1(a) Create a html web page that contains a button with which it is possible to change the text that is shown on the screen. The file ButtonDemo.html has a JavaScript function named change\_text() which is called after the button is pressed. When the button is pressed repeatedly the text changes Hello! ... Well done! ... Hello! ... Well done! ... Hello! ..

<html>

<script>

function change\_text()

{

var text=document.createTextNode("Hello!...well done!...");

var p=document.createElement(p);

var temp=p.appendChild(text);

document.body.appendChild(temp);

}

</script>

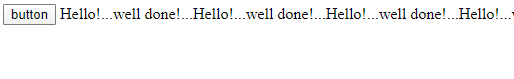
<body>

<input type="submit" value="button" onclick="change\_text()">

</body>

</html>

Output:



1(b) Modify the program so that the initial text shown on the screen is "Monday", and it will change in the following way when the button is

pressed repeatedly: Monday ... Tuesday ... Wednesday ... Thursday ... Friday ... Saturday ... Sunday ... Monday ... Tuesday ... etc. You should also change the button text so that there is written "Change day" on the button.

<html>

<head>

<script>

var p1=0;

function Change\_Days(){

var a=document.getElementById("m");

if(a.innerHTML==="Sunday"){

a.innerHTML="Monday";

}

else if(a.innerHTML==="Monday")

{

a.innerHTML="Tuesday";

}

else if(a.innerHTML==="Tuesday")

{

a.innerHTML="Wednesday";

}

else if(a.innerHTML==="Wednesday")

{

a.innerHTML="Thursday";

}

else if(a.innerHTML==="Thursday")

{

a.innerHTML="Friday";

}

else if(a.innerHTML==="Friday")

{

a.innerHTML="Saturday";

}

else

{

a.innerHTML="Sunday";

}

}

</script>

</head>

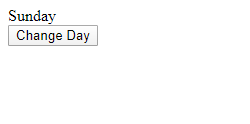
<body>

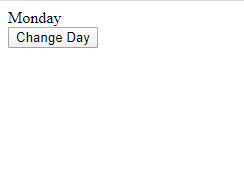
<div id="m">Sunday</div>

<input id="mybutton" type="button" value="Change Day" onclick="Change\_Days()">

</body>

</html>

3. 3. Using CSS properties create a calendar with following Calender Template 



3. Using CSS properties create a calendar with following Calender Template

<html>

<head>

<style>

\* {box-sizing: border-box;}

ul {list-style-type: none;}

body {font-family: Verdana, sans-serif;}

.month {

padding: 5px 25px;

width: 100%;

background:red;

text-align: center;

}

.month ul {

margin: 0;

padding: 0;

}

.month ul li {

color: white;

font-size: 20px;

text-transform: uppercase;

letter-spacing: 3px;

}

.month .prev {

float: left;

padding-top: 10px;

}

.month .next {

float: right;

padding-top: 10px;

}

.weekdays {

margin: 0;

padding: 10px 0;

background-color: #ddd;

}

.weekdays li {

display: inline-block;

width: 13.6%;

color: #666;

text-align: center;

}

.days {

padding: 10px 0;

background: #eee;

margin: 0;

}

.days li {

list-style-type: none;

display: inline-block;

width: 13.6%;

text-align: center;

margin-bottom: 5px;

font-size:12px;

color: #777;

}

.days li .active {

padding: 5px;

background: #FFC0CB;

color: white !important

}

/\* Add media queries for smaller screens \*/

@media screen and (max-width:720px) {

.weekdays li, .days li {width: 13.1%;}

}

@media screen and (max-width: 420px) {

.weekdays li, .days li {width: 12.5%;}

.days li .active {padding: 2px;}

}

@media screen and (max-width: 290px) {

.weekdays li, .days li {width: 12.2%;}

}

</style>

</head>

<body>

<div class="month">

<ul>

<li class="prev">&#10094;</li>

<li class="next">&#10095;</li>

<li>

MAY<br>

<span style="font-size:18px">2017</span>

</li>

</ul>

</div>

<ul class="weekdays">

<li>Mon</li>

<li>Tue</li>

<li>Wed</li>

<li>Thu</li>

<li>Fri</li>

<li>Sat</li>

<li>Sun</li>

</ul>

<ul class="days">

<li>1</li>

<li>2</li>

<li>3</li>

<li>4</li>

<li>5</li>

<li>6</li>

<li>7</li>

<li>8</li>

<li>9</li>

<li><span class="active">15</span></li>

<li>11</li>

<li>12</li>

<li>13</li>

<li>14</li>

<li>15</li>

<li>16</li>

<li>17</li>

<li>18</li>

<li>19</li>

<li>20</li>

<li>21</li>

<li>22</li>

<li>23</li>

<li>24</li>

<li>25</li>

<li>26</li>

<li>27</li>

<li>28</li>

<li>29</li>

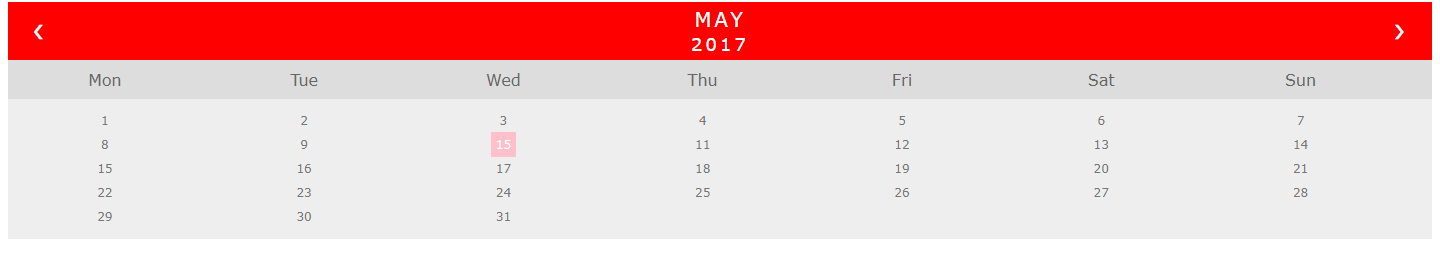
<li>30</li>

<li>31</li>

</ul>

</body>

</html>



4. Write the queries Based on following Loan Schema:

Create database loanschema;

Use loanschema;

CREATE TABLE customer\_master(

customer\_number VARCHAR(6),

firstname VARCHAR(30),

middlename VARCHAR(30),

lastname VARCHAR(30),

customer\_city VARCHAR(15),

customer\_contact\_no VARCHAR(10),

occupation VARCHAR(10),

customer\_date\_of\_birth DATE,

CONSTRAINT customer\_custid\_pk PRIMARY KEY (customer\_number));

CREATE TABLE branch\_master(

branch\_id VARCHAR(6),

branch\_name VARCHAR(30),

branch\_city VARCHAR(30),

CONSTRAINT branch\_bid\_pk PRIMARY KEY (branch\_id));

CREATE TABLE account\_master

(account\_number VARCHAR(255),

customer\_number VARCHAR(255),

branch\_id VARCHAR(255),

opening\_balance INT(20),

account\_opening\_date DATE,

account\_type VARCHAR(10),

account\_status VARCHAR(10),

PRIMARY KEY (account\_number),

FOREIGN KEY (customer\_number) references customer\_master(customer\_number),

FOREIGN KEY (branch\_id) references branch\_master(branch\_id));

CREATE TABLE transaction\_details(

transaction\_number VARCHAR(6),

account\_number VARCHAR(6),

date\_of\_transaction DATE,

medium\_of\_transaction VARCHAR(20),

transaction\_type VARCHAR(20),

transaction\_amount INT(7),

CONSTRAINT transaction\_details\_tnumber\_pk PRIMARY KEY (transaction\_number),

CONSTRAINT transaction\_details\_acnumber\_fk FOREIGN KEY (account\_number)

REFERENCES account\_master (account\_number));

CREATE TABLE loan\_details

(customer\_number varchar(255),

branch\_id varchar(255),

loan\_amount bigint(20),

foreign key(customer\_number) references customer\_master(customer\_number));

a. Write a query to display customer number, customer’s firstname , account number where the account status is terminated. Display the records sorted in ascending order based on customer number and then by account number.

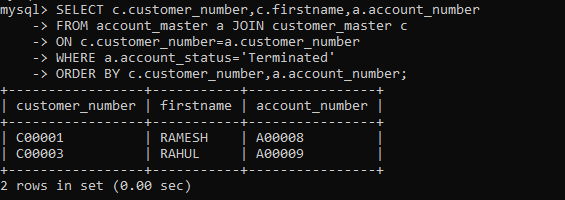
SELECT c.customer\_number,c.firstname,a.account\_number

FROM account\_master a JOIN customer\_master c

ON c.customer\_number=a.customer\_number

WHERE a.account\_status='Terminated'

ORDER BY c.customer\_number,a.account\_number;



b. Write a query to display the number of accounts opened in each city. The query should display the Branch city and the number of No\_of\_Accounts for the branch city where we don’t have any accounts opened display 0. Display the records in the sorted order based on branch city.

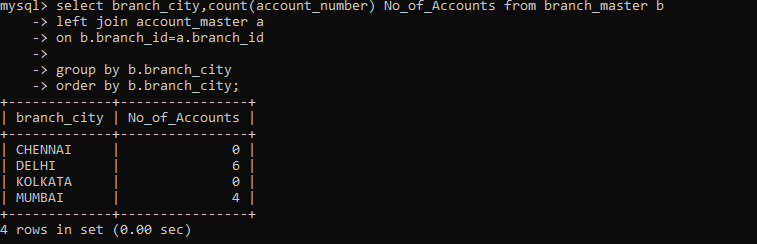
select branch\_city,count(account\_number) No\_of\_Accounts from branch\_master b

left join account\_master a

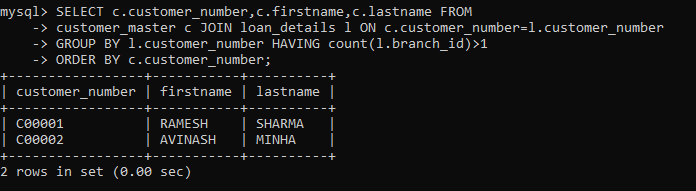
on b.branch\_id=a.branch\_id

group by b.branch\_city

order by b.branch\_city;

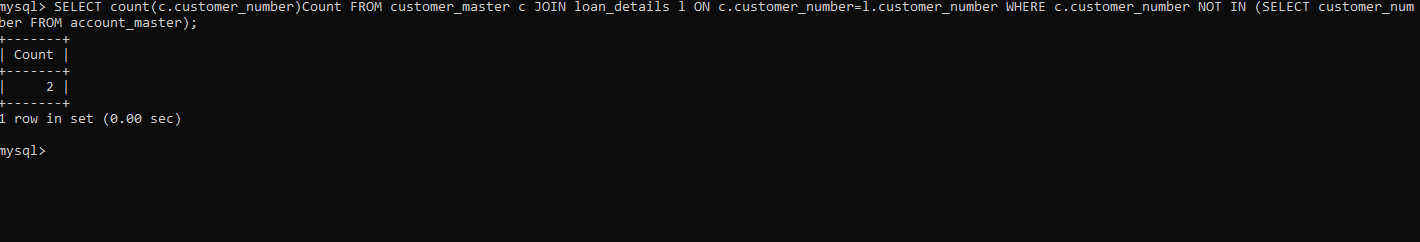


c. Write a query to display the customer number, customer firstname, customer lastname who has taken loan from more than 1 branch. Display the records sorted in order based on customer number.



d. Write a query to display the number of clients who have asked for loan but they don’t have any account in the bank though they are registered customers. Give the count an alias name of Count.

SELECT count(c.customer\_number)Count FROM customer\_master c JOIN loan\_details l ON c.customer\_number=l.customer\_number WHERE c.customer\_number NOT IN (SELECT customer\_number FROM account\_master);



e. Write a equerry to display customers firstname, city and account number whose occupation are not business , Services or Student. Display the records sorted in ascending order based on customer firstname and by account number.

SELECT c.firstname,c.customer\_city,a.account\_number FROM Customer\_master c JOIN account\_master a ON a.customer\_number=c.customer\_number WHERE c.occupation NOT IN ('Service','Student','Business') ORDER BY c.firstname,a.account\_number;

