

1

BCS-053

Ans 1 (a) Web 2.0 - Web 2.0 moved us from a read only Internet to what experts would call a "read/write" Internet.

Suddenly, users were able to enter a range of information into web fields and send it back to the Server, so that they could communicate with hosting servers in real time.

Web 3.0 :- As far as Web 3.0 Internet gurus dream of the day when semantic web and data mapping will transform the web into what you might call a "read write and execute" web where automation programs take over from purely user driven Internet activity.

Here you might theorize that Internet bots will do the actual work of generating the HTTP requests and responses and swift the web instead of human users.

We're not there yet - so it's safe to say that we're still the era of web 2.0.

Web 4.0 :- Web 4.0 is a new evolution of the web paradigm based on multiple models, technologies and social relationships. The concept of Web 4.0 is not totally clear.

and Unanimous in literature, because it is composed by several dimensions. In this sense this study uses a systematic review approach to clarify the concept of Web 4.0 and explore its various dimensions. Analyzing if they have elements in common.

Term www and its uses = Some people use the terms "internet" and world wide web interchangeably. They think they are the same things - but it is not so. Internet is entirely different from www. It is a worldwide network of devices like computers, laptops, tablets etc. It enables user to send emails to other users and chat with them online. For example, when you send

The term Blogging: The word blog is actually a shortened form of its original name, "weblog". These weblogs allowed early internet user to "log" the details of their day in diary-style entries. Blogs often allow readers to comment so as they became more common, communities sprung up around popular blogs.

10 Protocols / Software that can be used as Web 2.0 technologies:

1. Search
2. Blogging
3. Content Network
4. Podcasting
5. Social Networking
6. Content Hosting Services
7. Rich Internet Application
8. Web Services
9. Mashups
10. Social Bookmarking.

(b) Register html

```

<html>
  <head>
    <title> Register </title>
    <link rel="stylesheet" href="Style.css">
    <script src="Script.js"></script>
  </head>
  <body>
    <!-- from header -->
    <div class="form-header">
      <h1> Create Account </h1>
    </div>
    <!-- form body -->
  
```

<!-- Username and Password -->

<div class = "horizontal-group">

<div class = "form-group-left">

<form name = "form1">

<label for = "email" class = "label-title"> Email </label>

<input type = "email" name = "label-title" > Email

input" placeholder = "Enter Desired Email" required = "required" />

</form>

</div>

<div class = "form-group-right">

<label for = "Password" class = "label-title"> Password </label>

<input type = "password" id = "password" class = "form-input" placeholder = "enter Password" />

</div>

</div>

<!-- Name -->

<div class = "form-group">

<label for = "name" class = "label-title"> Name </label>

<input type = "text" id = "name" class = "form-input" placeholder = "Enter Your Name" required = "required" />

</div>

<!-- City and State -->

<div class = "horizontal-group">

<div class = "form-group-left">

<label for="City" class="label-title">City\*</label>  
 <Select class="form-input" id="level">  
   <option value="Mumbai">Mumbai</option>  
   <option value="New Delhi">New Delhi</option>  
   <option value="Bangalore">Bangalore</option>  
   <option value="Hyderabad">Hyderabad</option>  
   <option value="Jaipur" selected>Jaipur</option>  
   <option value="Chennai">Chennai</option>  
   <option value="Kolkata">Kolkata</option>  
   <option value="Surat">Surat</option>  
   <option value="Pune">Pune</option>  
   <option value="Lucknow">Lucknow</option>  
   <option value="Kanpur">Kanpur</option>  
   <option value="Nagpur">Nagpur</option>  
   <option value="Indore">Indore</option>  
 </Select>

</div>

<div class="form-group right">  
   <label for="State" class="label-title">State\*</label>  
   <Select class="form-input" id="level">  
     <option value="Maharashtra">Maharashtra</option>  
     <option value="Delhi">Delhi</option>  
     <option value="Karnataka">Karnataka</option>  
     <option value="Telangana">Telangana</option>  
     <option value="Rajasthan" selected>Rajasthan</option>  
     <option value="Tamil Nadu">Tamil Nadu</option>

```

</select>
</div>
<!--Desired--><div class="form-group">
<div class="form-group">
<label for="deniroeservice" class="label-title">Desired
Email Service </label>
<select class="form-input" id="level">
<option value="Basic"> Basic Service </option>
<option value="Advance"> Advance Service </option>
</select>
</div>
<!--Form Footer-->
<div class="form-footer">
<button type="submit" class="btn"
onclick="register()">> Submit </button>
</div>
</div>
</div>
</form>
</body>
</html>

```

Sign Style - CSS

body {  
background-color: #D3D3D3;  
}

Sign-up-form {

font-family: sans-serif;  
 width: 650px;  
 margin: 130px auto;  
 background: linear-gradient(to right, #fffff0%,  
 #fafafa 50% #fffff1 99%);  
 border-radius: 10px;  
 {

.form-header {  
 background-color: #EFF0F1;  
 border-top-left-radius: 10px;  
 border-top-right-radius: 10px;  
 }

.form-head h1 {  
 font-align: center;  
 color: #3498DB;  
 padding: 20px 0;  
 border-bottom: 1px solid #cccccc;  
 }

/\* Form Body \*/

.form-body {  
 padding: 10px 40px;  
 color: #666;  
 }

.form-group {  
 margin-bottom: 20px;  
 }

. form - body . label - title {

color : # 006400;

font size : 16px;

font weight : bold;

}

. form - body . form - input {

font-size : 12px;

box-sizing : border-box;

width : 100%;

height : 34px;

padding-left : 10px;

padding-right : 10px;

color : # 0000 8B;

text-align : left;

border : 1px solid # d6d6d6;

border-radius : 4px;

background : white;

outline : none;

}

. horizontal - group . left {

float : left;

width : 49%;

?

. horizontal - group . right {

float : right;

width : 49%;

{

9

!! - Webkit - input - Placeholder {

color: #d9d9d9;

}

\* form footer \*/

.form-footer {

clear: both;

}

\* form footer \*/

.Signup-form form-footer {

background-color: #EFF0F1;

border-bottom-left-radius: 10px;

border-bottom-right-radius: 10px;

padding:

Script.js

function CheckPass(password) {

if (password.match(/[a-zA-Z]/g) && password.match(/[^A-Z]/g) && password.match(/[^0-9]/g) && password.match(/[^a-zA-Z\d]/g) && password.length >= 8)

return true;

else

return false;

?

function CheckEmail(emailId) {

Var reg = /^[A-Za-z0-9]+@[A-Za-z0-9-]+\.[A-Za-z]{2,4}\$/;

```
if (reg. test (email / id) == false
    {  
    return false;  
}
```

```
} else {  
    return true;  
}  
}
```

function register () {

email = document.getElementById ("username").  
Value;

psw = document. getElementById ("Password").  
Value;

if (!email || !psw) {

alert ("Please enter the email and Password");

} else {

mail chk = checkEmail (email);

pswchk = checkPSW (psw);

if (mail chk == true & & pswchk == true) {

alert ("good to go");

} else {

alert ("Please check your email and  
follow password Criteria");

}

{

## (c) Schedule.html

```

<html>
  <head>
    <title> BCA Time table </title>
    <style>
      h1 { font-family: Cambria; }
      table, th, td { border-collapse: collapse; border: 1px solid black; font-size: 14pt; }
      font-family: Cambria; }
      tr:nth-child(even) { background-color: #98FB98; }
    </style>
  </head>
  <body>
    <center><br/><h1>BCA 1st Semester Assignment Schedule</h1><br/>
    <table cellpadding="10" border="1" style="width: 100%; border-collapse: collapse; border: none; text-align: center; margin-bottom: 10px;">
      <tr><th><i>Course Code</i></th><th><i>Course Name</i></th><th><i>Last Date</i></th><th><i>Viva-Voice</i></th><th><i>Assignment No</i></th></tr>
      <tr><td>CS-053</td><td>Computer Organization</td><td>15/01/2024</td><td>15/01/2024</td><td>1</td></tr>
      <tr><td>CS-054</td><td>Computer Architecture</td><td>15/01/2024</td><td>15/01/2024</td><td>2</td></tr>
      <tr><td>CS-055</td><td>Computer Organization</td><td>15/01/2024</td><td>15/01/2024</td><td>3</td></tr>
      <tr><td>CS-056</td><td>Computer Organization</td><td>15/01/2024</td><td>15/01/2024</td><td>4</td></tr>
    </table>
    </body>
</html>

```

## Suggestion.html

```

<html>
  <head>
    <title> Syllabus of BCS-053 </title>

```

< style >

h1 { font-family: 'Combia'; color: #000000; }  
 h2 { font-family: 'Combia'; color: #8C1AFT; }  
 . footer { font-size: 11pt; font-family: 'Times New Roman'; }

Subheader { font-size: 11pt; font-family: 'Times New Roman'; font-weight: bold; color: #FF3333; }

</style >

</head >

<body >

<center><h1>Li > Assignment writing suggestion </h1><br/>

<center><br/>

<div >

<ul > Use foolscap size paper for writing your answer.

<ul > Leave 4cm margin on the top, bottom and left of your answer sheet </ul >

<ul > Clearly indicate question no. and part of the question being solved while writing answer. </ul >

<ul > Assignment have to be sent to the Coordinator of your Study Centre </ul >

<ul > We strongly suggest that you should retain a copy of your assignment responses. </ul >

<ul > You are expected to complete all your assignment before writing the final term-end Examination </ul >

</div >

</body>  
</html>

(d) Customer.xml

```

<?xml Version="1.0"?>
<!DOCTYPE Customer SYSTEM "Customer.dtd">
<Customer List>
  <Customer>
    <CID>100001</CID>
    <Name>Akash</Name>
    <Address>New Delhi</Address>
    <Phone>8896542376</Phone>
    <Profession>Shopkeeper</Profession>
  </Customer>
  <Customer>
    <CID>100002</CID>
    <Name>Karan</Name>
    <Address>Mumbai</Address>
    <Phone>9997662233</Phone>
    <Profession>Engineer</Profession>
  </Customer>
  <Customer>
    <CID>110003</CID>
    <Name>Rajesh</Name>
    <Address>up</Address>
  
```

```

< Phone > 8822554411< /Phone >
< Profession > Lawyer < /Profession >
< /Customer >
< Customer >
< CID>100004< /CID >
< Name > Shubham < /Name >
< Address > Kolkata < /Address >
< Phone > 8956165154 < /Phone >
< Profession > Doctor < /Profession >
< /Customer >
< Customer >
< CID>100005< /CID >
< Name > Rehul >< /Name >
< Phone > 89765568 < /Phone >
< Profession > Teacher < /Profession >
< /Customer >
< /Customerlist >

```

### Customer. lxt

```

<!ELEMENT List (customer+)>
<!ELEMENT Customer (ID; Name, Address, Phone, Profession)
<!ELEMENT CID (#PCDATA)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Address (#PCDATA)>
<!ELEMENT Phone (#PCDATA)>
<!ELEMENT Profession (#PCDATA)>

```

(e)

```

Var text = "let us try Java script";
Var Counter = 0;
Var elem = document.getElementById("write
programs using Java Script");
Var inst = SetInterval(Change, 1000);
function Change() {
    elem.innerHTML = text[Counter];
    Counter++;
    if (Counter >= text.length) {
        Counter = 0;
        // ClearInterval(inst); // Uncomment
        // this if you want to stop refreshing
        // after one cycle
    }
}

```

(f)

WAP Model : WAP Gateway / Proxy is the entity that connects the wireless domain with the Internet. You should make a note that the request that is sent from the wireless client to the WAP Gateway / Proxy uses the wireless session protocol.

A markup language - the wireless language (WML) has been adapted to develop optimized WAP applications. In order to save valuable

bandwidth in the wireless network, WML can be encoded into a compact binary format. Encoding WML is one of the tasks performed by the WAP Gateway / Proxy.

1. WSP Request (URL)
2. WSP Response (Binary WML)
3. HTTP Request (URL)
4. HTTP Request (WML)

How WAP Model Works.

- The user selects an option on their mobile phone device that has a URL with wireless Markup language (WML) Content assigned to it.
- The phone sends the URL request via the phone network to a WAP gateway using the binary encoded WAP protocol.
- The gateway translates this WAP request into a conventional HTTP request for the specified URL and sends it on to the internet.
- The appropriate Web Server picks up the HTTP request.
- The server processes the request just as it would any other request.
- The Web Server adds the HTTP header to the WML Content and return it to the gateway.

- The WAP gateway Compiles the wml into binary form.
- The gateway then Sends the wml response back to the Phone.
- The Phone receive the wml via the wap protocol.
- The micro-browser processes the WML and displays the content on the Screen.

### WML elements

`<a>` element : The `<a>--</a>` tag pair can also be used to create an anchor link and always a preferred way of creating links. You can enclose Text or image inside `<a>--</a>` tags.

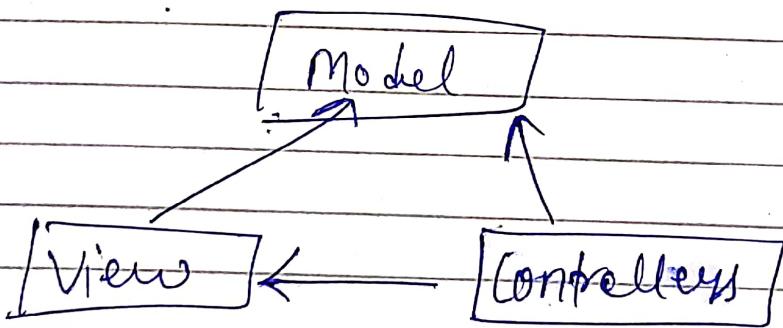
Anchor element: The `<anchor>--</anchor>` tag pair is used to create an anchor link. It is used together with other wml elements called `<go>`, `<refresh>` or `<script>`. These elements are called task elements and tell wap browsers what to do when a user selects the anchor link. You can enclose text or image along with a text tag inside `<anchor>` - `</anchor>` tag pair.

Select elements:- The <select>..</select> WML elements are used to define a selection list and the <option>..</option> tags are used to define an item in a selection list.

Items are represented as radio buttons in some WAP browsers. The <option></option> tag pair should be enclosed within the <select></select> tags.

Ans 2 = (a)

(i) MVC architecture : The Model-View Controller (MVC) is an architectural pattern that separates an application into three main logical components : the model, the view and controller . Each of these components are built to handle specific development aspect of an application . MVC is one of the most frequently used industry - Standard web development framework to create Scalable and extensible projects.



### Model

The model Component correspond to all the data related logic that the user works with. This can represent either the data that is being transferred between the view and controller components or any other business logic-related data.

### View

The View Component is used for all the UI logic of the application. !

### Controller

Controller act as an interface between model and view Components to Boxes all the business logic and incoming requests, Manipulate data using the Model Component and interact with the views to render the final output

### (ii) GET and POST methods:-

#### The GET Method

The GET Method Sends the encoded user information appended to the page request.

The Page and the encoded information are separated by the ? Character.

- Try out following example by putting the source code in test.php script.

```

<?php
if ($_GET["name"] || $_GET["age"]){
    echo "Welcome ". $_GET["name"]. "<br>";
    echo "You are ". $_GET['age']. " year old";
    exit();
}
?>
<html>
<body>
<form action = "<?php $_PHP_SELF?>" method="GET">
Name: <input type = "text" name = "name"/>
Age: <input type = "text" name = "age"/>
<input type = "submit" />
</form>
</body>
</html>

```

## The POST Method

The POST method transfers information via HTTP header. The information is encoded as described case of GET Method and put

into a header called QUERY-STRING

Try out following example by putting the source code in test.php script.

```
<?php
if ($_POST["name"] || $_POST["age"]) {
    if (preg_match ("/[A-Za-Z]/", $_POST['name'])) {
        die ("invalid name and name should be alpha");
    }
}
```

```
echo "Welcome ". $_POST['name']. "<br/>";
echo "You are ". $_POST['age']. "<br/>fear old";
exit();
?>
```

<html>

<body>

```
<form action = "<?php $_PHP_SELF ?>">
    method = "POST">
```

Name : <input type = "text" name = "name"/>

Age : <input type = "text" name = "age"/>

<input type = "Submit"/>

</form>

</body>

</html>

## (iii) Server side Scripting and its tools:

Server side Scripting is a method of designing Webside So that the process or user request is run on the originating.

## Server-side Scripting example

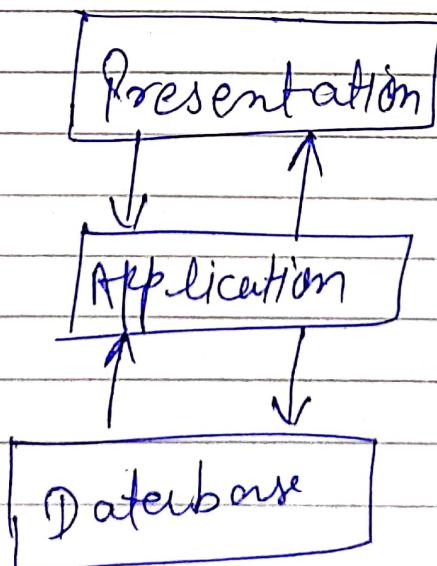
Many language may be used to create these Scripts. They include but are not limited to the example below.

- Active VFP
- ASP
- C
- DC
- JAVA
- JavaScript (Using SSJS (Server side Java script))
- Perl
- PHP
- Python
- R
- Ruby

## (iv) Web Container :- A web Container is the component of a web Server that interacts with Java Servlets. A web

Container manages the life cycle of Servlets; it maps a URL to the particular Servlet while ensuring that the requester has relevant access rights.

- (V) N Tier architecture:- N-tier architecture is also called multi-tier architecture because the software is engineered to have the processing, data management, and presentation function physically and logically separated. That means that these different functions are hosted on several machines or clusters, ensuring that service are provided without resources being shared and as such these services are delivered at top capacity. The "N" in the name n-tier architecture refers to any number form.



(b) (a) The Process of generation dynamic Content using JSP and advantages of using JSP.

As discussed JSP is used for creating dynamic webpages.

Advantages of JSP.

1. JSP has all the advantage of Servlet, like Better performance than CGI Built in Session features,
2. JSP Enables the Separation of Content generation from Content presentation
3. With the JSP, it is now easy for web designers to show Case the information What is needed.
4. Web Application programming can Concentrate on how to process / build the information Code:-

```
<%-- JSP Comment --%>
<HTML>
<HEAD>
<TITLE>&MESSAGE<TITLE>
<HEAD>
<BODY>
<% out.print("Hello, Sample JSP Code"); %>
<Body>
<HTML>
```

(b) Page and include directives of JSP:-  
The include directive is used to include a file during the translation phase. This directive tells the container to merge the content of other external files with the current JSP during the translation phase. You may code include directive anywhere in your JSP page.

Code:-

```
<%!
int PageCount=0;
Void add Count () {
    Page Count++;
}
%>
<% add Count (); %>
<html>
<head>
<title>The include Directive Example</title>
<head>
<body>
<center>
<h2>The include Directive Example</h2>
<p>This site has been visited <% = Page
Count %> times.</p>
</center>
<br><br/>
```

### (c) JSP Declarations:-

Declarations in JSP declare java language Statement. `<%! - %>` tags contain declaration. They can declare class / instances / inner classes / variable / method. Unlike the other two, this code doesn't go to `- JspService()` method. This code rather goes to the source file that gets generated outside the `- JspService` method.

#### Declaration.jsp

```
<html>
<head><title> Declaration </title>
</head>
<body>
<h3>-- Dataflow --</h3>
<h3> Use of Declaration in jsp </h3>
<%! int num1 = 2, num2 = 3, n = 0; %>
<% n = num1 + num2 + 1;
out.println ("The number after adding declares
variable is " + n);
%>
</body>
</html>
```

**JSP Expressions:-** JSP Expression tags, as the name suggests, evaluates expression. They incorporate arithmetic and logical expression so that they can be evaluated. They form an means to access the value of a Java Variable.

<html>

<head><title> Expression </title>

</head>

<body>

<h3> Use of expression : done by

<h3> -- Data flair --</h3>

<h3> Use of expression in JSP</h3>

<% String s1 = " Expression : done by  
keyjal "; %>

<% out.print(s1); %>

<% = s1. toUpperCase(); %>

</body>

</html>

**Scriptlets :-** JSP Scriptlets help in embedding Java in HTML for JSP Code. <% - %> tags contain the Java Code and are called as Scriptlet tags. Unlike expressions, Scriptlets have a set of Statement in Java language. It allows writing Java Code in it.

## Scriptlet . JSP

<html>

<head> <title> Scriptlets </title>

</head>

<body>

<h3> -- Data flavor -- </h3>

<h3> use of Scriptlet in JSP </h3>

<% int a = 3;

int b = 4

int c = 5

out.println("a is: " + a + "  
" + b + "  
" + c + ".");

out.println("Multiplication gives: " + a \* b \* c + "  
");

out.println("Addition gives: " + (a + b + c));

<%>

</body>

</html>

(d) <jsp: getProperty> and <jsp:setProperty>  
Active elements :-

The SetProperty and GetProperty action tags  
are used for developing web application  
with Java Bean. In web development,  
bean class is mostly used because it is  
a reusable software component that  
represents data.

`jsp:getProperty active tag`

The JSP's 'get Property' action tag returns the value of the property.

Syntax of `jsp:getProperty active tag`

`<jsp: getProperty name = "instance of Bean"`

`Property = "Property Name" />`

Simple example of `JSP: getProperty active tag`

`<jsp: getProperty name = "Obj" Property = "name">`

- (e) Session and Out implicit object:- Session is most frequently used implicit object in JSP. The main usage of it to gain access to all the user's data till the user session is active.

### Methods of Session implicit Object.

1. `Set Attribute (String, Object)` - This method is used to save an object in session by assigning a unique string to the object.
2. `getAttribute (String name)` - The object stored by Set Attribute method is fetched from session using get Attribute method.
3. `removeAttribute (String name)` - The objects which are stored in session can be removed from session using this method.

4. Get Attribute Name - It return all the objects objects stored in Session.
5. Get Creation Time - This method return the Session Creation Creation time, the time when session got initiated (became active)
6. getId - Servlet Container assigns a unique String Identifier to session while creation of it.
7. isNew() - used to check whether whether the session is new.
8. invalidate() = it kills a session and breaks the association of session with all the stored objects.
9. getMaxInactiveInterval - Return session's maximum inactivity time interval in seconds.
10. getLastAccessedTime - Generally used to know the last accessed time of a session.

index.html  
 <html>  
 <head>  
 <title> Welcome Page : Enter your name </title>  
 </head>  
 <body>  
 <form action = "session.jsp">  
 <input type = "text" name = "inputname">  
 <input type = "Submit" value = "Click here!">  
 </form>

</body>

</html>

(c) (i) <html>

'<head>

<title>Enter two number to add <br> </title>

</head>

<body>

<% = " <h1> The Value is "

" + Integer.parseInt(request.getParameter("a"))"

Integer.parseInt(request.getParameter("b"))

/ Integer.parseInt(request.getParameter("c"))

+ Integer.parseInt(request.getParameter("d")) + "<br>" %>

</body>

</html>

(ii) Cookies are created to identify you when you visit a new website. The web server - which stores the website's data - sends a short stream of identifying info to your web browser.

Brower Cookies are identified and read by "name-value" pairs = These tell Cookies where to be sent and what data to recall. The server only sends the Cookies when it wants the web browser to save it.

### (d) Purpose

Staffs are the backbone of any Company; their management plays a major role in deciding the success of the organization.

Our "QUEUE (our group name)" Understands this fact and therefore designed a Unique and 100% Functional Staff management System.

### Scope

In fixed scope, when you send us the enquiry we determine the scope of work for your project upfront and give you Pricing and timeline estimate upfront.

Once you send us the enquiry, we analyze the same and reply to you with our understanding of your requirement along with the queries and suggestions.

The unmodified "Waterfall mode". Progress flows from the top to the bottom, like a waterfall.

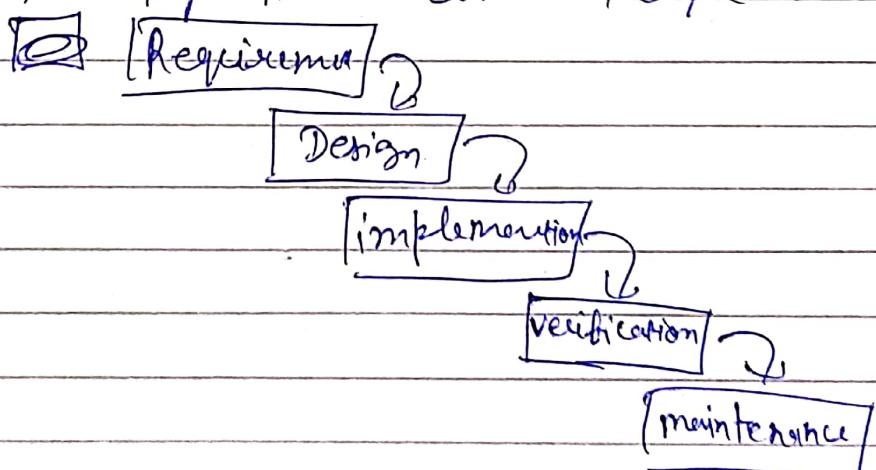


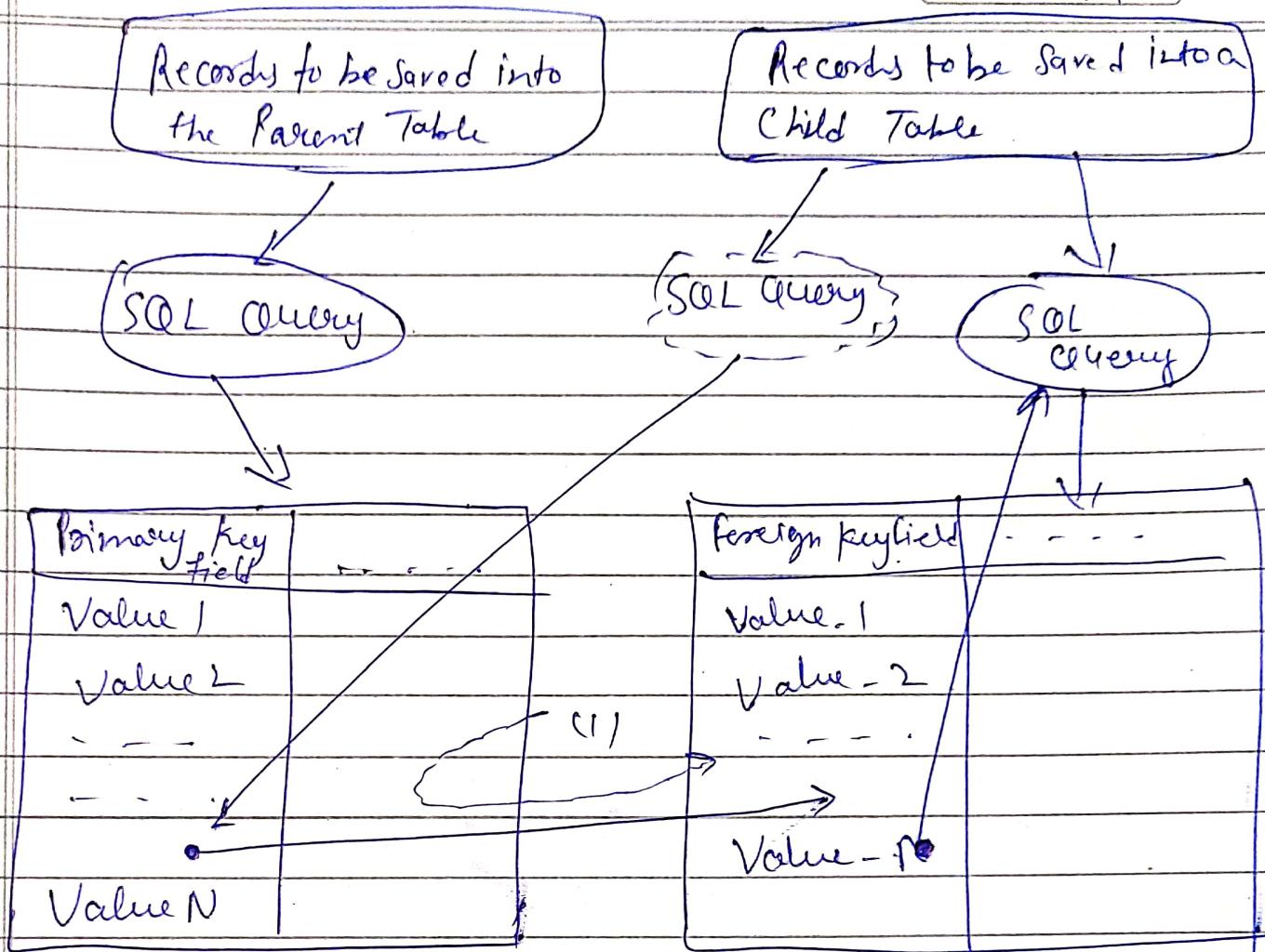
Table : Database

Staffmember ID	Name	Salary	Designation
1001	Mathesh	35000	Employee
1002	Kapil	30000	Employee
1003	Ravi	40500	Employee
1287	Mohan	41000	Employee
2343	Suresh	45000	Accountant

Saving data into the database.

This kind of operation upon the database is subdivided into two groups. Saving a new Staff's records and Add a record to an Staff's later records.

Saving new Staff's record. The whole process comprises a few actions, but not all of them are compulsory to be accomplished at once. First of all to unlock the fields in order to get them prepared for accepting new data the ("Add Staff") button has to be clicked.



Saving records Scheme