Management System (NDBMS)
- Hierarchical Database Management System(HDBMS)
- Object-Oriented Database Management System(OODBMS)

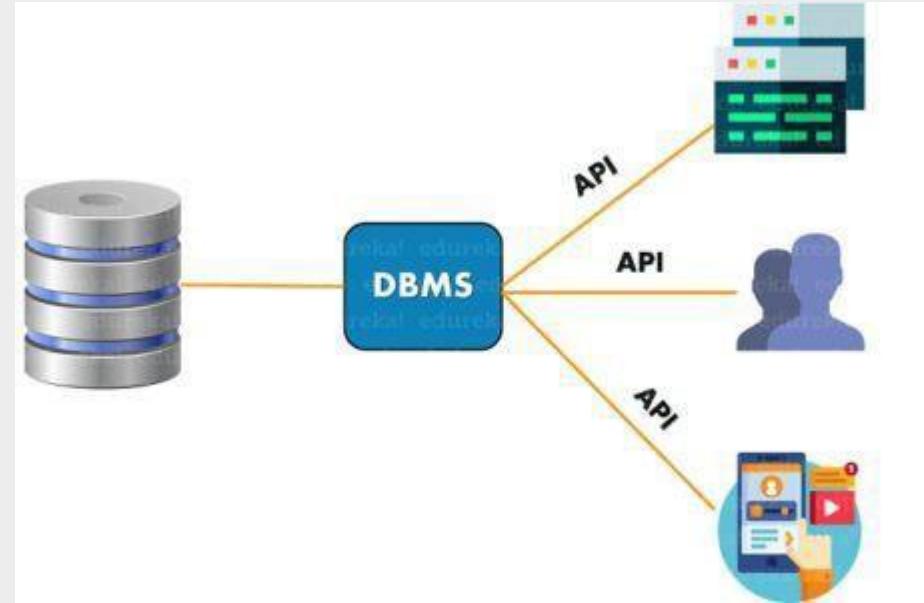


Image Source::

<https://d1jnx9ba8s6j9r.cloudfront.net/blog/wp-content/uploads/2019/10/DATABASES-457x300.png>

Types of DBMS

File Processing System

- File Processing System is a traditional approach of storing and managing data.
- Data Storage was being done by storing the information in files present in the computer.

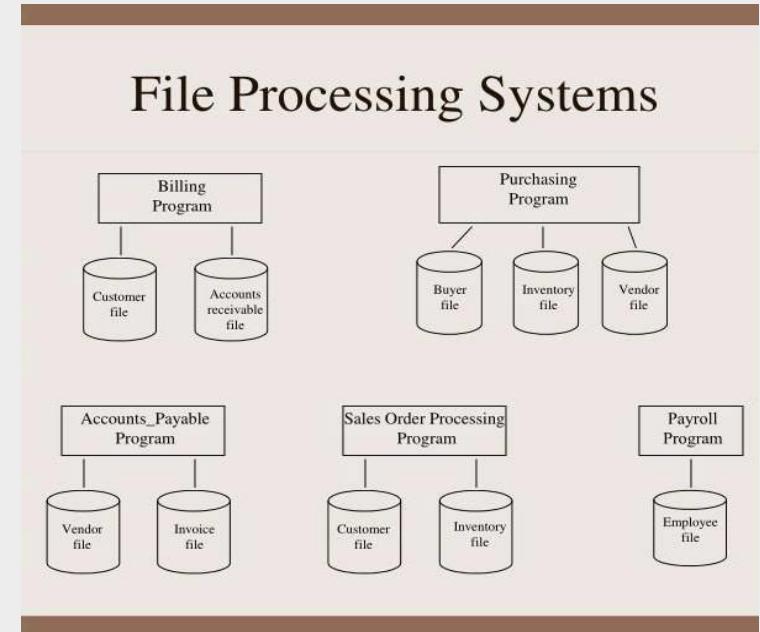


Image Source::
<https://image.slideserve.com/1272110/file-processing-systems-n.jpg>

Types of DBMS

Relational Database Management System (RDBMS)

- In Relational Database Management Systems, unlike file processing system the data is stored in the tables(having rows and columns)
- Any table can be related or linked to any other table using certain concepts.

Attribute	Value
A	1
B	2
C	3

Types Of DBMS : RDBMS

Types of DBMS

Network Database Management System(NDBMS)

- Network Database Management System is another type which was very popular in the late 1970's.
- These databases were known to be very adaptable and flexible in terms of data storage because of their data storage architecture.

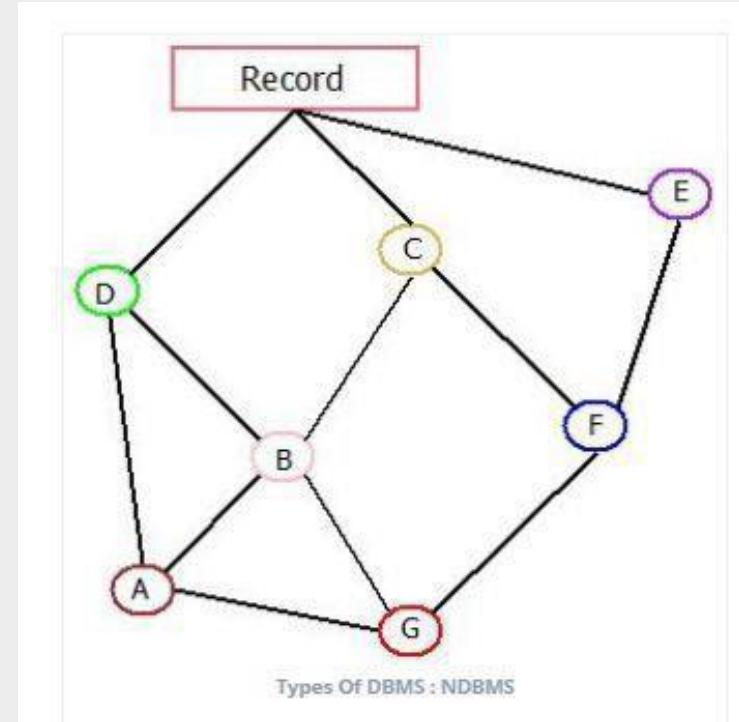


Image Source::

[https://www.minigranth.com/wp-content/u
ploads/2018/04/NDBMS.jpeg](https://www.minigranth.com/wp-content/uploads/2018/04/NDBMS.jpeg)

Types of DBMS

Hierarchical Database Management System(HDBMS)

- The Hierarchical Database Management System works on the concept similar to that of “Concept Hierarchy” .
- The whole database is structured and architecture in the form similar to the structure of tree, making it one of the simplest and fastest database present out there.

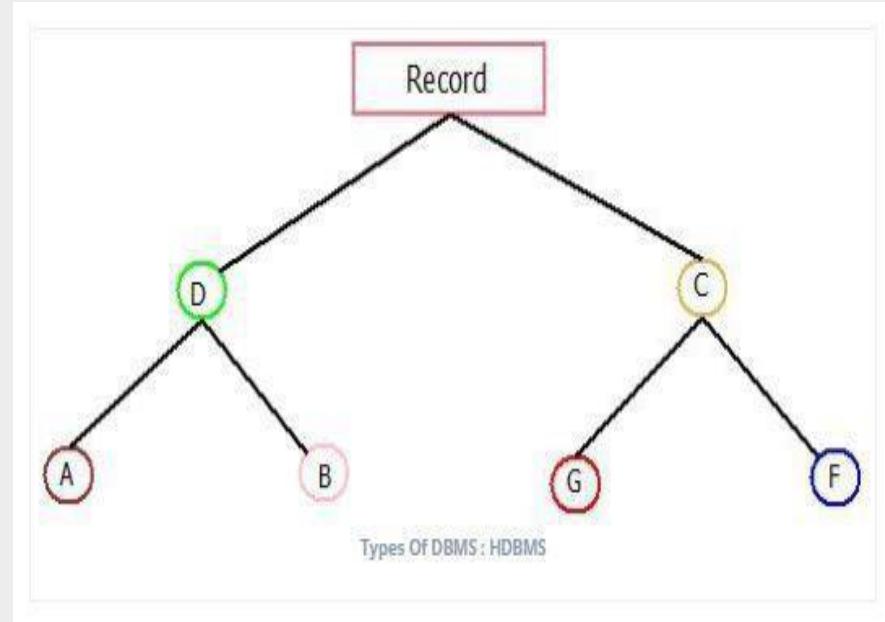


Image Source:
<https://www.minigranth.com/wp-content/uploads/2018/04/HDBMS.jpeg>

Types of DBMS

Object-Oriented Database Management Systems (OODBMS)

- They are different from other databases as Object-Oriented Database Management Systems revolves around “Object”(An object here referred to any real world entity)
- It is based upon the concept of various object oriented languages such as “C++, Java” etc.

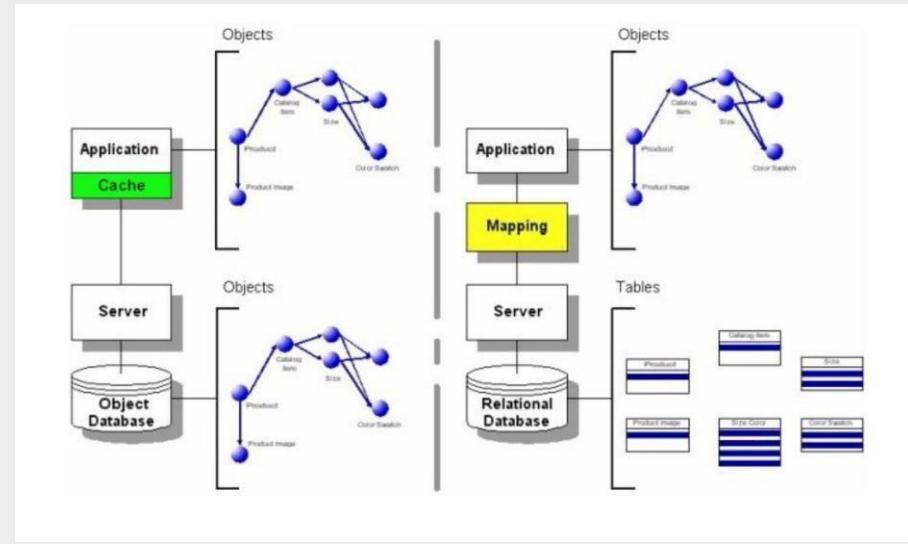
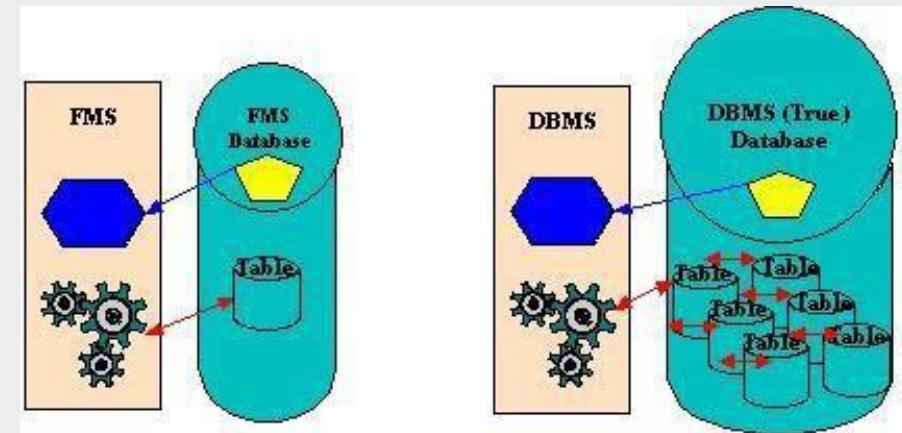


Image Source::

https://miro.medium.com/max/2988/1*C3XqvO_wmVzxBcZngRCjPg.png

Purpose of database systems

File Management System	Database Management System
File System is a general, easy-to-use system to store general files which require less security and constraints.	Database management system is used when security constraints are high.
Data Redundancy is more in file management system.	Data Redundancy is less in database management system.
Data Inconsistency is more in file system.	Data Inconsistency is less in database management system.
Centralisation is hard to get when it comes to File Management System.	Centralisation is achieved in Database Management System.



FMS versus DBMS Comparison Diagram (Figure 1)

Data abstraction

What is Data Abstraction

- Data system are made up of complex data structure
- The use of data abstraction relates to the user interaction with database
- The developer hide internal detail from the users.
- The process of hiding irrelevant detail from user is called data abstraction.



Data Abstraction

- **Separation of concept (interface) from implementation**
- **Main program needn't know internal implementation of list**
 - Program is given interface to conceptual operations
 - Does need to be aware of limitations of implementation
- **List implementation needn't know details of main program's data**
 - Data in list entry is encapsulated in data type defined by main program
 - In dynamic memory implementations, list entries needn't even be all the same data type

Image Source::

<https://blog.oureducation.in/data-abstraction-and-encapsulation/#!prettyPhoto/0/>

Data abstraction

Example of Data Abstraction

- Let's say we are storing customer information in a customer table.
- At **physical level** these records can be described as blocks of storage (bytes, gigabytes, terabytes etc.) in memory.
- These details are often hidden from the programmers.

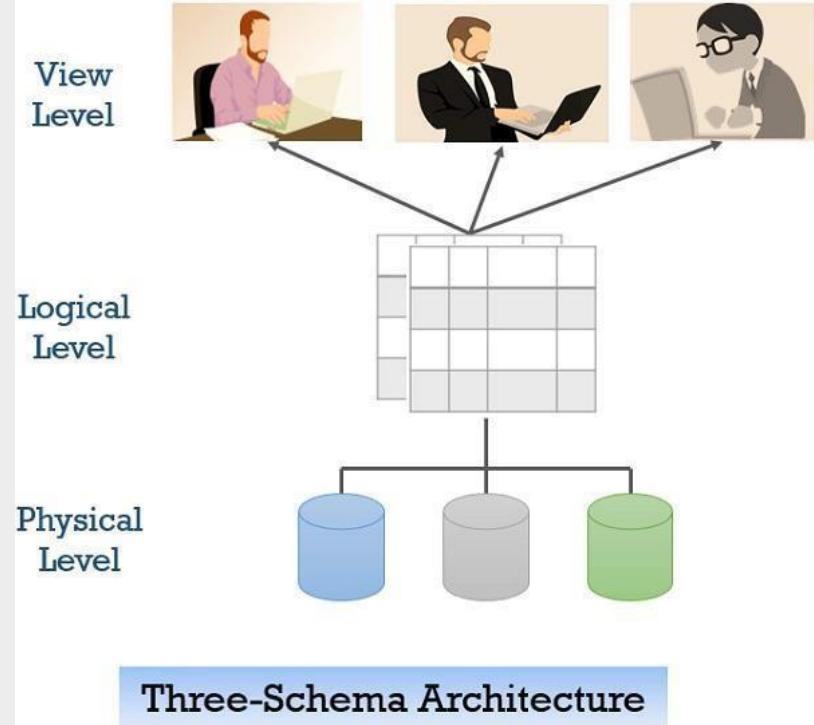


Image Source::
<https://binaryterms.com/wp-content/uploads/2019/11/View-of-data-three-schema-architecture.jpg>

Data abstraction

Example of Data Abstraction

- At the **logical level** these records can be described as fields and attributes along with their data types, their relationship among each other can be logically implemented.
- The programmers generally work at this level because they are aware of such things about database systems.

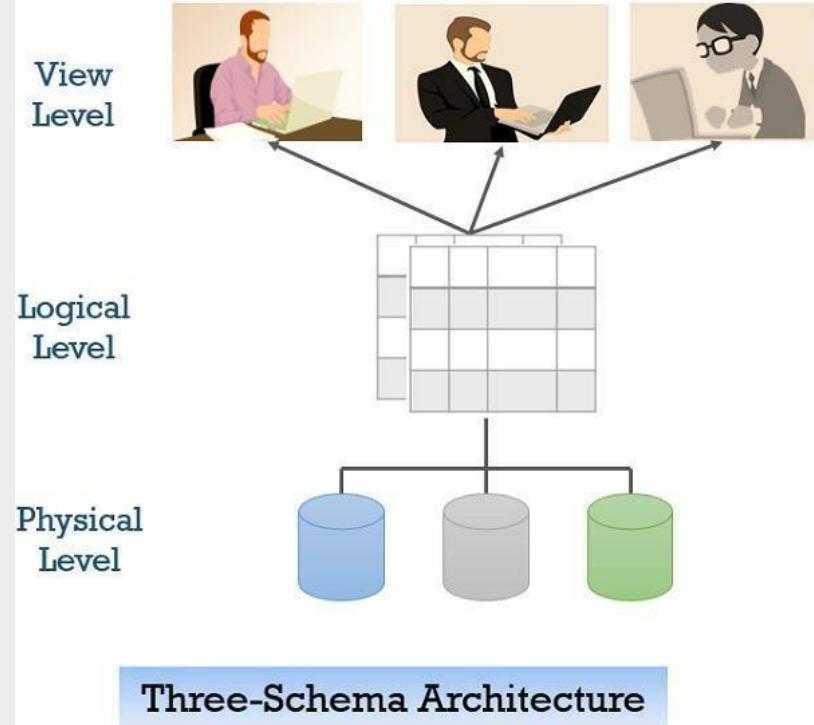


Image Source::
<https://binaryterms.com/wp-content/uploads/2019/11/View-of-data-three-schema-architecture.jpg>

Data abstraction

Example of Data Abstraction

- At **view level**, user just interact with system with the help of GUI and enter the details at the screen.
- They are not aware of how the data is stored and what data is stored; such details are hidden from them.

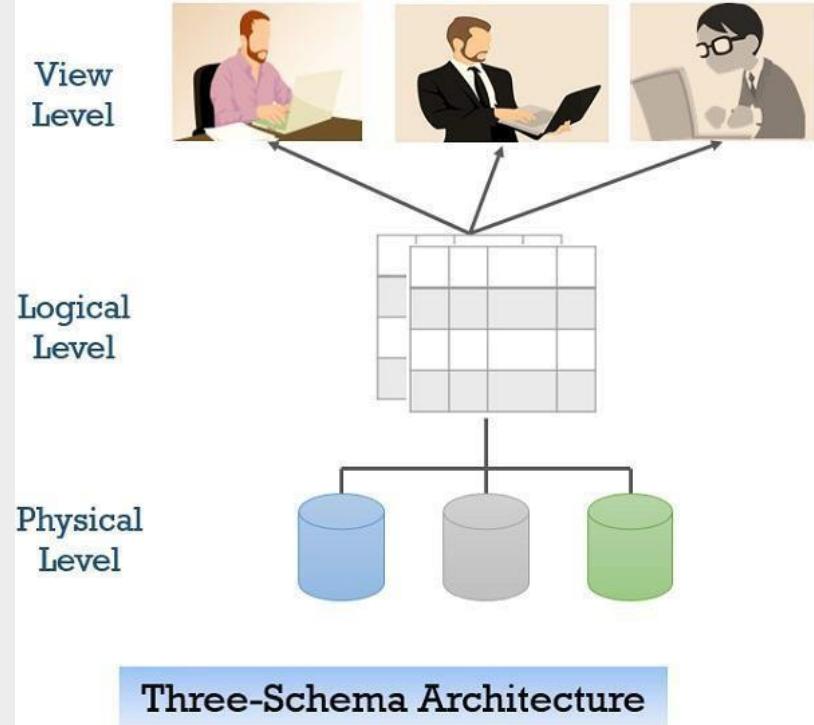


Image Source::
<https://binaryterms.com/wp-content/uploads/2019/11/View-of-data-three-schema-architecture.jpg>

Database Users

What is DBA?

- A database administrator (DBA) is the information technician responsible for directing or performing all activities related to maintaining a successful database environment.
- A DBA makes sure an organization's database and its related applications operate functionally and efficiently.

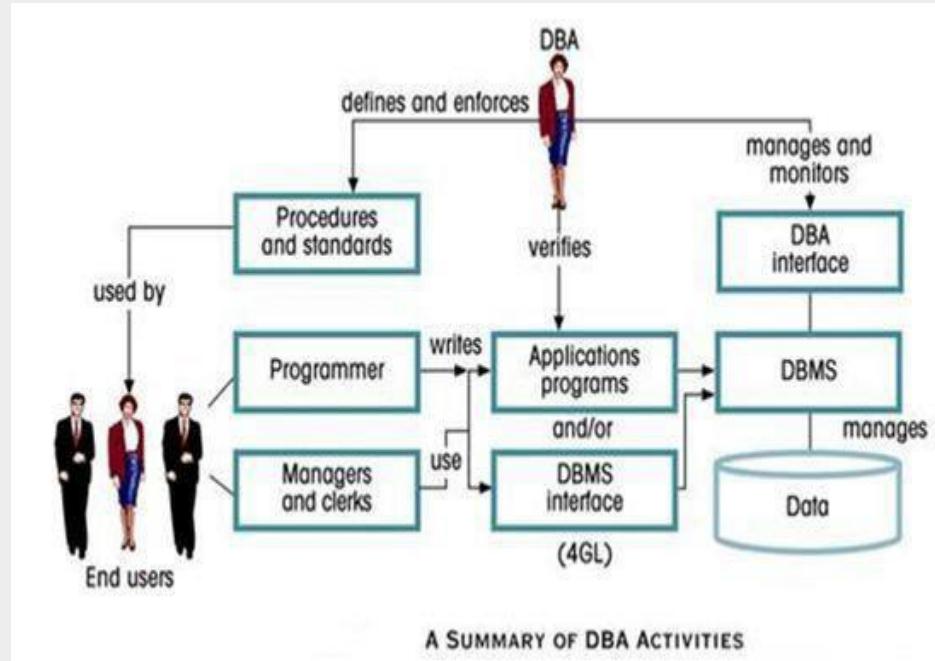


Image Source::
<https://www.myreadingroom.co.in/images/stories/docs/dbms/summary%20of%20dba%20activities.JPG>

Data Independence

What is Conceptual/Logical Schema?

- A **conceptual schema** is a high-level description of informational needs underlying the design of a database.
- It typically includes only the main concepts and the main relationships among them.
- Typically this is a first-cut model, with insufficient detail to build an actual database. This level describes the structure of the whole database for a group of users.
- The conceptual model is also known as the data model that can be used to describe the conceptual schema when a database system is implemented. It hides the internal details of physical storage and targets on describing entities, datatype, relationships.

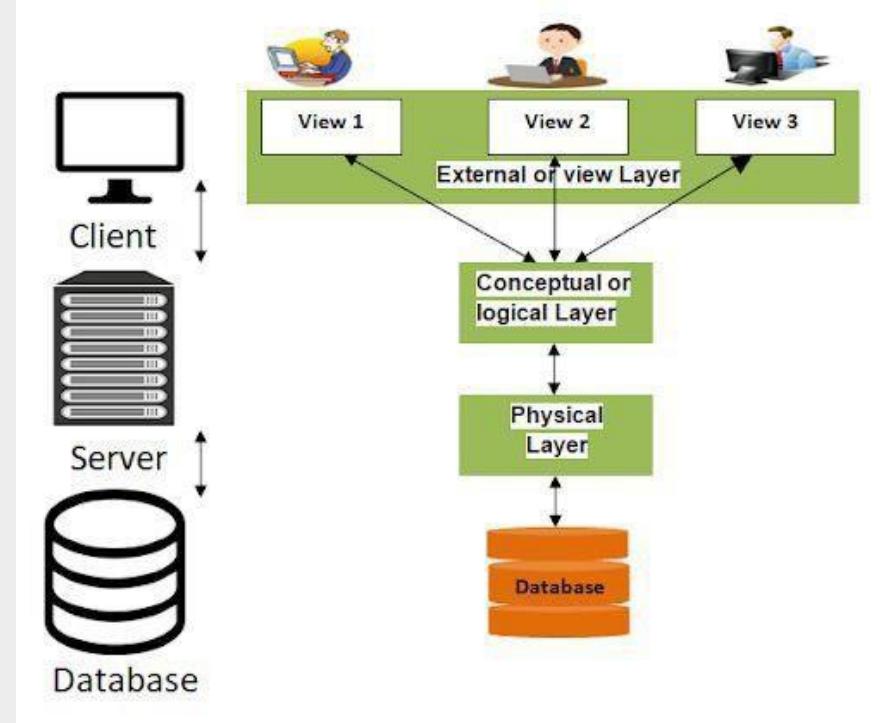


Image Source::

https://lh5.googleusercontent.com/proxy/z36HD9v_To7vjP1x-l-T9x_tjYK9xhqE-tlNenp55ykn54dGzLSCd6CPAVkis_h8njDRFqcnFw3GZpS9QbQU91qRCbA--A87zcVaK-8tZljQrlfTeIY1rcfse2qEVWu

Instance & Schemes

BASIS FOR COMPARISON	SCHEMA	INSTANCE
Basic	Description of the database.	Snapshot of a database at a specific moment.
Change occurrence	Rare	Frequent
Initial state	Empty	Always have some data.

Analogy

		Real World	Database		
Scheme			Template for a Table		
Instances			Data-filled Tables		
P-ID	Name	Pname			
102366	Smith	John		Pname	
102357	Potter	Harry		William	
			523646	Wood	Lucinda

Image Source::

https://lh6.googleusercontent.com/proxy/NmDIYKVTeT5KJ4cViogUgX8dpSmxe0VqqjILC0RXvd5oBtKoLd9gGvmWHU_088z8K2aKLQI-o6Qn74wprAmY14U6cHpYmaf5-hDK

Instance & Schemes

What is Sub-schema?

- **Subschema** is a sub part of a schema and inherits same properties of schema.
- Subschema describes different view of the database.
- Subschema is an application programmer's or user view of data item types and records type.
- *For example:* Suppose a table employee, programmer has access of all columns of table employee but user has access of only two or three columns of table employee. Subschema describes different view of the database.

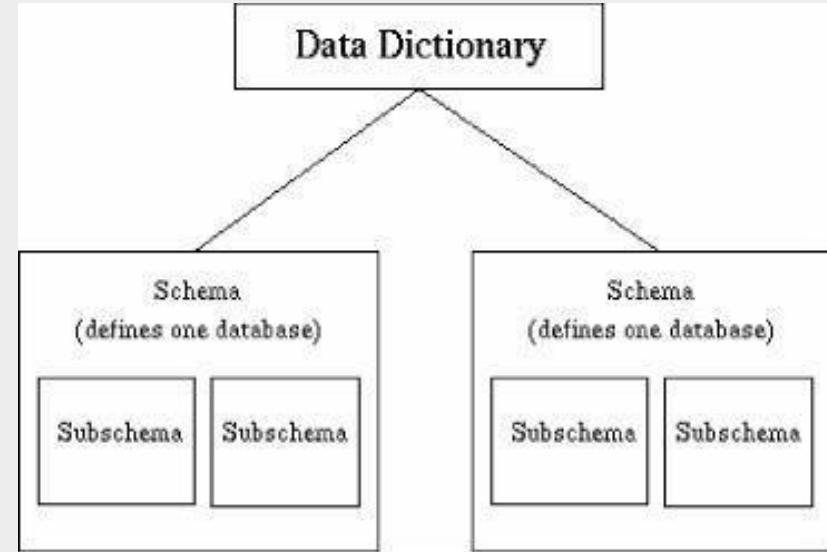


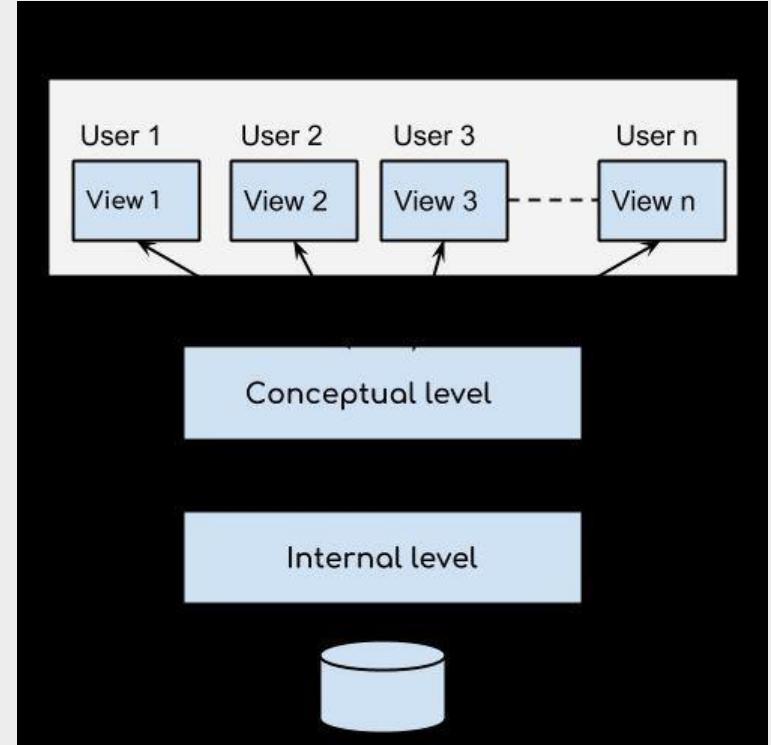
Image Source::
https://lh6.googleusercontent.com/proxy/NmDIYKVTeT5KJ4cViogUgX8dpSmxe0VqqjILC0RXvd5oBtKoLd9gGvmWHU_088z8K2aKLQI-o6Qn74wprAmY14U6cHpYmaf5-hDK

Three layered Architecture of DBMS

External, Conceptual & Internal Level

This architecture has three levels:

- External level
- Conceptual level
- Internal level



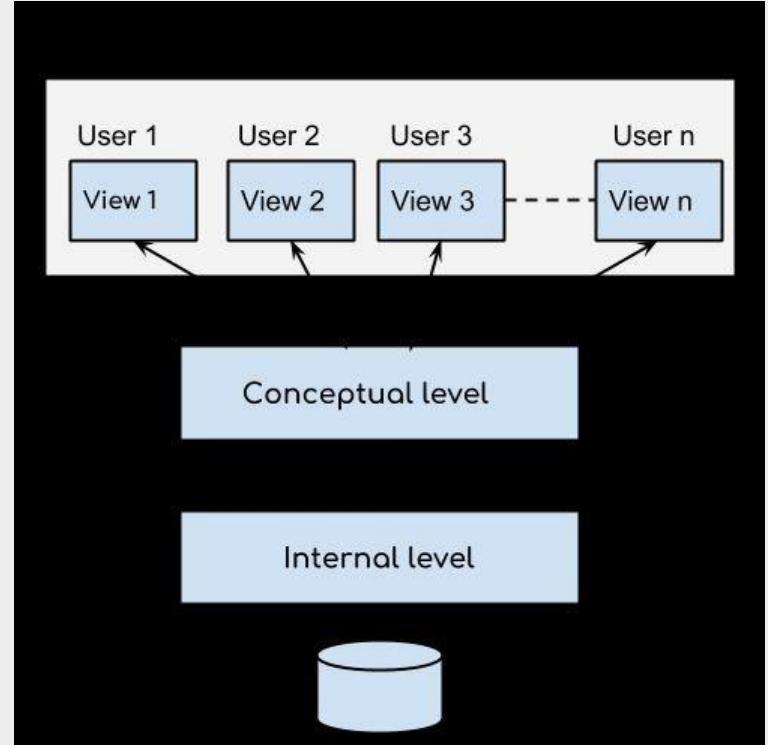
Image

Source: https://beginnersbook.com/wp-content/uploads/2018/11/dbms_three_level_architecture.png

Three layered Architecture of DBMS

External Level

- It is also called **view level**.
- The reason this level is called “view” is because several users can view their desired data from this level which is internally fetched from database
- The data is fetched with the help of conceptual and internal level mapping.



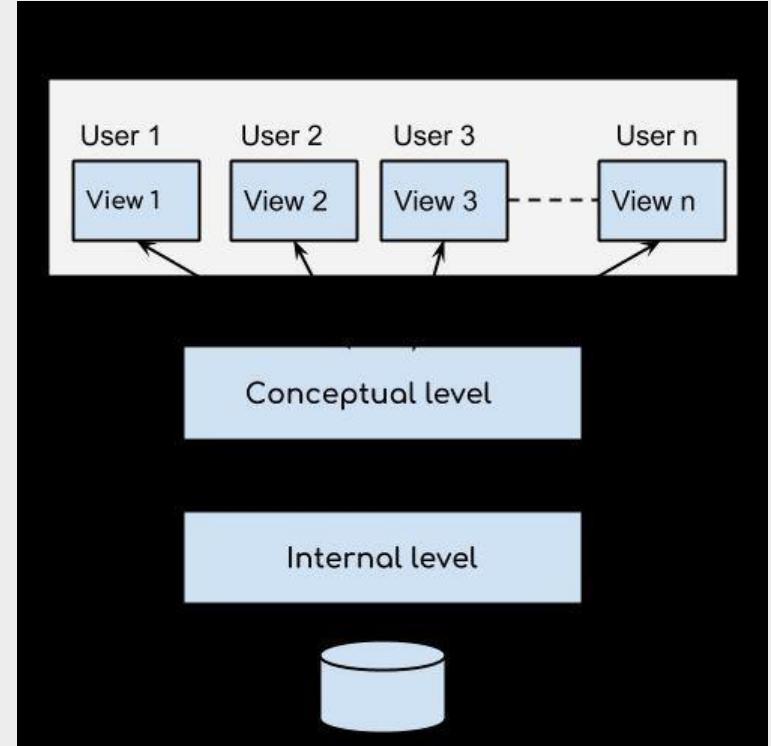
Image

Source: https://beginnersbook.com/wp-content/uploads/2018/11/dbms_three_level_architecture.png

Three layered Architecture of DBMS

Conceptual Level

- It is also called **logical level**.
- The whole design of the database such as relationship among data, schema of data etc. are described in this level.



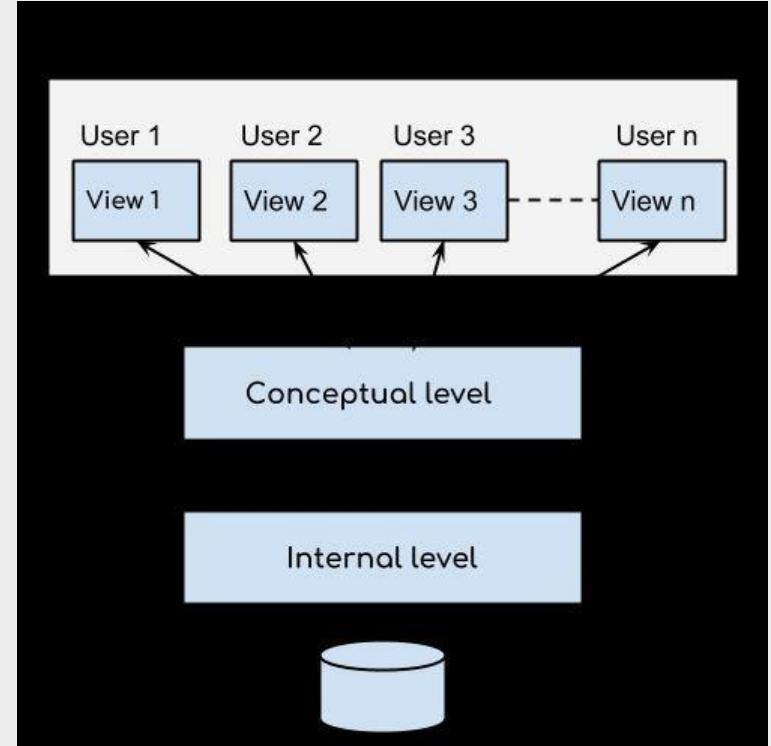
Image

Source: https://beginnersbook.com/wp-content/uploads/2018/11/dbms_three_level_architecture.png

Three layered Architecture of DBMS

Internal Level

- This level is also known as physical level.
- This level describes how the data is actually stored in the storage devices.
- This level is also responsible for allocating space to the data. This is the lowest level of the architecture.



Image

Source: https://beginnersbook.com/wp-content/uploads/2018/11/dbms_three_level_architecture.png

Different level of abstraction

Physical, Logical & View level

The three-schema architecture defines the view of data at three levels:

- Physical level (internal level)
- Logical level (conceptual level)
- View level (external level)

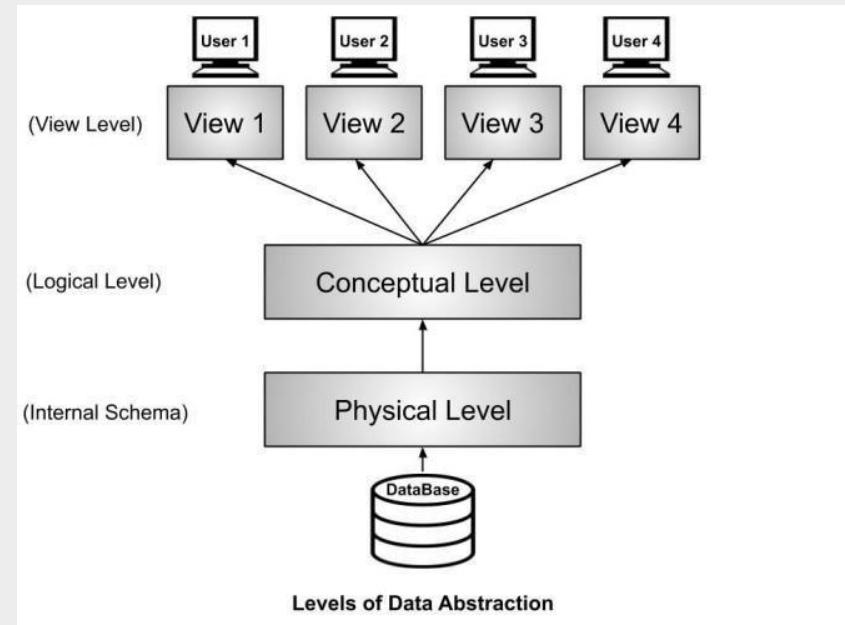


Image Source::
<https://binaryterms.com/wp-content/uploads/2019/11/View-of-data-three-schema-architecture.jpg>

Different level of abstraction

Physical level

- The physical or the internal level schema describes **how the data is stored in the hardware.**
- It also describes how the data can be accessed.

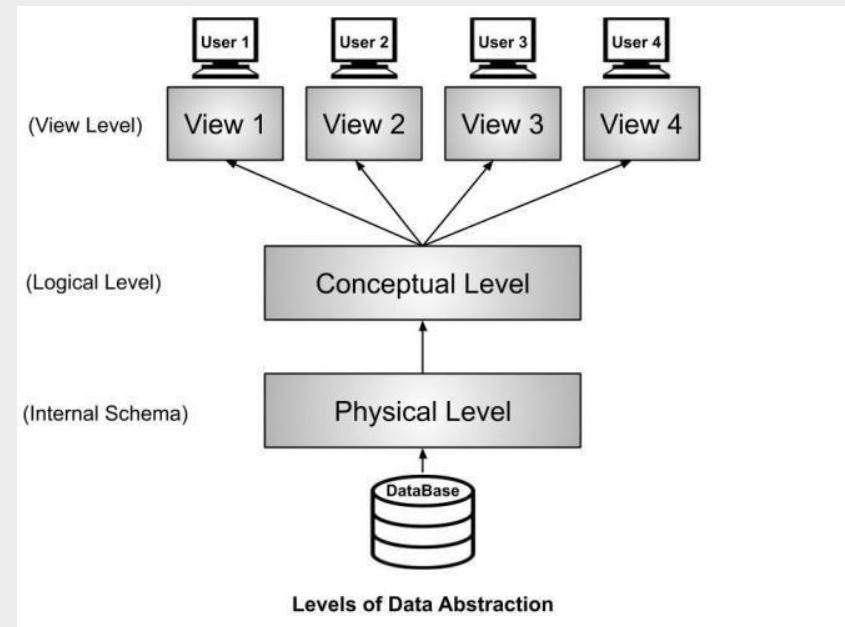


Image Source::
<https://binaryterms.com/wp-content/uploads/2019/11/View-of-data-three-schema-architecture.jpg>

Different level of abstraction

Logical level

- It is a level above the physical level. Here, the data is stored in the form of the **entity set, entities, their data types**.
- The **relationship** among the entity sets, **user operations** performed to retrieve or modify the data and certain **constraints on the data**.

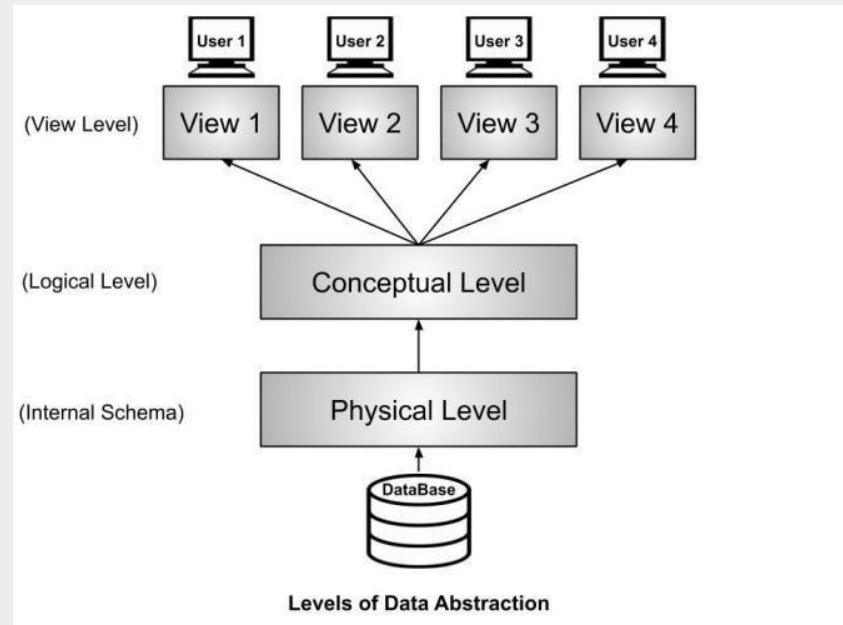


Image Source::
<https://binaryterms.com/wp-content/uploads/2019/11/View-of-data-three-schema-architecture.jpg>

Different level of abstraction

View level

- It is the highest level of data abstraction and exhibits only a part of the whole database.
- It exhibits the data in which the user is interested.

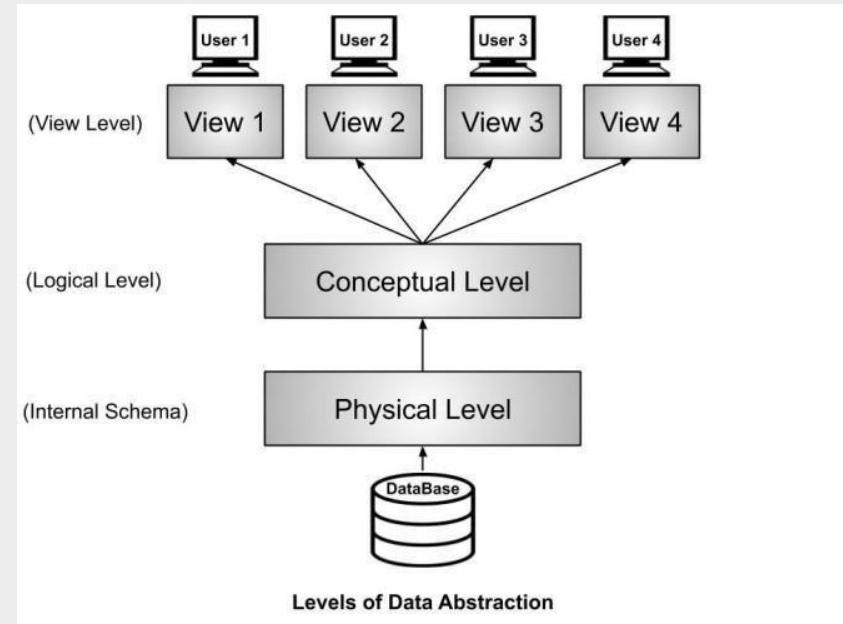


Image Source::
<https://binaryterms.com/wp-content/uploads/2019/11/View-of-data-three-schema-architecture.jpg>

Data Modeling

What is Data Modeling?

- Data models define how the logical structure of a database is modeled.
- Data Models are fundamental entities to introduce abstraction in a DBMS.

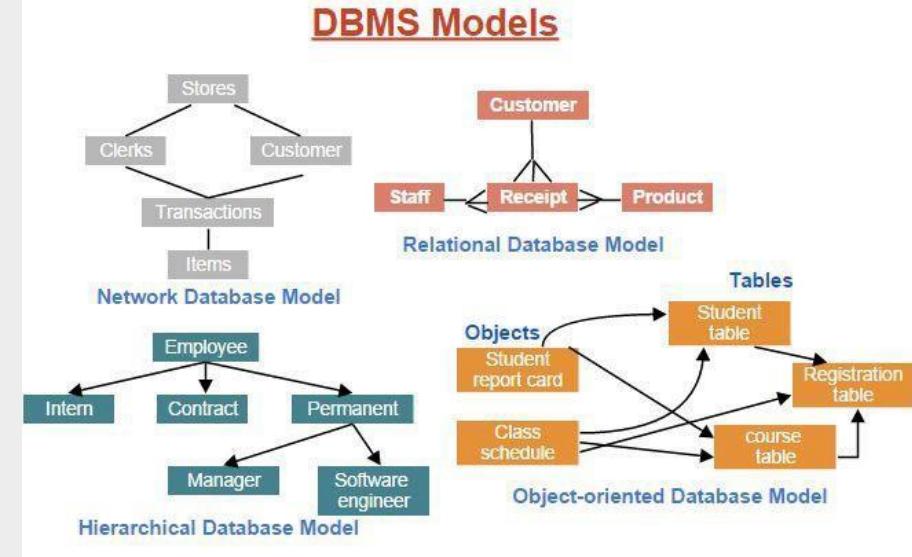


Image Source::

https://www.assignmenthelp.net/assignment_help/images/database-models.png

Data Modeling

Flat Data Model

- Flat data model is the first introduced traditional data model where data is kept in the same plane.
- This is a very old model which is not much scientific.

	Route No.	Miles	Activity
Record 1	I-95	12	Overlay
Record 2	I-495	05	Patching
Record 3	SR-301	33	Crack seal

Image Source::

https://www.assignmenthelp.net/assignment_help/images/database-models.png

Data Modeling

Relational Data Model

- The most popular and extensively used data model is the relational data model.
- The data model allows the data to be stored in tables called relation.
- The relations are normalized and the normalized relation values are known as atomic values. Each of the rows in a relation is called tuples which contains the unique value.
- The attributes are the values in each of the columns which are of the same domain.

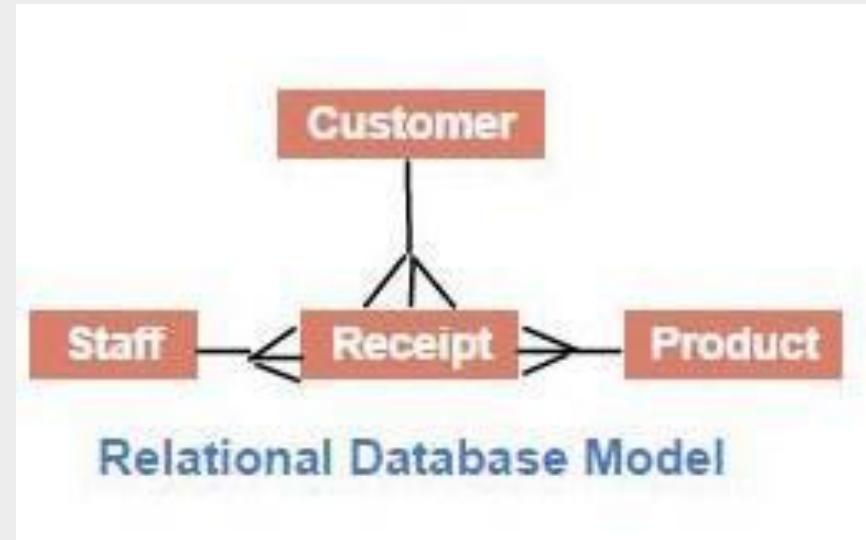


Image Source::

https://www.assignmenthelp.net/assignment_help/images/database-models.png

Data Modeling

Record base Data Model

- The record-based data model is used to determine the overall design of the database.
- This data model contains different kinds of record types.
- Each of the record types has a fixed length and a fixed number of fields.

EMPLOYEE					DEPARTMENT		PROJECT	
EMP_ID	EMP_NAME	ADDRESS	DEPT_ID	PROJ_ID	DEPT_ID	DEPT_NAME	PROJ_ID	PROJ_NAME
100	Joseph	Clinton Town	10	206	10	Accounting	201	C Programming
101	Rose	Fraser Town	20	205	20	Quality	202	Web development
102	Mathew	Lakeside Village	10	206	30	Design	204	Database Design
103	Stewart	Troy	30	204			205	Testing
104	William	Holland	30	202			206	Pay Slip Generation

Image Source::
<https://www.tutorialcup.com/images/dbms/record-based-data-models/Relational-Data-Models2.png>

E-R Modeling

Components of ER- Diagrams

- Entity-Relationship (ER) Model is based on the notion of real-world entities and relationships among them.
- While formulating real-world scenario into the database model, the ER Model creates entity set, relationship set, general attributes and constraints.
- ER Model is best used for the conceptual design of a database.
- ER Model is based on –
 - Entities** and their *attributes*.
 - Relationships** among entities.

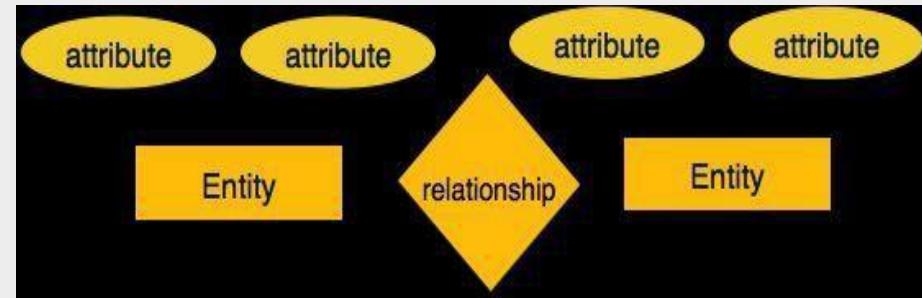


Image Source::

https://www.assignmenthelp.net/assignment_help/images/database-models.png

Logical Model: Object & Record based – Object oriented model

Steps of ER Model

- A logical data model describes the data in as much detail as possible, without regard to how they will be physical implemented in the database.
- Features of a logical data model include:
- Includes all entities and relationships among them.
 - All attributes for each entity are specified.
 - The primary key for each entity is specified.
 - Foreign keys (keys identifying the relationship between different entities) are specified.
 - Normalization occurs at this level.

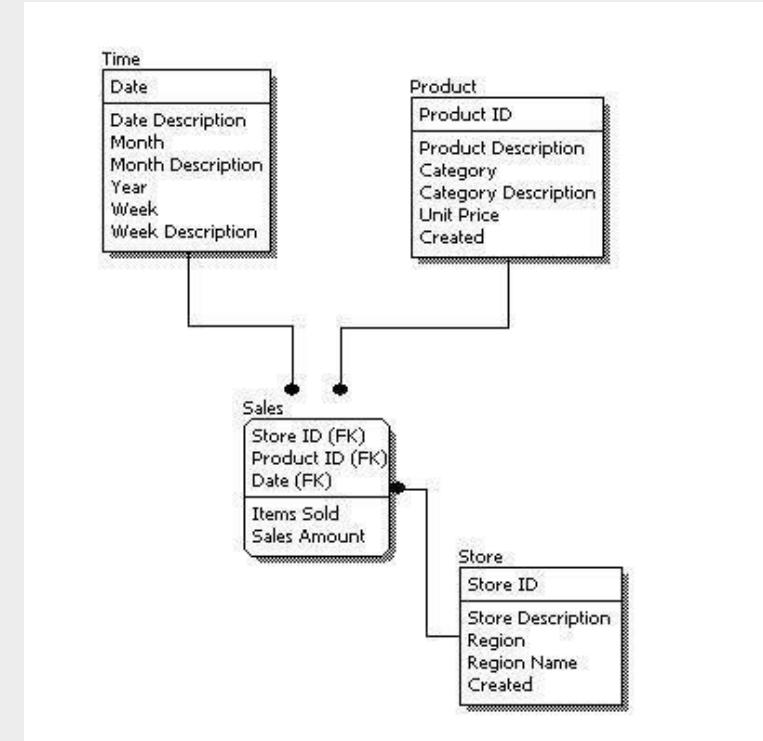


Image Source: <https://www.1keydata.com/datawarehousing/logical-data-model.jpg>

Logical Model: Object & Record based – Object oriented model

Steps of ER Model

- The steps for designing the logical data model are as follows:
 - Specify primary keys for all entities.
 - Find the relationships between different entities.
 - Find all attributes for each entity.
 - Resolve many-to-many relationships.
 - Normalization.

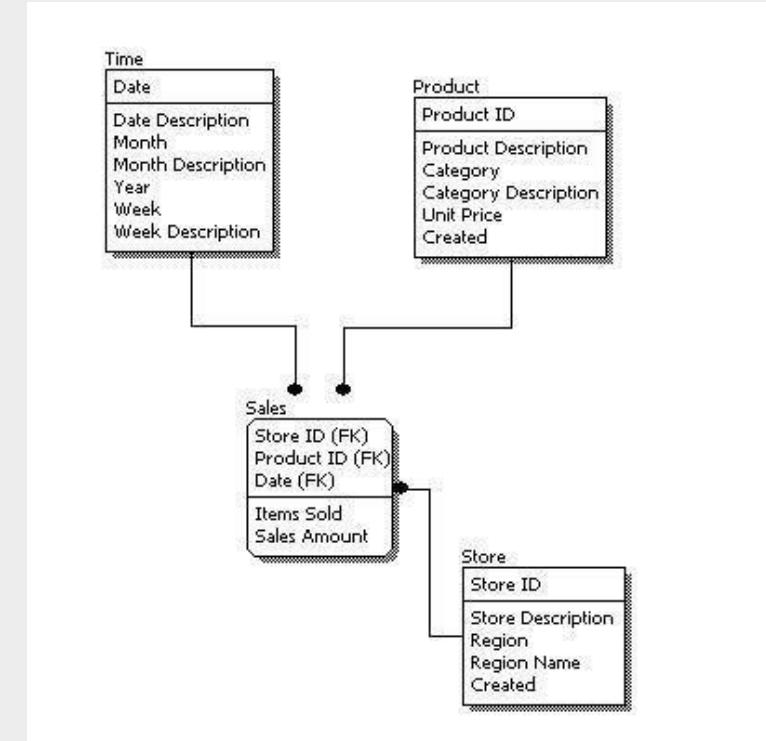


Image Source: <https://www.1keydata.com/datawarehousing/logical-data-model.jpg>

Entity sets & relationships sets

Entity

- An entity refers to any object having-
 - Either a physical existence such as a particular person, office, house or car.
 - Or a conceptual existence such as a school, a university, a company or a job.

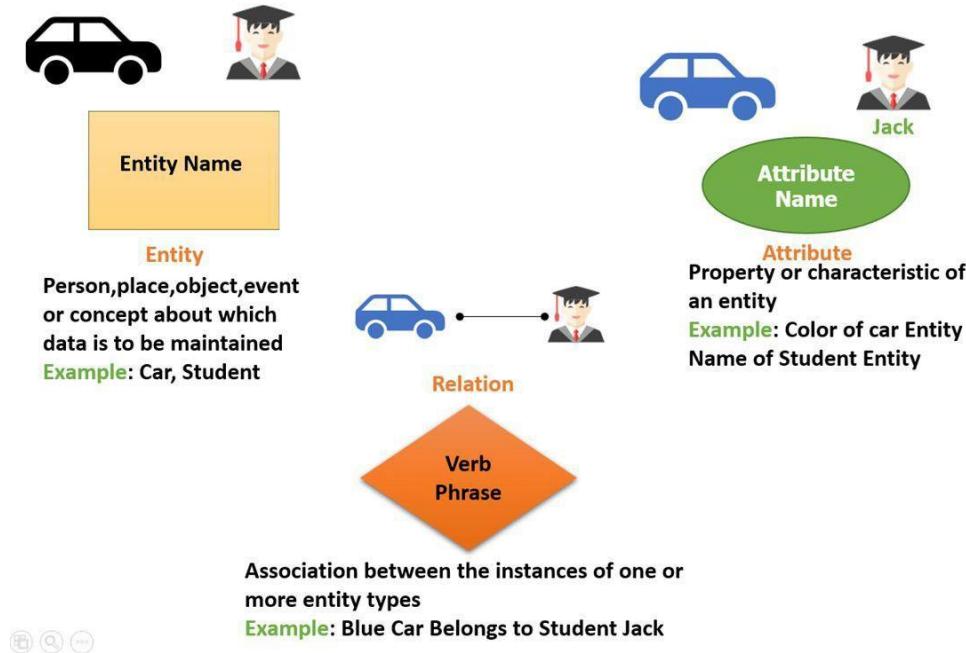


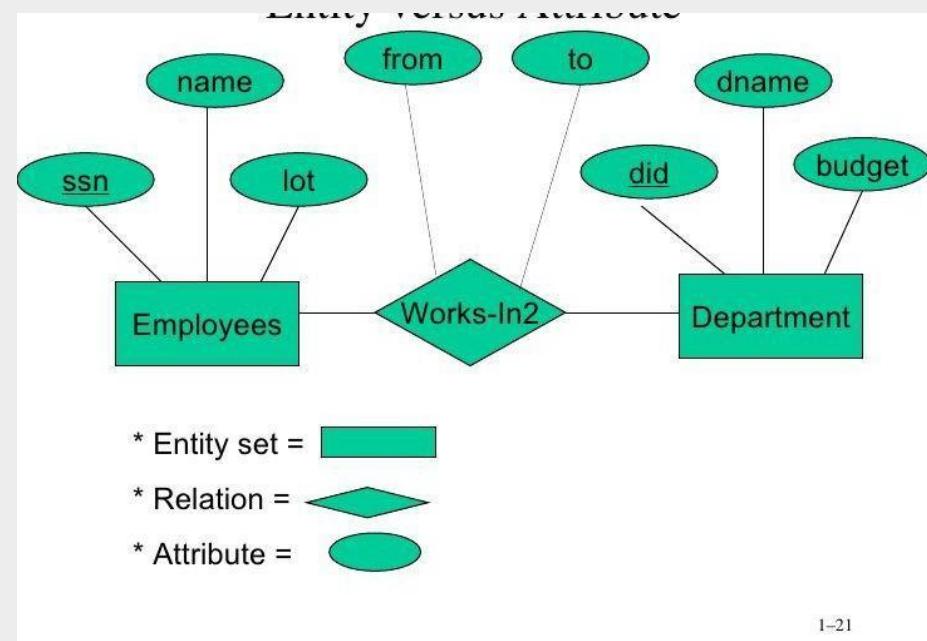
Image Source::

https://www.guru99.com/images/1/100518_0621_ERDiagramTu2.png

Entity sets & relationships sets

Entity Sets

- **Entity Set** is a collection or a group of 'entities' sharing exactly the 'same set of attributes'.
- All **entities** can be **distinctly identified** in an entity set.
- This is because all the entities have a different set of value for some set of attributes.
- We further classify the entity set into two basic categories **Strong** and **Weak entity set**.



1-21

Image Source::
<https://image.slidesharecdn.com/27fcs157a14-111005002541-phpapp01/95/dbms-unit-1-21-728.jpg?cb=1317774454>

Entity sets & relationships sets

Types of Entity Sets

- An entity set may be of the following two types-
 - Strong entity set
 - Weak entity set

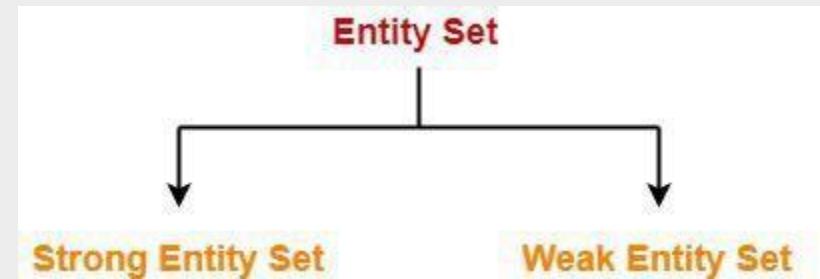


Image Source::
<https://www.gatevidyalay.com/wp-content/uploads/2018/05/Strong-Entity-Set-Example.png>

Types of Entity Sets

Strong Entity Set

- A strong entity set is an entity set that contains sufficient attributes to uniquely identify all its entities.
- In other words, a primary key exists for a strong entity set.
- Primary key of a strong entity set is represented by underlining it.

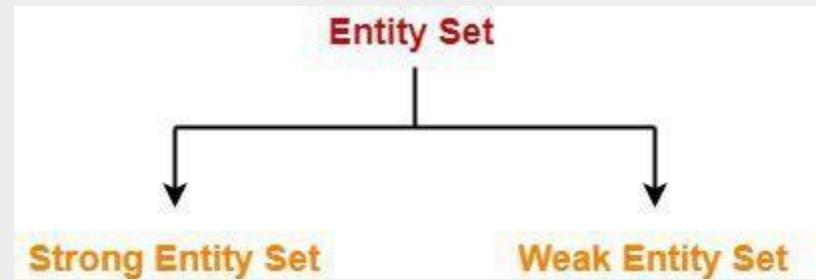


Image Source::
<https://www.gatevidyalay.com/wp-content/uploads/2018/05/Strong-Entity-Set-Example.png>

Types of Entity Sets

Strong Entity Set

Symbols Used-

- A single rectangle is used for representing a strong entity set.
 - A diamond symbol is used for representing the relationship that exists between two strong entity sets.
- A single line is used for representing the connection of the strong entity set with the relationship set.
- A double line is used for representing the total participation of an entity set with the relationship set.
 - Total participation may or may not exist in the relationship.

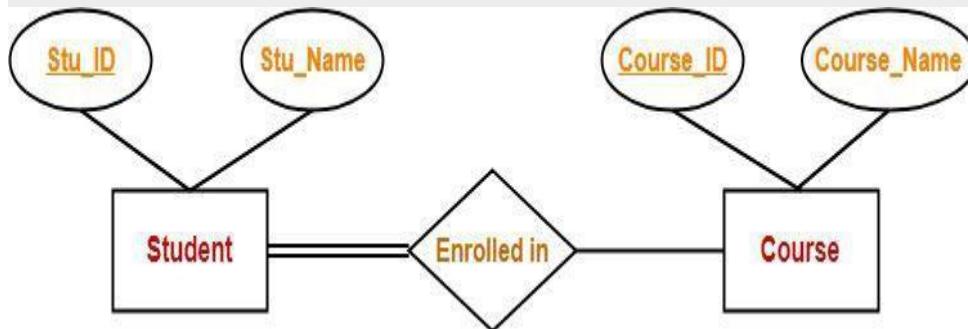


Image Source::
<https://www.gatevidyalay.com/wp-content/uploads/2018/05/Strong-Entity-Set-Example.e.png>

Types of Entity Sets

Weak Entity Set

- A weak entity set is an entity set that does not contain sufficient attributes to uniquely identify its entities.
- In other words, a primary key does not exist for a weak entity set.
- However, it contains a partial key called as a **discriminator**.
Discriminator can identify a group of entities from the entity set.
- Discriminator is represented by underlining with a dashed line.

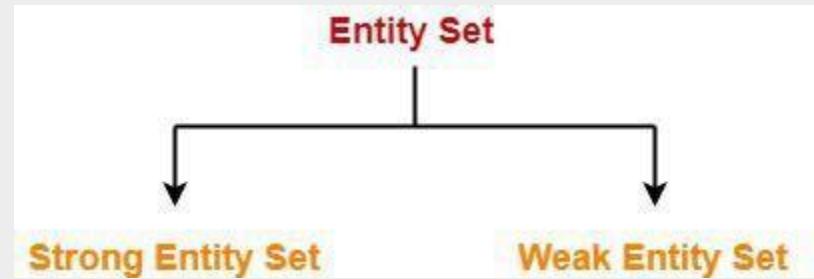


Image Source::
<https://www.gatevidyalay.com/wp-content/uploads/2018/05/Strong-Entity-Set-Example.png>

Types of Entity Sets

Weak Entity Set

Symbols Used-

- A double rectangle is used for representing a weak entity set.
- A double diamond symbol is used for representing the relationship that exists between the strong and weak entity sets and this relationship is known as **identifying relationship**.
- A double line is used for representing the connection of the weak entity set with the relationship set.
- Total participation always exists in the identifying relationship.

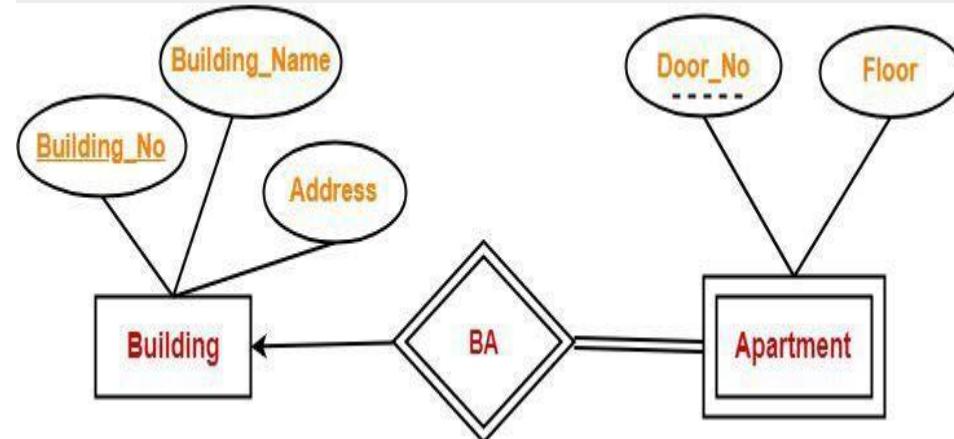


Image Source::
<https://www.gatevidyalay.com/wp-content/uploads/2018/05/Strong-Entity-Set-Example.png>

Entity sets & relationships sets

Relationship

- Entities may have several relationships among themselves.
- Whenever an attribute of one entity refers to another entity, there exists a relationship between the two entities.
- A relationship, in the context of databases, is a situation that exists between two relational database tables when one table has a foreign key that references the primary key of the other table.
- Relationships allow relational databases to split and store data in different tables, while linking disparate data items.

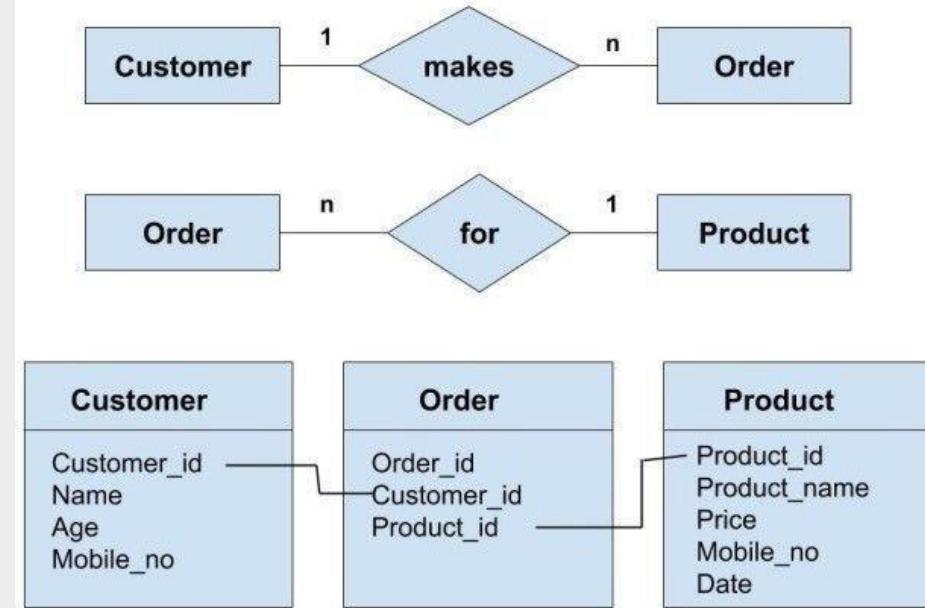


Image Source::
<https://s3.ap-south-1.amazonaws.com/afteracademy-server-uploads/what-are-the-different-types-of-relationships-in-dbms-many-to-many-relationship-examples-c961b29d4c8aa6a5.jpg>

Entity sets & relationships sets

Relationships sets

- A relationship set is a set of relationships of same type.
- Example - Set representation of above ER diagram is - Degree of a Relationship Set-.
- The number of different entity sets participating in a relationship set is called as degree of a relationship set.

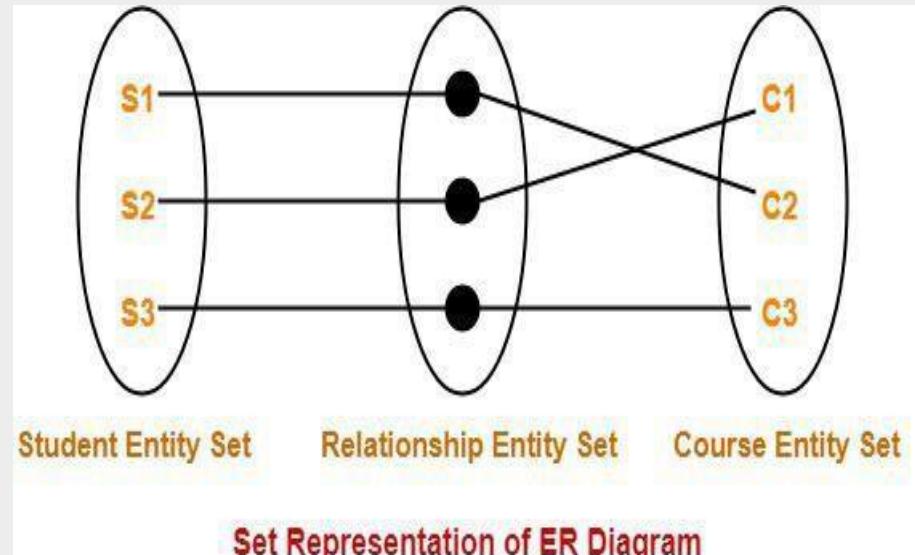


Image Source::
<https://www.gatevidyalay.com/wp-content/uploads/2018/05/Set-Representation-of-ER-Diagram.png>

Entity sets & relationships sets

Types of Relationships sets

- Unary Relationship
- Binary Relationship
- n-ary Relationship

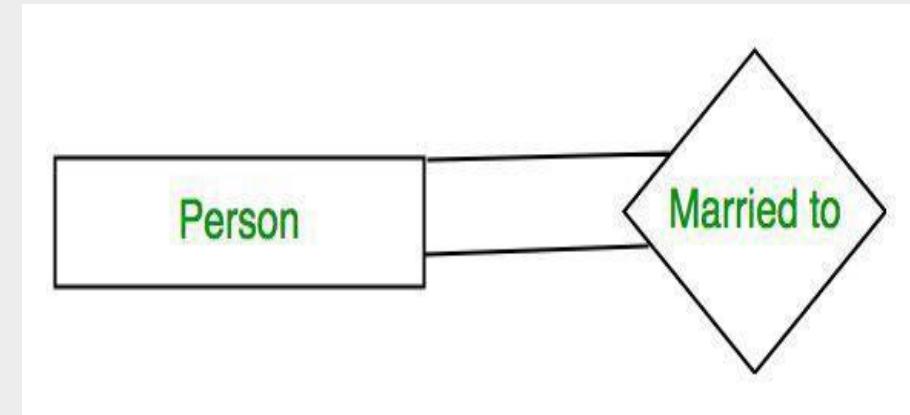


Image Source::
<https://media.geeksforgeeks.org/wp-content/uploads/Database-Management-System-ER-Model-10.png>

Types of Relationships sets

Unary Relationship

- When there is **only ONE entity set participating in a relation**, the relationship is called as unary relationship.
- For example, one person is married to only one person.

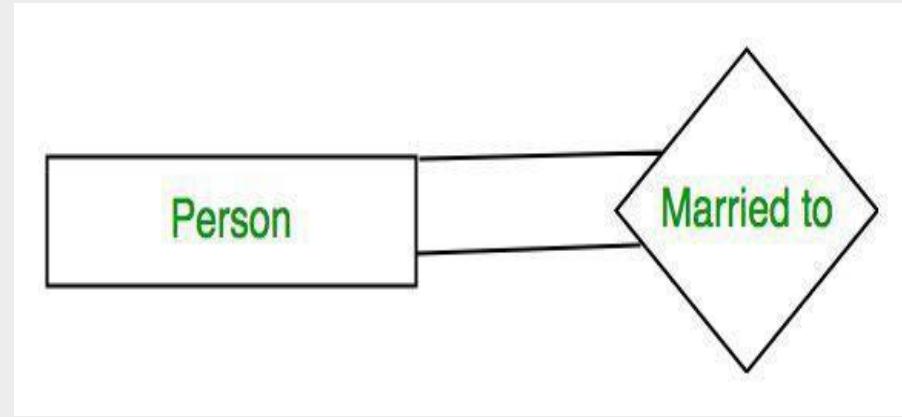


Image Source::
<https://media.geeksforgeeks.org/wp-content/uploads/Database-Management-System-ER-Model-10.png>

Types of Relationships sets

Binary Relationship

- When there are **TWO entities set participating in a relation**, the relationship is called as binary relationship.
- For example, Student is enrolled in Course.

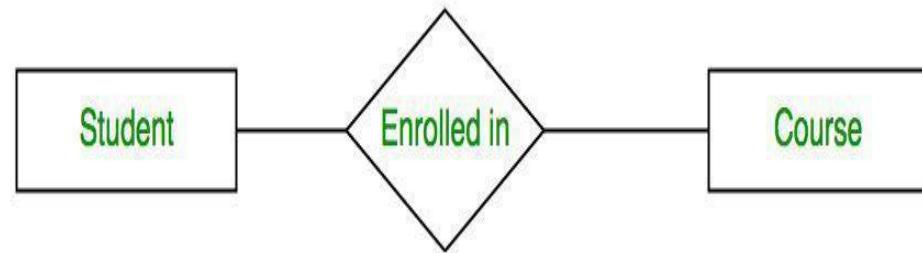


Image Source::
<https://media.geeksforgeeks.org/wp-content/uploads/Database-Management-System-ER-Model-10.png>

Types of Relationships sets

n-ary Relationship

- When there are n entities set participating in a relation, the relationship is called as n -ary relationship.

Entity sets & relationships sets

Cardinality of Relationships sets

- The **number of times an entity of an entity set participates in a relationship set** is known as cardinality.
- Cardinality can be of different types:
- One to One
- Many to One
- Many to Many

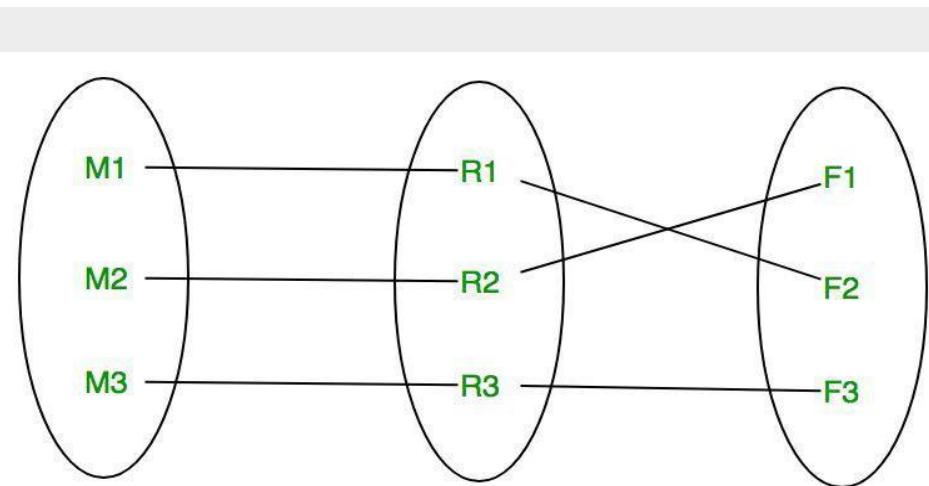
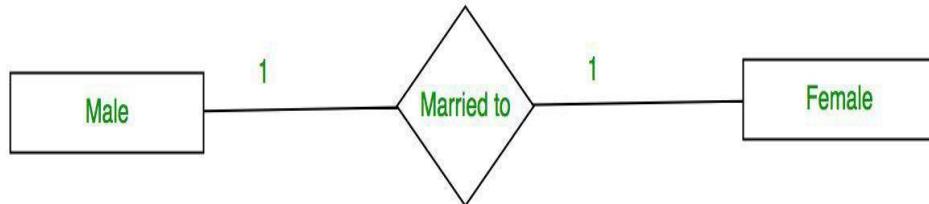


Image Source::
<https://media.geeksforgeeks.org/wp-content/uploads/Database-Management-System-ER-Model-13.png>

Cardinality of Relationships sets

One to one

- When each entity in each entity set can take part **only once in the relationship**, the cardinality is one to one.
- Let us assume that a male can marry to one female and a female can marry to one male.
- So the relationship will be one to one.

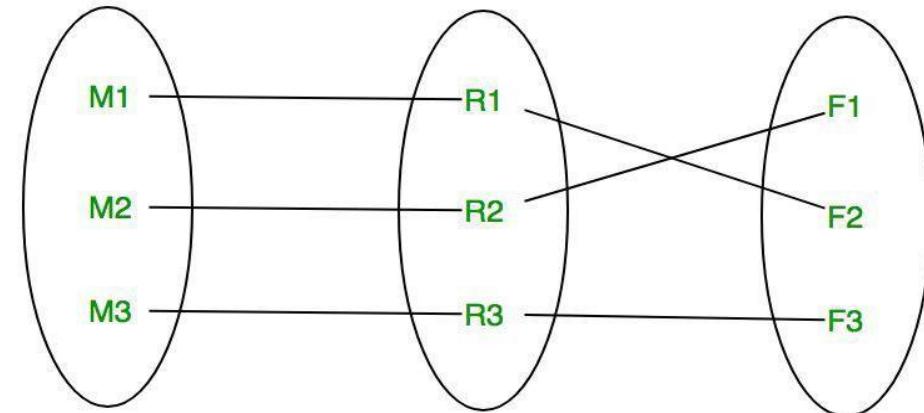
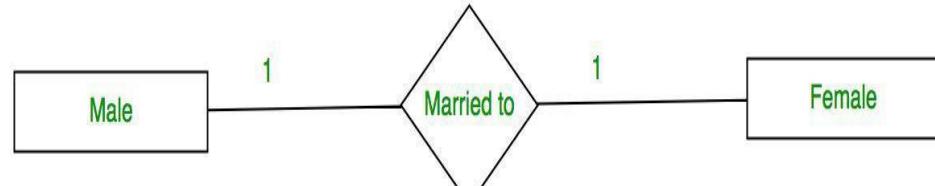


Image Source::
<https://media.geeksforgeeks.org/wp-content/uploads/Database-Management-System-ER-Model-13.png>

Cardinality of Relationships sets

Many to One

- When entities in one entity set can take part only **once in the relationship set and entities in other entity set can take part more than once in the relationship set**, cardinality is many to one.
- Let us assume that a student can take only one course but one course can be taken by many students.
- So the cardinality will be n to 1.
- It means that for one course there can be n students but for one student, there will be only one course.

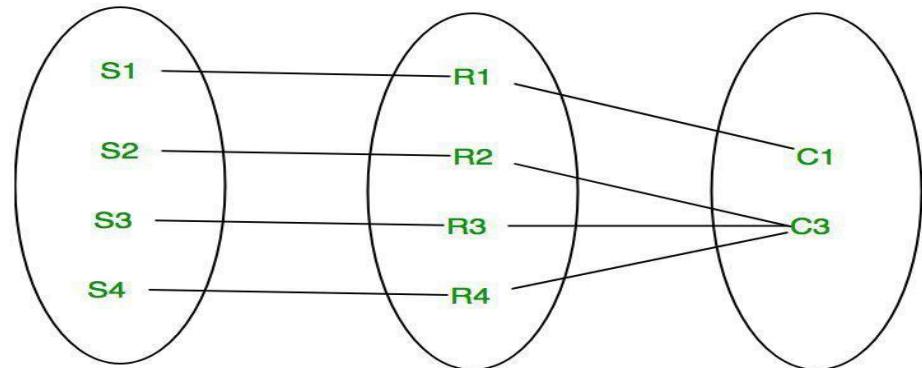
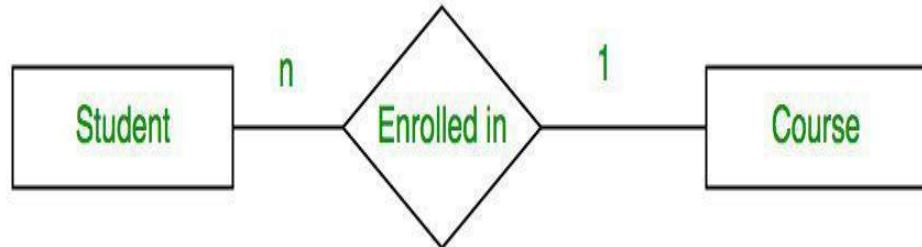


Image Source::
<https://media.geeksforgeeks.org/wp-content/uploads/Database-Management-System-ER-Model-13.png>

Cardinality of Relationships sets

Many to Many

- When entities in all entity sets can **take part more than once in the relationship** cardinality is many to many.
- Let us assume that a student can take more than one course and one course can be taken by many students.
- So the relationship will be many to many.

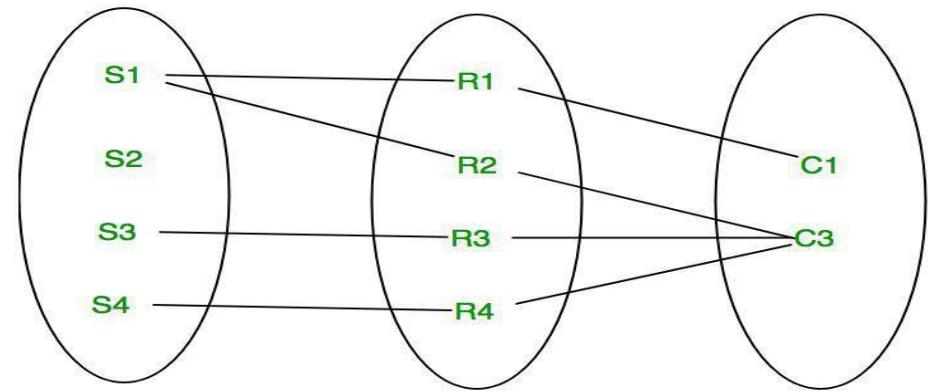
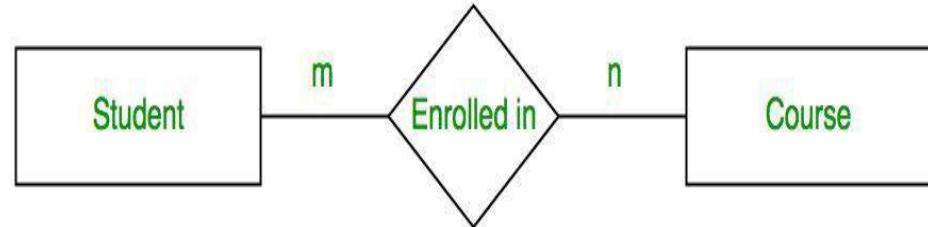
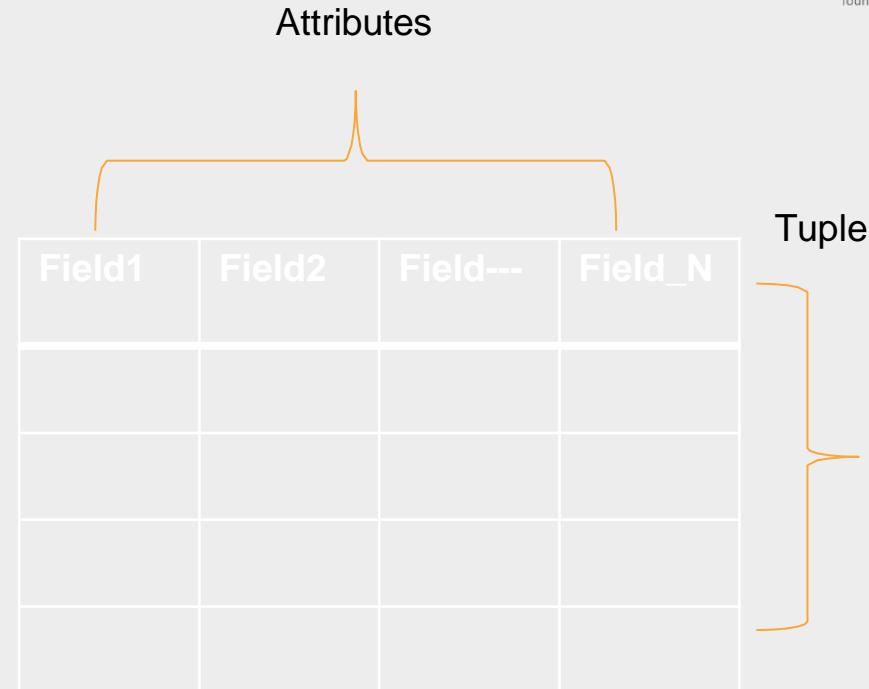


Image Source::
<https://media.geeksforgeeks.org/wp-content/uploads/Database-Management-System-ER-Model-13.png>

Concept of Attribute

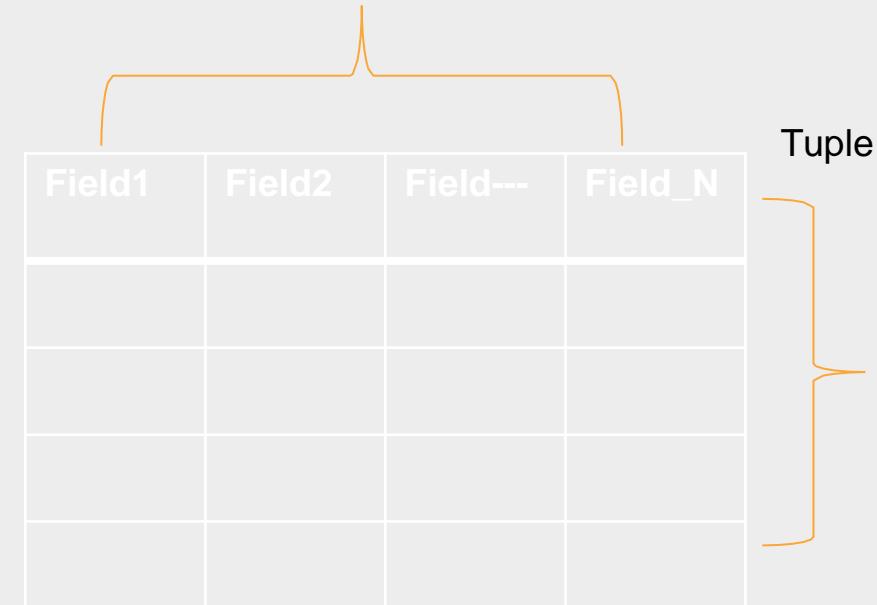


Concept of Attribute

Concept

- Attribute describes property of entity.
- It is used in entity relationship diagram.
- Graphical presentation.

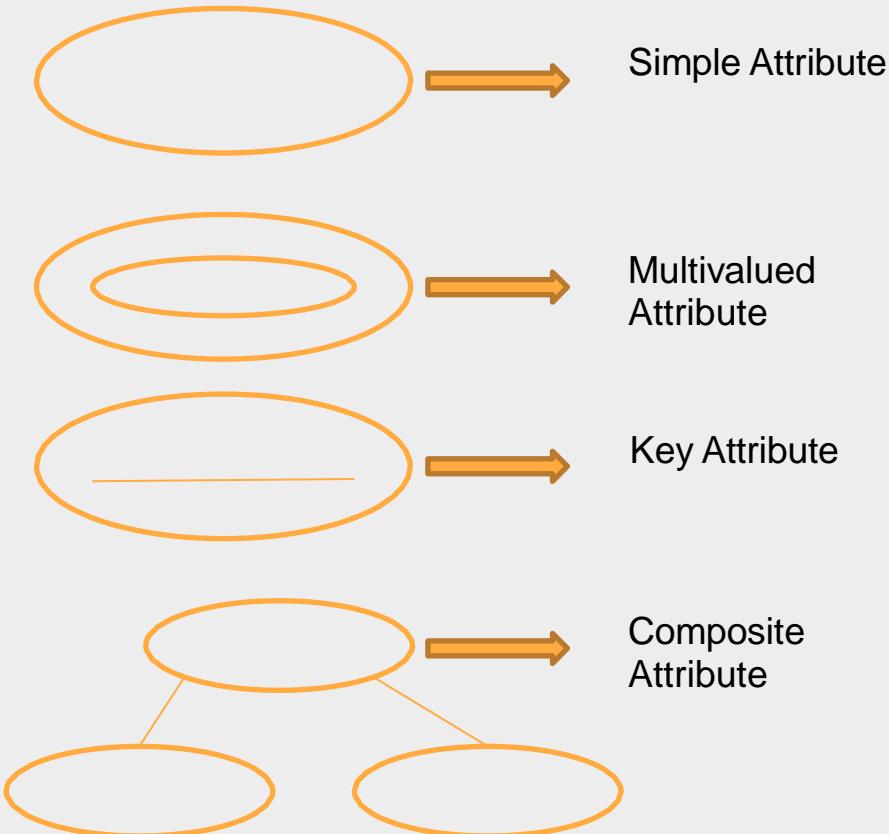
Attributes



Concept of Attribute

Symbols used to represent attributes

- Symbols



Concept of Attribute

Symbols used to represent attributes

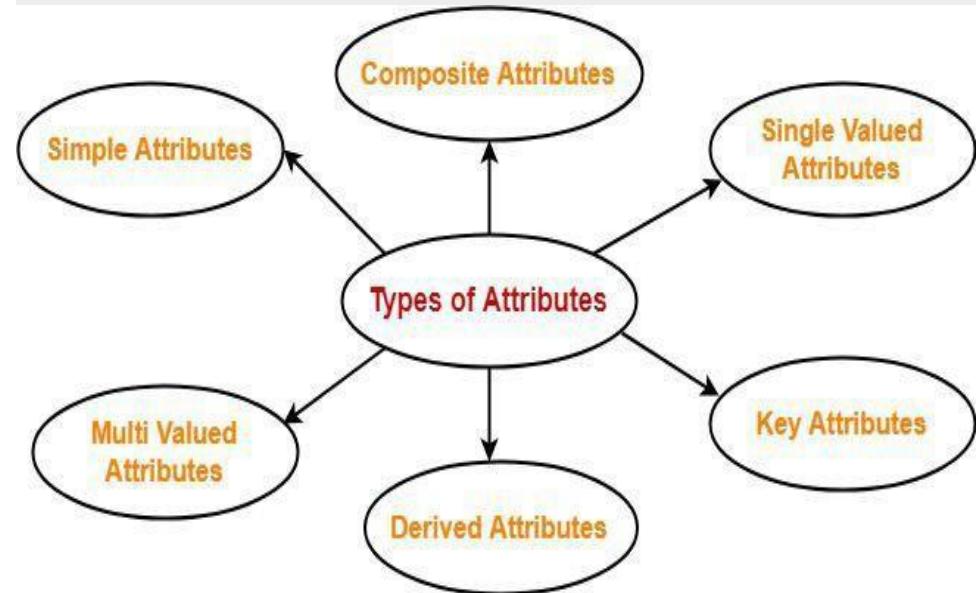


- Symbols

Concept of Attribute

Types of Attributes

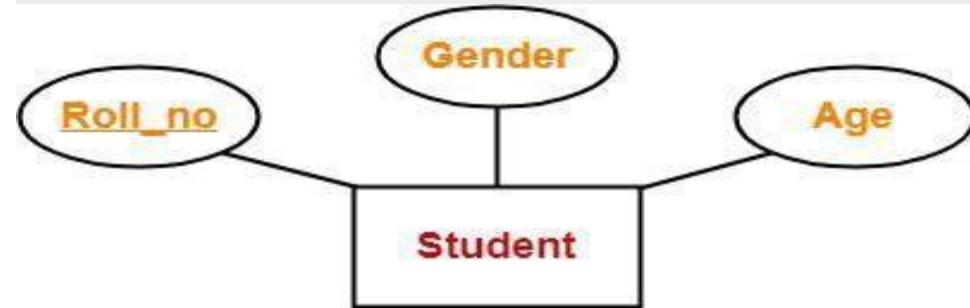
- Simple Attributes
- Composite Attributes
- Single Valued Attributes
- Key Attributes
- Derived Attributes
- Multivalued Attributes



Concept of Attribute

Types of Attributes Simple Attributes

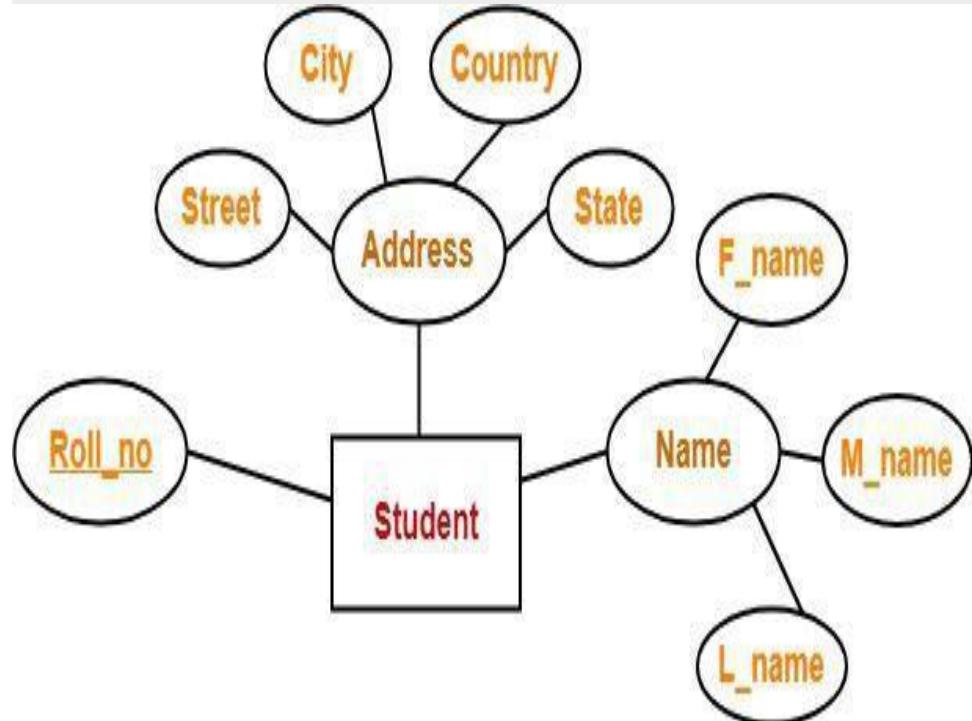
- A simple attribute is identify entity from an entity set.
- Simple attribute represent oval symbol with its value.



Concept of Attribute

Types of Attributes
Composite Attributes

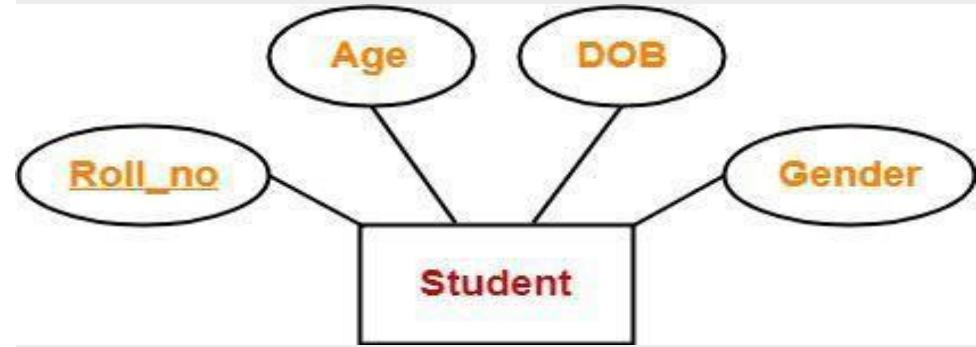
- Composite attributes are those attributes which are composed of many other simple attributes.
- COMPOSITE KEY is a combination of two or more columns that uniquely identify rows in a table.



Concept of Attribute

Types of Attributes
Single Valued Attributes

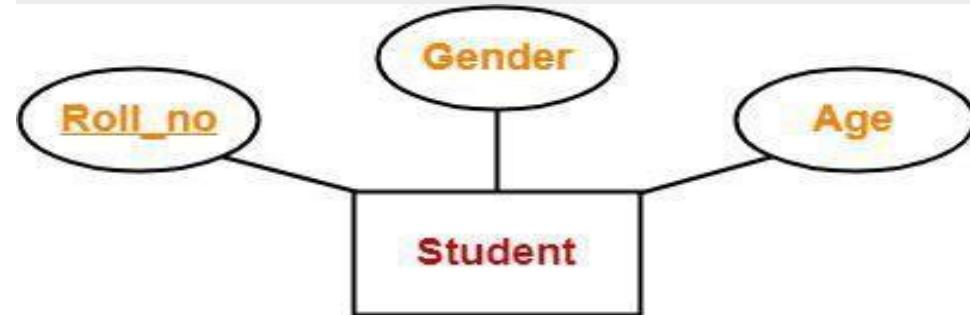
- Single valued attributes are those attributes which can take only one value for a given entity from an entity set.



Concept of Attribute

Types of Attributes
Key Attributes

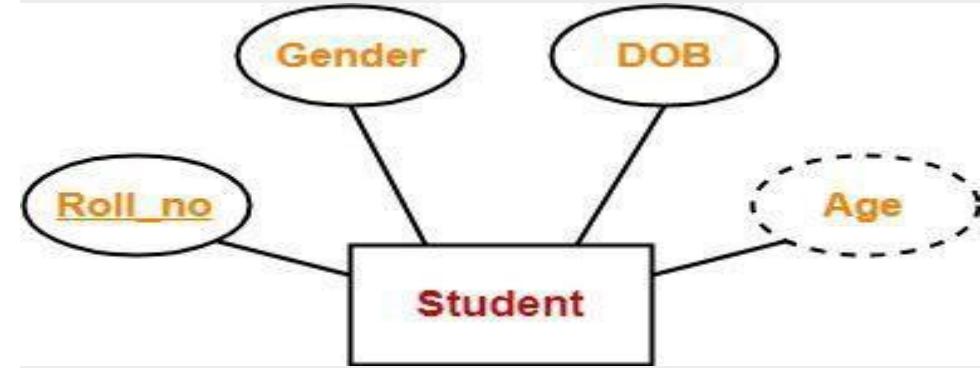
- A key attribute is uniquely identify entity from an entity set.
- Key attribute represent oval symbol same as like other with underline.



Concept of Attribute

Types of Attributes
Derived Attributes

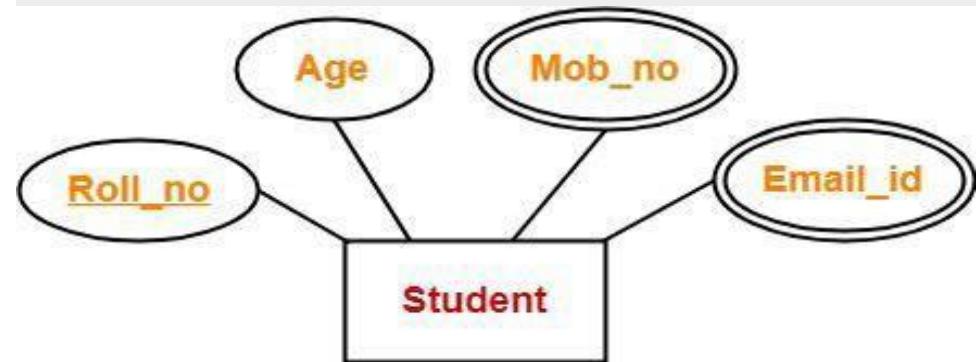
- Derived attributes are those attributes which can be derived from other attribute(s).



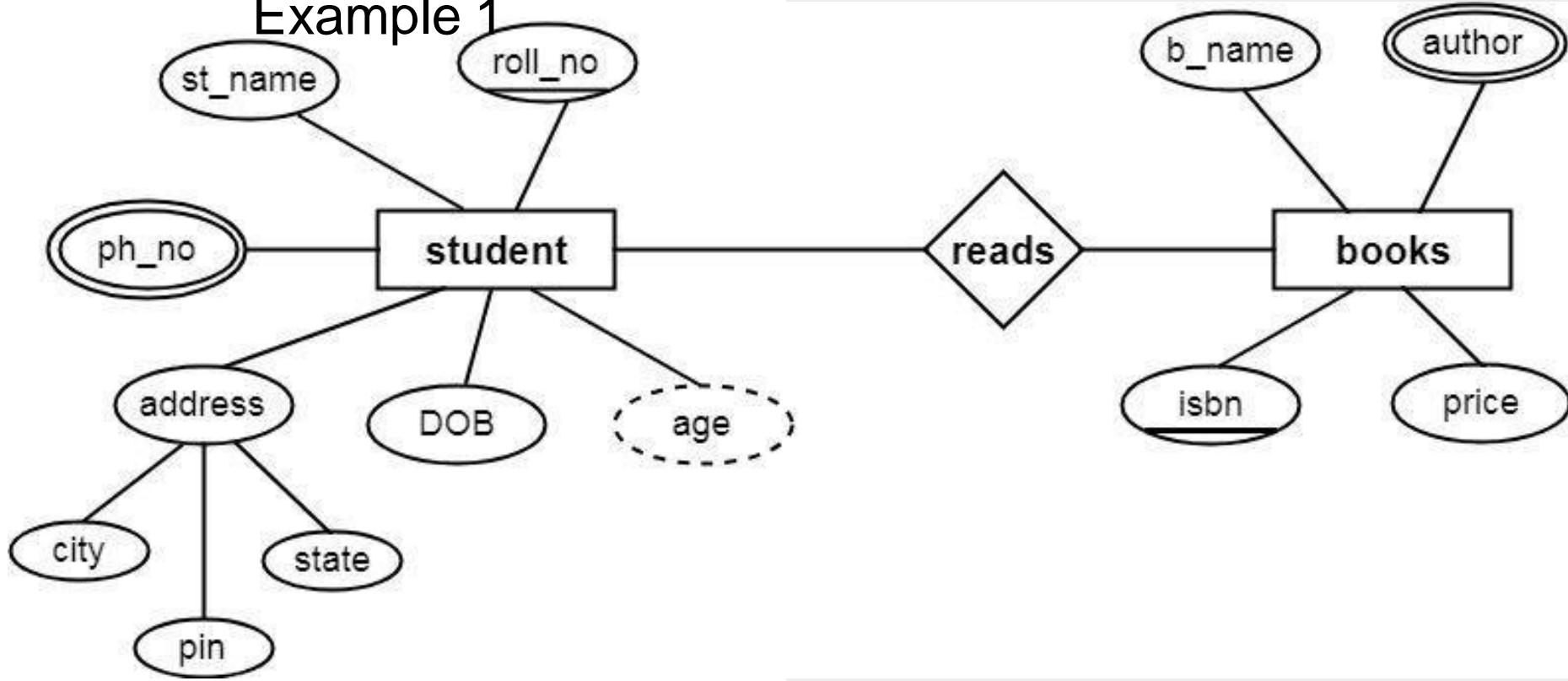
Concept of Attribute

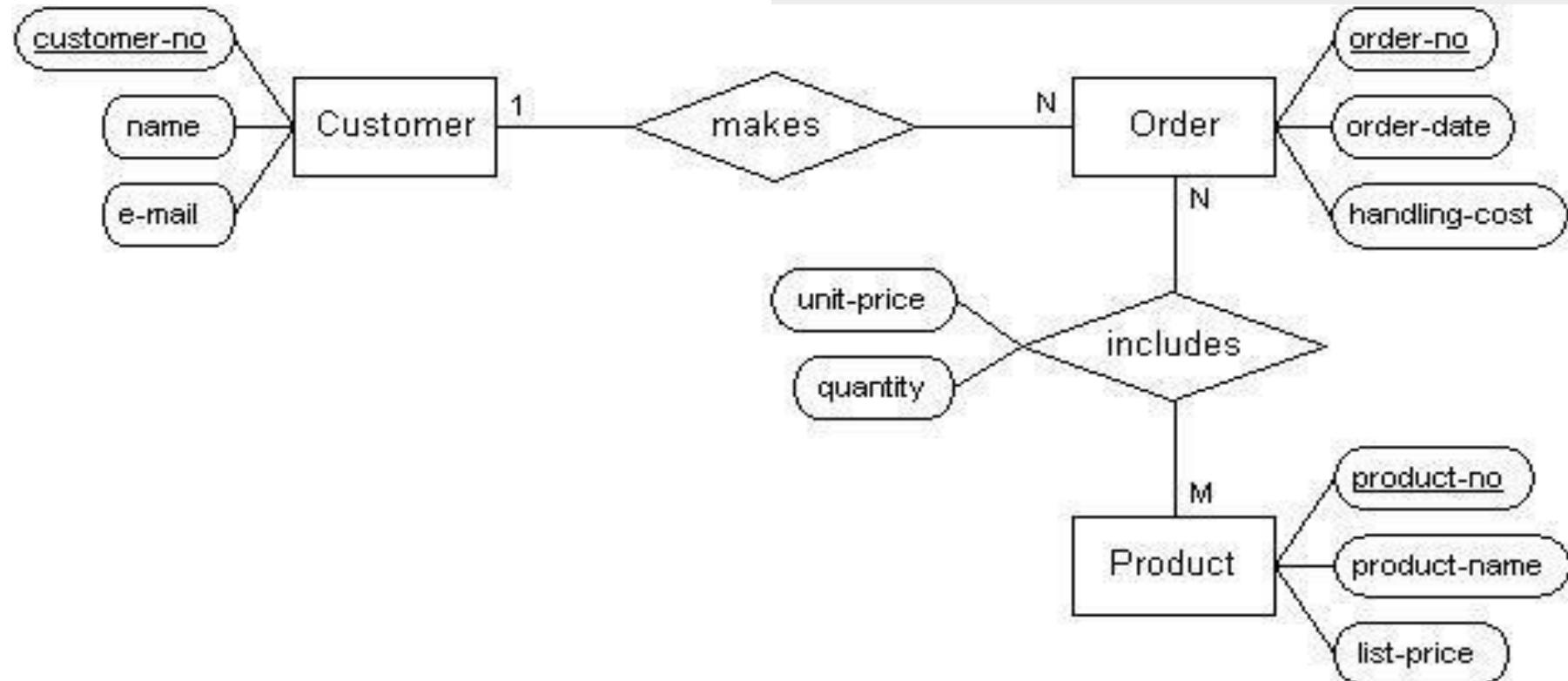
Types of Attributes
Multivalued Attributes

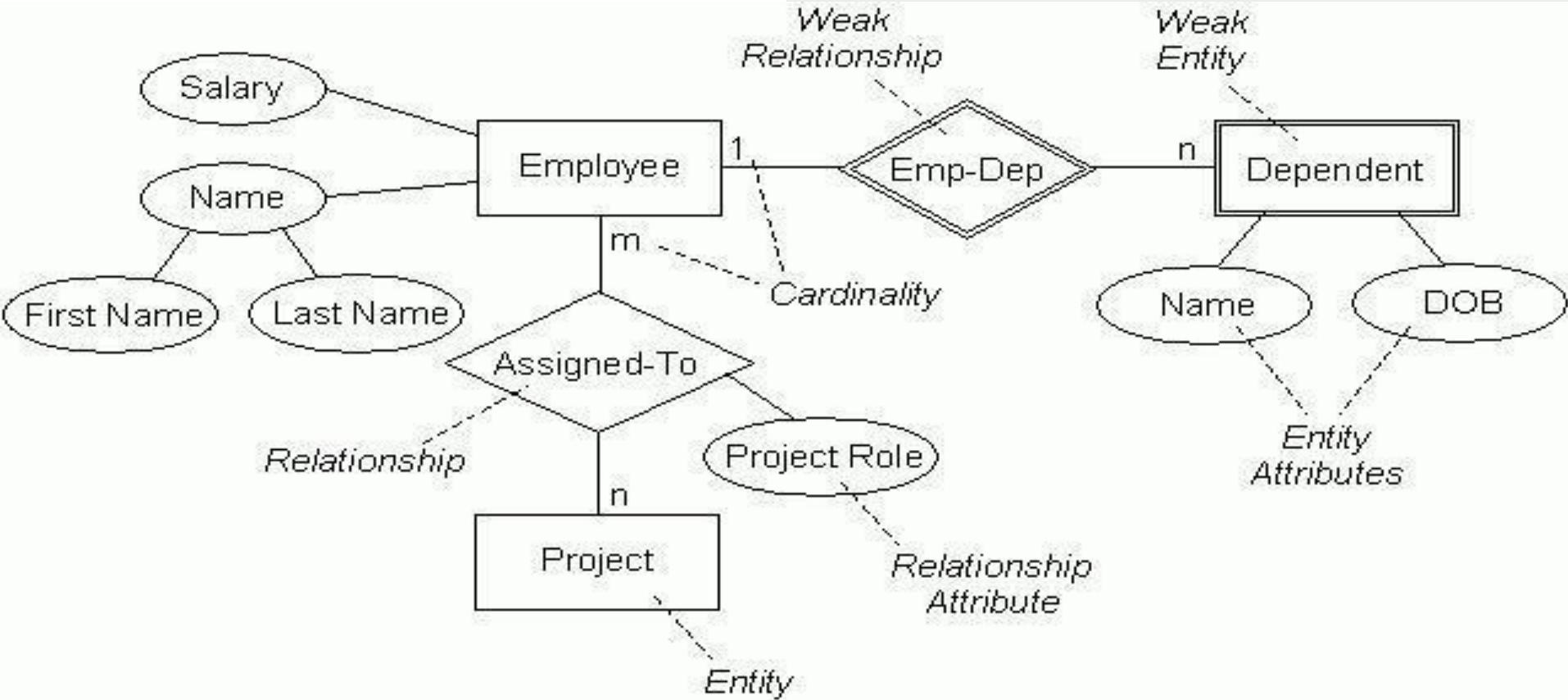
- Multi valued attributes are those attributes which can take more than one value for a given entity from an entity set.



Example 1







Concept of Relationship

Concept of Relationship

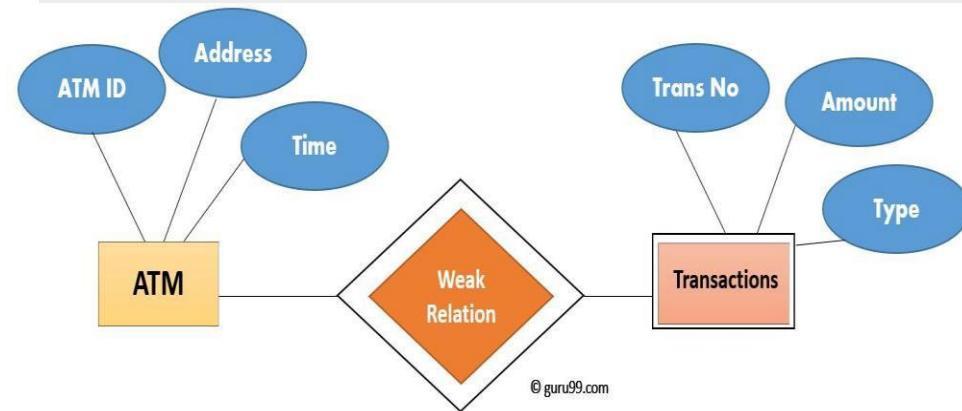
- Relationship is nothing but an association among two or more entities.
- Entities take part in relationships. We can often identify relationships with verbs or verb phrases.



Concept of Relationship

Concept of Relationship

- A weak entity is a type of entity which doesn't have its key attribute.
- It can be identified uniquely by considering the primary key of another entity. For that, weak entity sets need to have participation



Types of Keys

Keys

- Keys play an important role in the relational database.
- It is used to uniquely identify any record or row of data from the table.
- It is also used to establish and identify relationships between tables.

- Keys
 - Primary
 - Foreign
 - Super
 - Candidate
 - Unique

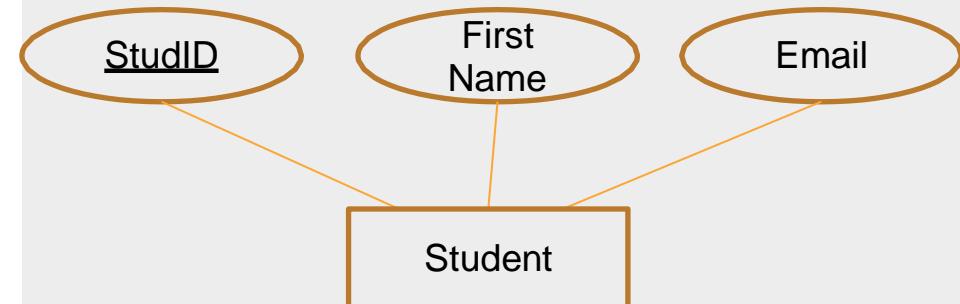
Example

Types of Keys

Primary Key

- **PRIMARY KEY** It is the first key which is used to identify one and only one instance of an entity uniquely.

StudID	First Name	Email
1	Tom	abc@gmail.com
2	Nick	xyz@gmail.com
3	Dana	mno@yahoo.com



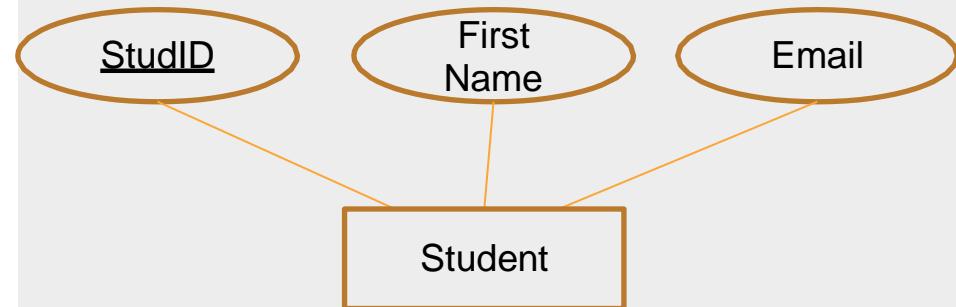
Example

Types of Keys

Rule for defining primary key

- Two rows can't have the same primary key value
- It must for every row to have a primary key value.
- The primary key field cannot be null.
- The value in a primary key column can never be modified or updated if any foreign key refers to that primary key.

StudID	First Name	Email
1	Tom	abc@gmail.com
2	Nick	xyz@gmail.com
3	Dana	mno@yahoo.com



Types of Keys

Foreign Key

- FOREIGN KEY is a column that creates a relationship between two tables.
- The purpose of Foreign keys is to maintain data integrity and allow navigation between two different instances of an entity.

Example

DeptCode	DeptName
001	Science
002	English
005	Computer

Teacher ID	Fname	Lname
B002	David	Warner
B017	Sara	Joseph
B009	Mike	Brunton

Types of Keys

Rule for defining foreign key

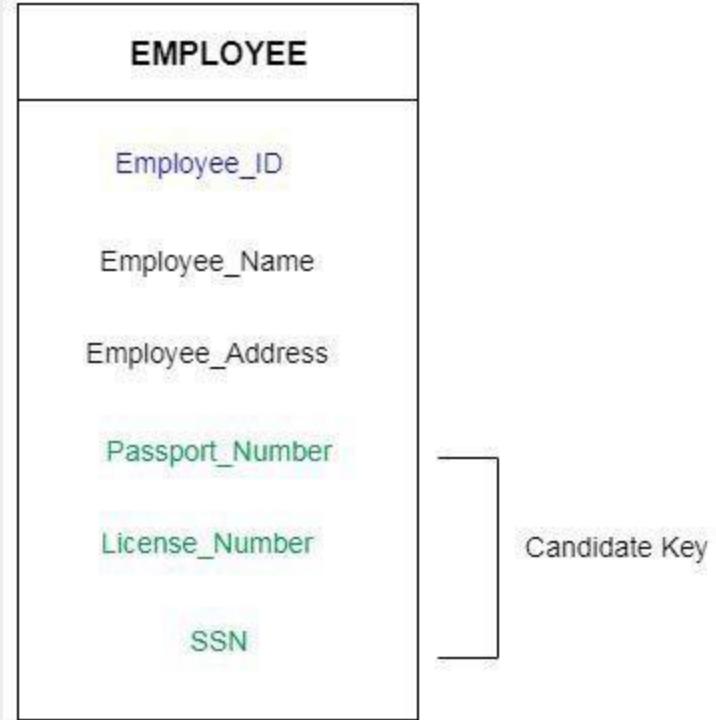
- Foreign key columns must use their referenced column's type.
- Each column cannot belong to more than 1 Foreign Key constraint.
- Cannot be a computed column.
- Foreign key columns must be indexed.

Teacher ID	DeptCode	Fname	Lname
B002	002	David	Warner
B017	002	Sara	Joseph
B009	001	Mike	Brunton

Types of Keys

Candidate Key

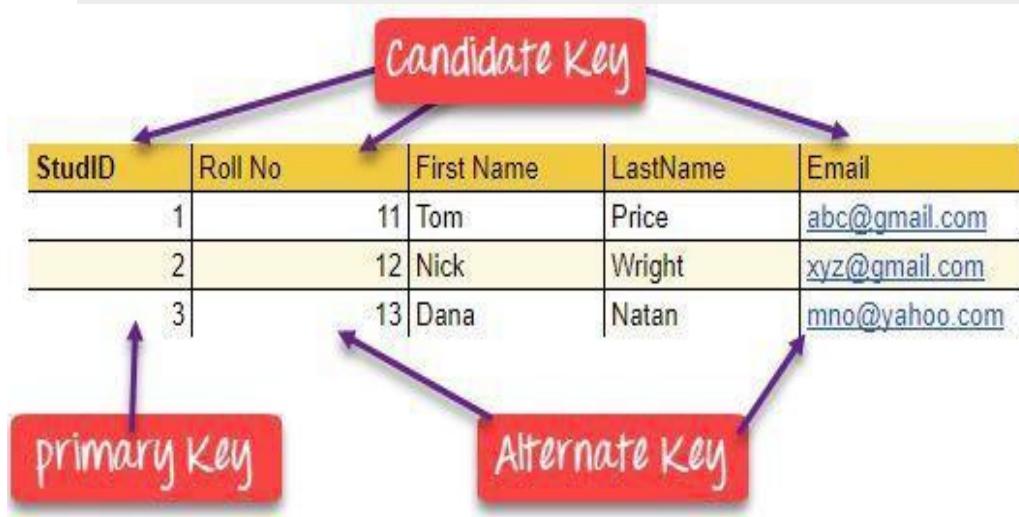
- A candidate key is an attribute or set of an attribute which can uniquely identify a tuple.
- The remaining attributes except for primary key are considered as a candidate key. The candidate keys are as strong as the primary key.



Types of Keys

Rule for defining candidate key

- It must contain unique values
- Candidate key may have multiple attributes
- Must not contain null values
- It should contain minimum fields to ensure uniqueness
- Uniquely identify each record in a table



Types of Keys

Super Key

- A superkey is a group of single or multiple keys which identifies rows in a table.
- A Super key may have additional attributes that are not needed for unique identification.
- In the above-given example, EmpNo and Emp_Name are superkeys.

EmpSSN	Emp_No	Emp_Name
9812345098	AB05	Satish
9876512345	AB06	Anip
199937890	AB07	Khan

Types of Keys

Compound Key

- COMPOUND KEY has two or more attributes that allow you to uniquely recognize a specific record.
- It is possible that each column may not be unique by itself within the database
- Order ID and Product ID could be used as it uniquely identified each record.

OrderNo	ProductID	Product Name	Quantity
B005	JAP102459	Mouse	5
B005	DKT32157 3	USB	10
B005	OMG44678 9	LCD Monitor	20
B004	DKT32157 3	USB	15
B002	OMG44678 9	Laser Printer	3

Types of Keys

Alternate Key

- ALTERNATE KEYS is a column or group of columns in a table that uniquely identify every row in that table.
- A table can have multiple choices for a primary key but only one can be set as the primary key.
- All the keys which are not primary key are called an Alternate Key.

OrderNo	ProductID	Product Name	Quantity
B005	JAP102459	Mouse	5
B005	DKT32157 3	USB	10
B005	OMG44678 9	LCD Monitor	20
B004	DKT32157 3	USB	15
B002	OMG44678 9	Laser Printer	3

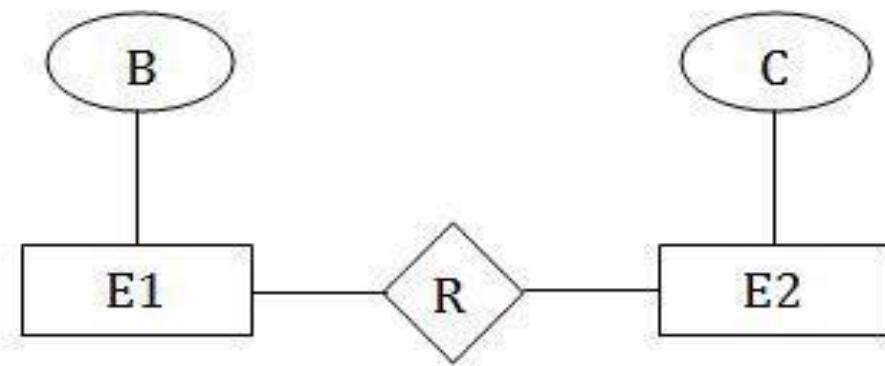
Types of Keys

Unique Key

- A unique key is a set of one or more than one fields/columns of a table that uniquely identify a record in a database table.

OrderNo	ProductID	Product Name	Quantity
B005	JAP102459	Mouse	5
B005	DKT32157 3	USB	10
B005	OMG44678 9	LCD Monitor	20
B004	DKT32157 3	USB	15
B002	OMG44678 9	Laser Printer	3

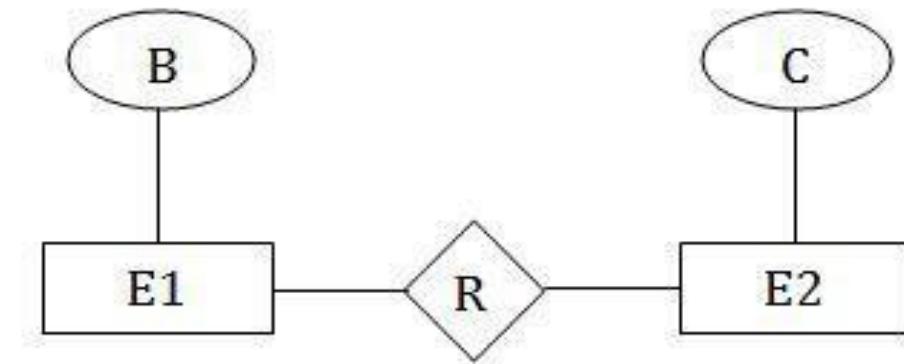
Introduction to Mapping Constraints



Introduction to Mapping Constraints

Introduction

- A mapping constraint is a data constraint that expresses the number of entities to which another entity can be related via a relationship set.
- Mapping constraints defines how many entities can be related to another entity to a relationship



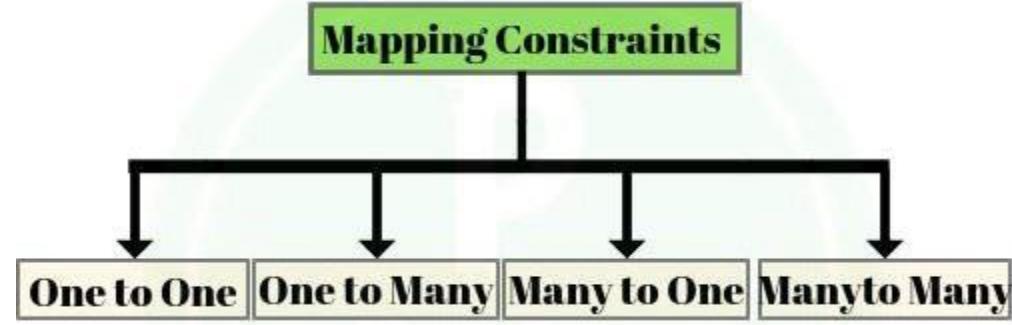
Introduction to Mapping Constraints

Types of Mapping Constraints

- It is most useful in describing the relationship sets that involve more than two entity sets.

Binary Relationship

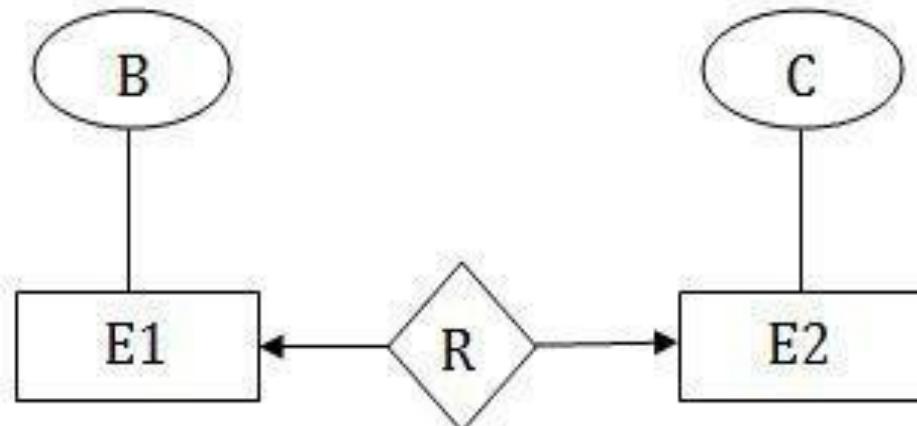
- One to one (1:1)
- One to many (1:M)
- Many to one (M:1)
- Many to many (M:M)



Introduction to Mapping Constraints

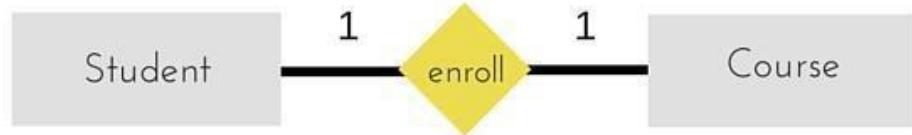
One to One (1:1)

- An entity in E1 is associated with at most one entity in E2, and an entity in E2 is associated with at most one entity in E1.



Introduction to Mapping Constraints

One to One (1:1)

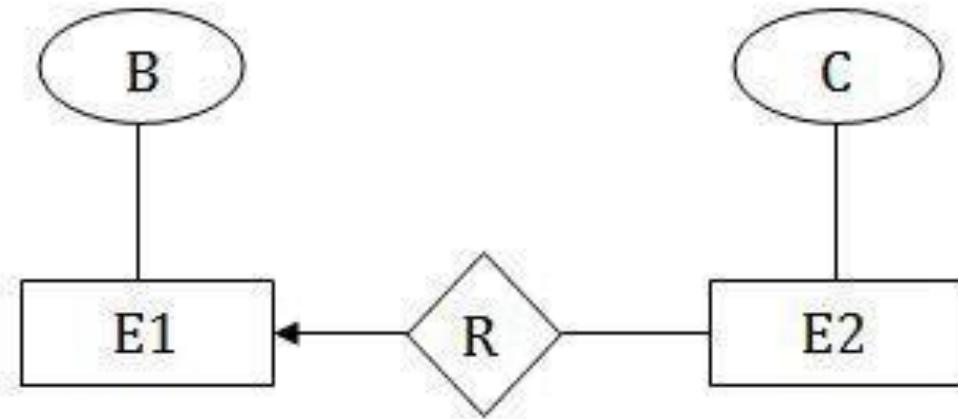


- The above example describes that one student can enroll only for one course and a course will also have only one Student.

Introduction to Mapping Constraints

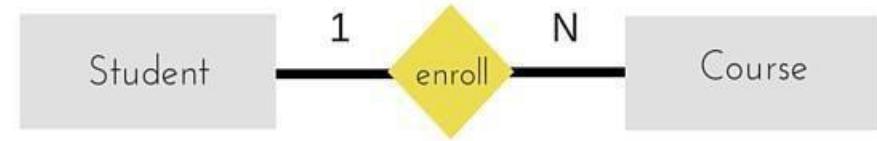
One to Many (1:M)

- An entity in E1 is associated with many entities of E2, and an entity in E2 can be associated with atmost one entity in E1.



Introduction to Mapping Constraints

One to Many (1:M)

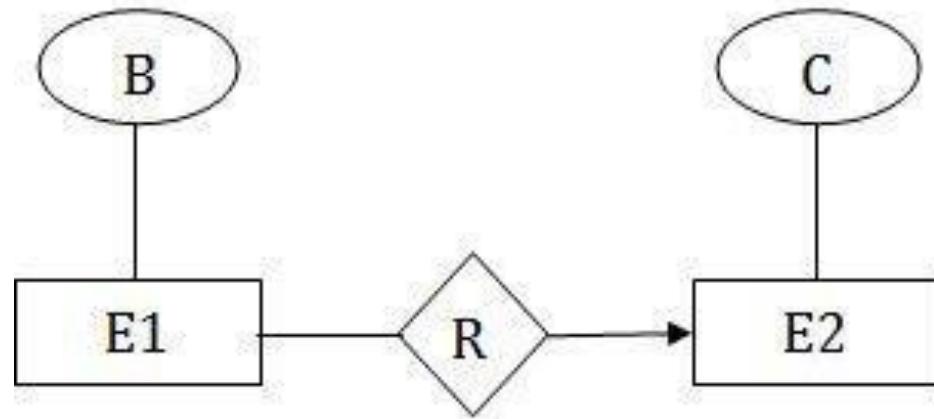


- this relationship, which means that 1 student can opt for many courses, but a course can only have 1 student.

Introduction to Mapping Constraints

Many to One (M:1)

- An entity in E1 is associated with at most one entity in E2, and an entity in E2 is associated with any number of entities in E1.



Introduction to Mapping Constraints

Many to One (M:1)

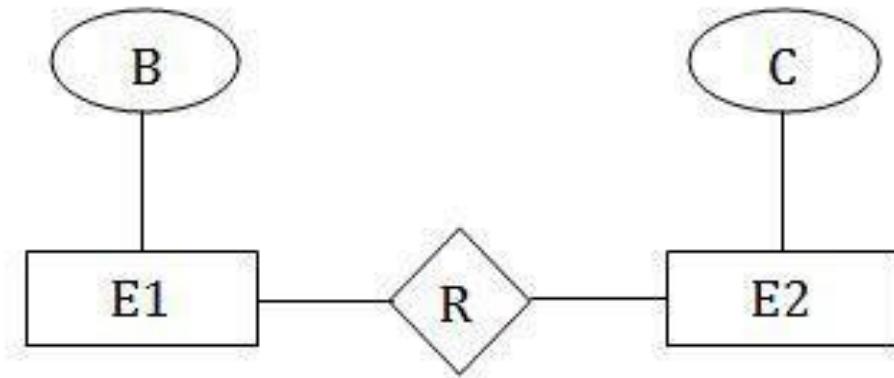


- Student enrolls for only one Course but a Course can have many Students.

Introduction to Mapping Constraints

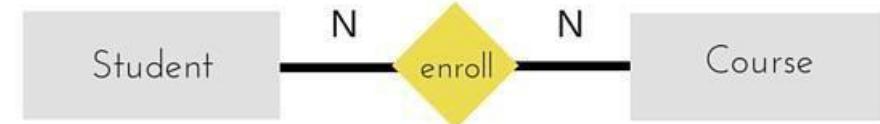
Many to Many (M:M)

- An entity in E1 is associated with any number of entities in E2, and an entity in E2 is associated with any number of entities in E1.

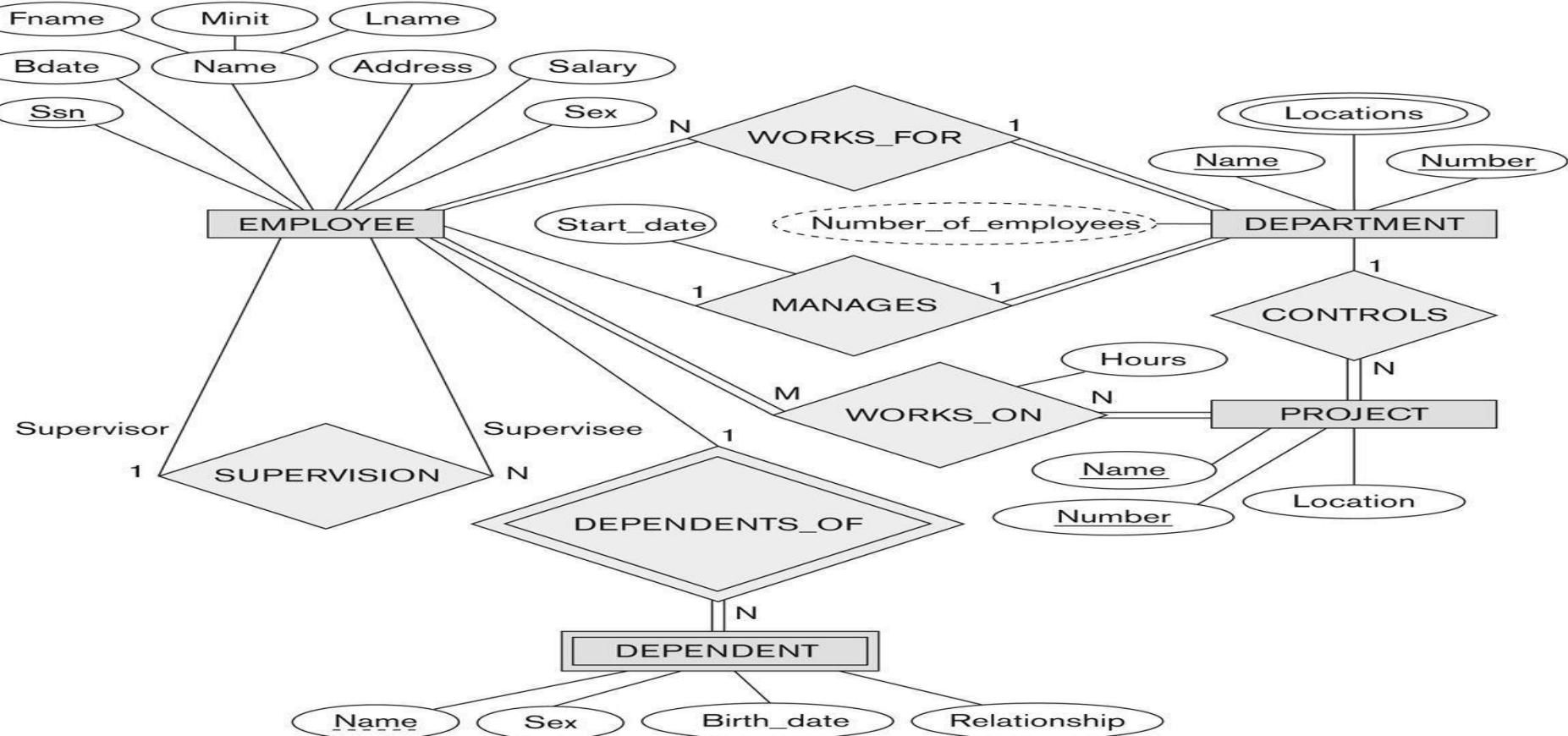


Introduction to Mapping Constraints

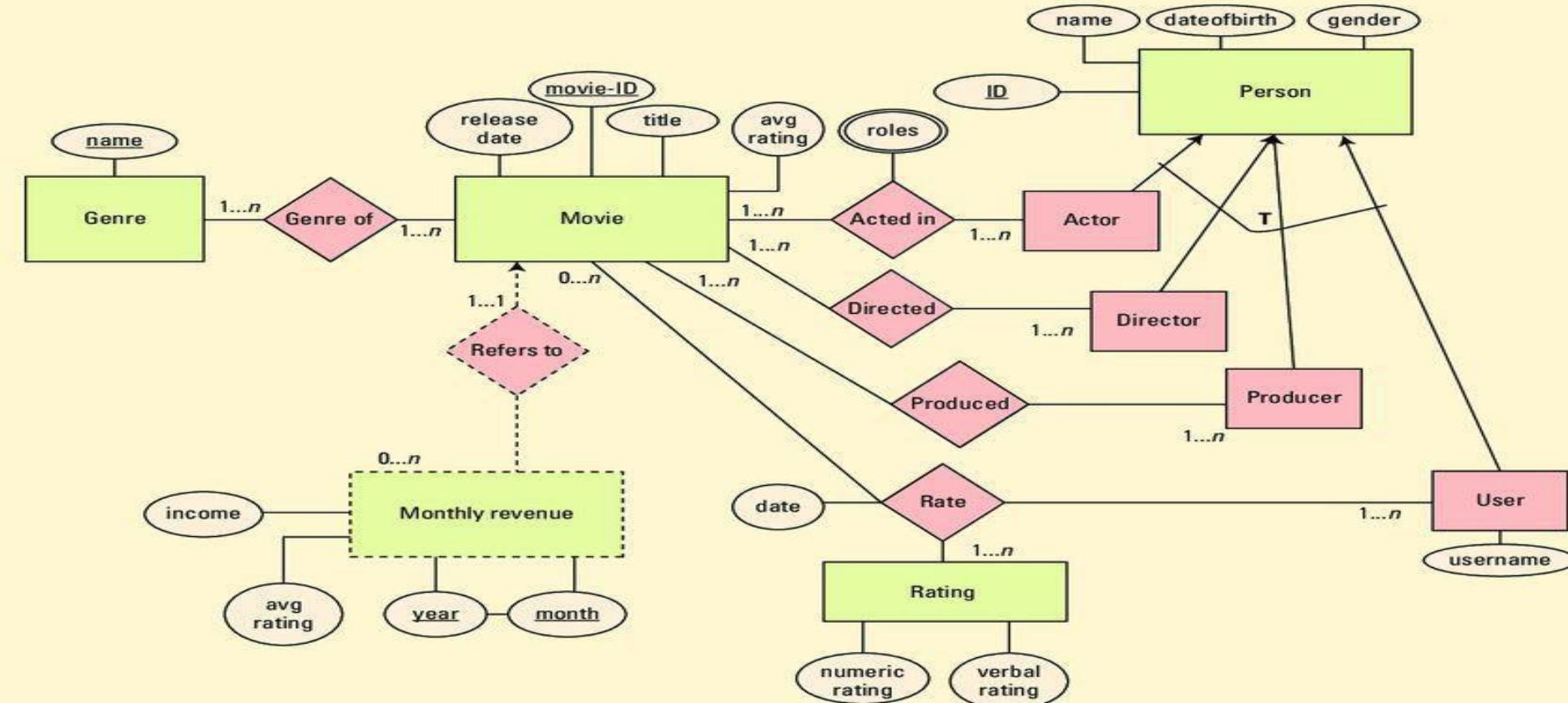
Many to Many (M:M)



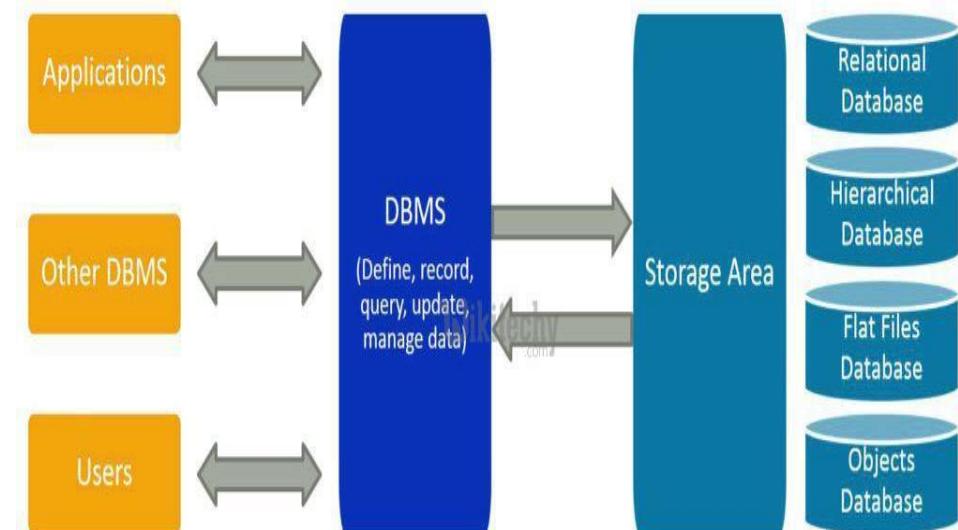
- One student can enroll for more than one courses. And a course can have more than 1 student enrolled in it.



Example 2



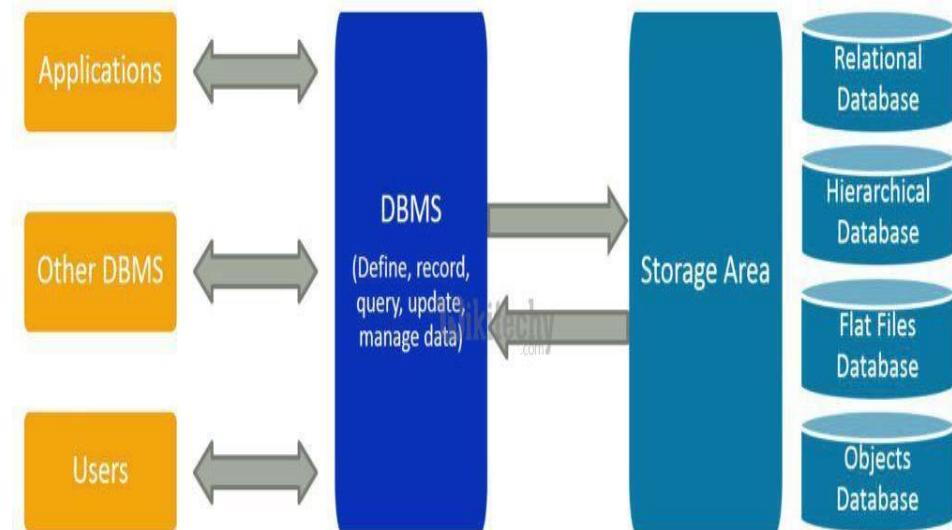
Introduction to DBMS



Introduction to DBMS

Overview of DBMS

- **A database management system** (DBMS) is a software tool that makes it possible to organize data in a database.
- A DBMS consists of the following three elements:
- Physical Database
- Database Engine
- Database Schema



Introduction to DBMS

Advantages of DBMS

- Improved data sharing.
 - Improved data security.
 - Better data integration
 - Minimized data inconsistency
 - Improved data access.
 - Improved decision making.
 - Increased end-user productivity



Image Source:

https://cdn5.vectorstock.com/i/1000x1000/21/59/dbms-database-management-system-computer-data-vector_1115959.jpg

Introduction to DBMS

Disadvantages of DBMS

- Increased costs.
- Management complexity.
- Maintaining currency.
- Frequent upgrade/replacement cycles.



Image Source:

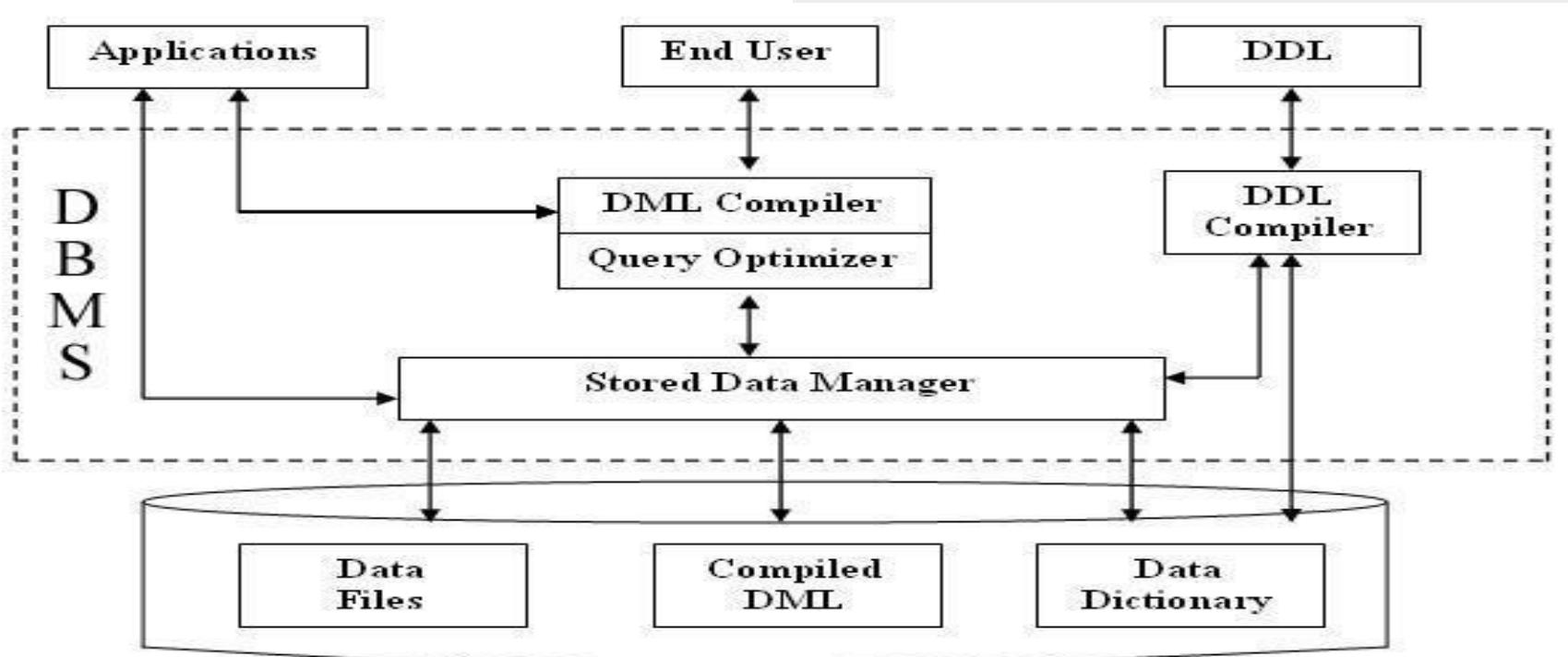
<https://www.qsstudy.com/wp-content/uploads/2018/10/Database-Management-System-3.jpg>

Introduction to DBMS

Applications of DBMS



Structure to DBMS



Introduction to RDBMS

What is RDBMS

- A Relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by E. F. Codd.

RDBMS - What Is RDBMS ?

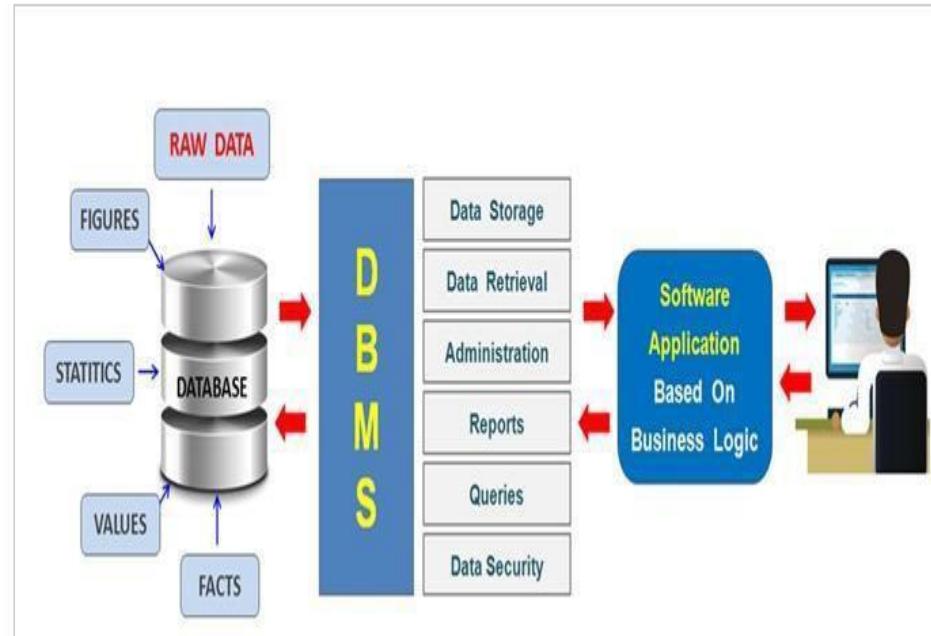


Image Source:

<https://www.learncomputerscienceonline.com/wp-content/uploads/2019/08/What-Is-RDBMS.jpg>

Introduction to RDBMS

Database Architecture

- A Relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by E. F. Codd.

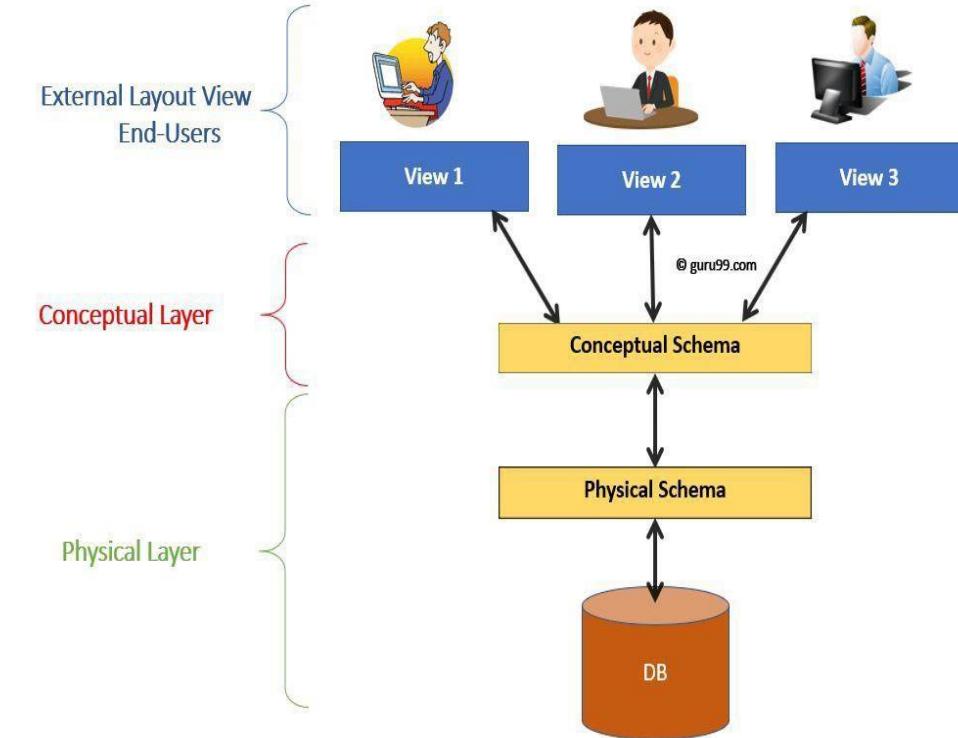
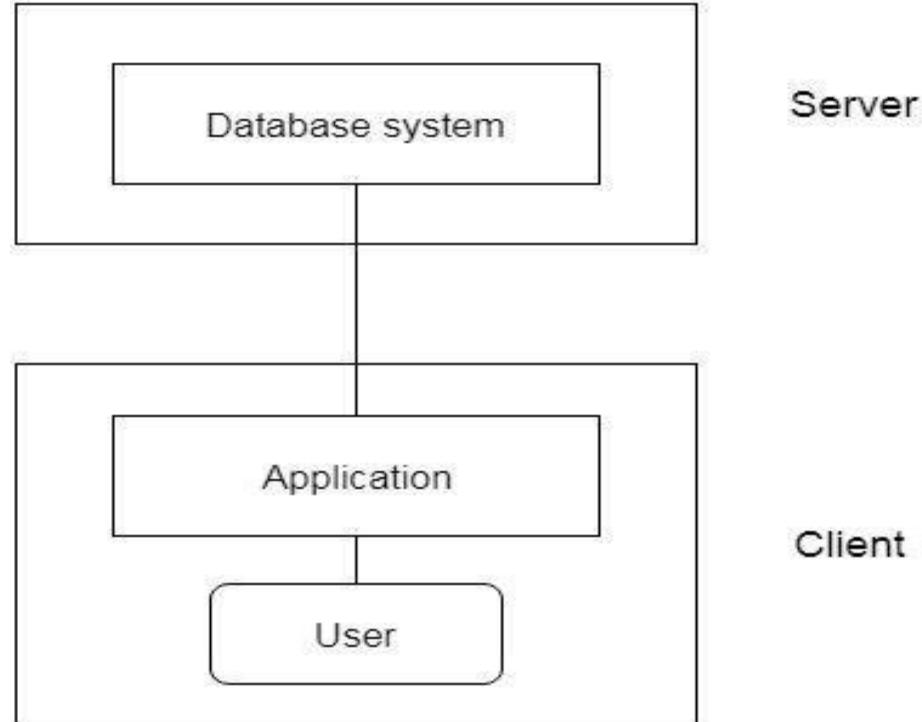


Image Source: https://www.guru99.com/images/1/042919_0417_DataIndep1.png

Introduction to RDBMS

Logical two-tier Client / Server Architecture

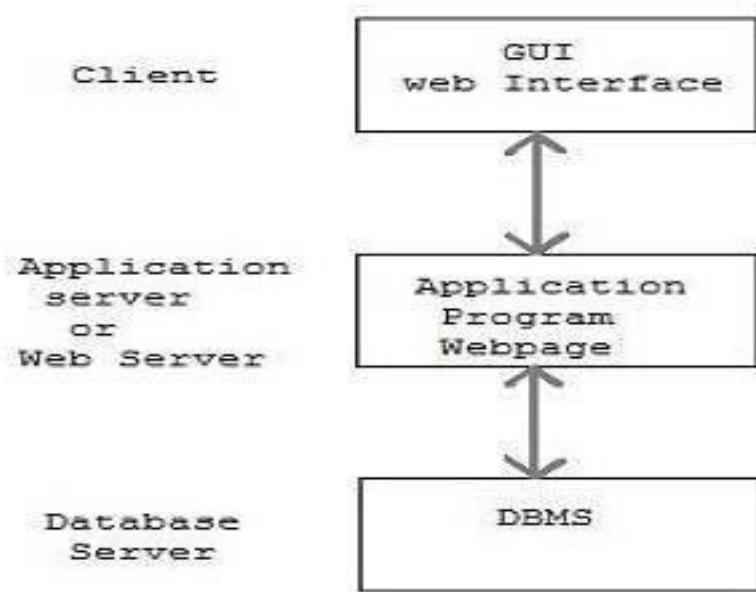
- The user interfaces and application programs are run on the client-side.
- The server side is responsible to provide the functionalities like: query processing and transaction management.



Introduction to RDBMS

Logical two-tier Client / Server Architecture

- The 3-Tier architecture is used in case of large web application.
- The application on the client-end interacts with an application server which further communicates with the database system.



Relational Model

Table also called Relation

Primary Key Domain
Ex: NOT NULL
© guru99.com

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

Tuple OR Row
Total # of rows is Cardinality

Column OR Attributes
Total # of column is Degree

Relational Model

Relational Model Concepts

- Attribute
- Table
- Tuple
- Relational Schema
- Degree Cardinality
- Column
- Relational Instance
- Relational Key
- Attribute Domain

Table also called Relation

Primary Key Domain
Ex: NOT NULL © guru99.com

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

Tuple OR Row

Total # of rows is Cardinality

Column OR Attributes

Total # of column is Degree

Image Source: www.guru99.com/images/1/091318_0803_RelationalD1.png

Relational Model

Relational Integrity Constraints

- Relational Integrity constraints is referred to conditions which must be present for a valid relation.
- Categories
- Domain constraints
- Key constraints
- Referential integrity constraints

Table also called Relation

Primary Key Domain
Ex: NOT NULL © guru99.com

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

Tuple OR Row

Total # of rows is **Cardinality**

Column OR Attributes

Total # of column is **Degree**

Relational Model

Domain Constraints

- Domain constraints can be violated if an attribute value is not appearing in the corresponding domain or it is not of the appropriate data type.
- Domain constraints specify that within each tuple, and the value of each attribute must be unique.

Example:

Create DOMAIN CustomerName CHECK (value not NULL)

The example shown demonstrates creating a domain constraint such that CustomerName is not NULL

Relational Model

Key Constraints

- An attribute that can uniquely identify a tuple in a relation is called the key of the table. The value of the attribute for different tuples in the relation has to be unique.

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

In the given table, CustomerID is a key attribute of Customer Table. It is most likely to have a single key for one customer, CustomerID =1 is only for the CustomerName =" Google".

Relational Model

Referential Integrity Constraints

- Referential integrity constraints is base on the concept of Foreign Keys.

The diagram illustrates referential integrity constraints between two relational tables: Customer and Billing. The Customer table (top) contains three rows with CustomerID 1 (Google, Active), 2 (Amazon, Active), and 3 (Apple, Inactive). The Billing table (bottom) contains three rows with InvoiceNo 1, 2, and 3. The CustomerID column in the Billing table is a foreign key that references the CustomerID column in the Customer table. A dashed green arrow points from the CustomerID value '1' in the first row of the Billing table to the corresponding row in the Customer table. A red box highlights the '1' in the CustomerID column of the first Billing row, and another red box highlights the '1' in the CustomerID column of the first Customer row. A red box also highlights the '1' in the CustomerID column of the second Billing row, which corresponds to CustomerID 2 in the Customer table.

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

Customer

InvoiceNo	CustomerID	Amount
1	1	\$100
2	1	\$200
3	2	\$150

Billing

Relational Model

Advantages

- Simplicity
- Structural Independence
- Easy to use
- Query Capability
- Data Independence
- Scalable

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

InvoiceNo	CustomerID	Amount
1	1	\$100
2	1	\$200
3	2	\$150

Relational Model

Properties of Relations

- Name of the relation is distinct from all other relations.
- Each relation cell contains exactly one atomic (single) value
- Each attribute contains a distinct name
- Attribute domain has no significance
- tuple has no duplicate value
- Order of tuple can have a different sequence

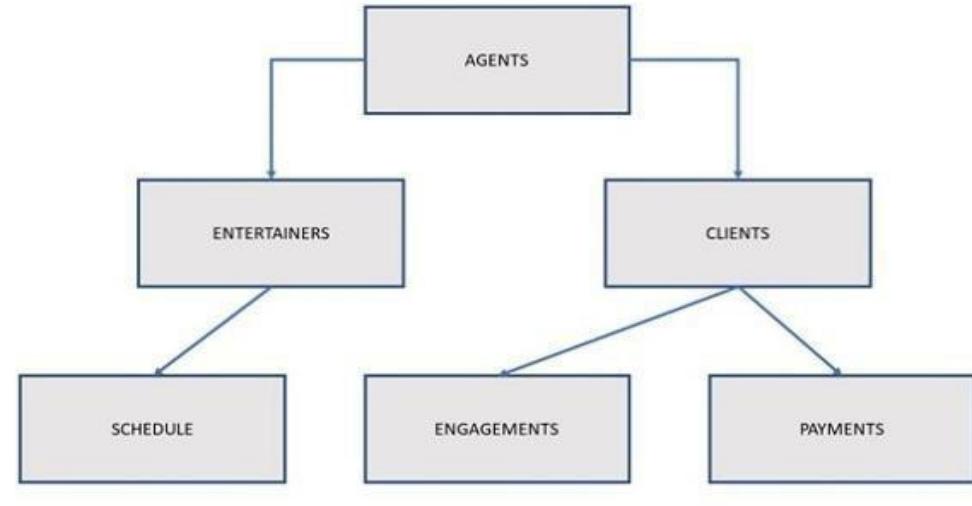
CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

Customer

InvoiceNo	CustomerID	Amount
1	1	\$100
2	1	\$200
3	2	\$150

Billing

Introduction to Hierarchical Model



Introduction to Hierarchical Model

Overview of Hierarchical Model

- A hierarchical model represents the data in a tree-like structure in which there is a single parent for each record.
- The hierarchy starts from the **Root** data, and expands like a tree, adding child nodes to the parent nodes.

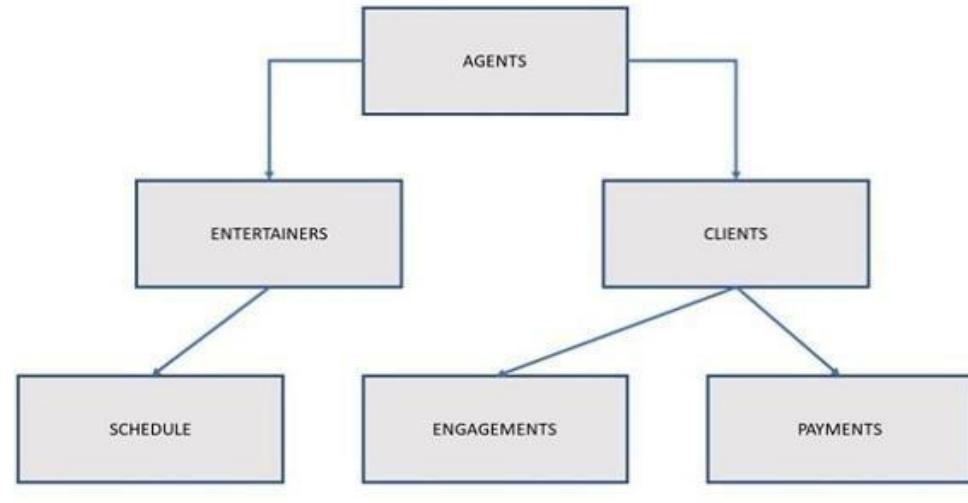
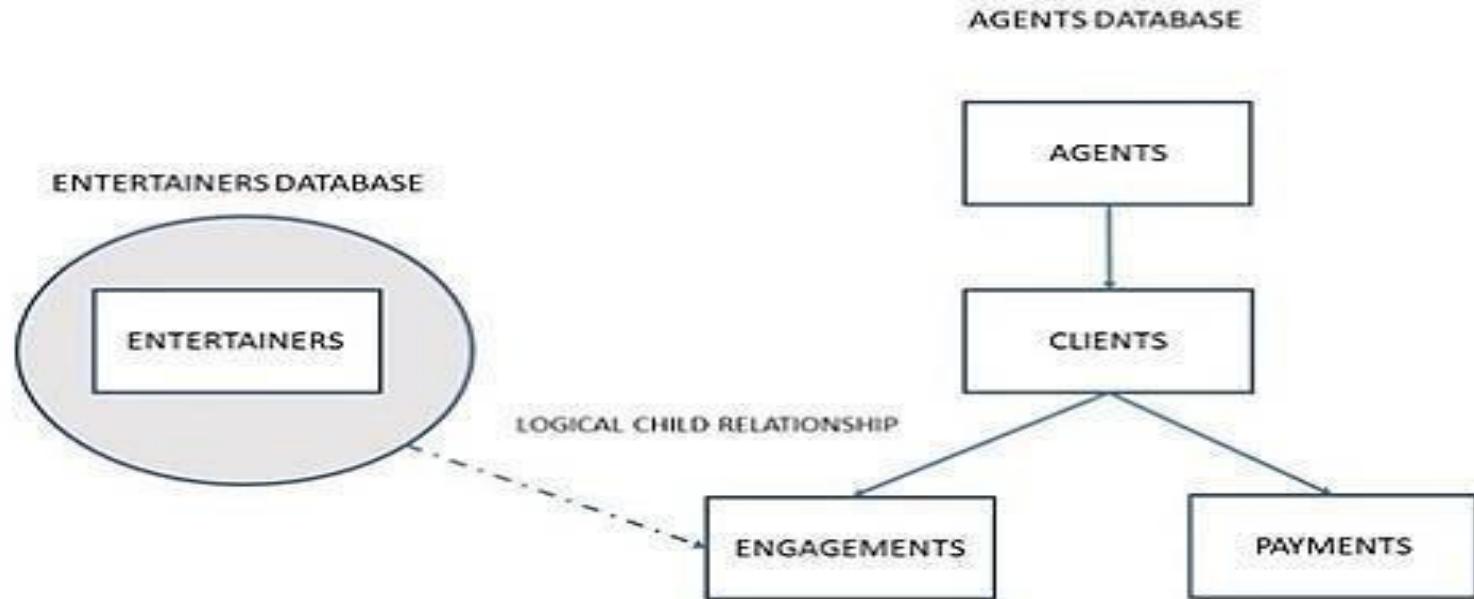


Image Source <https://www.tutorialspoint.com/assets/questions/images/154411-1532346635.jpg>

Introduction to Hierarchical Model



Introduction to Hierarchical Model

Disadvantages

- When a user needs to store a record in a child table that is currently unrelated to any record in a parent table, it gets difficulty in recording and user must record an additional entry in the parent table.
- This type of database cannot support complex relationships, and there is also a problem of redundancy.

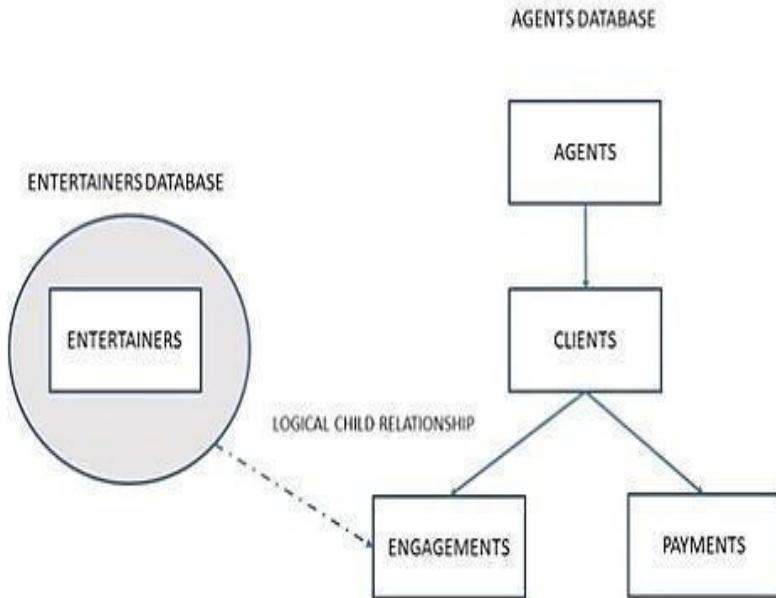
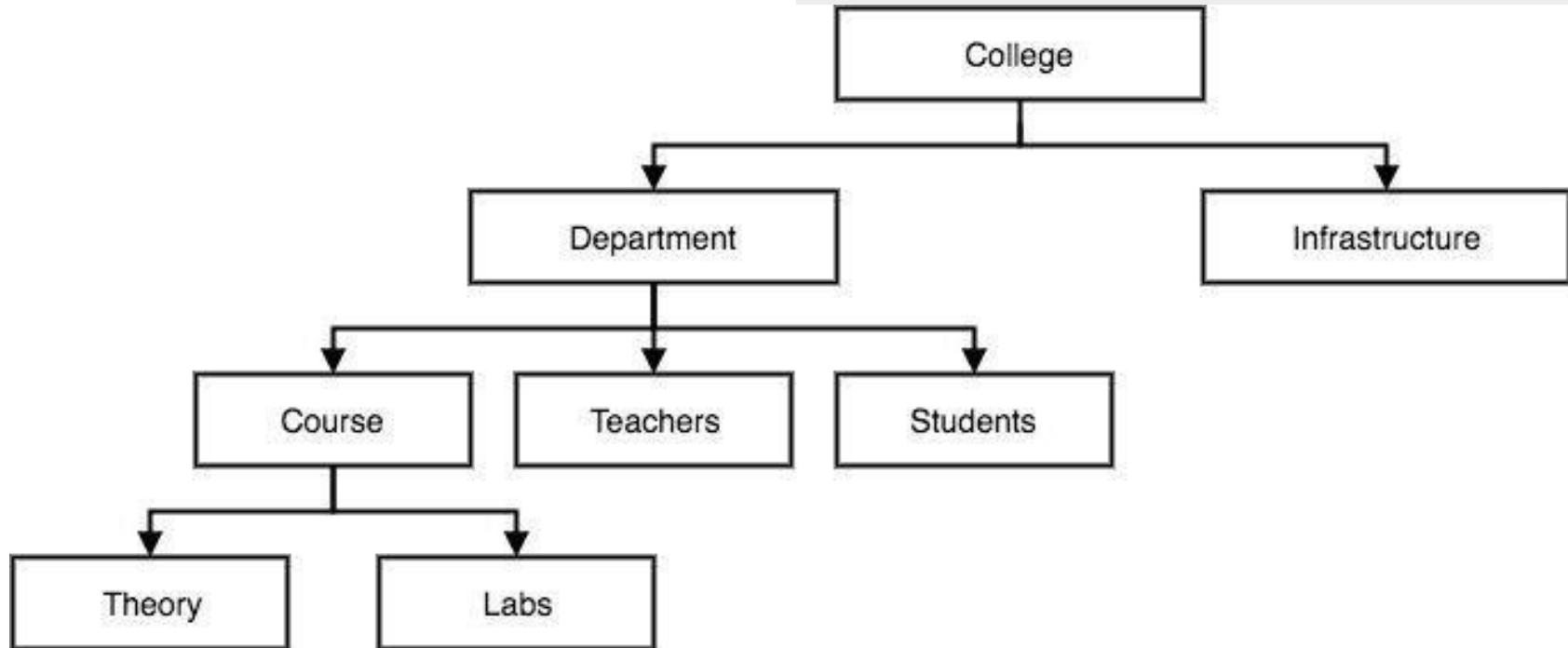


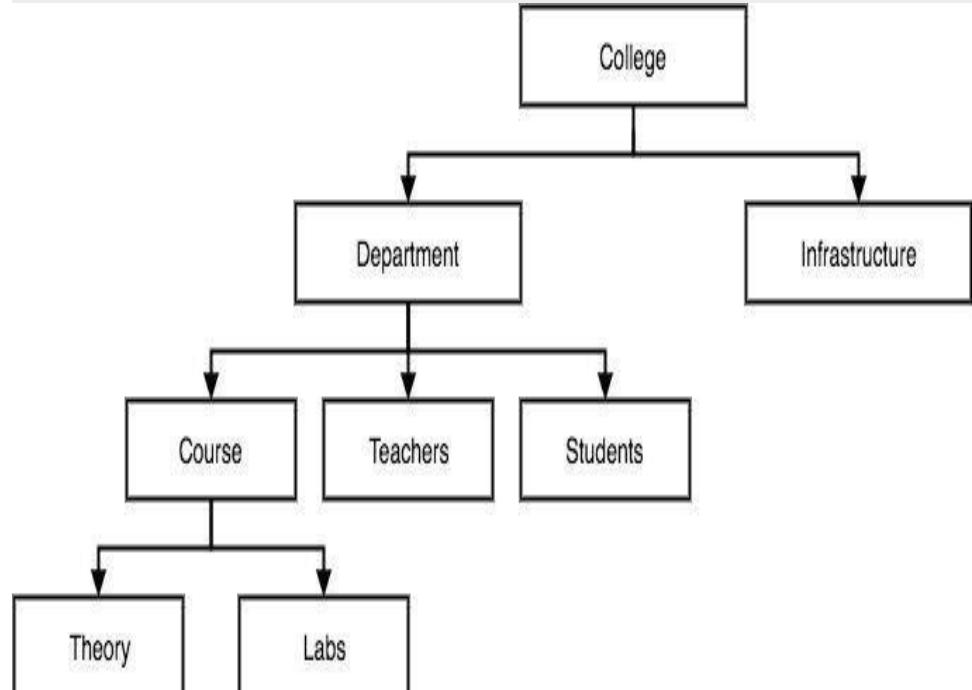
Image Source <https://www.tutorialspoint.com/assets/questions/images/154411-1532346635.jpg>

Introduction to Hierarchical Model

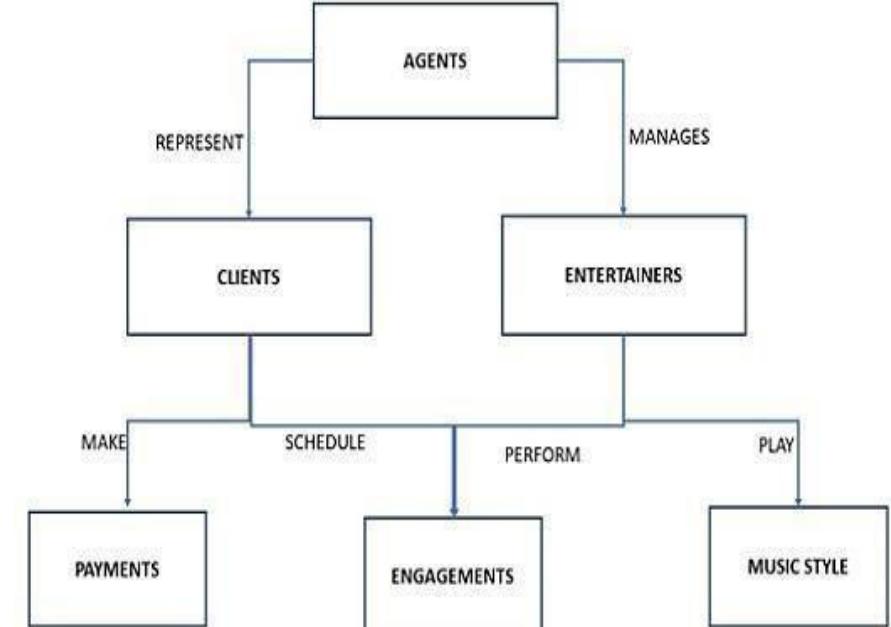


Examples

- Employee Management
- Project Management
- Event Management



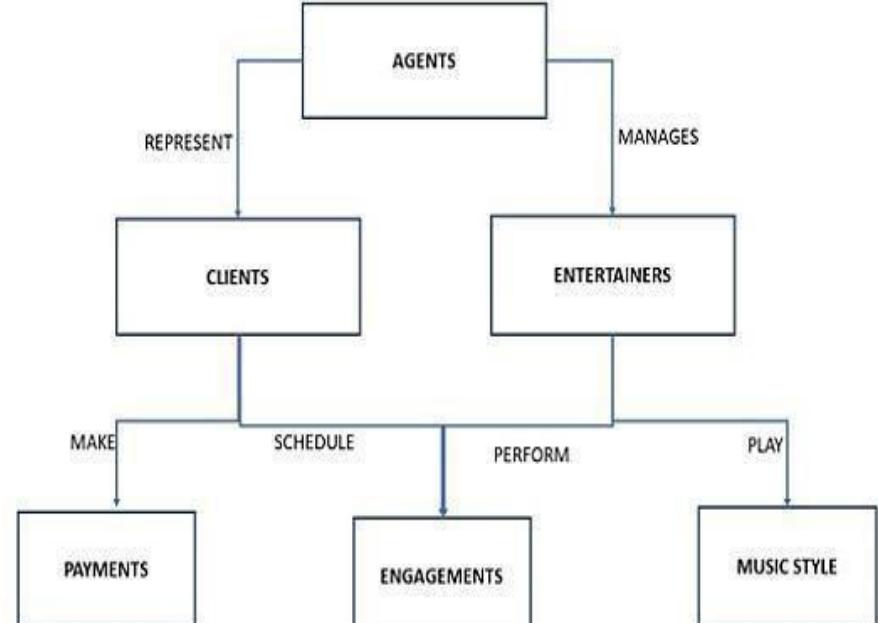
Introduction to Network Model



Introduction to Network Model

Overview of Network Model

- The network model is the extension of the hierarchical structure because it allows many-to-many relationships to be managed in a tree-like structure that allows multiple parents.



Introduction to Network Model

Overview of Network Model

- It can represent redundancy in data more efficiently than that in the hierarchical model.
- There can be more than one path from a previous node to successor node/s.

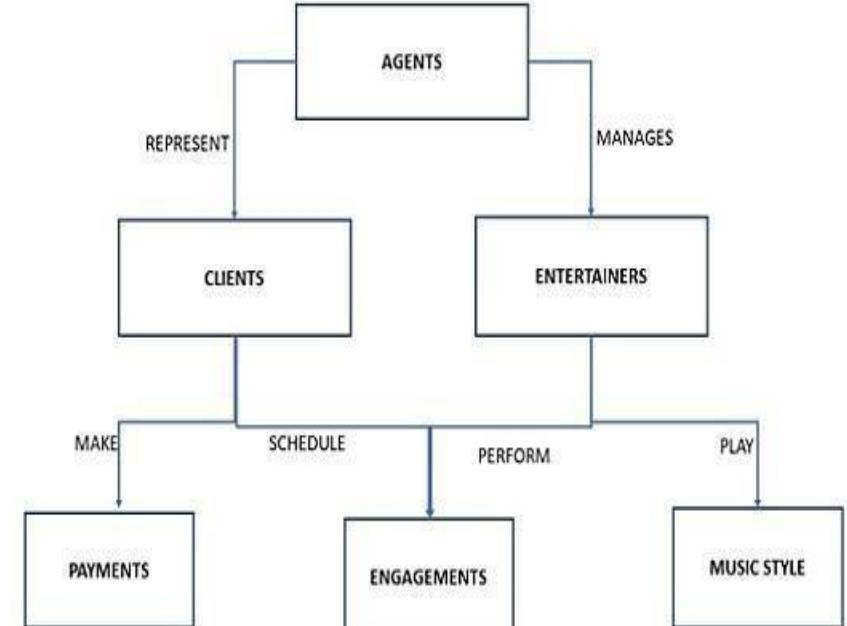


Image Source

<https://www.tutorialspoint.com/assets/questions/images/120543-1532343127.>

Introduction to Network Model

Overview of Network Model

- The operations of the network model are maintained by indexing structure of linked list (circular) where a program maintains a current position and navigates from one record to another by following the relationships in which the record participates.
- Records can also be located by supplying key values.

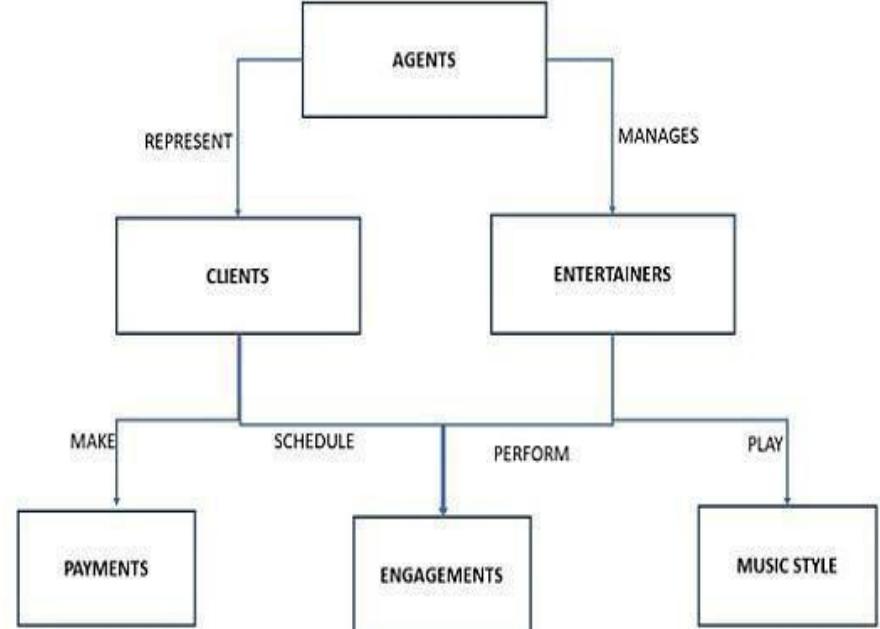


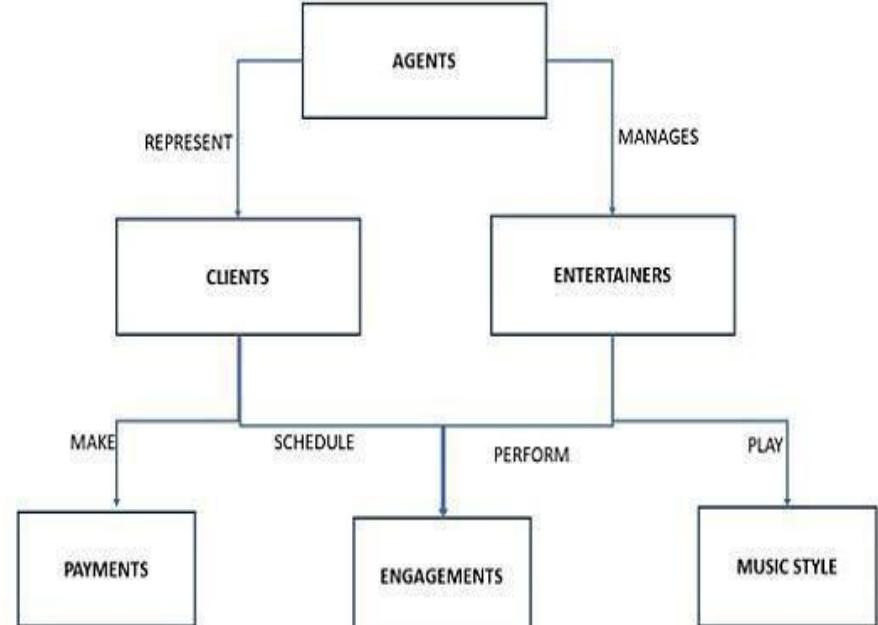
Image Source

<https://www.tutorialspoint.com/assets/questions/images/120543-1532343127.jpg>

Introduction to Network Model

Advantages

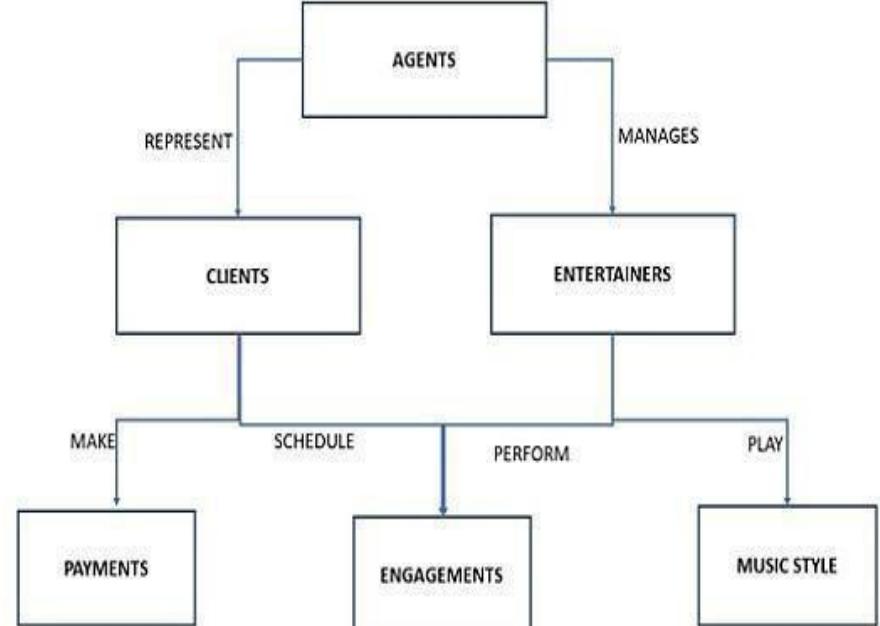
- Fast data access.
- It also allows users to create queries that are more complex than those they created using a hierarchical database. So, a variety of queries can be run over this model.



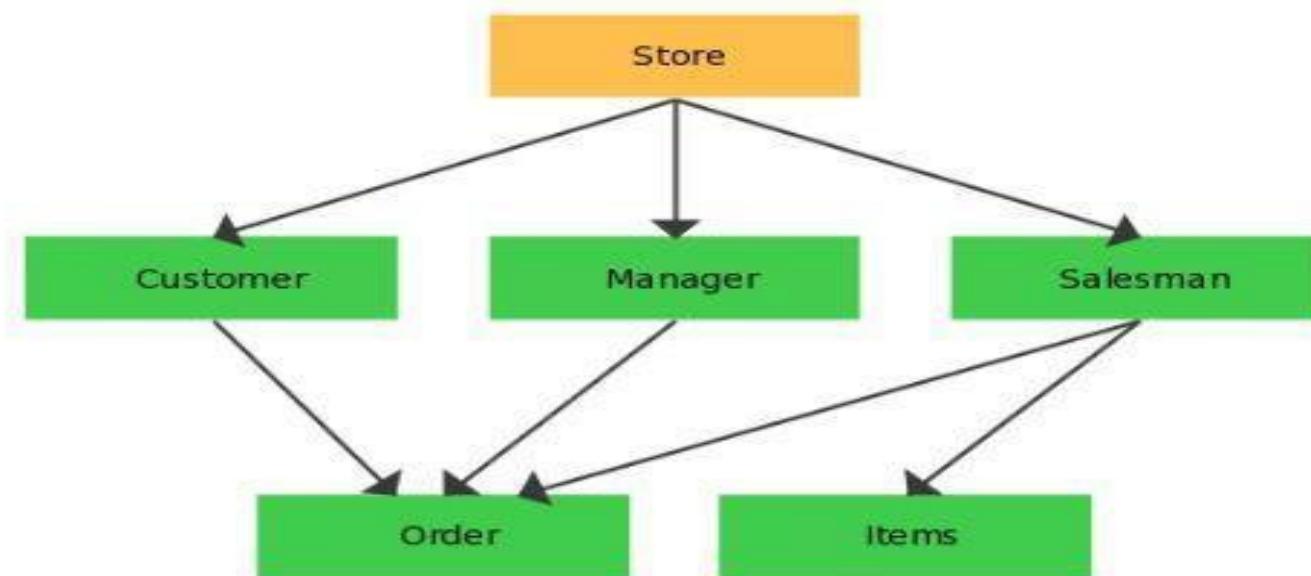
Introduction to Network Model

Disadvantages

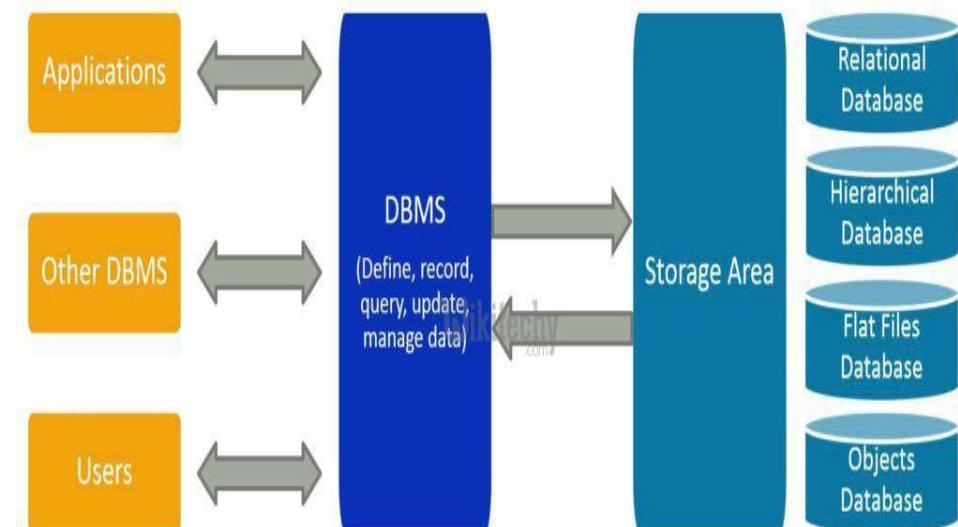
- A user must be very familiar with the structure of the database to work through the set structures.
- Updating inside this database is a tedious task. One cannot change a set structure without affecting the application programs that use this structure to navigate through the data.



Example



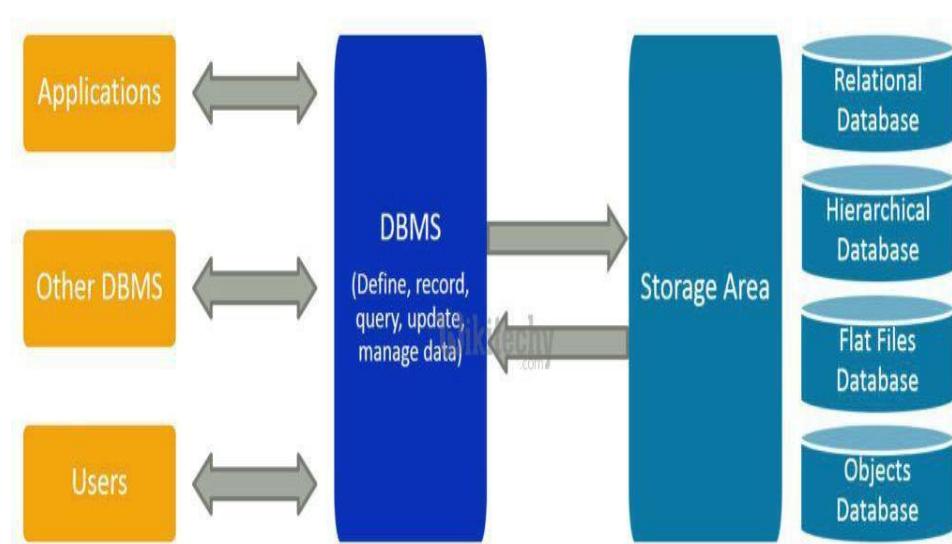
Introduction to DBMS



Introduction to DBMS

Overview of DBMS

- **A database management system** (DBMS) is a software tool that makes it possible to organize data in a database.
- A DBMS consists of the following three elements:
- Physical Database
- Database Engine
- Database Schema



Introduction to DBMS

Advantages of DBMS

- Improved data sharing.
- Improved data security.
- Better data integration
- Minimized data inconsistency
- Improved data access.
- Improved decision making.
- Increased end-user productivity.



Image Source:

<https://cdn5.vectorstock.com/i/1000x1000/21/59/dbms-database-management-system-computer-data-vect>

Introduction to DBMS

Disadvantages of DBMS

- Increased costs.
- Management complexity.
- Maintaining currency.
- Frequent upgrade/replacement cycles.



Image Source:

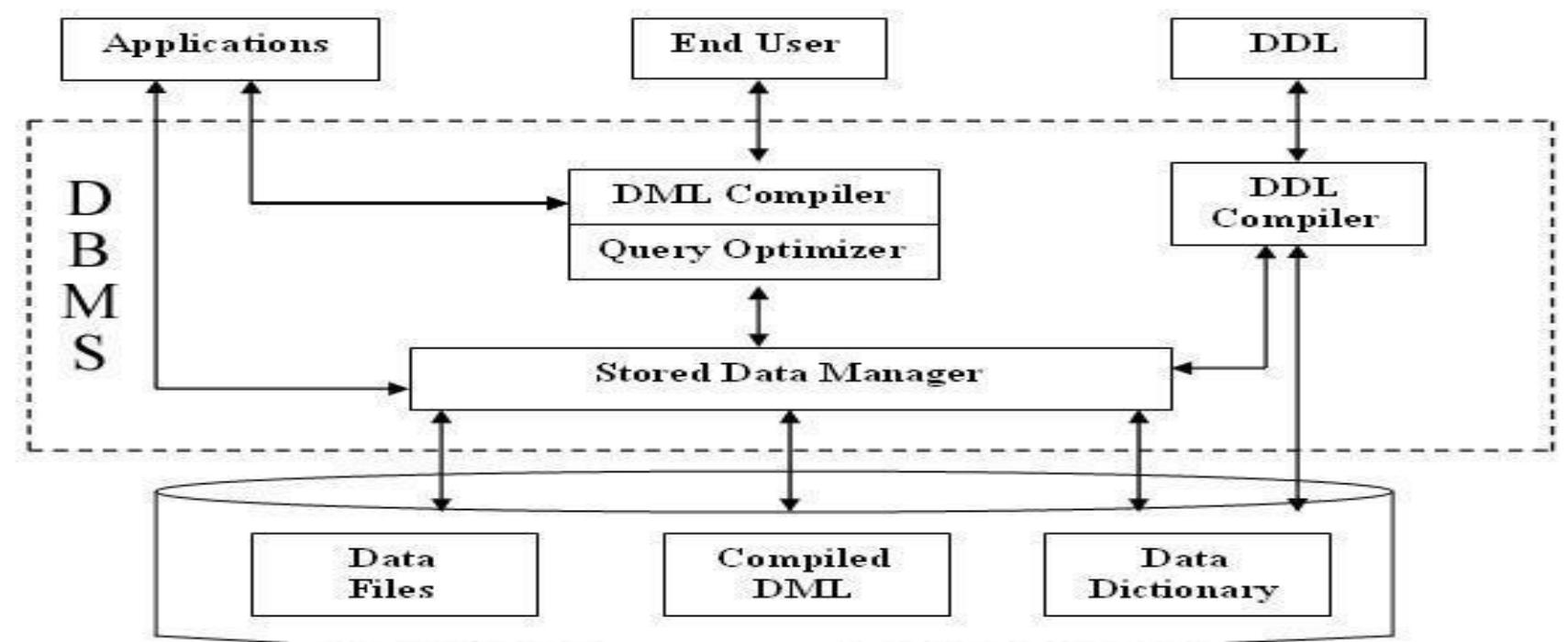
<https://www.qsstudy.com/wp-content/uploads/2018/10/Database-Management-System-3.jpg>

Introduction to DBMS

Applications of DBMS



Structure to DBMS



DBMS VS File System

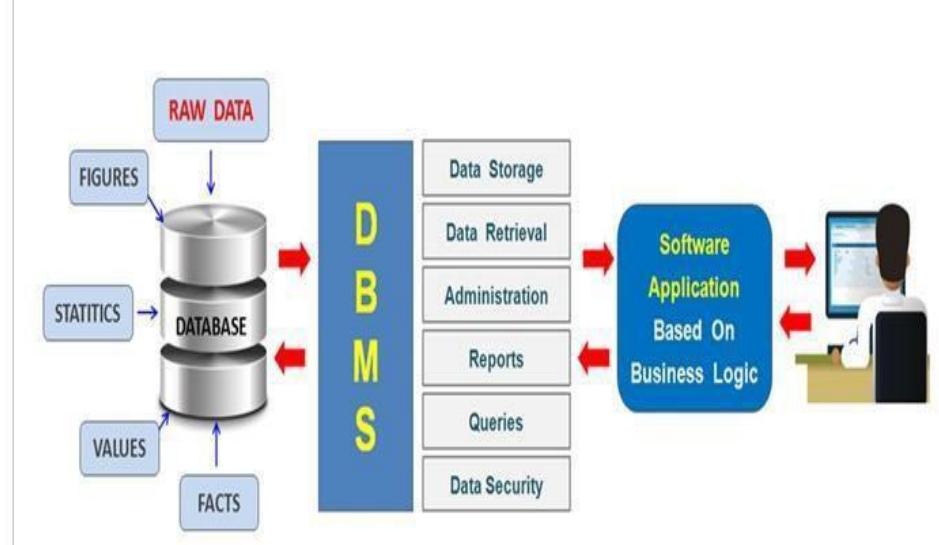
DBMS	File System
In DBMS, the user is not required to write the procedures.	In this system, the user has to write the procedures for managing the database.
DBMS gives an abstract view of data that hides the details.	File system provides the detail of the data representation and storage of data.
DBMS provides a crash recovery mechanism, i.e., DBMS protects the user from the system failure.	File system doesn't have a crash mechanism, i.e., if the system crashes while entering some data, then the content of the file will be lost.
DBMS provides a good protection mechanism.	It is very difficult to protect a file under the file system.
DBMS contains a wide variety of sophisticated techniques to store and retrieve the data.	File system can't efficiently store and retrieve the data.
DBMS takes care of Concurrent access of data using some form of locking.	In the File system, concurrent access has many problems like redirecting the file while other deleting some information or updating some information.

DBMS VS RDBMS

No.	DBMS	RDBMS
1)	DBMS applications store data as file .	RDBMS applications store data in a tabular form .
2)	In DBMS, data is generally stored in either a hierarchical form or a navigational form.	In RDBMS, the tables have an identifier called primary key and the data values are stored in the form of tables.
3)	Normalization is not present in DBMS.	Normalization is present in RDBMS.
4)	DBMS does not apply any security with regards to data manipulation.	RDBMS defines the integrity constraint for the purpose of ACID (Atomocity, Consistency, Isolation and Durability) property.
5)	DBMS uses file system to store data, so there will be no relation between the tables .	in RDBMS, data values are stored in the form of tables, so a relationship between these data values will be stored in the form of a table as well.
6)	DBMS does not support distributed database .	RDBMS supports distributed database .
7)	DBMS is meant to be for small organization and deal with small data . it supports single user .	RDBMS is designed to handle large amount of data . it supports multiple users .
8)	Examples of DBMS are file systems, xml etc.	Example of RDBMS are mysql, postgres, sql server, oracle etc.

Introduction to RDBMS

RDBMS - What Is RDBMS ?



Introduction to RDBMS

What is RDBMS

- A Relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by E. F. Codd.

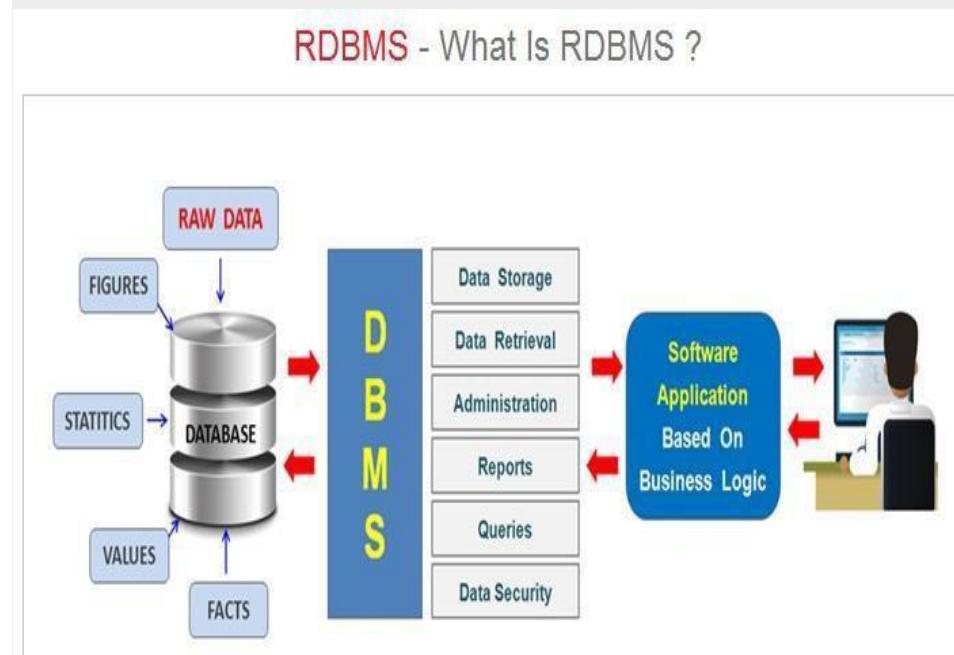


Image Source:

<https://www.learncomputerscienceonline.com/wp-content/uploads/2019/08/What-Is-RDBMS.jpg>

Introduction to RDBMS

Database Architecture

- A Relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by E. F. Codd.

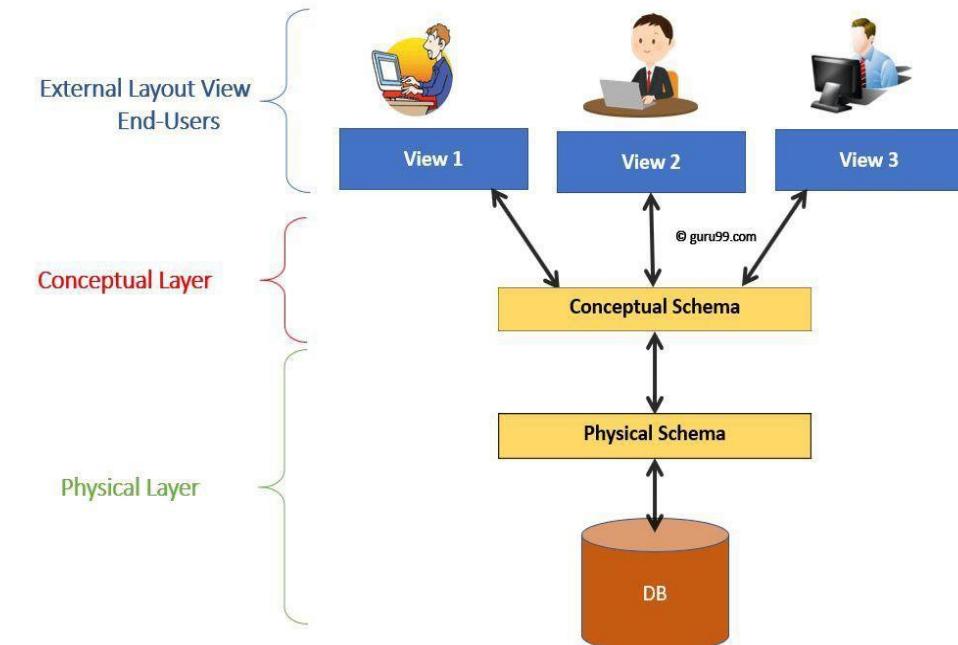
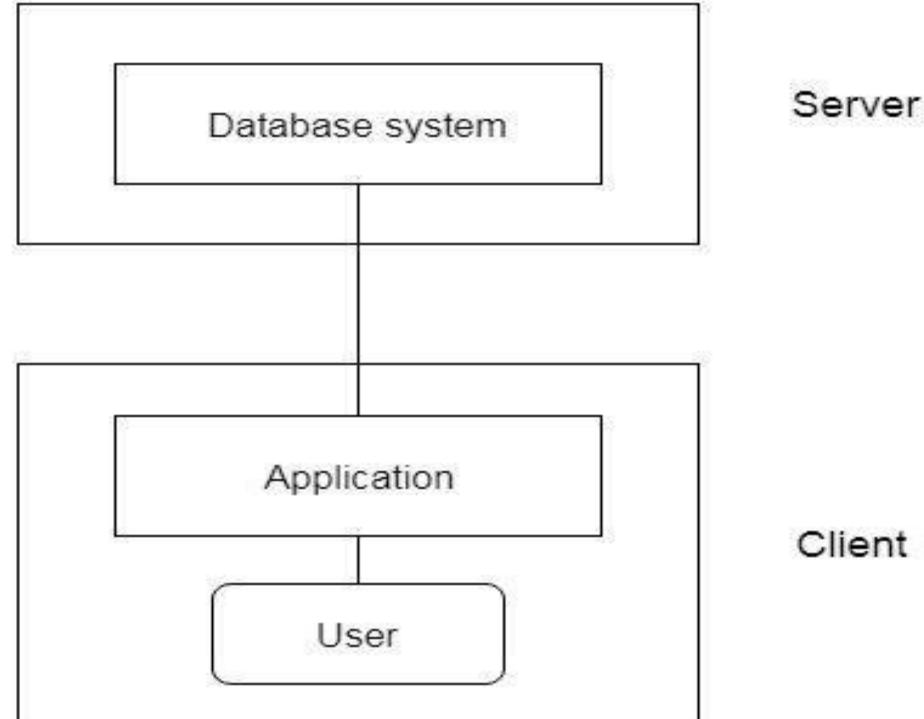


Image Source: https://www.guru99.com/images/1/042919_0417_DataIndep1.png

Introduction to RDBMS

Logical two-tier Client / Server Architecture

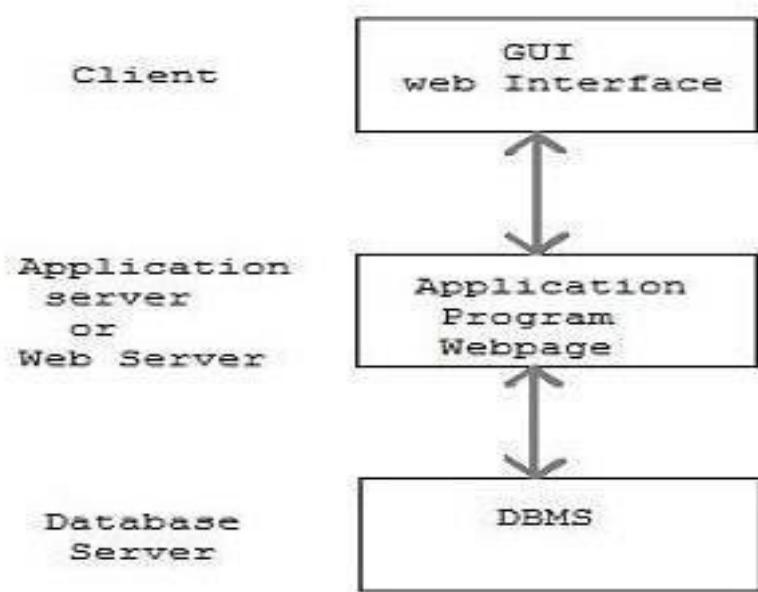
- The user interfaces and application programs are run on the client-side.
- The server side is responsible to provide the functionalities like: query processing and transaction management.



Introduction to RDBMS

Logical two-tier Client / Server Architecture

- The 3-Tier architecture is used in case of large web application.
- The application on the client-end interacts with an application server which further communicates with the database system.



Introduction to RDBMS

Three schema Architecture

- This framework is used to describe the structure of a specific database system.
- The three schema architecture is also used to separate the user applications and physical database.
- The three schema architecture contains three-levels. It breaks the database down into three different categories.

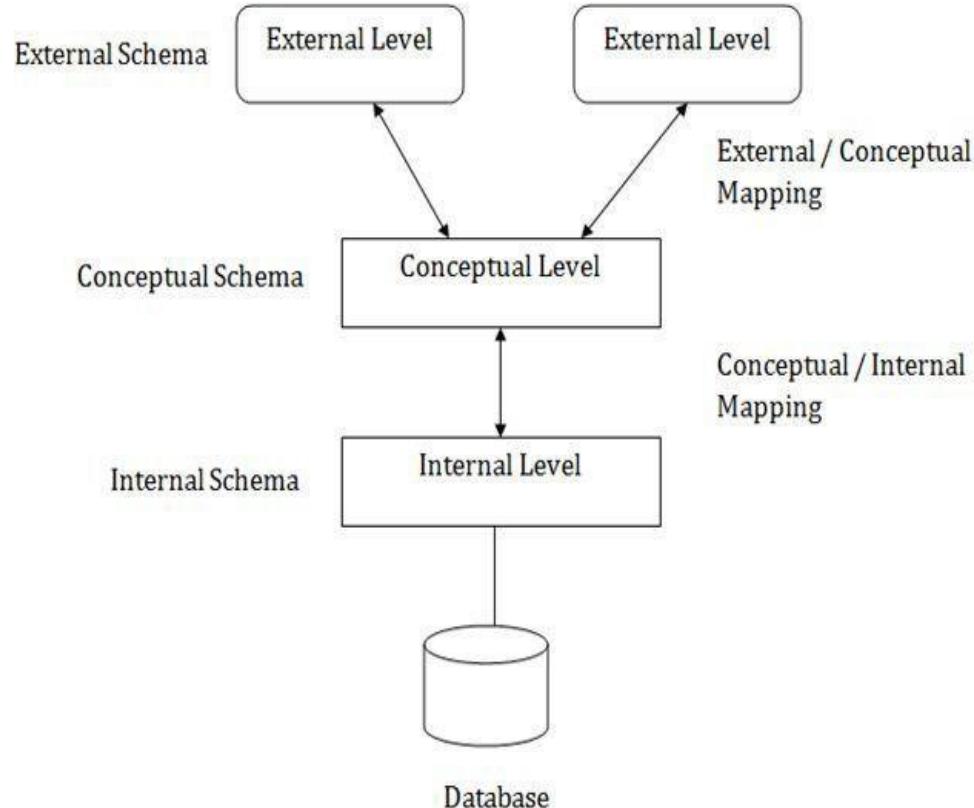


Image Source: <https://static.javatpoint.com/dbms/images/dbms-three-schema-architecture.png>

Introduction to RDBMS

Internal Level

- The internal level has an internal schema which **describes the physical storage structure** of the database.
- The internal schema is also known as a **physical schema**.
- It uses the physical data model.
- It is used to define that how the data will be **stored in a block**.
- The physical level is used to describe complex **low-level data structures** in detail.

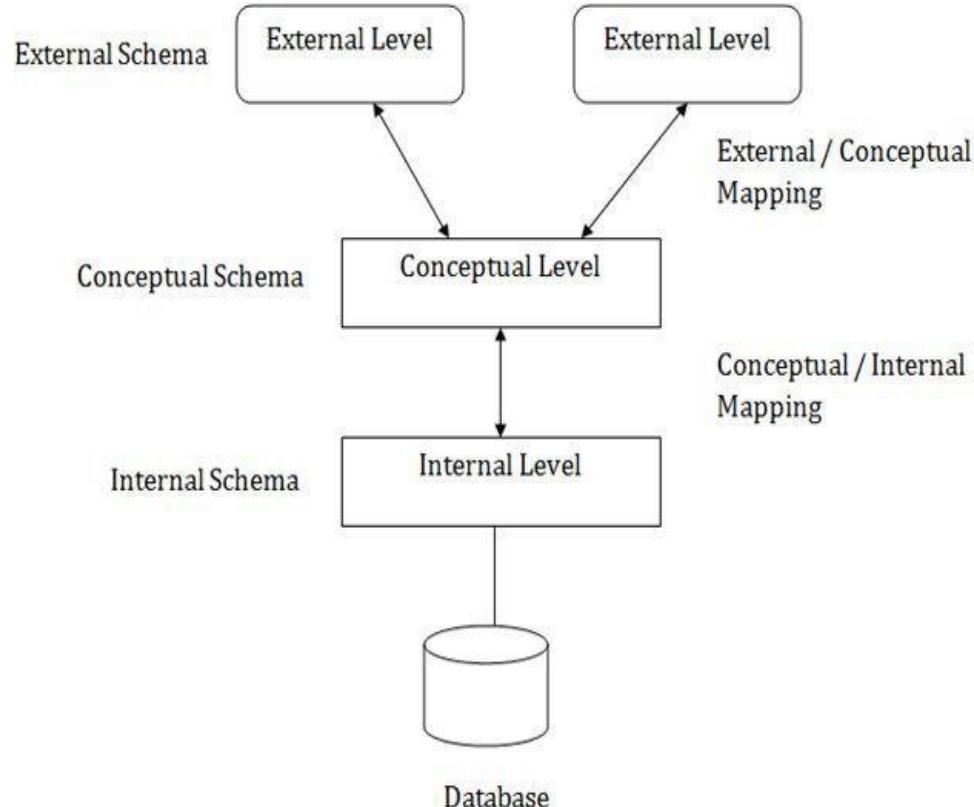


Image Source: <https://static.javatpoint.com/dbms/images/dbms-three-schema-architecture.png>

Introduction to RDBMS

Conceptual Level

- The conceptual schema describes the **design of a database** at the conceptual level. Conceptual level is also known as **logical level**.
- The conceptual schema describes the **structure of the whole database**.
- The conceptual level describes what data are to be stored in the database and also describes what **relationship exists among those data**.
- Programmers and database administrators work at this level.

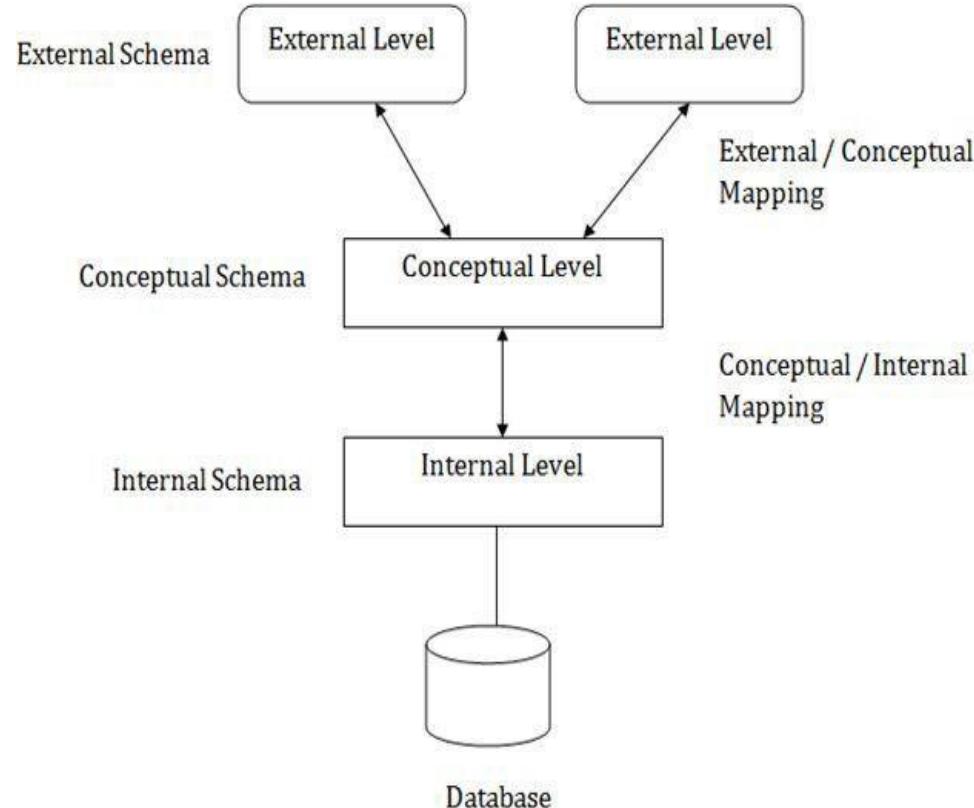
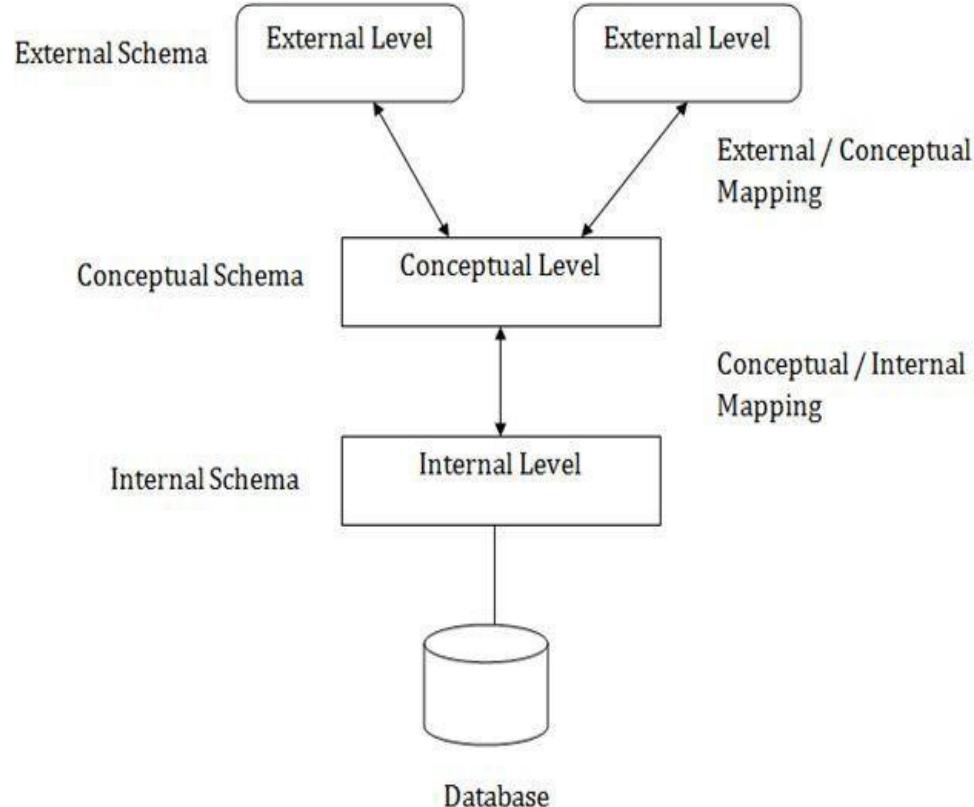


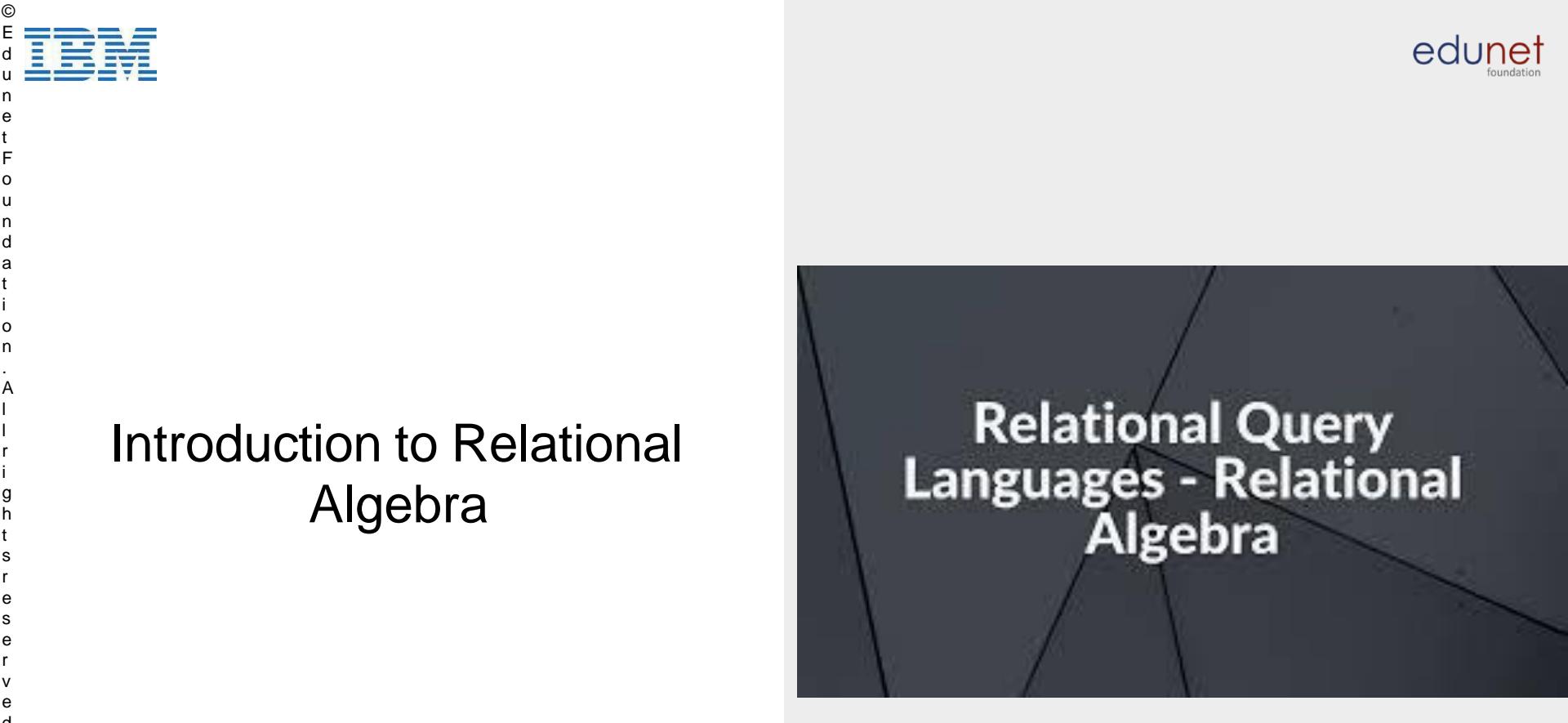
Image Source: <https://static.javatpoint.com/dbms/images/dbms-three-schema-architecture.png>

Introduction to RDBMS

External Level

- At the external level, a database contains several schemas that sometimes called as subschema. The subschema is used to describe the different view of the database.
- An external schema is also known as **view schema**.
- The view schema describes the end **user interaction with database systems**.



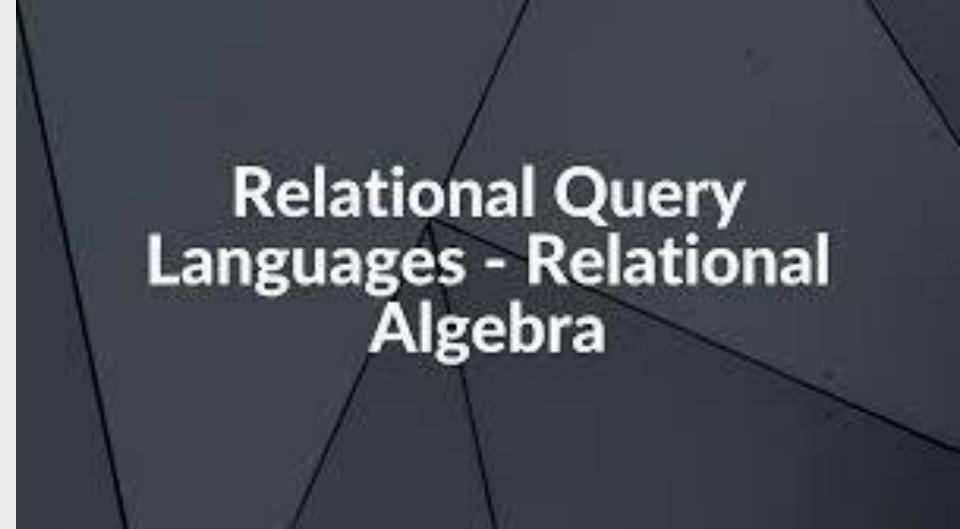


Introduction to Relational Algebra

Introduction to Relational Algebra

Introduction

- Relational Algebra is procedural query language, which takes Relation as input and generate relation as output.
- Relational algebra operations are performed recursively on a relation

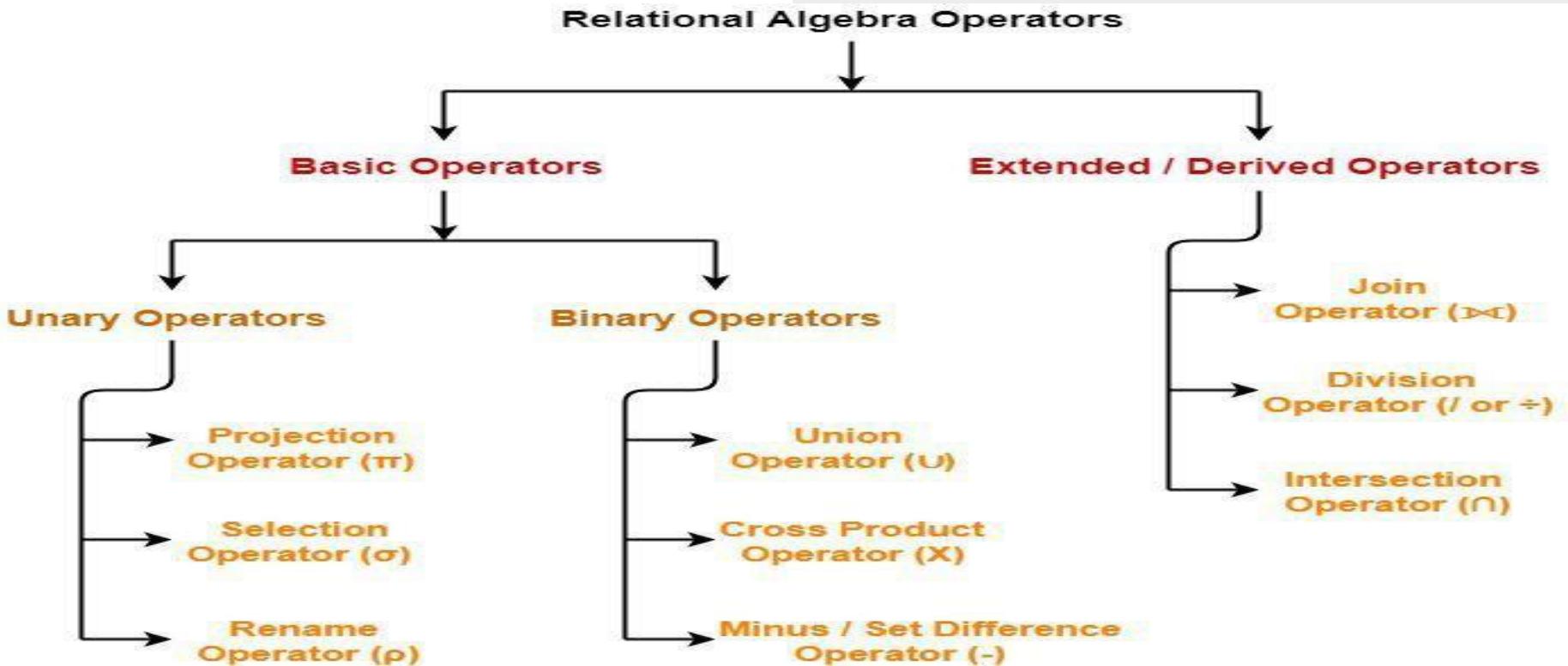


**Relational Query
Languages - Relational
Algebra**

Image Source:

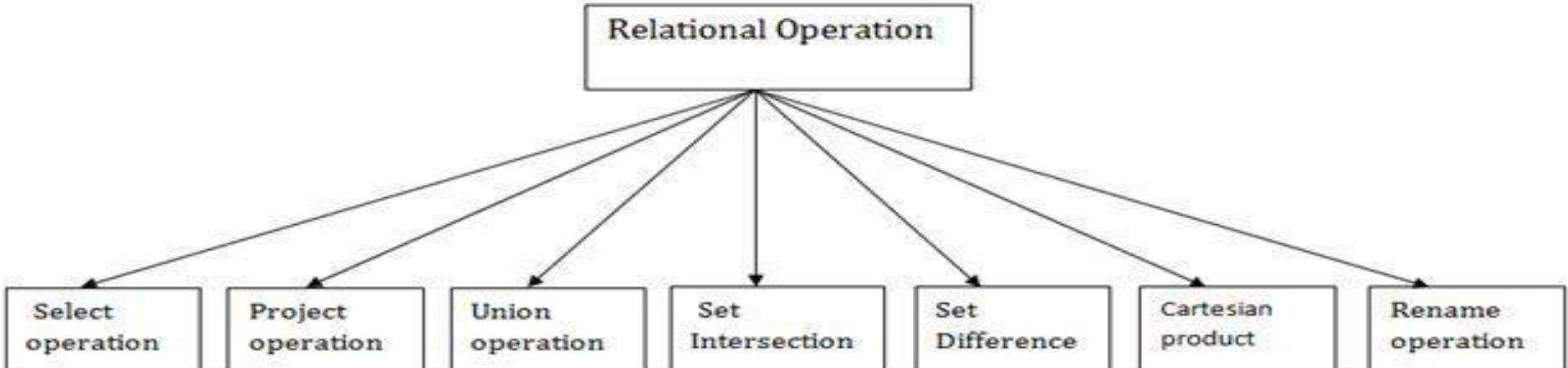
https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcRrt8Z79xBi_fs1wYROUo33pDaGmXFbTyRB

Introduction to Relational Algebra



Introduction to Relational Algebra

Types of Operations



Introduction to Relational Algebra

SELECT (σ)

- The SELECT operation is used for selecting a subset of the tuples according to a given selection condition.
- Σ (σ)Symbol denotes it.
- Notation: $\sigma p(r)$
- σ is used for selection prediction
 r is used for relation
 p is used as a propositional logic

Loan Relation: Loan Table

BRANCH_NAME	LOAN_NO	AMOUNT
Mumbai	L-17	10000
Delhi	L-23	20000
Kolkata	L-15	15000

Input:

σ BRANCH_NAME = "Mumbai" (LOAN)

Introduction to Relational Algebra

SELECT (σ)

- **Example 1**

$$\sigma_{\text{topic} = \text{"Database"}}(\text{Courses})$$

Output - Selects tuples from Courses where topic = 'Database'.

- **Example 2**

$$\sigma_{\text{topic} = \text{"Database"} \text{ and } \text{author} = \text{"Satish"}}(\text{Courses})$$

Output - Selects tuples from Courses where the topic is 'Database' and 'author' is Satish.

- **Example 3**

$$\sigma_{\text{sales} > 50000}(\text{Customers})$$

Output - Selects tuples from Customers where sales is greater than 50000

Table Courses

Topic	Author	Sales
Database	Satish	10000
Programming	Sandip	15000

Introduction to Relational Algebra

Projection(π)

- This operation shows the list of those attributes that we wish to appear in the result. Rest of the attributes are eliminated from the table.
- It is denoted by Π .

Customer Relation: Customers Table

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive
4	Alibaba	Active

Input:

$\Pi_{CustomerName, Status} (Customers)$

Introduction to Relational Algebra

Projection(π)

- Projection operator does not obey commutative property i.e.
 $\pi_{<\text{list2}>}(\pi_{<\text{list1}>}(R)) \neq \pi_{<\text{list1}>}(\pi_{<\text{list2}>}(R))$
 $\pi_{\text{Sales}}(\pi_{\text{Topic}}(\text{Courses})) \neq \pi_{\text{Topic}}(\pi_{\text{Sales}}(\text{Courses}))$
- Following expressions are equivalent because both finally projects columns of list-1
 - $\pi_{<\text{list1}>}(\pi_{<\text{list2}>}(R)) = \pi_{<\text{list1}>}(R)$
 $\pi_{\text{Sales}}(\pi_{\text{Author}}(\text{Courses})) = \pi_{\text{Sales}}(\text{Courses})$

Table Courses

Topic	Author	Sales
Database	Satish	10000
Programming	Sandip	15000

Introduction to Relational Algebra

Union operation (u)

- Suppose there are two tuples R and S. The union operation contains all the tuples that are either in R or S or both in R & S.
- It eliminates the duplicate tuples. It is denoted by U .

DEPOSITOR RELATION

CUSTOMER_NAME	ACCOUNT_NO
Satish Pise	A-101
Quanith Khan	A-121
Anip Sharma	A-321

BORROW RELATION

CUSTOMER_NAME	LOAN_NO
Rakesh Sharma	L-17
Quanith Khan	L-23
Vinod	L-15

Input:

$$\sqcup \text{CUSTOMER_NAME} \text{ (BORROW)} \cup \sqcup \text{CUSTOMER_NAME} \text{ (DEPOSITOR)}$$

Introduction to Relational Algebra

Intersection Operator (\cap)

- Let R and S be two relations.
- Then-
- $R \cap S$ is the set of all tuples belonging to both R and S.
- In $R \cap S$, duplicates are automatically removed.
- Intersection operation is both commutative and associative

DEPOSITOR RELATION

CUSTOMER_NAME	ACCOUNT_NO
Satish Pise	A-101
Quanith Khan	A-121
Anip Sharma	A-321

BORROW RELATION

CUSTOMER_NAME	LOAN_NO
Rakesh Sharma	L-17
Quanith Khan	L-23
Vinod	L-15

Input:

$$\Pi_{\text{CUSTOMER_NAME}} (\text{BORROW}) \cap \Pi_{\text{CUSTOMER_NAME}} (\text{DEPOSITOR})$$

Introduction to Relational Algebra

Difference Operator (-)

- Let R and S be two relations.
- Then-
- $R - S$ is the set of all tuples belonging to R and not to S.
- In $R - S$, duplicates are automatically removed.
- Difference operation is associative but not commutative.

DEPOSITOR RELATION

CUSTOMER_NAME	ACCOUNT_NO
Satish Pise	A-101
Quanith Khan	A-121
Anip Sharma	A-321

BORROW RELATION

CUSTOMER_NAME	LOAN_NO
Rakesh Sharma	L-17
Quanith Khan	L-23
Vinod	L-15

Input:

$$\Pi \text{CUSTOMER_NAME} (\text{BORROW}) - \Pi \text{CUSTOMER_NAME} (\text{DEPOSITOR})$$

Introduction to Relational Algebra

Cartesian product(X)

- The Cartesian product is used to combine each row in one table with each row in the other table. It is also known as a cross product.
- It is denoted by X.

DEPOSITOR RELATION

CUSTOMER_NAME	ACCOUNT_NO
Satish Pise	A-101
Quanith Khan	A-121
Anip Sharma	A-321

BORROW RELATION

CUSTOMER_NAME	LOAN_NO
Rakesh Sharma	L-17
Quanith Khan	L-23
Vinod	L-15

Input:
DEPOSITOR X BORROW

Introduction to Relational Algebra

Rename product(ρ)

- The rename operation is used to rename the output relation. It is denoted by **rho** (ρ).

DEPOSITOR RELATION

CUSTOMER_NAME	ACCOUNT_NO
Satish Pise	A-101
Quanith Khan	A-121
Anip Sharma	A-321

Input:

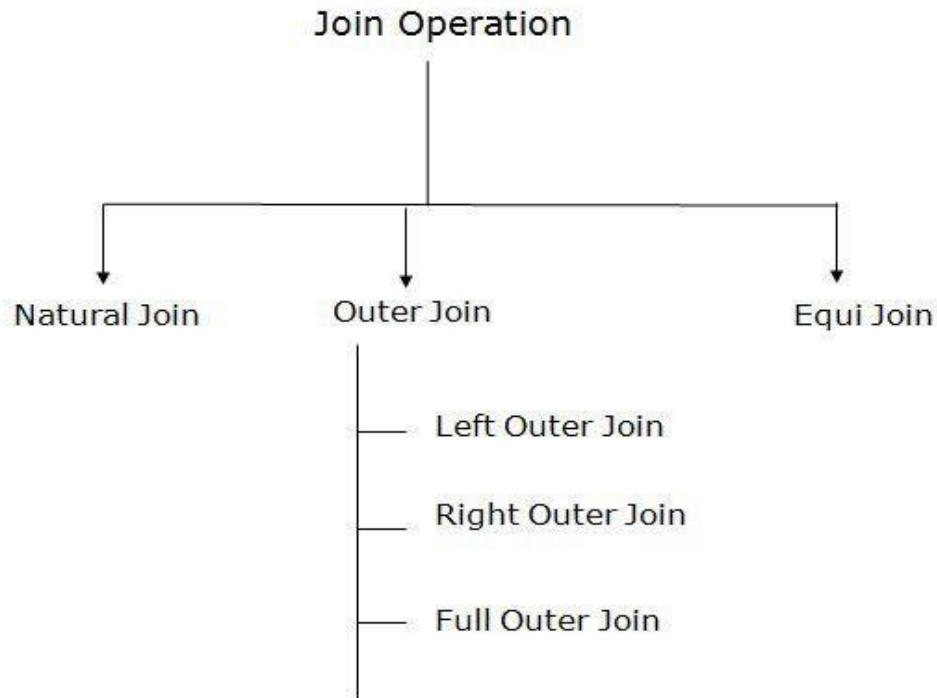
We can use the rename operator to rename STUDENT relation to STUDENT1.
 $\rho(\text{STUDENT1}, \text{STUDENT})$

Rename the Member relation as Library Member
 $\rho \text{ librarymember} (\text{Member})$

Introduction to Relational Algebra

Join Operation

- Join operation is essentially a Cartesian product followed by a selection criterion.
- Join operation denoted by \bowtie .
- JOIN operation also allows joining variously related tuples from different relations.



Introduction to Relational Algebra

Inner Join / Natural Join

- Natural join between two or more relations will result in all the combination of tuples where they have equal values for the common attribute

Member \bowtie Borrow

Member ID	Name	Date of Birth	Book ID	Borrow Date	Return Date
1	Alice	03/03/1995	1	02/03/2020	12/03/2020
3	Charlie	21/10/1997	5	05/03/2020	15/03/2020
3	Charlie	21/10/1997	3	10/03/2020	20/03/2020
4	Mike	16/09/1992	2	13/03/2020	23/03/2020
5	Katie	21/10/1997	4	13/03/2020	13/03/2020

Introduction to Relational Algebra

Theta Join (θ)

- The general case of JOIN operation is called a Theta join. It is denoted by symbol θ
- Example
- $A \bowtie_{\theta} B$ Theta join can use any conditions in the selection criteria.

For example:

$A \bowtie_{A.\text{column 2} > B.\text{column 2}} (B)$

Column1	Column2
1	2

Introduction to Relational Algebra

EQUI Join (=)

- When a theta join uses only equivalence condition, it becomes a equi join.
- For example:
- $A \bowtie_{A.\text{column 2} = B.\text{column 2}} (B)$

For example:

$$A \bowtie_{A.\text{column 2} = B.\text{column 2}} (B)$$

Column1	Column2
1	2

Introduction to Relational Algebra

Left Outer Join (\bowtie)

- In the left outer join, operation allows keeping all tuple in the left relation. However, if there is no matching tuple is found in right relation, then the attributes of right relation in the join result are filled with null values.



Example : $A \bowtie B$

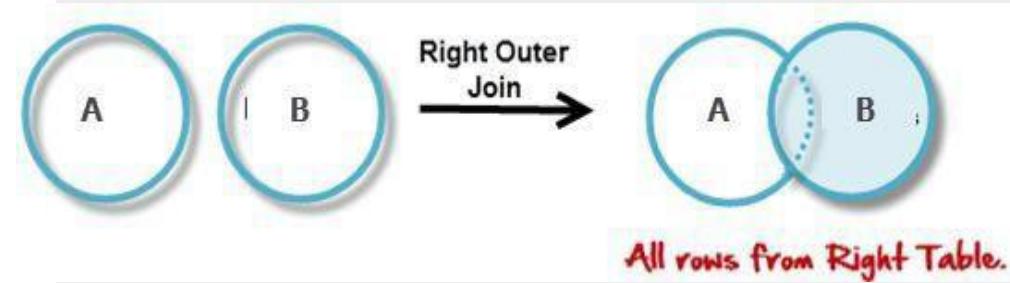
Example: Select students whose ROLL_NO is greater than EMP_NO of employees and details of other students as well

$STUDENT \bowtie_{STUDENT.ROLL_NO > EMPLOYEE.EMP_NO} EMPLOYEE$

Introduction to Relational Algebra

Right Outer Join (\bowtie)

- In the left outer join, operation allows keeping all tuple in the left relation. However, if there is no matching tuple is found in right relation, then the attributes of right relation in the join result are filled with null values.



Example : A \bowtie B

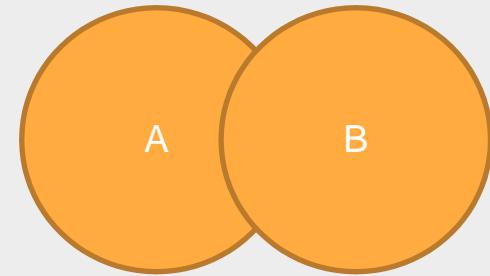
Example: Select students whose ROLL_NO is greater than EMP_NO of employees and details of other Employees as well

STUDENT \bowtie STUDENT.ROLL_NO > EMPLOYEE.EMP_NO
EMPLOYEE

Introduction to Relational Algebra

Full Outer Join (\bowtie)

- In the left outer join, operation allows keeping all tuple in the left relation. However, if there is no matching tuple is found in right relation, then the attributes of right relation in the join result are filled with null values.



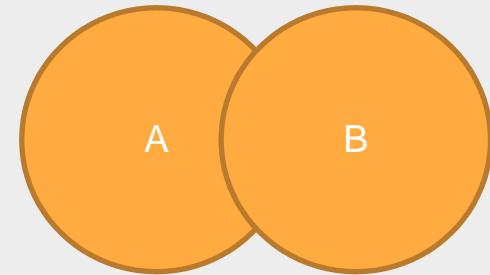
Example : $A \bowtie B$

Example: Select students whose ROLL_NO is greater than EMP_NO of employees and details of other Employees as well and other Students as well

STUDENT \bowtie **EMPLOYEE**
o **STUDENT.ROLL_NO > EMPLOYEE.EMP_N**

Introduction to Relational Algebra

- Division Operator (\div)
- Division operator $A \div B$ can be applied if and only if:
- Attributes of B is proper subset of Attributes of A.
- The relation returned by division operator will have attributes = (All attributes of A – All Attributes of B)
- The relation returned by division operator will return those tuples from relation A which are associated to every B's tuple.

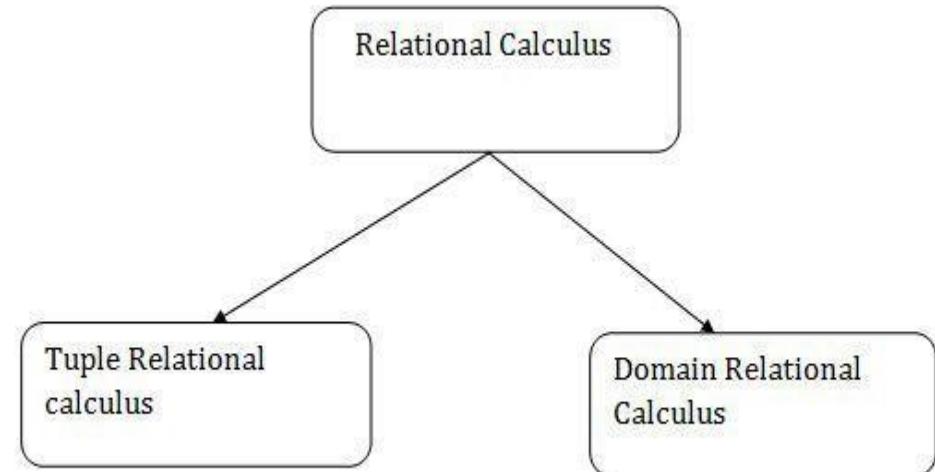


Example : $A \div B$

Consider the relation STUDENT_SPORTS and ALL_SPORTS given in Table 2 and Table 3 above.

To apply division operator as
STUDENT_SPORTS \div ALL_SPORTS

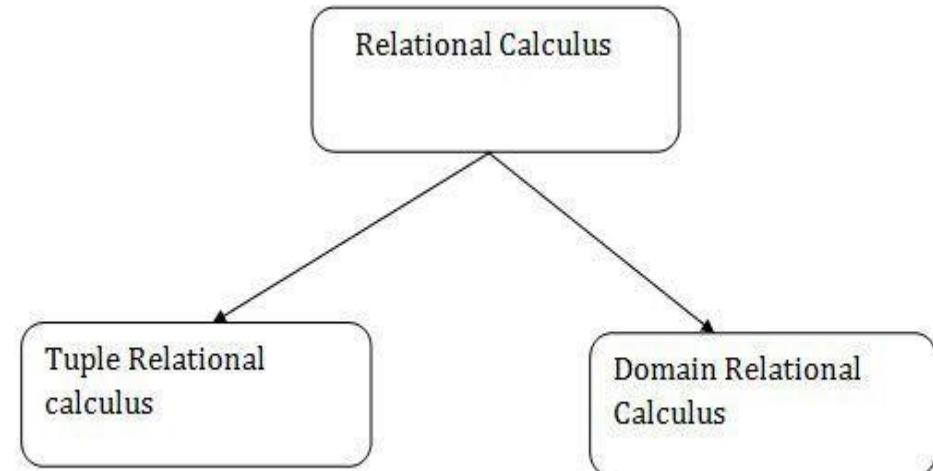
Introduction to Relational Calculus



Introduction to Relational Calculus

Introduction: What is Relational Calculus?

- Relational calculus is a non-procedural query language, and instead of algebra, it uses mathematical predicate calculus.
- Tuple relational calculus which was originally proposed by Codd in the year 1972 and
- Domain relational calculus which was proposed by Lacroix and Pirotte in the year 1977



Introduction to Relational Calculus

Types of Relational Calculus

- In tuple relational calculus, we work on filtering tuples based on the given condition.

Syntax: { T | Condition }

- In domain relational calculus, filtering is done based on the domain of the attributes and not based on the tuple values.

Syntax: { c1, c2, c3, ..., cn | F(c1, c2, c3, ..., cn) }

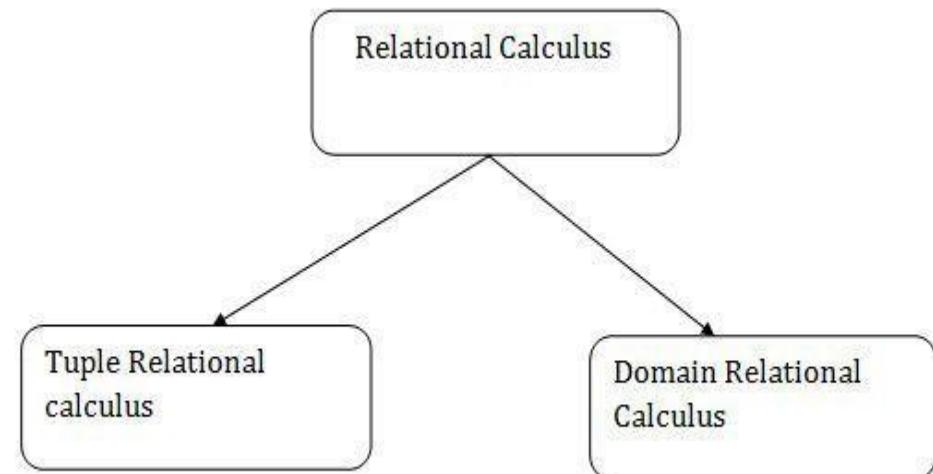
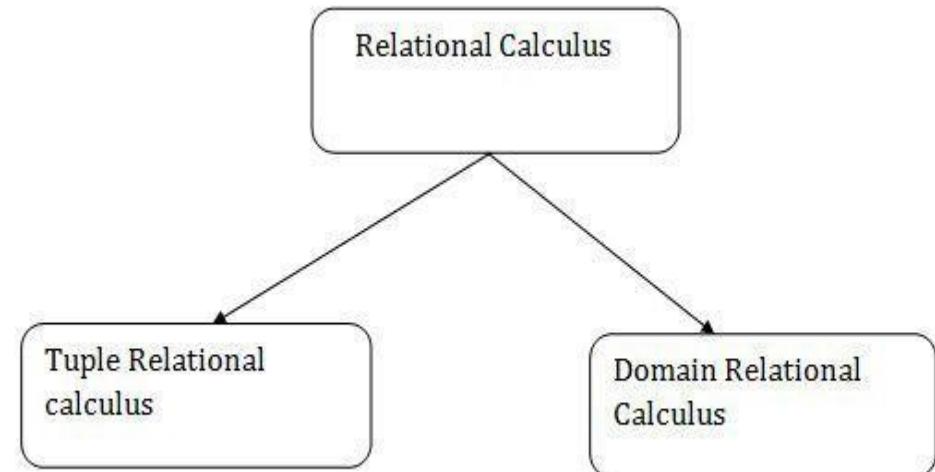


Image Source: <https://static.javatpoint.com/dbms/images/dbms-relational-calculus.png>

Introduction to Relational Calculus

Tuple Relational Calculus

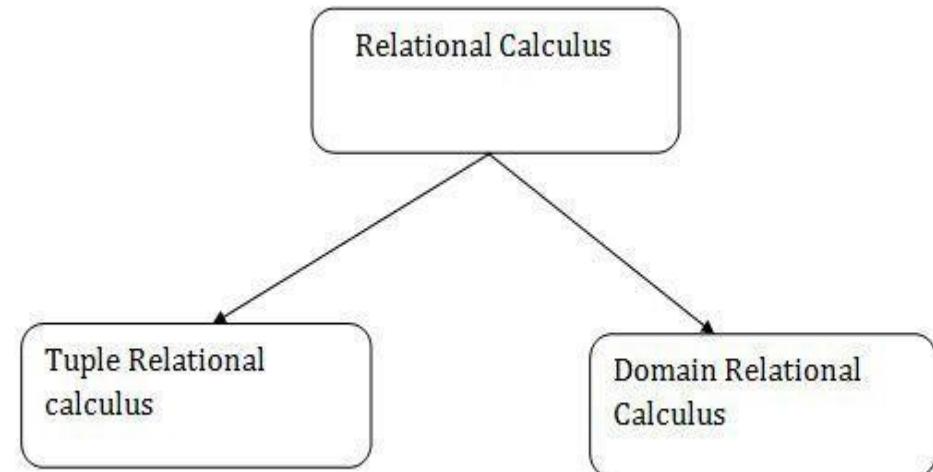
- The tuple relational calculus is specified to select the tuples in a relation.
- In TRC, filtering variable uses the tuples of a relation.
- The result of the relation can have one or more tuples.



Introduction to Relational Calculus

Tuple Relational Calculus

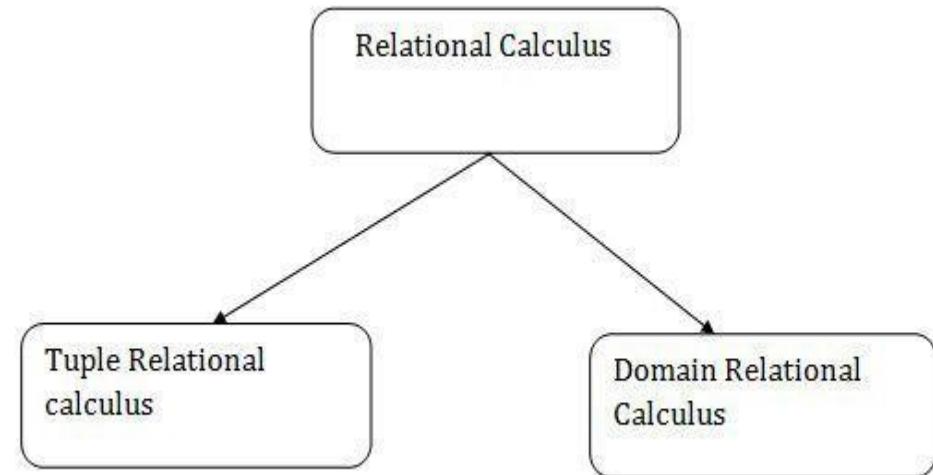
- **Notation:**
• $\{T \mid P(T)\}$ or $\{T \mid \text{Condition}(T)\}$
Where
 - **T** is the resulting tuples
 - **P(T)** is the condition used to fetch T.
 - **For example:**
 - $\{ T.name \mid \text{Author}(T) \text{ AND } T.article = 'database' \}$
 - **OUTPUT:** This query selects the tuples from the AUTHOR relation. It returns a tuple with 'name' from Author who has written an article on 'database'.



Introduction to Relational Calculus

Tuple Relational Calculus

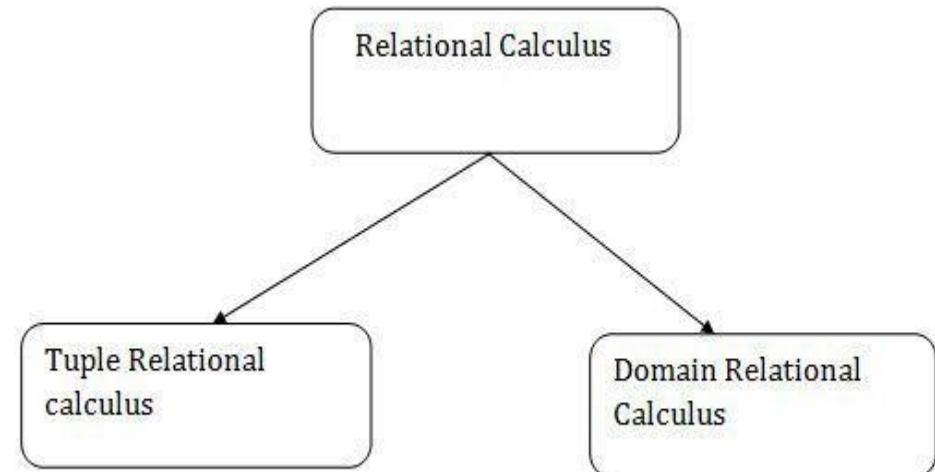
- $\{t \mid P(t)\}$ or $\{t \mid \text{condition}(t)\}$ —
- this is also known as expression of relational calculus
- Where t is the resulting tuples, $P(t)$ is the condition used to fetch t .



Introduction to Relational Calculus

Tuple Relational Calculus

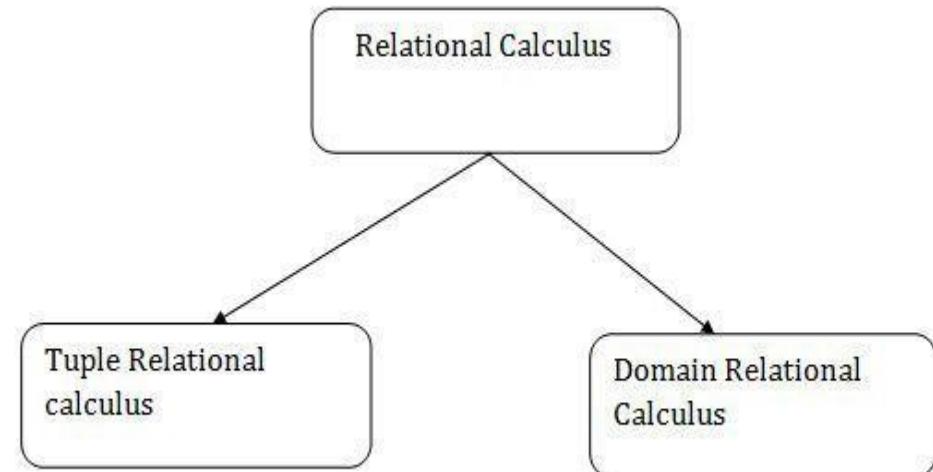
- **{t | EMPLOYEE (t) and t.SALARY>10000} –**
- implies that it selects the tuples from EMPLOYEE relation such that resulting employee tuples will have salary greater than 10000. It is example of selecting a range of values.
- **{t | EMPLOYEE (t) AND t.DEPT_ID = 10} –**
- this select all the tuples of employee name who work for Department 10.



Introduction to Relational Calculus

Tuple Relational Calculus

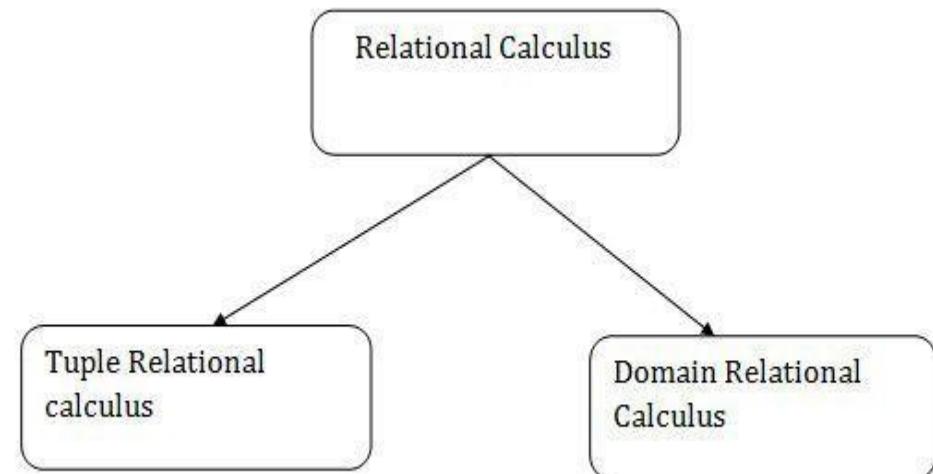
- $\{ T.name \mid \text{FACULTY}(T) \text{ AND } T.DeptId = 0 \text{ CS0} \}$
- can be read as: “Find all tuples T field such that T is a tuple in the FACULTY relation and the value of DeptId field is ‘CS’. Return a tuple with a single field name which is equivalent to the name field of one such T tuple”.



Introduction to Relational Calculus

Tuple Relational Calculus

- $\{R \mid \exists T \in \text{FACULTY} \ (T.DeptId = 0 \text{ CS0 AND } R.name = T.name)\}$
- can be read as: “Find all tuples R such that there exists a tuple T in FACULTY with the DeptId field value ‘CS’, and the value of the name field of R is equivalent to the name field of this tuple T.”



Introduction to Relational Calculus

Tuple Relational Calculus

- Example 1 Find the loan number, branch, amount of loans of greater than or equal to 10000 amount.
- $\{t | t \in \text{loan} \wedge t[\text{amount}] \geq 10000\}$
- Example 2 Find the loan number for each loan of an amount greater or equal to 10000.
- $\{t | \exists s \in \text{loan} (t[\text{loan number}] = s[\text{loan number}] \wedge s[\text{amount}] \geq 10000)\}$

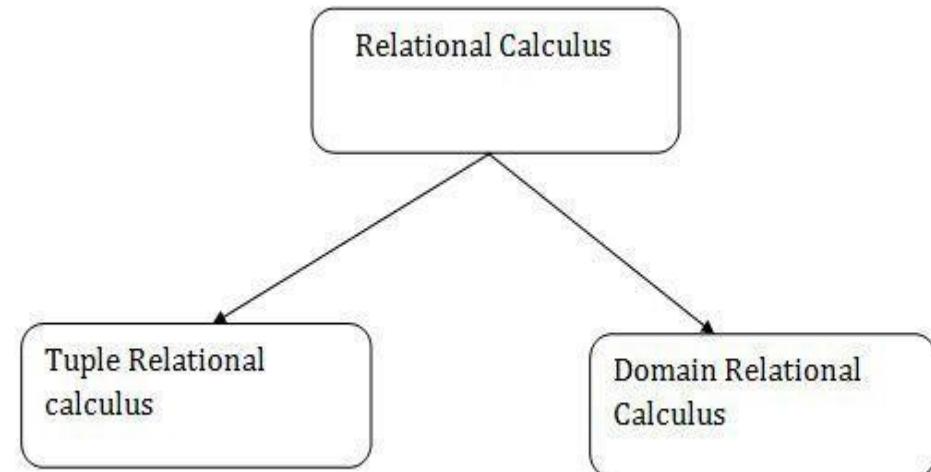
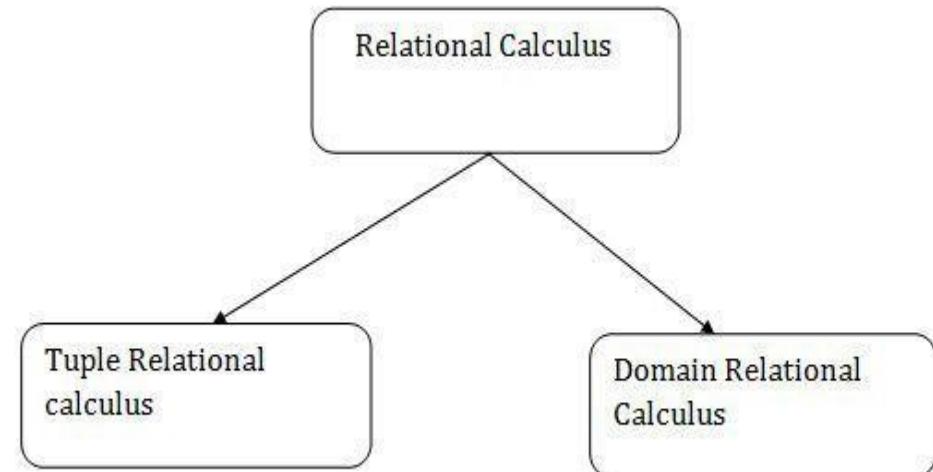


Image Source: <https://static.javatpoint.com/dbms/images/dbms-relational-calculus.png>

Introduction to Relational Calculus

Tuple Relational Calculus

- Example 3 Find the names of all customers who have a loan and an account at the bank.
- $\{t \mid \exists s \in \text{borrower}(t[\text{customer-name}] = s[\text{customer-name}]) \wedge \exists u \in \text{depositor}(t[\text{customer-name}] = u[\text{customer-name}])\}$

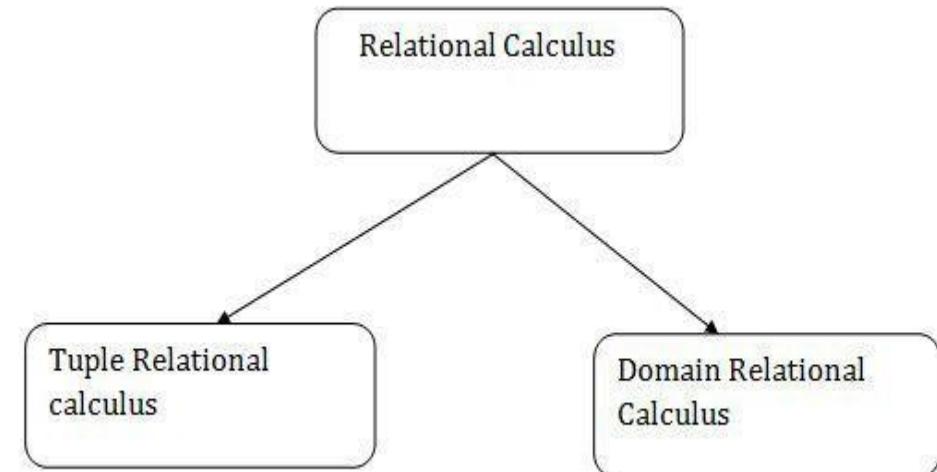


Introduction to Relational Calculus

Domain Relational Calculus

- In domain relational calculus, filtering variable uses the domain of attributes.
- Domain relational calculus uses the same operators as tuple calculus. It uses logical connectives \wedge (and), \vee (or) and \neg (not).
- It uses Existential (\exists) and Universal Quantifiers (\forall) to bind the variable.
- **Notation:**
- $\{ a_1, a_2, a_3, \dots, a_n \mid P (a_1, a_2, a_3, \dots, a_n) \}$
- Where
- **a1, a2** are attributes

P stands for formula built by inner
attributes



Introduction to Relational Calculus

Domain Relational Calculus

- **Example 1**
- Find the loan number, branch, amount of loans of greater than or equal to 100 amount.
- $\{<l, b, a> \mid <l, b, a> \in \text{loan} \wedge (a \geq 100)\}$
- **Example 2**
- Find the loan number for each loan of an amount greater or equal to 150.
- $\{<l> \mid \exists b, a (<l, b, a> \in \text{loan} \wedge (a \geq 150))\}$

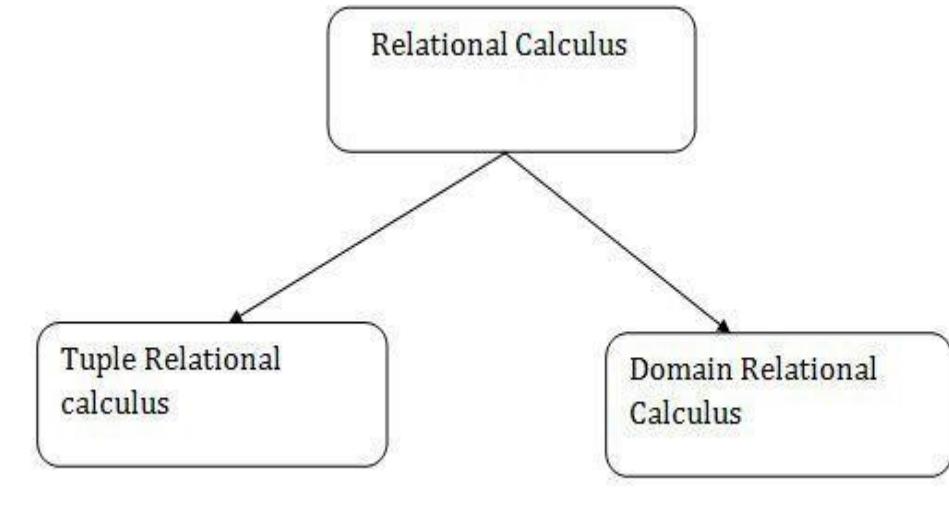
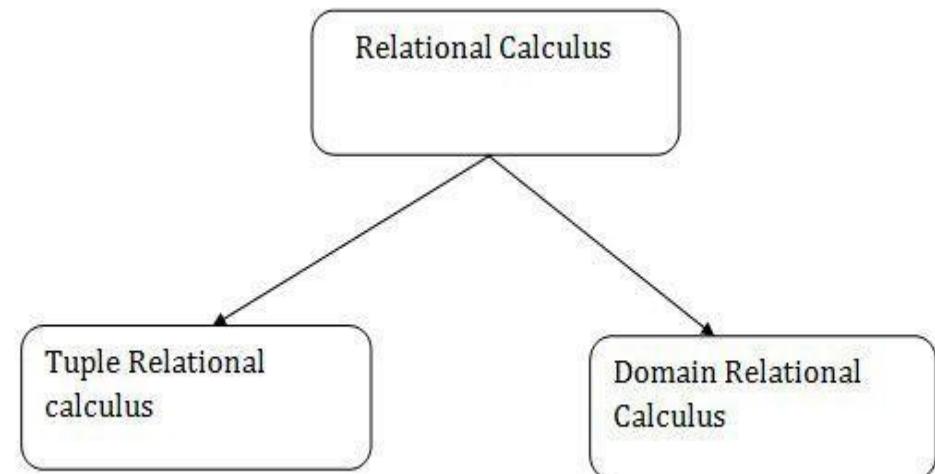


Image Source: <https://static.javatpoint.com/dbms/images/dbms-relational-calculus.png>

Introduction to Relational Calculus

Domain Relational Calculus

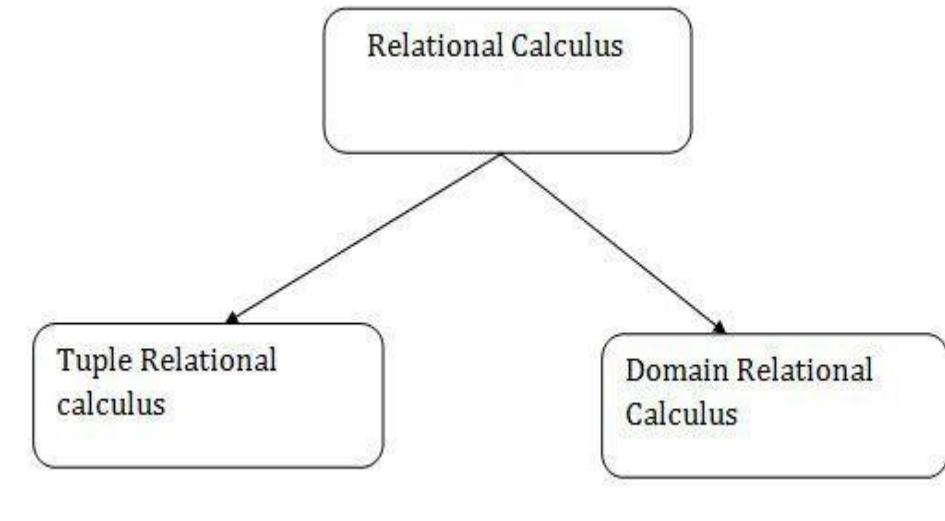
- **Example 3**
- Find the names of all customers having a loan at the “Main” branch and find the loan amount .
- $\{ \langle c, a \rangle \mid \exists l (\langle c, l \rangle \in \text{borrower} \wedge \exists b (\langle l, b, a \rangle \in \text{loan} \wedge (b = \text{"Main"}))) \}$



Introduction to Relational Calculus

Domain Relational Calculus

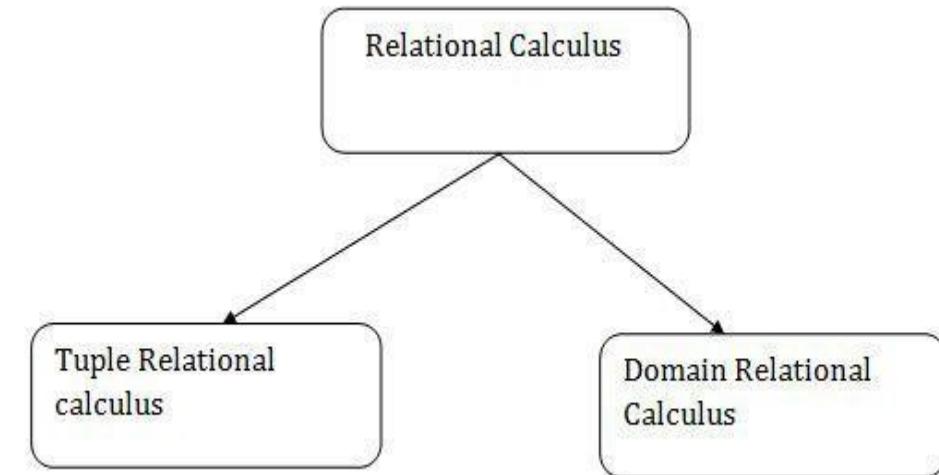
- **Example 4**
- $\{< \text{name}, \text{age} > \mid \text{Student} \wedge \text{age} < 21\}$
- Again, the above query will return the *names and ages of the students in the table Student who not greater than 21 years old*



Introduction to Relational Calculus

Domain Relational Calculus

- **Example 5**
- $\{<\text{Fname, Emp_ID}> \mid \in \text{Employee} \wedge \text{Salary} > 10000\}$
- The result here will be returning the *Fname* and *Emp_ID* values for all the rows in the employee table where salary is greater than 10000.



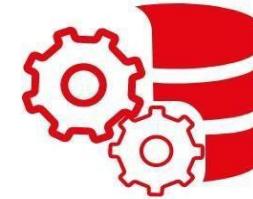
RDBMS Technologies



RDBMS Technologies

Oracle Database Technology

- Data Concurrency and Consistency
- Manageability
- Backup & Recovery
- Business Intelligence
- High Availability
- Very Large Databases
- Content Management etc.



ORACLE®
DB Management

RDBMS Technologies

MySQL Database Technology

- Data Concurrency and Consistency
- Scalability and Limit
- Backup & Recovery
- Connectivity
- Clients and Tools
- Workbench tool



RDBMS Technologies

MongoDB Database Technology

- Indexing
- Replication
- Backup & Recovery
- Load Balancing
- Map Reducing and Aggregation
- Stores files of any size easily without complicating your stack.
- Cloud Support



RDBMS Technologies

MongoDB Database Technology

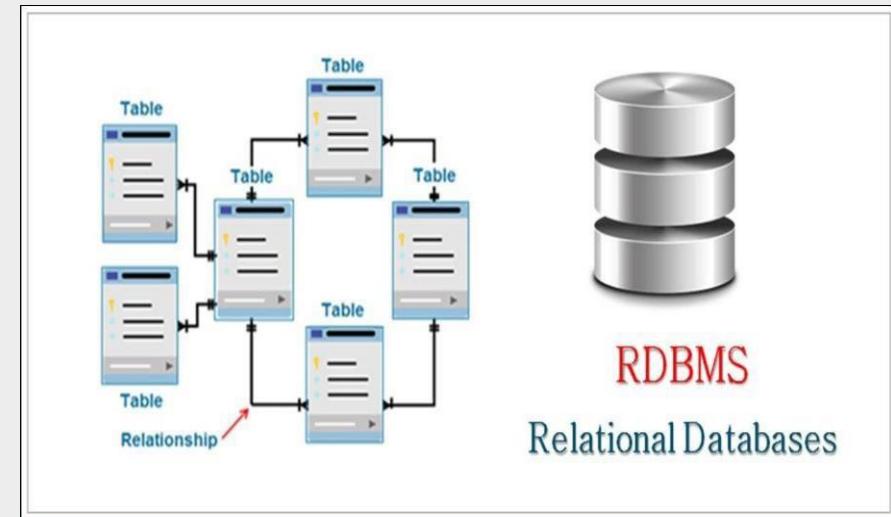
- End to End Mobile BI
- Backup & Recovery
- Load Balancing
- Built-in Analytics
- Mission Critical Availability
- Cloud Support



Image Source:

https://www.openshift.com/hs-fs/hubfs/Imported_Blog_Media/sql-server-2019.png?width=800&height=390

Relational Data Structure



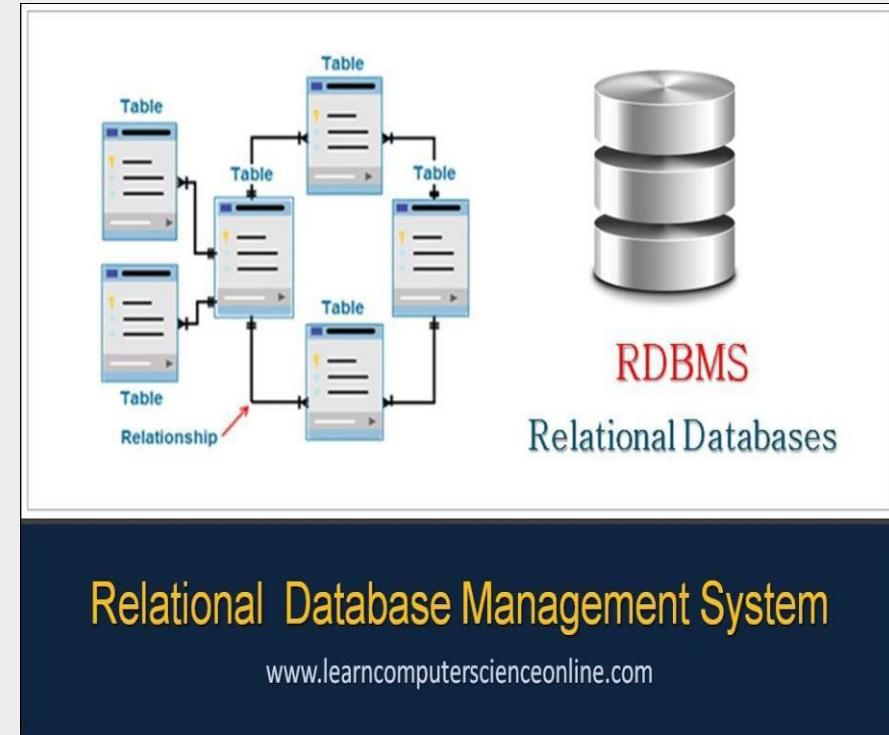
Relational Database Management System

www.learncomputerscienceonline.com

Relational Data Structure

Definition

- The relational model represents the database as a collection of relations.
- A relation is nothing but a table of values.
- Every row in the table represents a collection of related data values.
- These rows in the table denote a real-world entity or relationship.



Relational Data Structure

Relational Model Concepts

- **Attribute:** The properties which define a relation.
- **Tables:** Relations are saved in the table format.
- **Tuple:** Single row of a table
- **Relation Schema:** Represents the name of the relation with its attributes.

Example of a Relation

The diagram illustrates the structure of a relation. At the top, 'Relation Name' points to 'STUDENT'. Below it, 'Attributes' points to the columns: Name, Ssn, Home_phone, Address, Office_phone, Age, and Gpa. Arrows from 'Tuples' point to each row of the table, which contains data for five students: Benjamin Bayer, Chung-cha Kim, Dick Davidson, Rohan Panchal, and Barbara Benson.

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	749-1253	25	3.53
Rohan Panchal	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	NULL	19	3.25

Figure 5.1

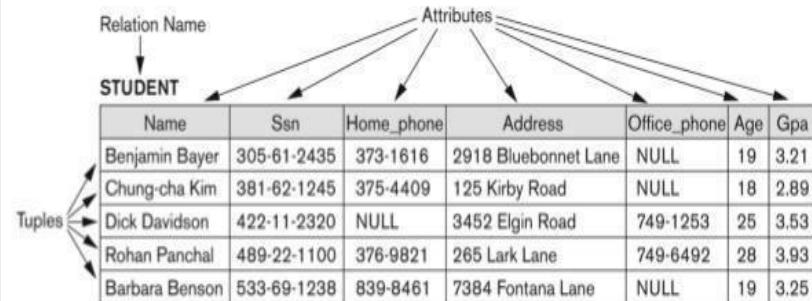
The attributes and tuples of a relation STUDENT.

Relational Data Structure

Relational Model Concepts (Contd...)

- **Degree:** The total number of attributes which in the relation.
- **Cardinality:** Total number of rows present in the Table.
- **Column:** Represents the set of values for a specific attribute.
- **Relation Instance:** Is a finite set of tuples in the RDBMS system

Example of a Relation



The diagram illustrates a relation named STUDENT. At the top, 'Relation Name' points to 'STUDENT'. Below it, 'Attributes' points to the columns: Name, Ssn, Home_phone, Address, Office_phone, Age, and Gpa. At the bottom, 'Tuples' points to the data rows. The table structure is as follows:

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	749-1253	25	3.53
Rohan Panchal	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	NULL	19	3.25

Figure 5.1

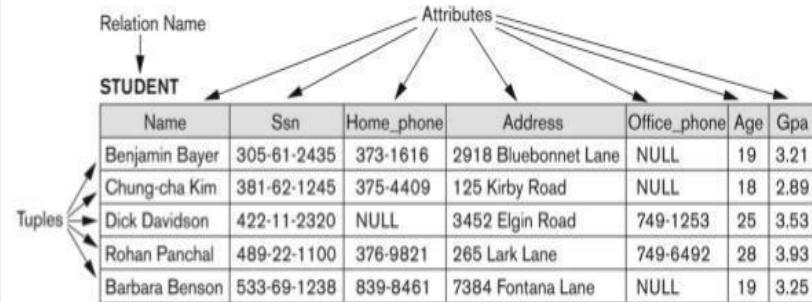
The attributes and tuples of a relation STUDENT.

Relational Data Structure

Relational Model Concepts (Contd...)

- **Relation key:** Every row has one, two or multiple attributes , which is called relation key.
- **Attribute domain:** Every attribute has some pre-defined value and scope which is known as attribute domain.

Example of a Relation



The diagram illustrates a relation named STUDENT. It shows the structure of the relation with arrows pointing from labels to the corresponding parts of the table. The 'Relation Name' arrow points to the label 'STUDENT' above the table. The 'Attributes' arrow points to the column headers: Name, Ssn, Home_phone, Address, Office_phone, Age, and Gpa. The 'Tuples' arrow points to the data rows in the table. The table itself is a grid with 7 columns and 6 rows of data.

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	749-1253	25	3.53
Rohan Panchal	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	NULL	19	3.25

Figure 5.1

The attributes and tuples of a relation STUDENT.

Relational Data Structure

Relational Integrity constraints

- Every relation has some conditions that must hold for it to be a valid relation.
Domain constraints.
- These conditions are called Relational Integrity Constraints.
- There are three main integrity constraints

Relational Integrity Constraints

- Constraints are **conditions** that must hold on **all** valid relation states.
- There are three *main types* of constraints in the relational model:
 - **Key** constraints
 - **Entity integrity** constraints
 - **Referential integrity** constraints
- Another implicit constraint is the **domain** constraint
 - Every value in a tuple must be an atomic value from the *domain of its attribute* (or it could be **null**, if allowed for that attribute)
 - Sub range of values from data type

Relational Data Structure

Relational Integrity constraints

1. Key Constraints

- There must be at least one minimal subset of attributes in the relation.
- which can identify a tuple uniquely.
- This minimal subset of attributes is called key for that relation.
- If there are more than one such minimal subsets, these are called candidate keys.

Key Constraints

■ Superkey of R:

- Is a set of attributes SK of R with the following condition:
 - No two tuples in any valid relation state $r(R)$ will have the same value for SK
 - That is, for any distinct tuples t_1 and t_2 in $r(R)$, $t_1[SK] \neq t_2[SK]$
 - This condition must hold in *any valid state* $r(R)$
 - Every relation has at least one default SK (set of all attributes)

■ Key of R:

- A "minimal" superkey
- That is, a key is a superkey K such that removal of any attribute from K results in a set of attributes that is not a superkey (does not possess the superkey uniqueness property)

Relational Data Structure

Relational Integrity constraints

2. Domain Constraint

- Domain constraints can be violated if an attribute value is not appearing in the corresponding domain
- Or it is not of the appropriate data type.
- Domain constraints specify that within each tuple, and the value of each attribute must be unique.

Domain Constraint

- A domain is defined as the set of all unique values that can be allowed for an attribute.
- For example, a domain of day-of-week is Sunday, Monday, Tuesday ... Saturday.
- Domain Constraint is one of the elementary form of integrity constraint that helps to maintain accuracy and consistency,
- Helps to avoid common errors such as date : 30th February, 2010...
- **CHECK, UNIQUE, NOT NULL, PRIMARY KEY** are the examples of Domain Constraints

Relational Data Structure

Relational Integrity constraints

3. Referential Integrity Constraints

- Works on the concept of Foreign Keys.
- A foreign key is a key attribute of a relation that can be referred in other relation.
- Referential integrity constraints.
- It states that if a relation refers to a key attribute of a different or same relation, then that key element must exist.

Referential Integrity Constraint

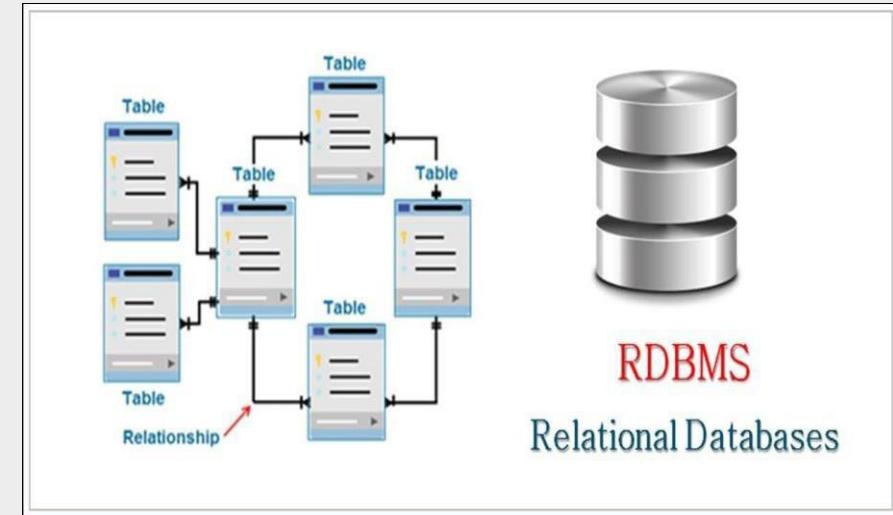
- Has sturdy link with entity integrity,
- The referential integrity relies on the entity integrity,
- **Referential integrity** is a property of data which, when satisfied, requires every value of one attribute (column) of a relation (table) to exist as a value of another attribute in a different (or the same) relation (table).
- Referential Integrity is based on the concept of foreign key,

Relational Data Structure

Relational Integrity constraints

Advantages

- **Simplicity:** Simpler than the hierarchical and network model.
- **Structural Independence:** Is only concerned with data and not with a structure.
- **Easy to use:** is easy as tables consisting of rows and columns is quite natural and simple to understand



Relational Database Management System

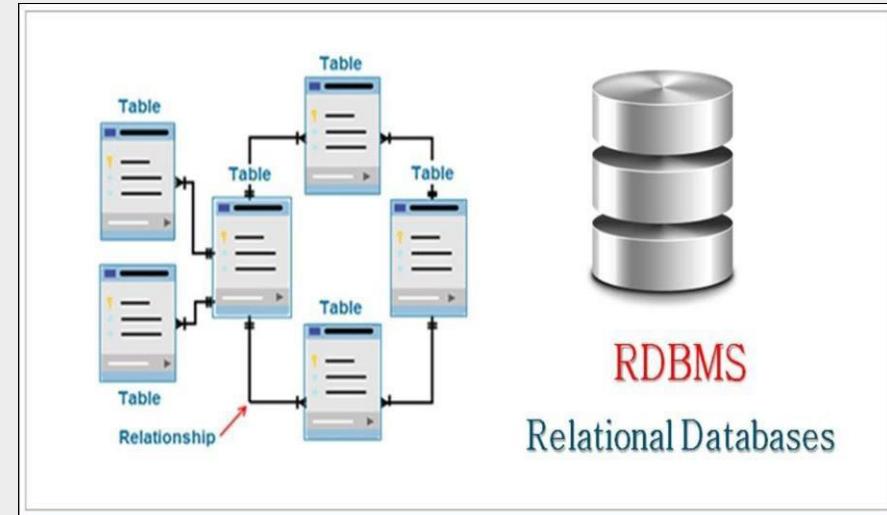
www.learncomputerscienceonline.com

Relational Data Structure

Relational Integrity constraints

Advantages
(Contd...)

- **Query capability:** It makes possible for a high-level query language like SQL to avoid complex database navigation.
- **Data independence:** The structure of a database can be changed without having to change any application.



Relational Database Management System

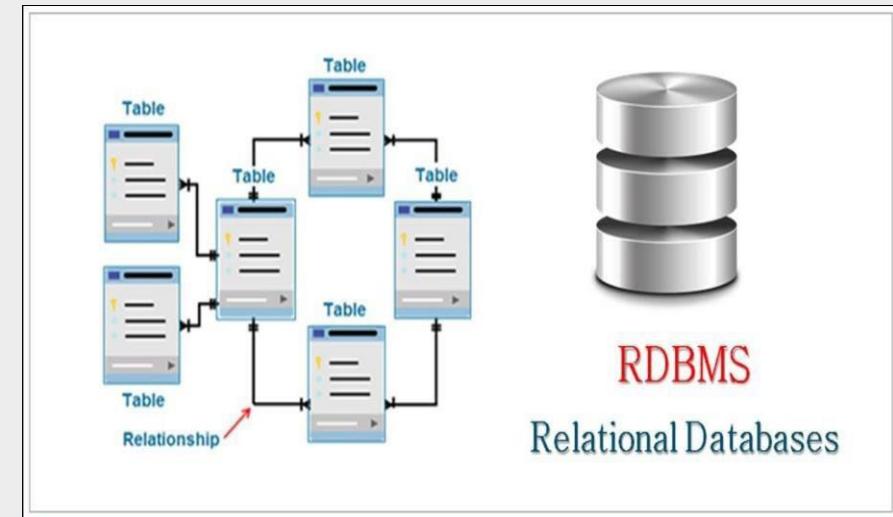
www.learncomputerscienceonline.com

Relational Data Structure

Relational Integrity constraints

Advantages
(Contd...)

- **Scalable:** Regarding a number of records, or rows, and the number of fields, a database should be enlarged to enhance its usability.



Relational Database Management System

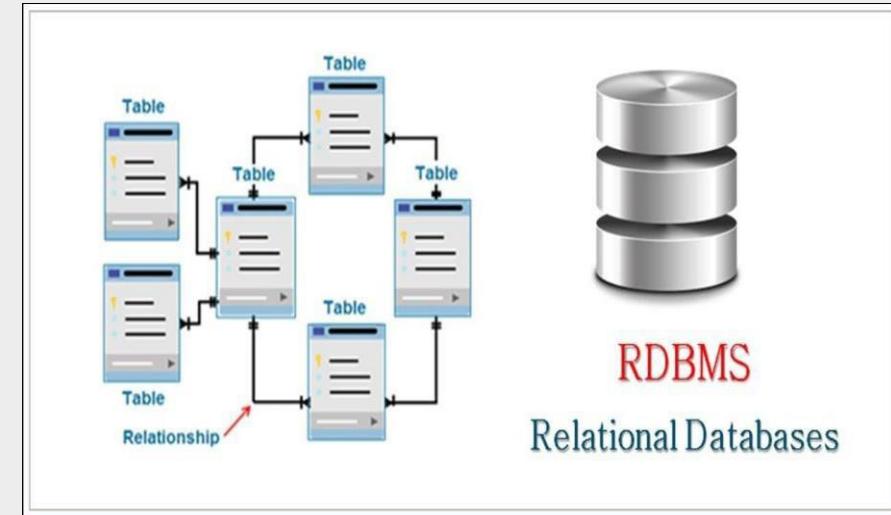
www.learncomputerscienceonline.com

Relational Data Structure

Relational Integrity constraints

Disadvantages

- Few relational databases have limits on field lengths which can't be exceeded.
- Can sometimes become complex as the amount of data grows, and the relations between pieces of data become more complicated.



Relational Database Management System

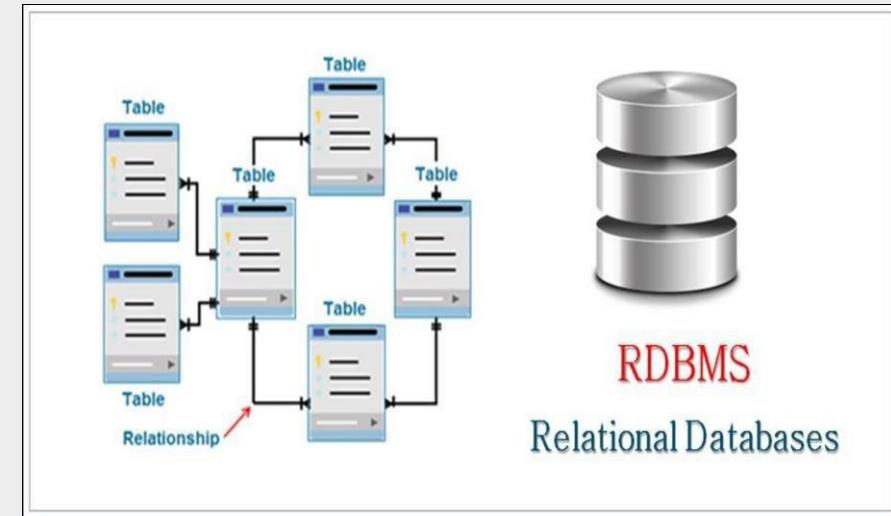
www.learncomputerscienceonline.com

Relational Data Structure

Relational Integrity constraints

Disadvantages
(Contd...)

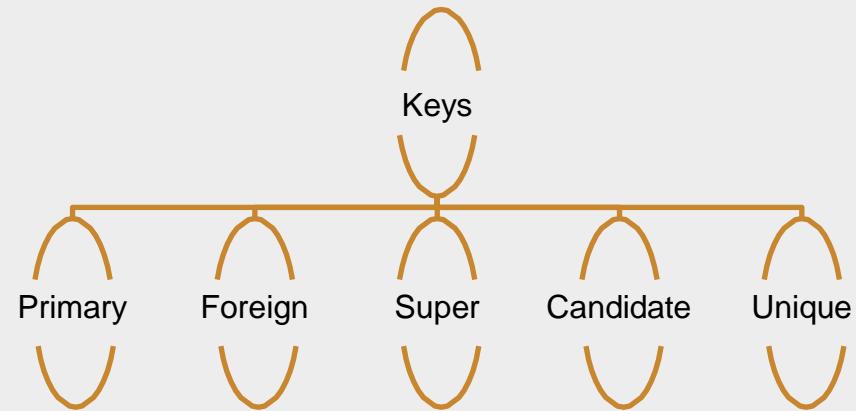
- Complex relational database systems may lead to isolated databases where the information cannot be shared from one system to another.



Relational Database Management System

www.learncomputerscienceonline.com

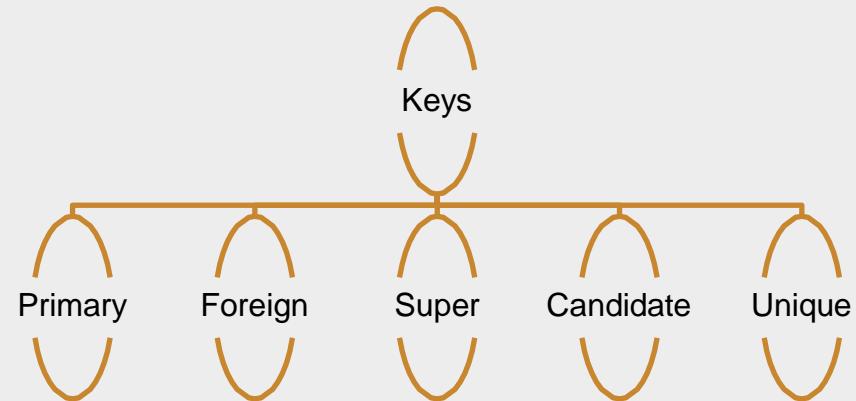
Keys and Relational Data Manipulation



Keys

Keys

- Key is an attribute or set of an attribute which helps to identify a row in a relation
- Allows to find the relation between two tables.
- Keys help you uniquely identify a row in a table
- Used to establish and identify relationships between tables.



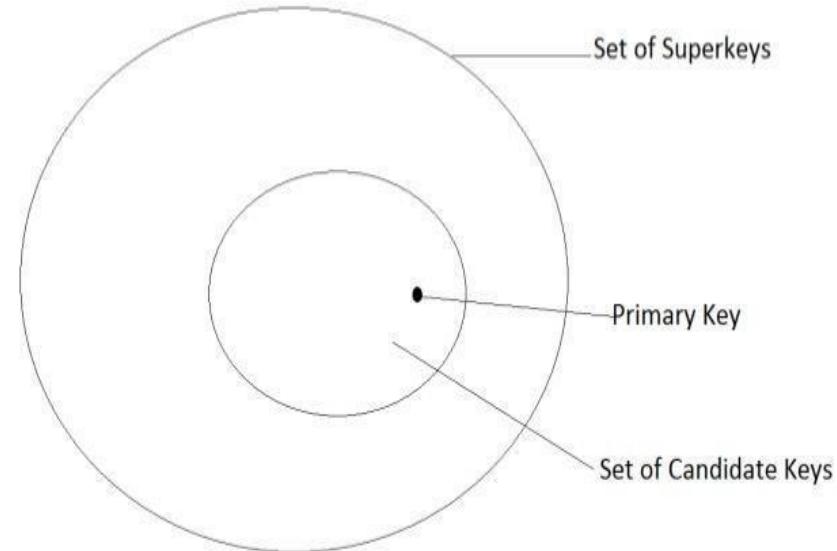
Keys

Keys(Contd...)

Super key:

- A Super key is a group of single or multiple keys which identifies rows in a table.
- A Super key may have additional attributes that are not needed for unique identification.

Relationship



<http://techtud.com>

Follow us on [f](#)

Image

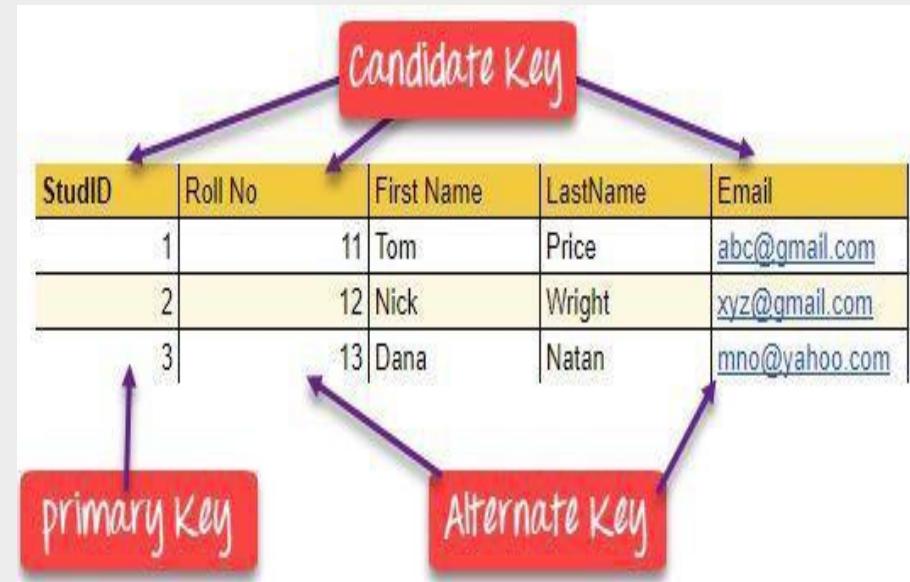
Source: <https://www.slideshare.net/TechtudNetwork/relationship-between-super-key-candidate-key-and-primary>

Keys

Keys(Contd...)

Primary Key:

- Is a column or group of columns in a table that uniquely identify every row in that table.
- Can't be a duplicate.
- A table cannot have more than one primary key.

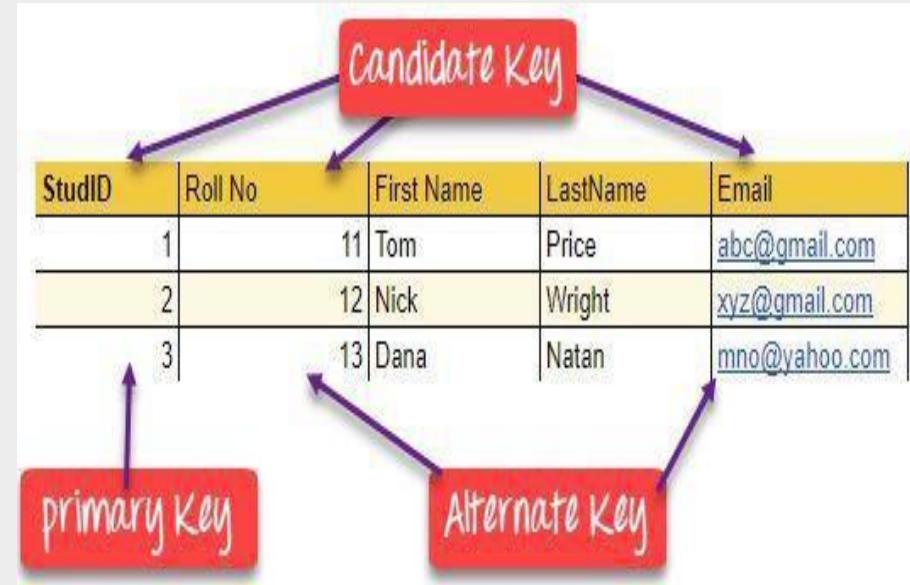


Keys

Keys(Contd...)

Alternate key:

- Is a column or group of columns in a table that uniquely identify every row in that table.
- A table can have multiple choices for a primary key but only one can be set as the primary key.
- All the keys which are not primary key are called an Alternate Key.

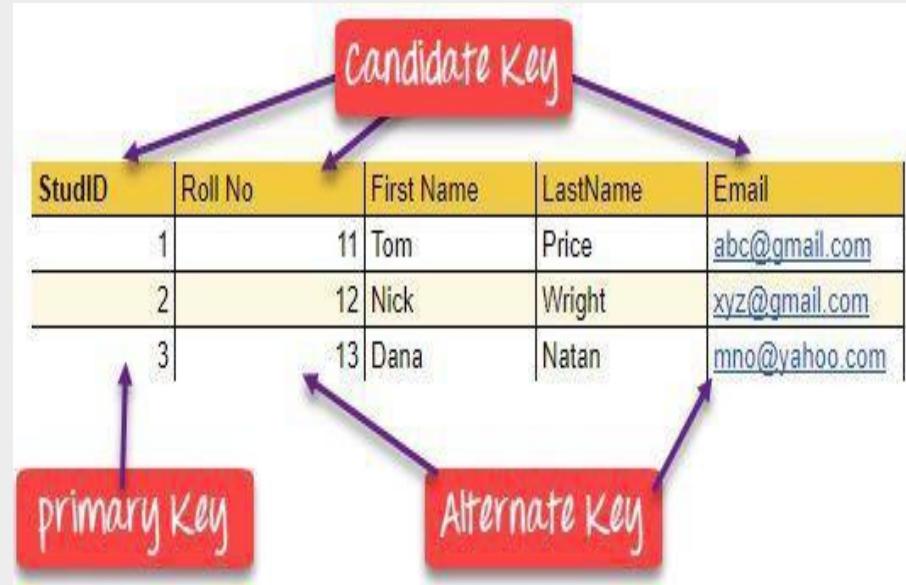


Keys

Keys(Contd...)

Candidate Key:

- Is a set of attributes that uniquely identify tuples in a table.
- Is a super key with no repeated attributes.
- The Primary key should be selected from the candidate keys.
- Every table must have at least a single candidate key.



Keys

Keys(Contd...)

Foreign key:

- Is a column that creates a relationship between two tables.
- The purpose of Foreign keys is to maintain data integrity
- And allow navigation between two different instances of an entity.
- It acts as a cross-reference between two tables



Image Source:

<http://etutorials.org/SQL/Database+design+for+mere+mortals/Part+I+Relational+Database+Design/Chapter+1+Relational+Database+Design+Concepts/1.1+Relational+Database+Design+Concepts.htm>

Keys

Keys(Contd...)

Compound key:

- Has two or more attributes that allow you to uniquely recognize a specific record.
- The purpose of compound key is to uniquely identify each record in the table.

Compound Key

It is possible in some tables that no one column can act as a primary key, and that a combination of columns are required to create the key. This is known as a compound key.

Order ID	Date	Product ID	Product Name	Price (€)	Quantity
X2405	22/11/15	ZAB102455	Modem	49.99	1
X2409	22/11/15	DFT921573	Laser Printer	199.99	1
X2409	22/11/15	OYG446789	Router	124.99	1
X2410	23/11/15	LTY942365	Keyboard	8.99	2
X2411	24/11/15	OYG446789	Router	124.99	1
X2412	24/11/15	DFT921573	Laser Printer	199.99	2
X2412	24/11/15	KUR331094	Mouse	4.99	6

A compound key of **Order ID** and **Product ID** could be used because this is unique for each record.

Keys

Keys(Contd...)

Composite key:

- Is a combination of two or more columns that uniquely identify rows in a table.
- The combination of columns guarantees uniqueness, though individually uniqueness is not guaranteed.
- Hence, they are combined to uniquely identify records in a table.

Student Id	First name of student	Last name of student	Course Id
123456	Jasmine	Shaik	001
123457	Rose	Mary	002
123458	Lily	Holmes	003



Keys

Keys(Contd...)

Compound v/s composite key

- The difference between compound and the composite key is that any part of the compound key can be a foreign key, but the composite key may or maybe not a part of the foreign key.

Student Id	First name of student	Last name of student	Course Id
123456	Jasmine	Shaik	001
123457	Rose	Mary	002
123458	Lily	Holmes	003

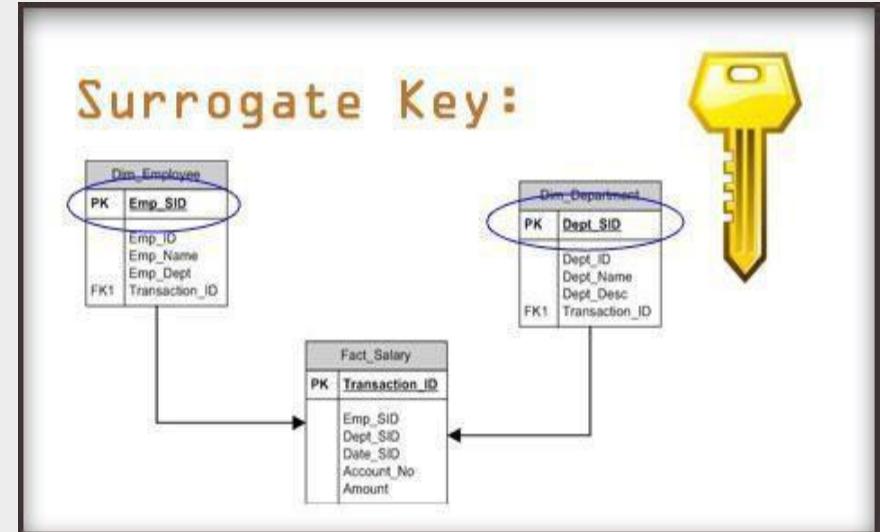


Keys

Keys(Contd...)

Surrogate Key:

- An artificial key which aims to uniquely identify each record is called a surrogate key.
- These kind of key are unique because they are created when you don't have any natural primary key
- They do not lend any meaning to the data in the table.
- Surrogate key is usually an integer.



Relational Data Manipulation

Data Manipulation Language(DML)

- DML stands for Data Manipulation Language. It is used for accessing and manipulating data in a database.
- It handles user requests.

DML
SELECT
INSERT
UPDATE
DELETE
MERGE
CALL
EXPLAIN PLAN
LOCK TABLE

Relational Data Manipulation

DML (Contd...)

Here are some tasks that come under DML:

- Select: It is used to retrieve data from a database.
- Insert: It is used to insert data into a table.

DML
SELECT
INSERT
UPDATE
DELETE
MERGE
CALL
EXPLAIN PLAN
LOCK TABLE

Relational Data Manipulation

DML (Contd...)

Here are some tasks that come under DML:

- Update: It is used to update existing data within a table.
- Delete: It is used to delete all records from a table.
- Merge: It performs UPSERT operation, i.e., insert or update operations.

DML
SELECT
INSERT
UPDATE
DELETE
MERGE
CALL
EXPLAIN PLAN
LOCK TABLE

Relational Data Manipulation

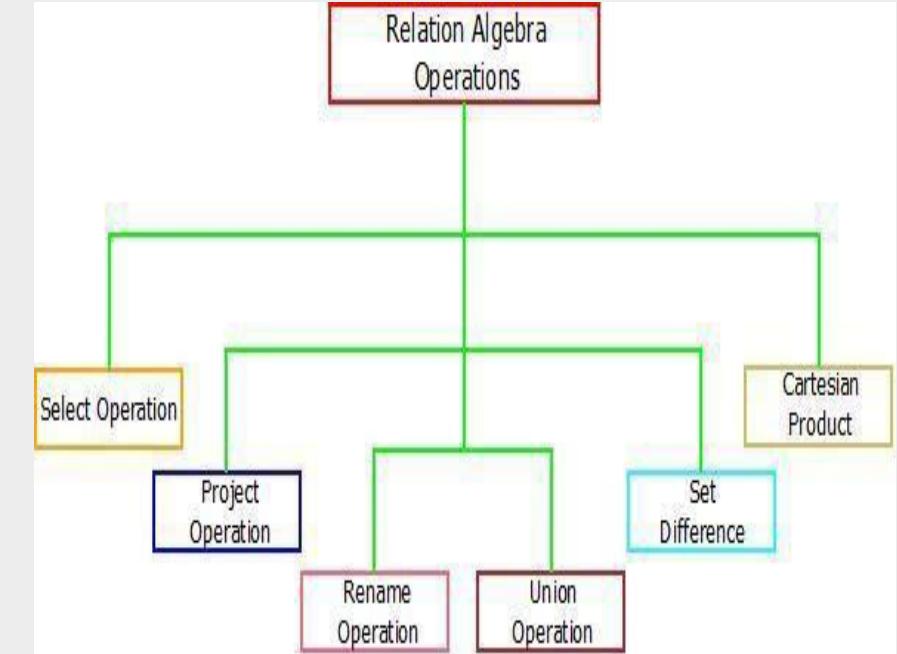
DML (Contd...)

Here are some tasks that come under DML:

- Call: It is used to call a structured query language or a Java subprogram.
- Explain Plan: It has the parameter of explaining data.
- Lock Table: It controls concurrency.

DML
SELECT
INSERT
UPDATE
DELETE
MERGE
CALL
EXPLAIN PLAN
LOCK TABLE

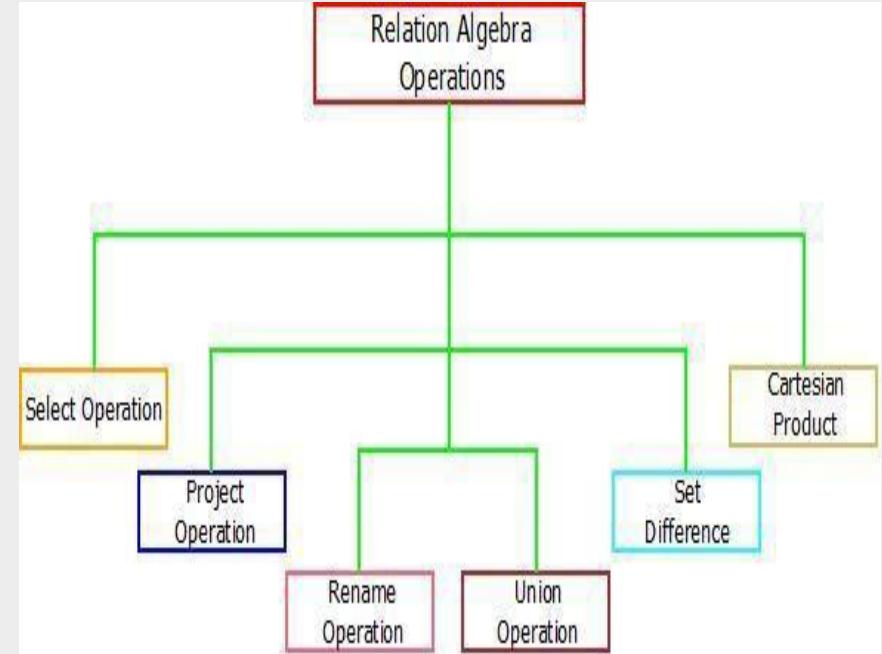
Relational Algebra



Relational Algebra

Definition

- RELATIONAL ALGEBRA is a widely used procedural query language.
- It collects instances of relations as input and gives occurrences of relations as output.
- Relational algebra operations are performed recursively on a relation.
- The output of these operations is a new relation.



Relational Algebraic Operations

Unary Relational Operations

SELECT (symbol: σ)

- The SELECT operation is used for selecting a subset of the tuples according to a given selection condition.
- Sigma(σ)Symbol denotes it
- It is used as an expression to choose tuples which meet the selection condition.

$\sigma_p(r)$

σ is the predicate

r stands for relation which is the name of the table

p is prepositional logic

Example 1

```
 $\sigma$  topic = "Database" (Tutorials)
```

Output - Selects tuples from Tutorials where topic = 'Database'.

Relational Algebraic Operations

Unary Relational Operations (Contd...)

PROJECT (symbol: π)

- The projection eliminates all attributes of the input relation but those mentioned in the projection list.
- The projection method defines a relation that contains a vertical subset of Relation.
- This helps to extract the values of specified attributes to eliminates duplicate values

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive
4	Alibaba	Active

Here, the projection of CustomerName and stat

$\pi_{CustomerName, Status} (Customers)$

CustomerName	Status
Google	Active
Amazon	Active
Apple	Inactive
Alibaba	Active

Relational Algebraic Operations

Unary Relational Operations (Contd...)

RENAME (symbol: ρ)

- The results of relational algebra are also relations but without any name.
- The rename operation allows us to rename the output relation.
- Is denoted with small Greek letter **rho** ρ .
- **Notation** – $\rho_x (E)$
- Where the result of expression **E** is saved with name of **x**.

$\rho_S(R)$ changes:

the *relation name* only to S

$\rho_{(B1, B2, \dots, Bn)}(R)$ changes:

the *column (attribute) names* only to B1, B1,Bn

$\rho_{(B1, B2, \dots, Bn)}(R)$ changes both:

the *relation name* to S, and

the *column (attribute) names* to B1, B1,Bn

Relational Algebraic Operations

From Set Theory

UNION (\cup)

- UNION is symbolized by \cup symbol.
- It includes all tuples that are in tables A or in B.
- It also eliminates duplicate tuples.
- set A UNION set B would be expressed as

$A \cup B$

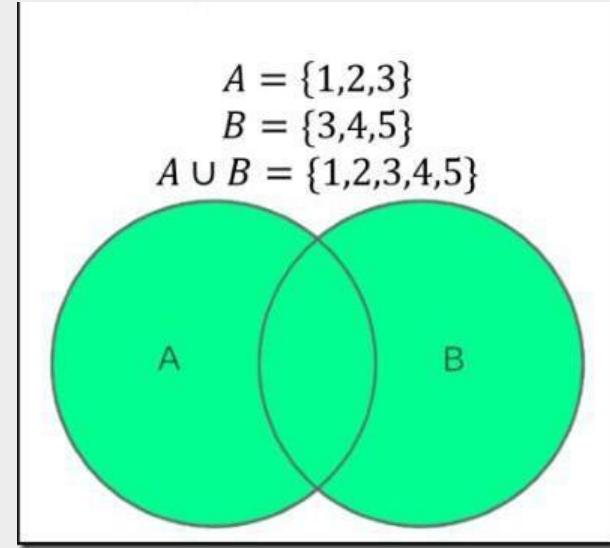


Image Source:

<https://www.codeproject.com/articles/1172312/just-enough-set-theory-when-sets-collide-part-of>

Relational Algebraic Operations

From Set Theory
(Contd...)

INTERSECTION (\cap)

- Intersection is defined by the symbol \cap
- Defines a relation consisting of a set of all tuple that are in both A and B.
- However, A and B must be union-compatible.
- Set A Intersection set B would be expressed as

$$A \cap B$$

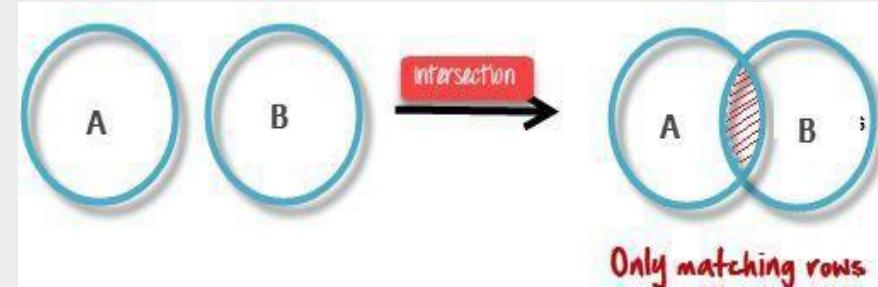


Image Source: <https://www.guru99.com/relational-algebra-dbms.html>

Relational Algebraic Operations

From Set Theory
(Contd...)

Set DIFFERENCE (-)

- The result of $A - B$, is a relation which includes all tuples that are in A but not in B.
- The attribute name of A has to match with the attribute name in B.
- The two-operand relations A and B should be either compatible or Union compatible.

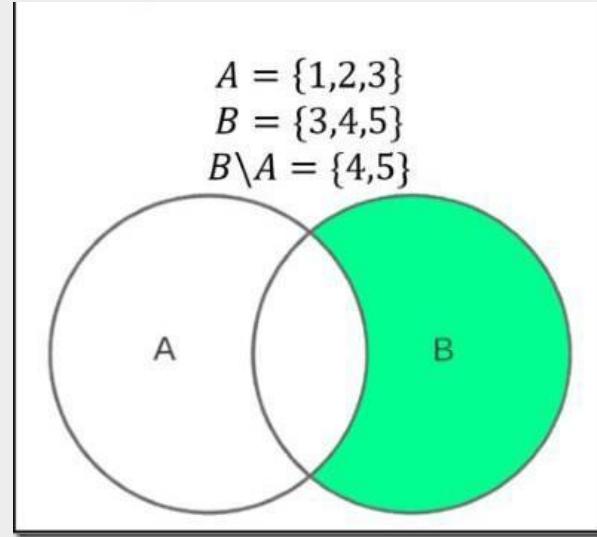


Image Source: <https://www.guru99.com/relational-algebra-dbms.html>

Relational Algebraic Operations

From Set Theory
(Contd...)

CARTESIAN PRODUCT (x)

- Is helpful to merge columns from two relations.
- Is never a meaningful operation when it performs alone.
- It becomes meaningful when it is followed by other operations.
- $\sigma_{\text{column 2} = '1'} (A \times B)$

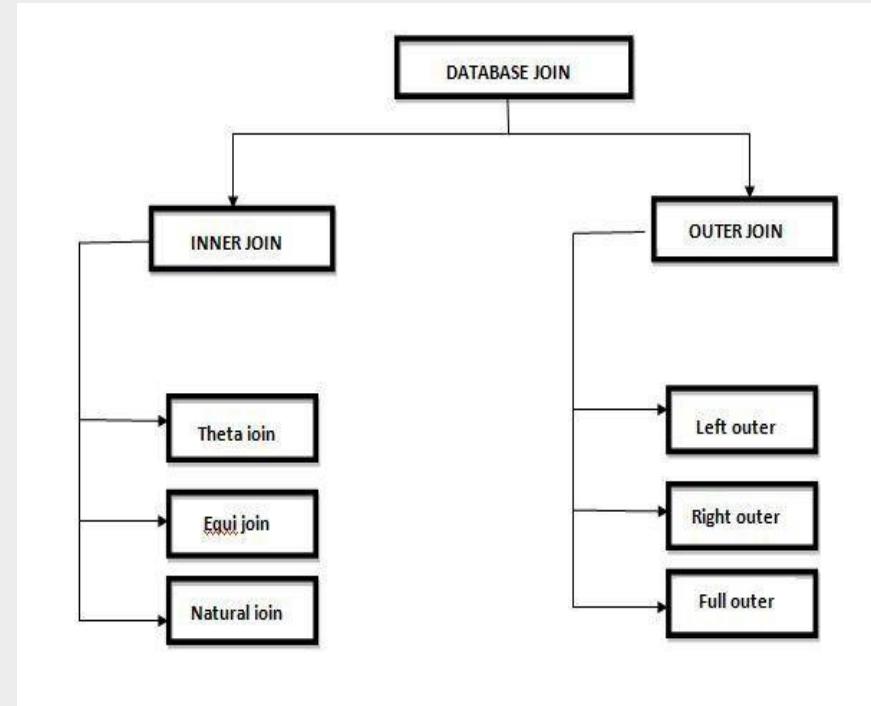


Relational Algebraic Operations

Binary Relational Operations

JOIN(\bowtie)

- Join operation is essentially a cartesian product followed by a selection criterion.
- Join operation denoted by \bowtie .
- JOIN operation also allows joining variously related tuples from different relations.



Relational Algebraic Operations

Binary Relational Operations

JOIN (Contd...)

Inner Joins

- In an inner join, only those tuples that satisfy the matching criteria are included.
- The rest are excluded.

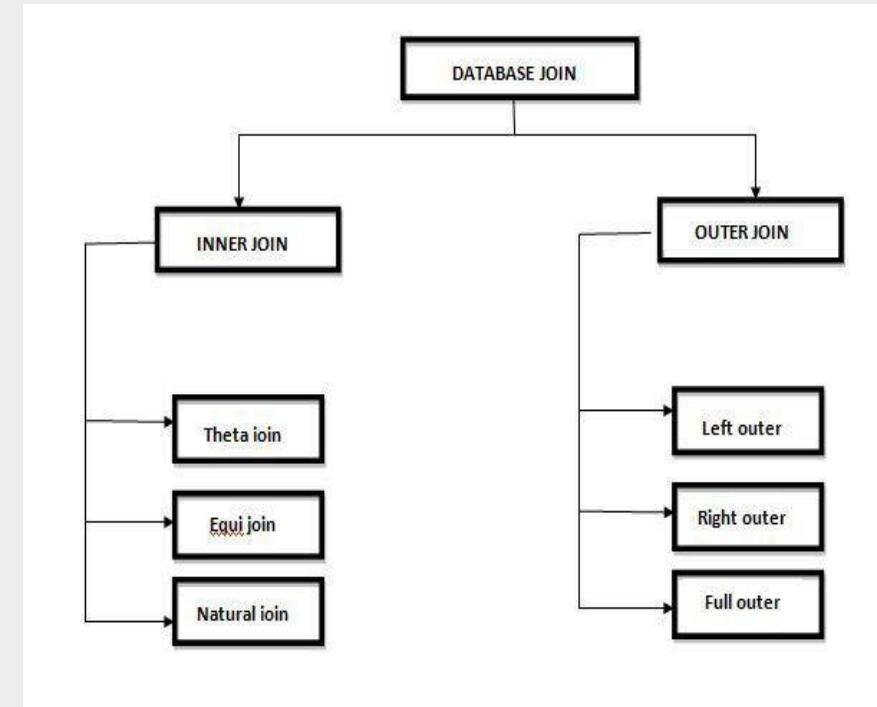


Image Source: <https://www.guru99.com/relational-algebra-dbms.html>

Relational Algebraic Operations

Binary Relational Operations

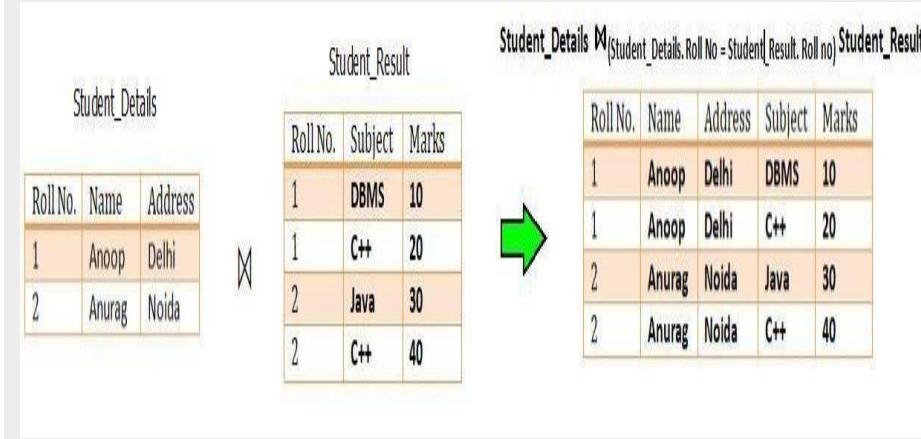
JOIN (Contd...)

Inner Joins

1. Theta join

- The general case of JOIN operation is called a Theta join.
- It is denoted by symbol θ
- Example

$$A \bowtie_{\theta} B$$



Relational Algebraic Operations

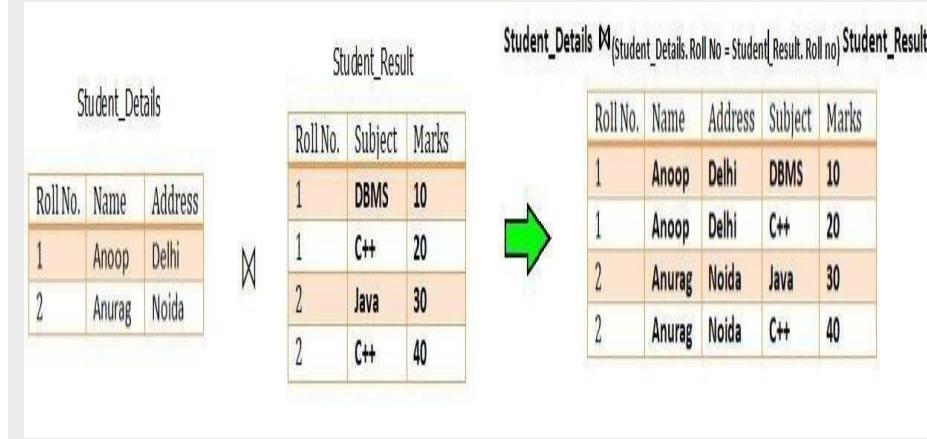
Binary Relational Operations

JOIN(Contd...)

Inner Joins

2. EQUI join

- When a theta join uses only equivalence condition, it becomes a equi join.
- When Theta join uses only equality comparison operator, it is said to be equijoin.
- EQUI join is the most difficult operations to implement efficiently in an RDBMS.



Relational Algebraic Operations

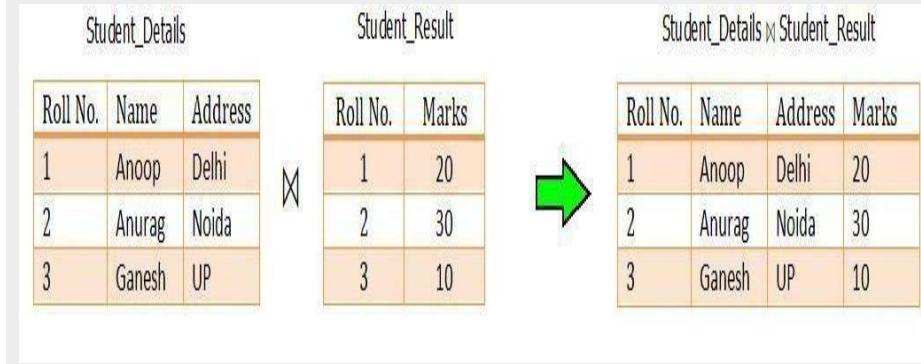
Binary Relational Operations

JOIN (Contd...)

Inner Joins

3. Natural Join

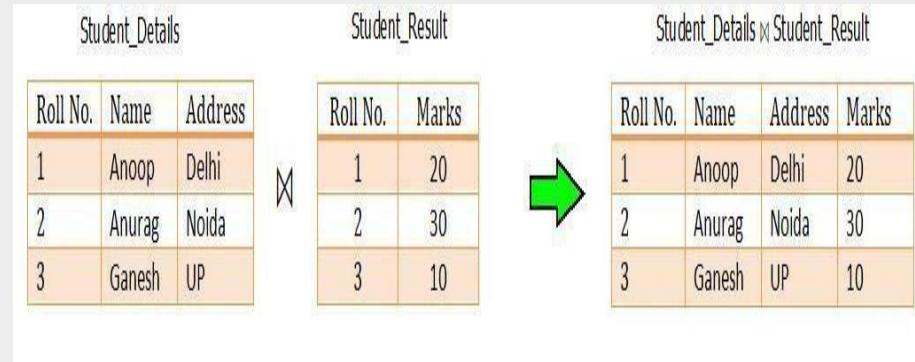
- Natural join does not utilize any of the comparison operators.
- Attributes should have the same name and domain.
- There should be at least one common attribute between two relations.



Relational Algebraic Operations

Binary Relational Operations

JOIN (Contd...)



Outer join

- In an outer join, along with tuples that satisfy the matching criteria, we also include some or all tuples that do not match the criteria.

Relational Algebraic Operations

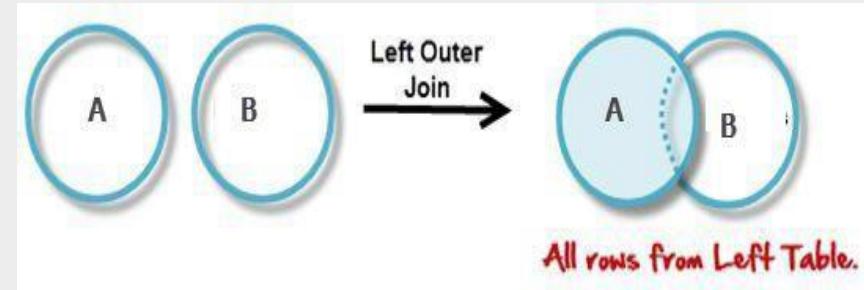
Binary Relational Operations

JOIN (Contd...)

Outer join

1. Left Outer Join

- In the left outer join, operation allows keeping all tuple in the left relation.
- However, if there is no matching tuple is found in right relation, then the attributes of right relation in the join result are filled with null values.



A \bowtie B

Image Source: <https://www.guru99.com/relational-algebra-dbms.html>

Relational Algebraic Operations

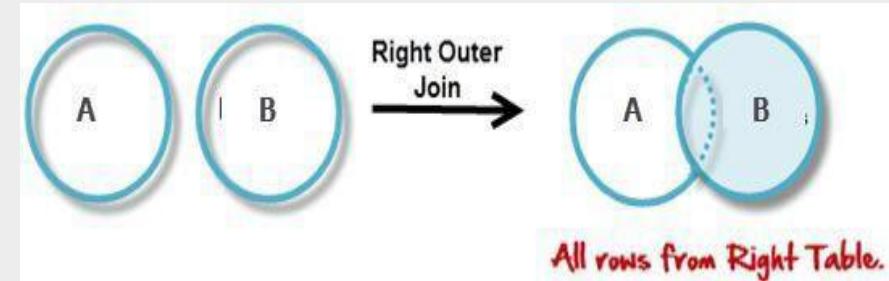
Binary Relational Operations

JOIN (Contd...)

Outer join

2. Right Outer Join

- In the right outer join, operation allows keeping all tuple in the right relation.
- However, if there is no matching tuple is found in the left relation, then the attributes of the left relation in the join result are filled with null values.



A \bowtie B

Image Source: <https://www.guru99.com/relational-algebra-dbms.html>

Relational Algebraic Operations

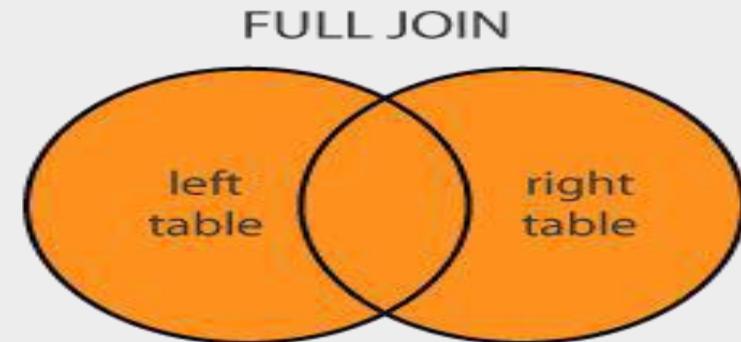
Binary Relational Operations

JOIN (Contd...)

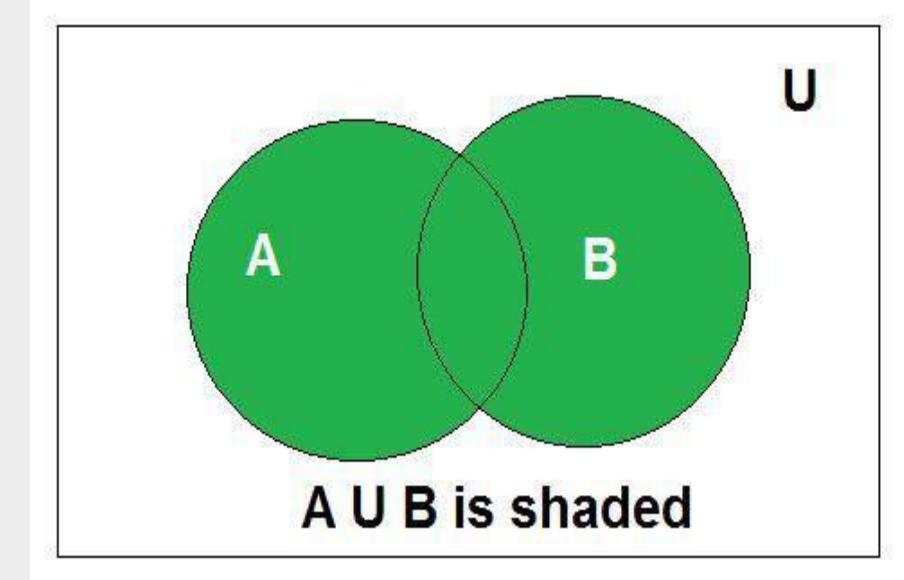
Outer join

3. Full Outer Join

- In a full outer join, all tuples from both relations are included in the result, irrespective of the matching condition.
- However, if there is no matching tuple is found in the left relation, then the attributes of the left relation in the join result are filled with null values.



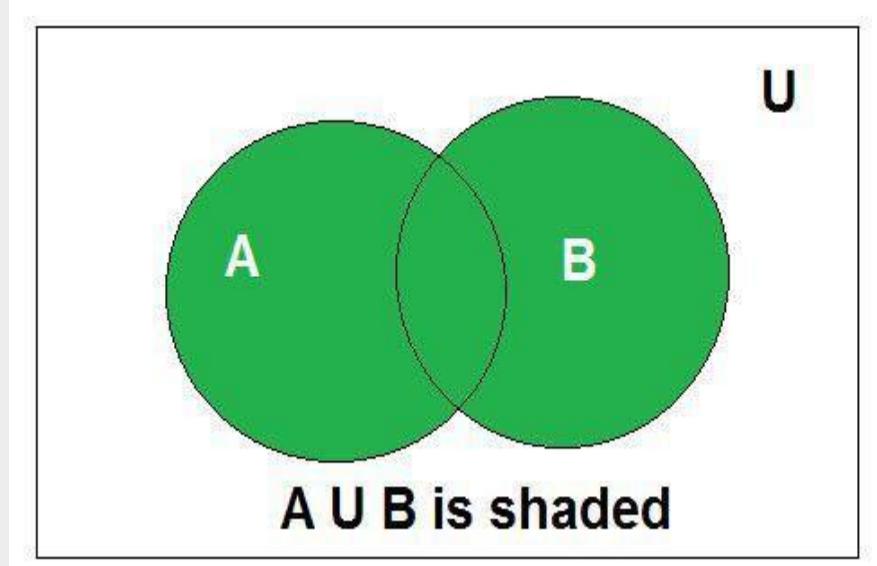
Set Operations



Set Operations

Union

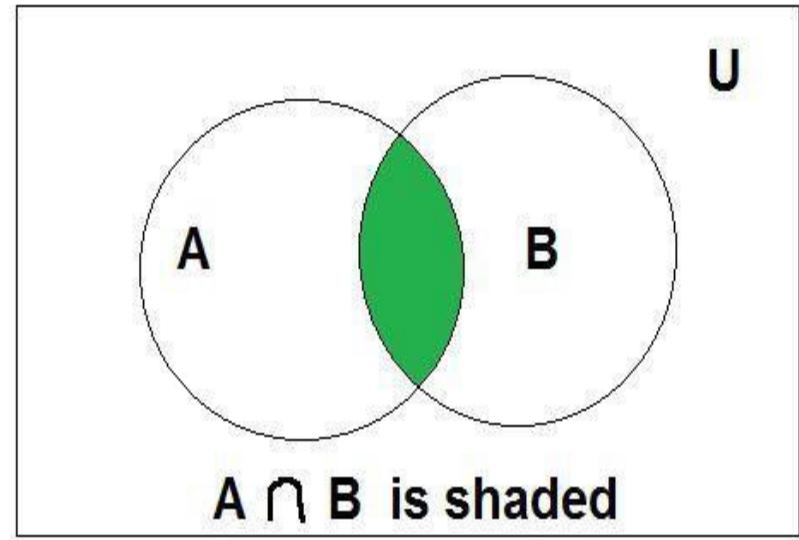
- Union of the sets A and B, denoted by $A \cup B$, is the set of distinct element belongs to set A or set B, or both.
- Example – If $A = \{ 10, 11, 12, 13 \}$ and $B = \{ 13, 14, 15 \}$, then $A \cup B = \{ 10, 11, 12, 13, 14, 15 \}$. (The common element occurs only once)



Set Operations

Intersection

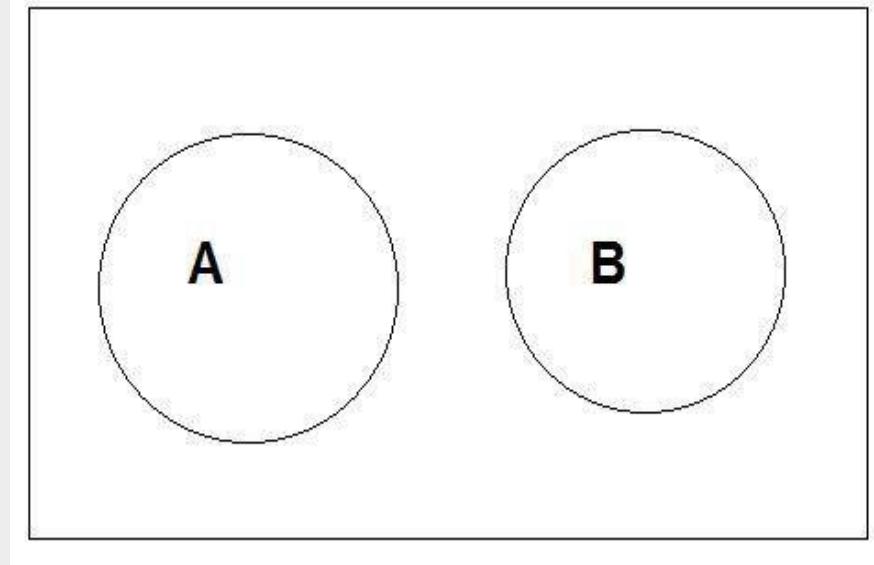
- The intersection of the sets A and B, denoted by $A \cap B$, is the set of elements belongs to both A and B i.e. set of the common element in A and B.
- Example – If $A = \{ 11, 12, 13 \}$ and $B = \{ 13, 14, 15 \}$, then $A \cap B = \{ 13 \}$



Set Operations

Disjoint

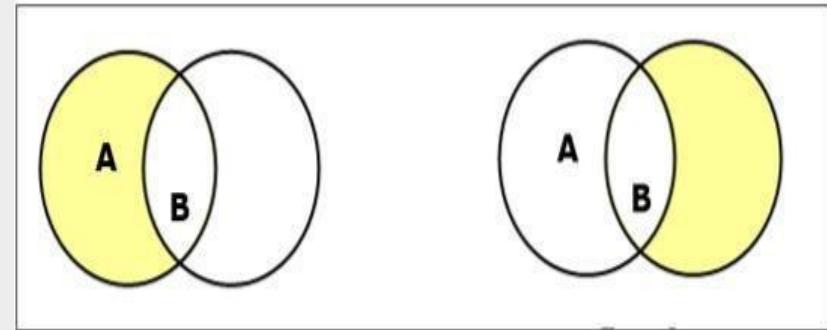
- Two sets are said to be disjoint if their intersection is the empty set .i.e sets have no common elements.
- Example: Let $A = \{1, 3, 5, 7, 9\}$ and $B = \{2, 4, 6, 8\}$.
A and B are disjoint set both of them have no common elements.



Set Operations

Set Difference/ Relative Complement

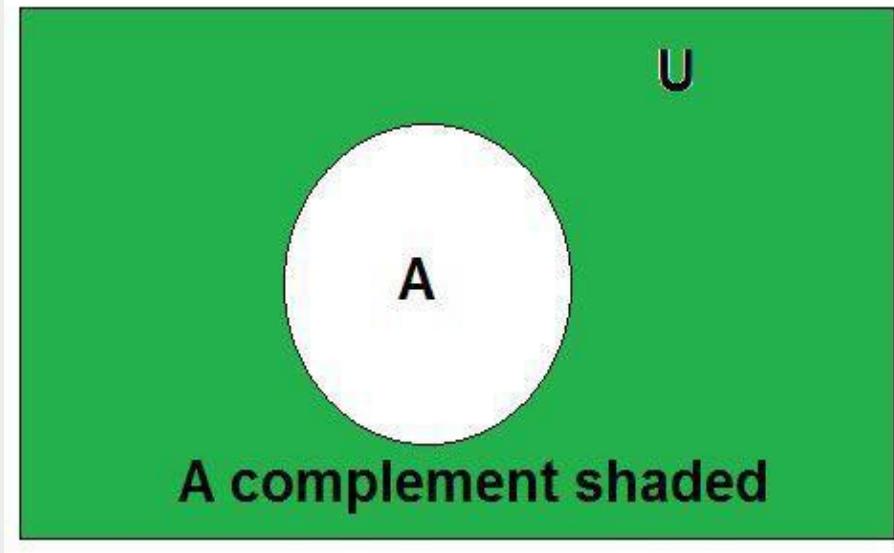
- Difference between sets is denoted by ' $A - B$ ', is the set containing elements of set A but not in B. i.e all elements of A except the element of B.
- Example – If $A = \{ 10, 11, 12, 13 \}$ and $B = \{ 13, 14, 15 \}$, then $(A - B) = \{ 10, 11, 12 \}$ and $(B - A) = \{ 14, 15 \}$. Here, we can see $(A - B) \neq (B - A)$



Set Operations

Complement of a Set

- The complement of a set A (denoted by A') is the set of elements which are not in set A .
- More specifically, $A' = (U - A)$ where U is a universal set that contains all objects.



Fundamental Operations

 $\sigma_p(r)$

σ is the predicate

r stands for relation which is the name of the table

p is prepositional logic

Example 1

```
 $\sigma_{topic = "Database"} (Tutorials)$ 
```

Output - Selects tuples from Tutorials where topic = 'Database'.

Fundamental Operations

Selection

SELECT (symbol: σ)

- The SELECT operation is used for selecting a subset of the tuples according to a given selection condition.
- Sigma(σ)Symbol denotes it
- It is used as an expression to choose tuples which meet the selection condition.

 $\sigma_p(r)$ σ is the predicate r stands for relation which is the name of the table p is prepositional logic

Example 1

```
 $\sigma_{topic = "Database"} (Tutorials)$ 
```

Output - Selects tuples from Tutorials where topic = 'Database'.

Fundamental Operations

Projection

PROJECT (symbol: π)

- The projection eliminates all attributes of the input relation but those mentioned in the projection list.
- The projection method defines a relation that contains a vertical subset of Relation.
- This helps to extract the values of specified attributes to eliminates duplicate values.

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive
4	Alibaba	Active

Here, the projection of CustomerName and stat

π CustomerName, Status (Customers)	
CustomerName	Status
Google	Active
Amazon	Active
Apple	Inactive
Alibaba	Active

Image Source: <https://www.guru99.com/relational-algebra-dbms.html>

Fundamental Operations

CARTESIAN PRODUCT

CARTESIAN PRODUCT (x)

- Is helpful to merge columns from two relations.
- Is never a meaningful operation when it performs alone.
- It becomes meaningful when it is followed by other operations.
- $\sigma_{\text{column 2} = '1'} (A \times B)$



Image Source: [Image Source: https://www.minigranth.com/dbms-tutorial/relational-algebra/](https://www.minigranth.com/dbms-tutorial/relational-algebra/)

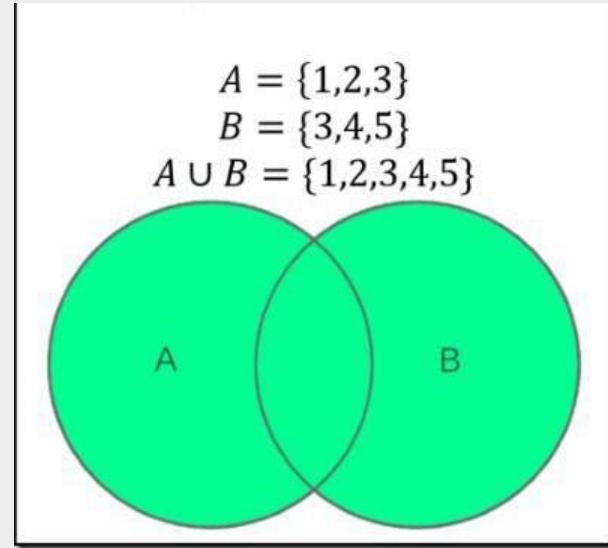
Fundamental Operations

Union

UNION (u)

- UNION is symbolized by U symbol.
- It includes all tuples that are in tables A or in B.
- It also eliminates duplicate tuples.
- set A UNION set B would be expressed as

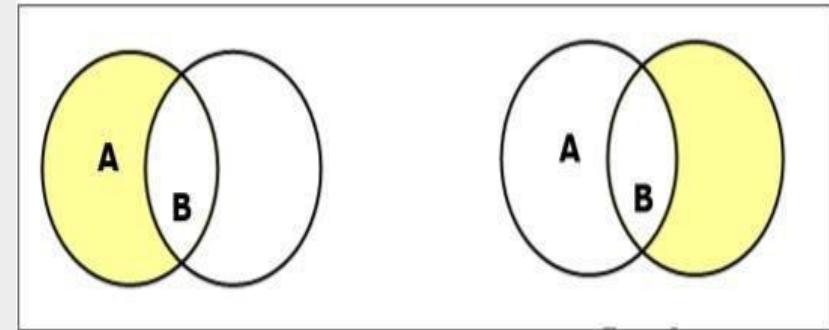
$A \cup B$



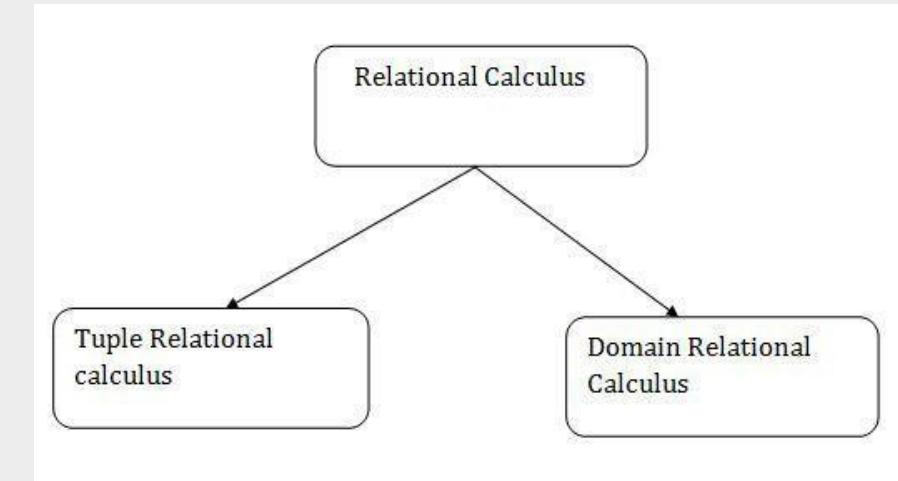
Fundamental Operations

Set Difference

- Difference between sets is denoted by ' $A - B$ ', is the set containing elements of set A but not in B. i.e all elements of A except the element of B.
- Example – If $A = \{ 10, 11, 12, 13 \}$ and $B = \{ 13, 14, 15 \}$, then $(A - B) = \{ 10, 11, 12 \}$ and $(B - A) = \{ 14, 15 \}$. Here, we can see $(A - B) \neq (B - A)$



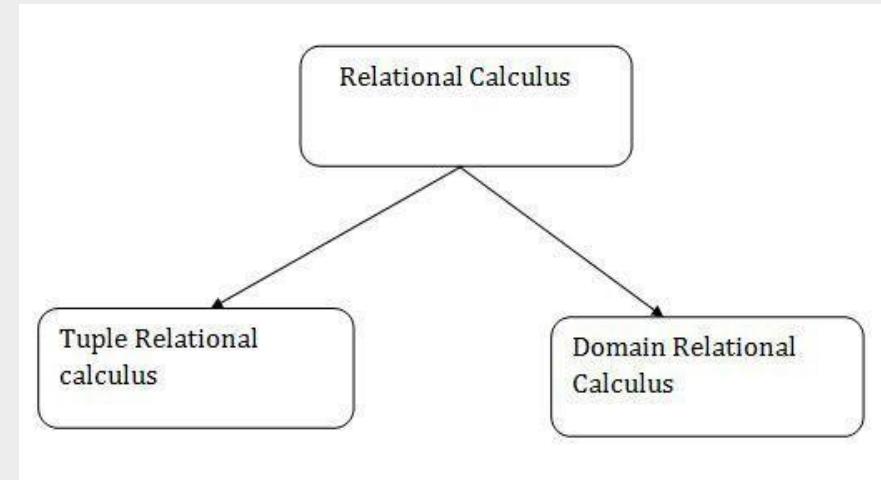
Relational Calculus



Relational Calculus

Definition

- In contrast to Relational Algebra, Relational Calculus is a non-procedural query language, that is, it tells what to do but never explains how to do it.

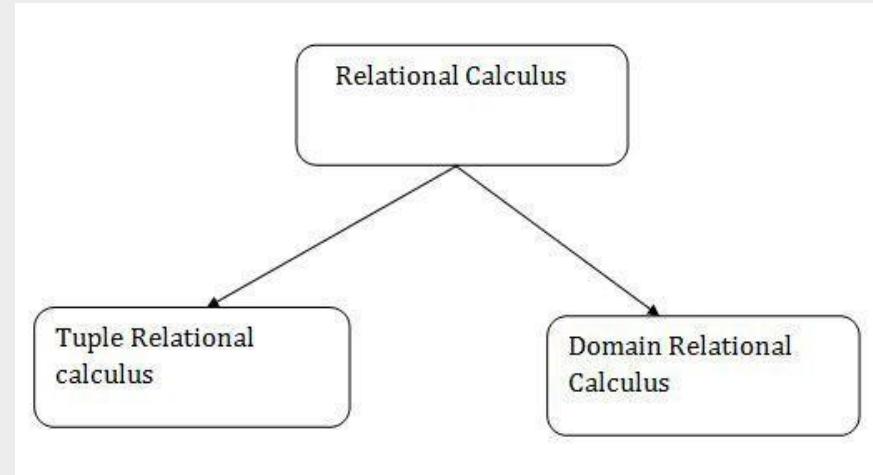


Relational Calculus

Relational Calculus forms

Tuple Relational Calculus (TRC)

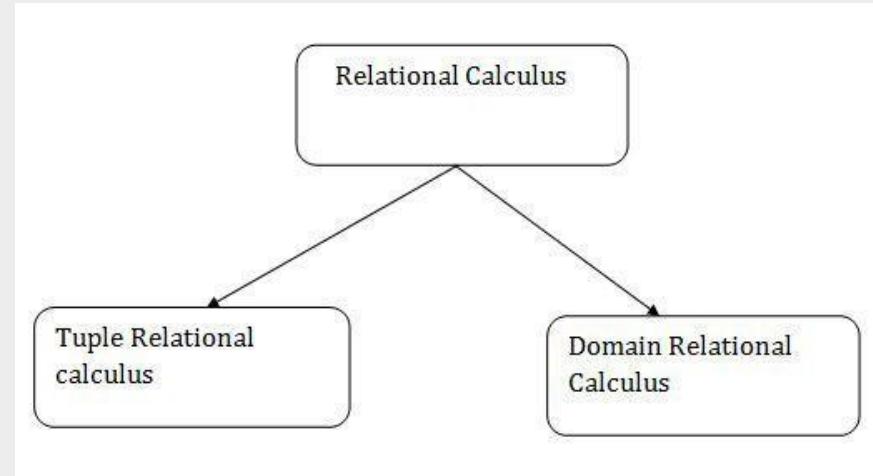
- is a non-procedural query language unlike relational algebra.
- Tuple Calculus provides only the description of the query
- it does not provide the methods to solve it.
- Thus, it explains what to do but not how to do.



Relational Calculus

TRC (Contd..)

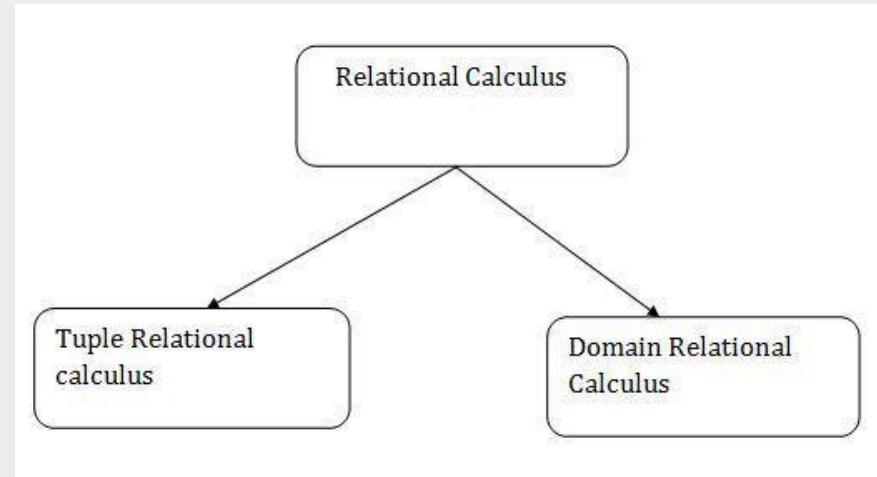
- In Tuple Calculus, a query is expressed as $\{t \mid P(t)\}$
- where t = resulting tuples,
 $P(t)$ = known as Predicate and these are the conditions that are used to fetch t
- Thus, it generates set of all tuples t , such that Predicate $P(t)$ is true for t .
- $P(t)$ may have various conditions logically combined with OR (\vee), AND (\wedge), NOT (\neg).



Relational Calculus

TRC (Contd..)

- It also uses quantifiers:
 $\exists t \in r (Q(t))$ = "there exists" a tuple in t in relation r such that predicate Q(t) is true.
 $\forall t \in r (Q(t))$ = Q(t) is true "for all" tuples in relation r.

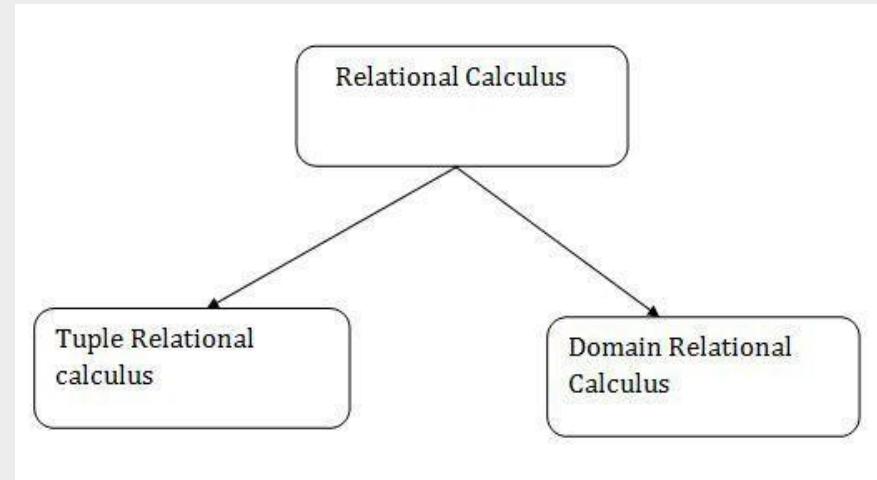


Relational Calculus

Domain Relational Calculus

Domain Relational Calculus (DRC)

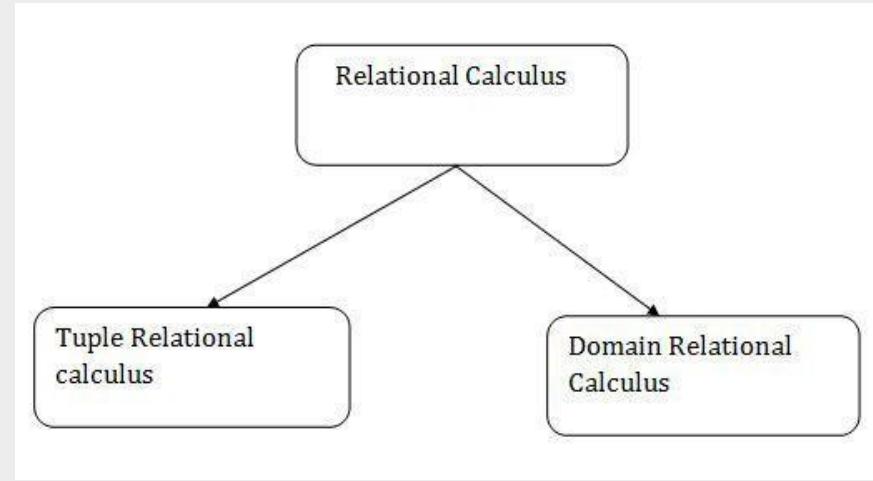
- is a non-procedural query language equivalent in power to Tuple Relational Calculus.
- provides only the description of the query but it does not provide the methods to solve it.



Relational Calculus

DRC (Contd...)

- In Domain Relational Calculus, a query is expressed as,
$$\{ < x_1, x_2, x_3, \dots, x_n > \mid P(x_1, x_2, x_3, \dots, x_n) \}$$
- where, $< x_1, x_2, x_3, \dots, x_n >$ represents resulting domains variables
- And $P(x_1, x_2, x_3, \dots, x_n)$ represents the condition
- Or formula equivalent to the Predicate calculus.



Data Definition Language

DDL
CREATE
ALTER
DROP
TRUNCATE
COMMENT
RENAME

Data Definition Language

Data Definition Language (DDL)

- DDL actually consists of the SQL commands that can be used to define the database schema.
- It simply deals with descriptions of the database schema.
- And is used to create and modify the structure of database objects in the database.

DDL
CREATE
ALTER
DROP
TRUNCATE
COMMENT
RENAME

Data Definition Language

DDL(Contd...)

DDL
CREATE
ALTER
DROP
TRUNCATE
COMMENT
RENAME

Examples of DDL commands

- **CREATE** – is used to create the database or its objects
- **DROP** – is used to delete objects from the database.
- **ALTER** -is used to alter the structure of the database.

Data Definition Language

DDL(Contd...)

Examples of DDL commands

- TRUNCATE –is used to remove all records from a table, including all spaces allocated for the records are removed.
- COMMENT –is used to add comments to the data dictionary.
- RENAME –is used to rename an object existing in the database.

DDL
CREATE
ALTER
DROP
TRUNCATE
COMMENT
RENAME

Operators

 $\sigma_p(r)$

σ is the predicate

r stands for relation which is the name of the table

p is prepositional logic

Example 1

```
 $\sigma$  topic = "Database" (Tutorials)
```

Output - Selects tuples from Tutorials where topic = 'Database'.

Operators

Select

SELECT (symbol: σ)

- The SELECT operation is used for selecting a subset of the tuples according to a given selection condition.
- Sigma(σ)Symbol denotes it
- It is used as an expression to choose tuples which meet the selection condition.

 $\sigma_p(r)$ σ is the predicate r stands for relation which is the name of the table p is prepositional logic

Example 1

 $\sigma_{topic = "Database"}(Tutorials)$

Output - Selects tuples from Tutorials where topic = 'Database'.

Operators

Project

PROJECT (symbol: π)

- The projection eliminates all attributes of the input relation but those mentioned in the projection list.
- The projection method defines a relation that contains a vertical subset of Relation.
- This helps to extract the values of specified attributes to eliminates duplicate values.

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive
4	Alibaba	Active

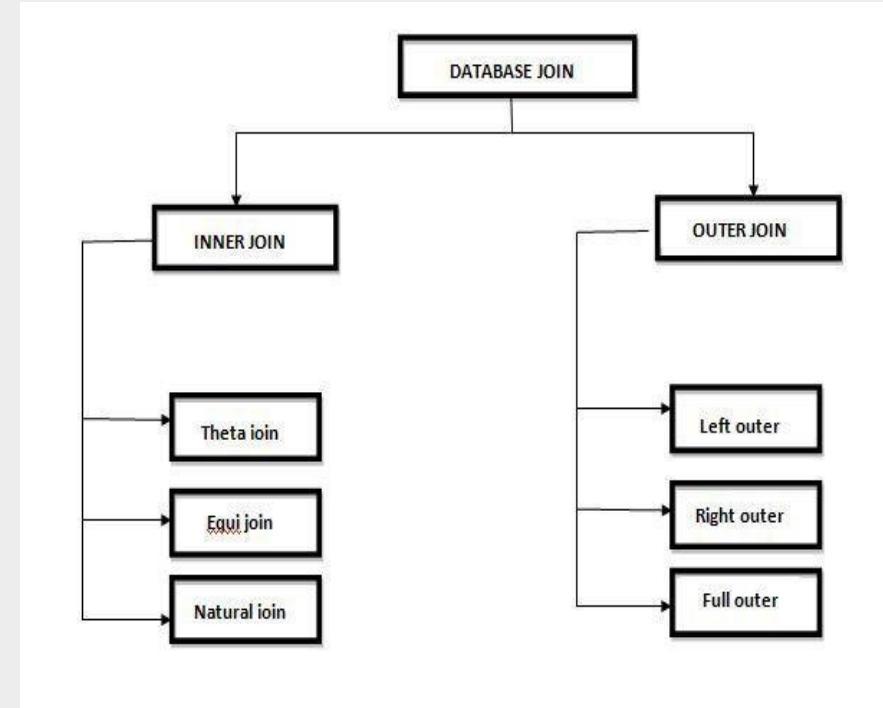
Here, the projection of CustomerName and stat

π CustomerName, Status (Customers)	
CustomerName	Status
Google	Active
Amazon	Active
Apple	Inactive
Alibaba	Active

Operators

Join (\bowtie)

- Join operation is essentially a cartesian product followed by a selection criterion.
- Join operation denoted by \bowtie .
- JOIN operation also allows joining variously related tuples from different relations.

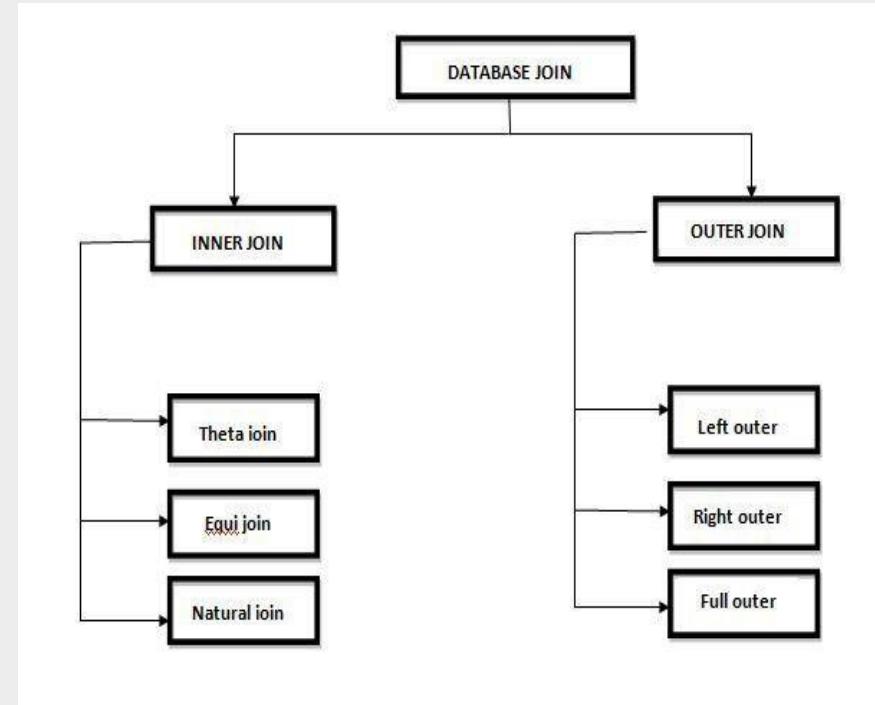


Operators

Join

Inner Joins

- In an inner join, only those tuples that satisfy the matching criteria are included.
- The rest are excluded.



Operators

Join

Inner Joins

1. Theta join

- The general case of JOIN operation is called a Theta join.
- It is denoted by symbol θ
- Example

$$A \bowtie_{\theta} B$$

Student_Details

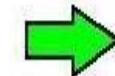
Roll No.	Name	Address
1	Anoop	Delhi
2	Anurag	Noida

Student_Result

Roll No.	Subject	Marks
1	DBMS	10
1	C++	20
2	Java	30
2	C++	40



Student_Details $\bowtie_{(Student_Details.Roll\ No = Student_Result.Roll\ no)}$ Student_Result



Roll No.	Name	Address	Subject	Marks
1	Anoop	Delhi	DBMS	10
1	Anoop	Delhi	C++	20
2	Anurag	Noida	Java	30
2	Anurag	Noida	C++	40

Operators

Join

Inner Joins

2. EQUI join

- When a theta join uses only equivalence condition, it becomes a equi join.
- When Theta join uses only equality comparison operator, it is said to be equijoin.
- EQUI join is the most difficult operations to implement efficiently in an RDBMS.

Student_Details

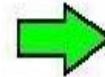
Roll No.	Name	Address
1	Anoop	Delhi
2	Anurag	Noida

Student_Result

Roll No.	Subject	Marks
1	DBMS	10
1	C++	20
2	Java	30
2	C++	40



Student_Details $\bowtie_{(Student_Details.Roll\ No = Student_Result.Roll\ no)}$ Student_Result



Roll No.	Name	Address	Subject	Marks
1	Anoop	Delhi	DBMS	10
1	Anoop	Delhi	C++	20
2	Anurag	Noida	Java	30
2	Anurag	Noida	C++	40

Operators

Join

Inner Joins

3. Natural Join

- Natural join does not utilize any of the comparison operators.
- Attributes should have the same name and domain.
- There should be at least one common attribute between two relations.

Student_Details

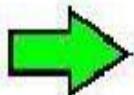
Roll No.	Name	Address
1	Anoop	Delhi
2	Anurag	Noida
3	Ganesh	UP

Student_Result

Roll No.	Marks
1	20
2	30
3	10



Student_Details \bowtie Student_Result



Roll No.	Name	Address	Marks
1	Anoop	Delhi	20
2	Anurag	Noida	30
3	Ganesh	UP	10

Operators

Joins

Outer join

- In an outer join, along with tuples that satisfy the matching criteria, we also include some or all tuples that do not match the criteria.

Student_Details

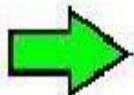
Roll No.	Name	Address
1	Anoop	Delhi
2	Anurag	Noida
3	Ganesh	UP

Student_Result

Roll No.	Marks
1	20
2	30
3	10



Student_Details \bowtie Student_Result



Roll No.	Name	Address	Marks
1	Anoop	Delhi	20
2	Anurag	Noida	30
3	Ganesh	UP	10

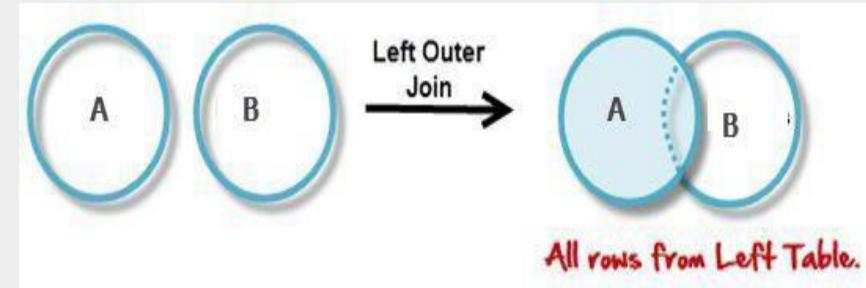
Operators

Joins

Outer join

1. Left Outer Join

- In the left outer join, operation allows keeping all tuple in the left relation.
- However, if there is no matching tuple is found in right relation, then the attributes of right relation in the join result are filled with null values.

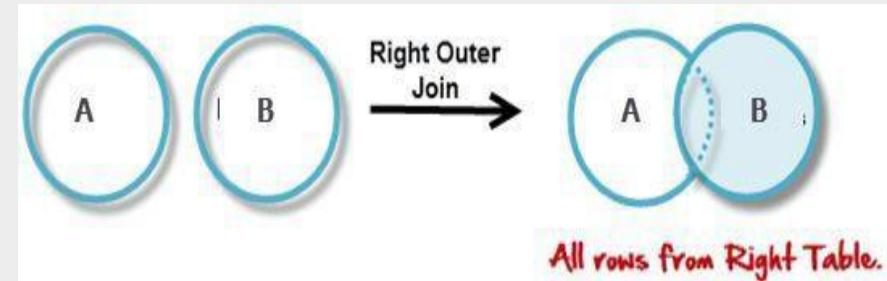


Operators

Join

Outer join 2. Right Outer Join

- In the right outer join, operation allows keeping all tuple in the right relation.
- However, if there is no matching tuple is found in the left relation, then the attributes of the left relation in the join result are filled with null values.



A \bowtie B

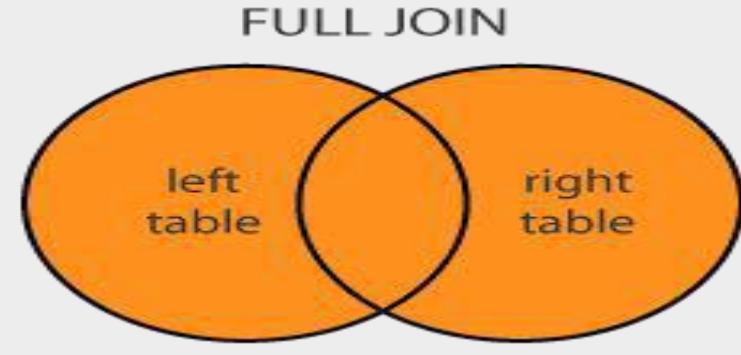
Operators

Join

Outer join

3. Full Outer Join

- In a full outer join, all tuples from both relations are included in the result, irrespective of the matching condition.
- However, if there is no matching tuple is found in the left relation, then the attributes of the left relation in the join result are filled with null values.



A \bowtie B

Operators

Rename

RENAME (symbol: ρ)

- The results of relational algebra are also relations but without any name.
- The rename operation allows us to rename the output relation.
- Is denoted with small Greek letter rho ρ .
- Notation – $\rho_x (E)$
- Where the result of expression E is saved with name of x.

$\rho_S(R)$ changes:

the *relation name* only to S

$\rho_{(B1, B2, \dots, Bn)}(R)$ changes:

the *column (attribute) names* only to B1, B1,Bn

$\rho_{(B1, B2, \dots, Bn)}(R)$ changes both:

the relation name to S, and

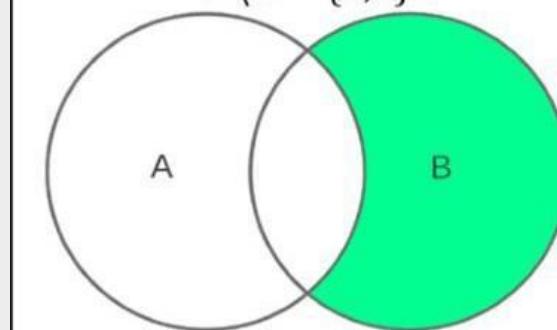
the column (attribute) names to B1, B1,Bn

Operators

Set Difference

- Difference between sets is denoted by ' $A - B$ ', is the set containing elements of set A but not in B. i.e all elements of A except the element of B.
- Example – If $A = \{ 10, 11, 12, 13 \}$ and $B = \{ 13, 14, 15 \}$, then $(A - B) = \{ 10, 11, 12 \}$ and $(B - A) = \{ 14, 15 \}$. Here, we can see $(A - B) \neq (B - A)$

$$\begin{aligned}A &= \{1,2,3\} \\B &= \{3,4,5\} \\B \setminus A &= \{4,5\}\end{aligned}$$



Software Development Life Cycle(SDLC)

(5 hours)

In this section, we will discuss:

- SDLC Overview
- Phases of Software Development Life Cycle (SDLC)
- Software Development Life Cycle Models
- Software Test Levels

SDLC Overview

What is SDLC?

- SDLC is a process followed for a software project, within a software organization.
- It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software..

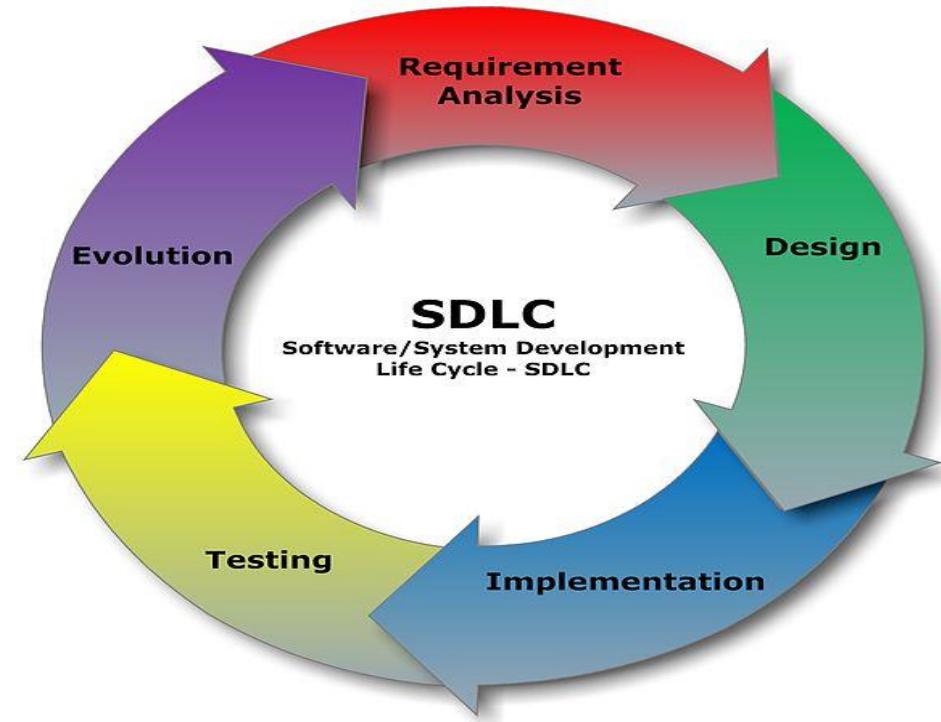


Image Source:

https://commons.wikimedia.org/wiki/File:SDLC_-_Software_Development_Life_Cycle.jpg

SDLC Overview

Why SDLC?

- Improved client relations
- Provides a framework for a standard set of activities and deliverables
- Increased and enhance development speed



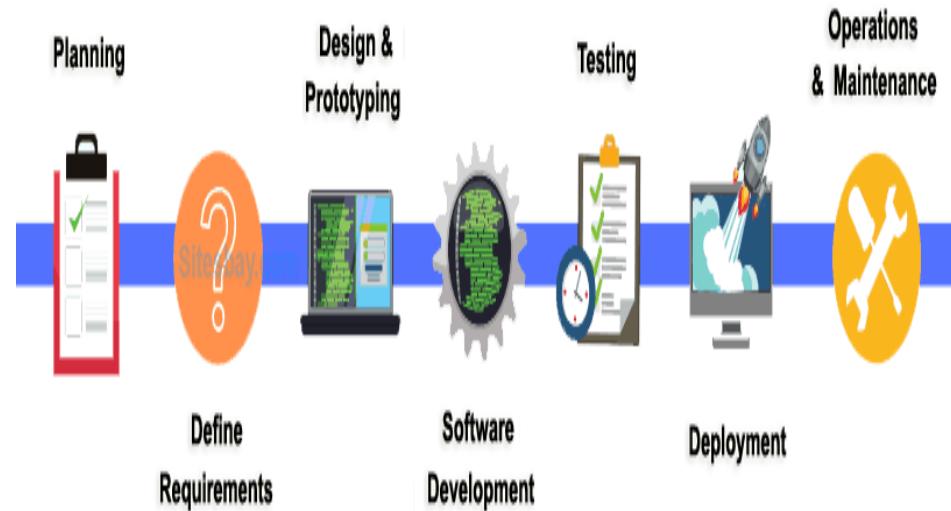
Image Source:

<https://clarusway.com/what-is-software-development-life-cycle/>

Phases of Software Development Life Cycle (SDLC)

Phases of SDLC

- Phase 1: Requirement collection and analysis
- Phase 2: Feasibility study
- Phase 3: Design
- Phase 4: Coding
- Phase 5: Testing
- Phase 6: Installation/Deployment
- Phase 7: Maintenance



Software Development Life Cycle models

SDLC Models

- Waterfall Model
- Iterative Model
- V-Shaped Model
- Spiral Model
- Agile Model
- Big Bang Model

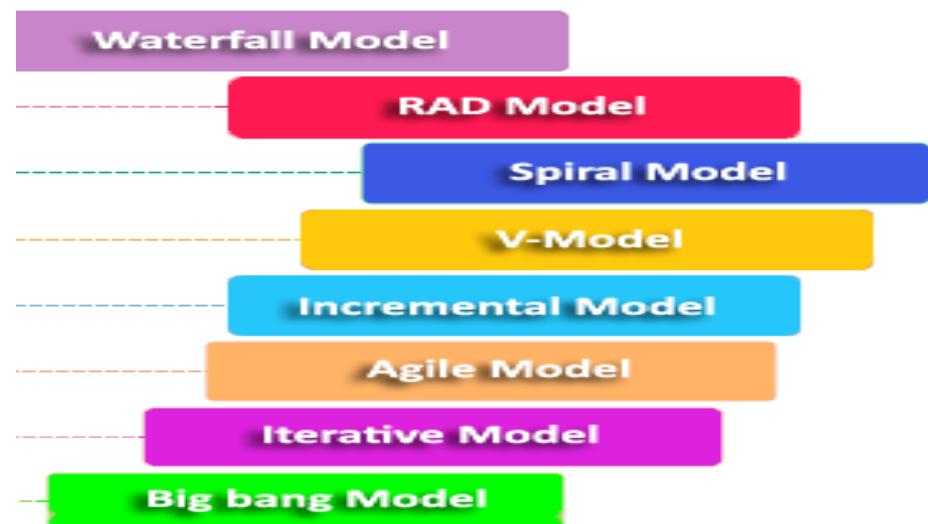


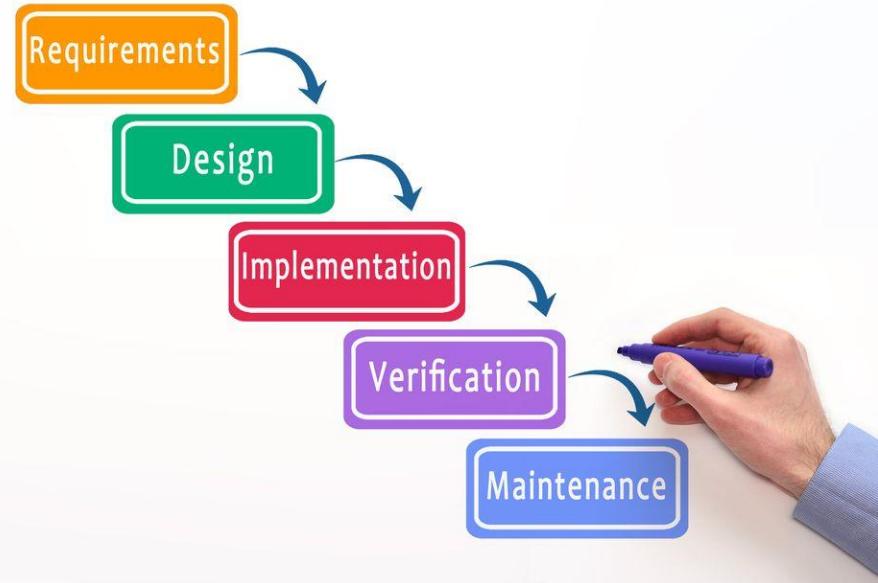
Image Source:

<https://www.sitesbay.com/software-engineering/se-what-is-sdlc-model>

SDLC Models

Waterfall Model

- Waterfall Model is a sequential model that divides software development into pre-defined phases.
- Each phase must be completed before the next phase can begin with no overlap between the phases



SDLC Models

Iterative Model

- A subset of the final product under development, which grows from iteration to iteration to become the final product or software

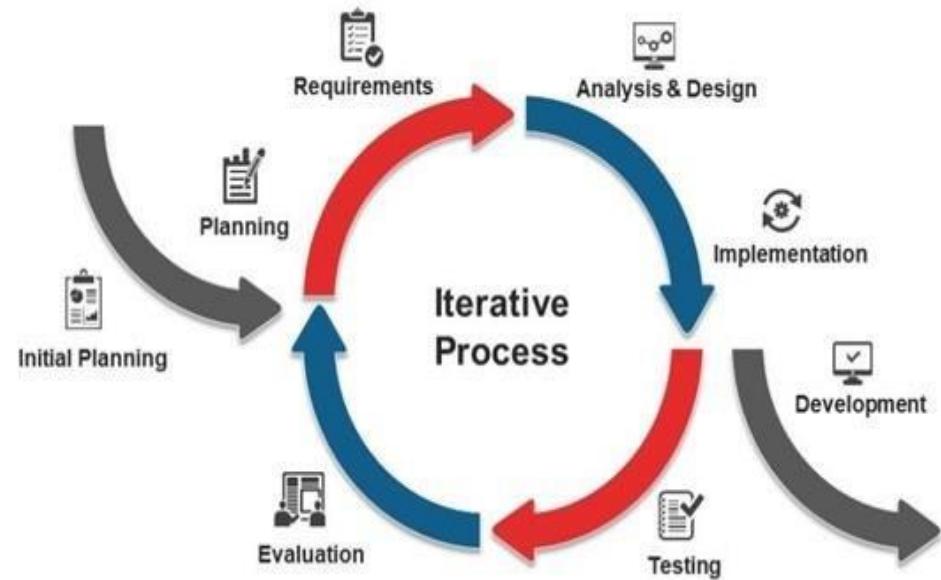


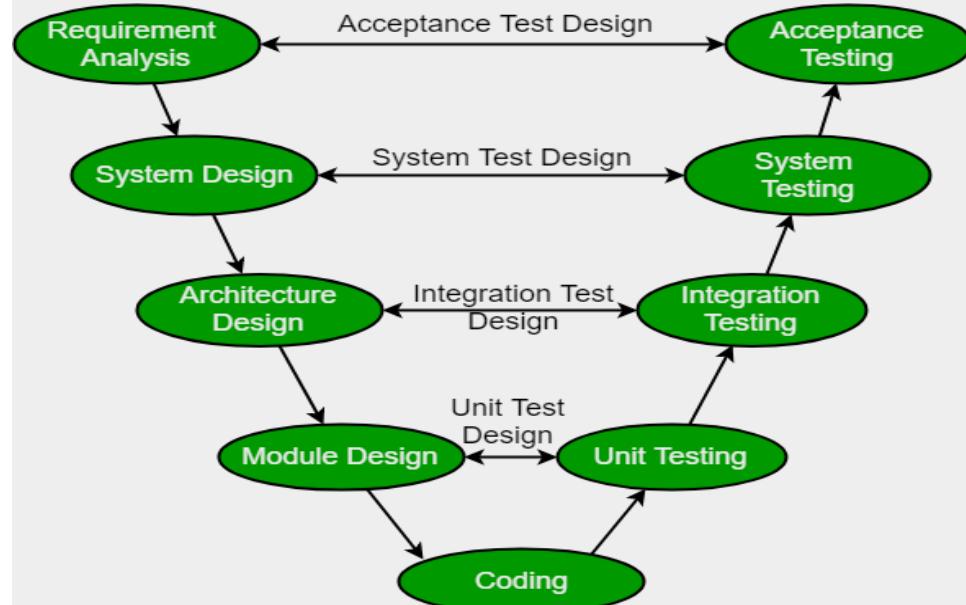
Image Source:

https://www.researchgate.net/figure/SDLC-Iterative-Model-2_fig4_338710620

SDLC Models

V-Shaped Model

- The V-model is a type of SDLC model where process executes in a sequential manner in V-shape.
- It is also known as Verification and Validation model.



SDLC Models

Spiral Model

- The spiral model is a systems development lifecycle (SDLC) method used for risk management that combines the iterative development process model with elements of the Waterfall model.



Image Source:

<https://eternalsunshineoftheismind.files.wordpress.com/2013/02/i-s-spiral.jpg>

SDLC Models

Agile Model

- Agile SDLC methodology is based on collaborative decision making between requirements and solutions teams, and a cyclical, iterative progression of producing working software.

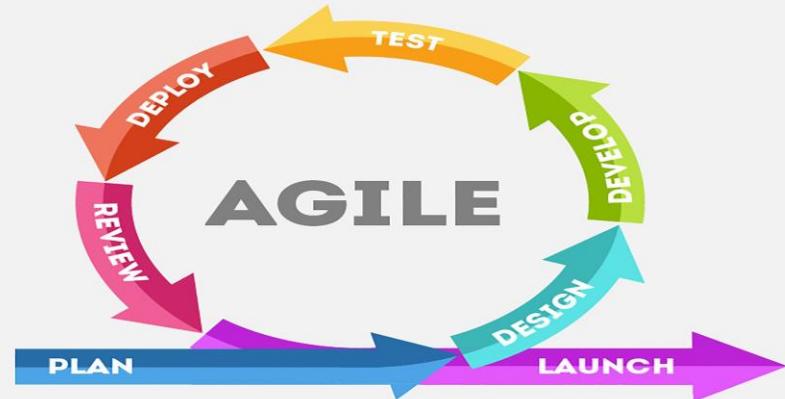


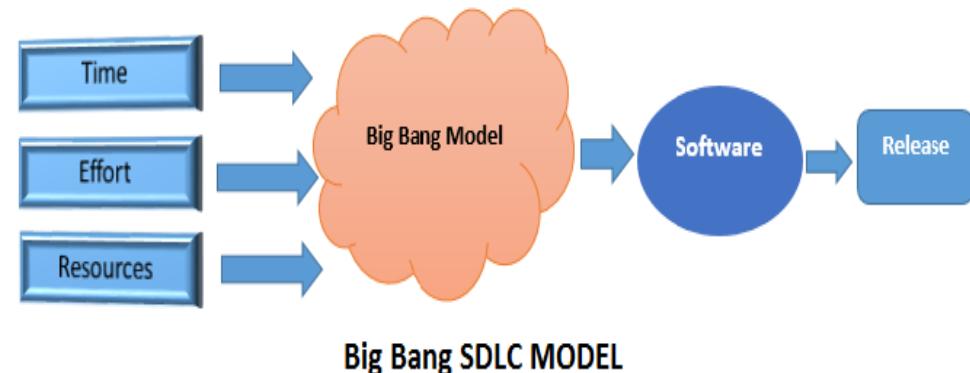
Image Source:

<https://gearheart.io/articles/7-phases-software-development-life-cycle-sdlc/>

SDLC Models

Bing Bang Model

- The Big bang model is an SDLC model that starts from nothing.
- It is the simplest model in SDLC as it requires almost no planning



Software Test Levels

What is Software Testing?

- Software testing is the process of evaluating and verifying that a software product or application does what is suppose to do.



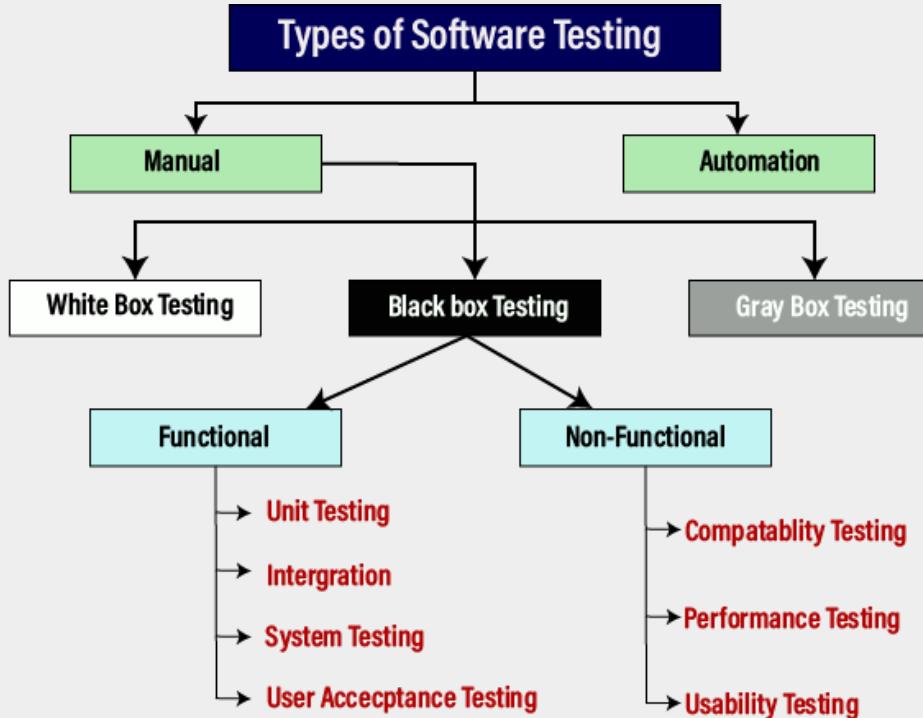
Image Source:

<https://www.inwizards.com/blog/software-testing-type-testing-introduction-basics-importance/>

Software Test Levels

Types of Software Testing

- Manual Testing
- Automation Testing



Software Test Levels

Levels of Testing

- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing

