BMS

INSTITUTE OF TECHNOLOGY AND MANAGEMENT

Yelahanka, Bengaluru - 560 064.

RECORD OF PRACTICAL WORK

NAME	Fiusha Kausar	UNIVERSITY SEAT NUMBER (USN)	1BY18SCS15
PROGRAMME	MTech	SEMESTER/ SECTION	I
COURSE CODE	18SCSL16	COURSE NAME	ADBMS & JDT

B.M.S. INSTITUTE OF TECHNOLOGY AND MANAGEMENT

Yelahanka, Bengaluru - 560 064

RECORD OF PRACTICAL WORK

NAME	FASIHA KAUSAR	UNIVERSITY SEAT NUMBER (USN)	1BY18SCS04
PROGRAMME	Mtech	SEMESTER/ SECTION	I
COURSE CODE	18SCSL16	COURSE NAME	ADBMS & IOT Laboratory

Record = 10

Open Ended Experiment = 10

Internal = 16

36

36/40

11/1/19

B.M.S. INSTITUTE OF TECHNOLOGY AND MANAGEMENT

Yelahanka, Bengaluru - 560 064

LABORATORY CERTIFICATE

This is to Certify that Mr./ Ms. FASIHA KAUSAR

has Satisfactorily completed the course of experiments in Practical

ADAMS and IOT LABORATORY Prescribed

by the Visvesvaraya Technological University for I

Semester M.tech [CSE] Course in the Laboratory of the college

in the year 2018 - 2019

Head of the Department 4/2/19

Staff incharge of the Batch 4/2/19

Date : 4/2/2019

Marks	
Maximum	Obtained
40	36

Name of the Candidate : Fasiha Kausar

Roll No : 4 USN : 1BY18SCS04

Fasiha

Signature of the Candidate

Particulars of the Experiments Performed

CONTENTS

40

Type Created

Table Created

1 row created

Name	Column_Value
Customer A	PO1
Customer A	PO2
Customer A	PO3
Customer A	PO4

Date 6/11/2018

Expt. No. 1 Page No. 1

1. Develop a database application to demonstrate the representation of multi-valued attributes and the use of nested tables to represent complex objects. Write suitable queries to demonstrate these use Consider Purchase order Example: This example is based on a typical business activity: managing customer orders. Need to demonstrate how the application might evolve from relational to object relational and how you could write it from scratch using a pure object oriented approach.

a. Show how to implement the schema implementing the application under the Relational model using only Oracle built in data types. Build an object oriented application on top of this relational schema using object views.

Multivalued Attribute Representation

```
create type PURCHASE_ORDERS as varray(5) of varchar2(25);
/
create table CUSTOMER ( Name varchar2(25),
                        Orders PURCHASE_ORDERS,
                        constraint Customer_PK primary key (Name));
/
insert into Customer values ('Customer A', PURCHASE_ORDERS ('PO1', 'PO2', 'PO3',
                            'PO4'));

select C.Name, PO.* From Customer C, Table (C.Orders) PO;
```

Teacher's Signature : _____

Type Created

Type Created

Table Created

Name	PO	OrderDate	ShipDate
A	1	1-DEC-16	1-JAN-17
A	2	1-DEC-17	1-JAN-18
A	3	1-DEC-18	1-JAN-19

Name	PO	OrderDate	ShipDate
A	1	1-NOV-16	1-JAN-17
A	2	1-DEC-17	1-JAN-18
A	3	1-DEC-18	1-JAN-18

Name	PO	OrderDate	ShipDate
A	2	1-DEC-17	1-JAN-18
A	3	1-DEC-18	1-JAN-19

Date 4/11/2018

Expt. No. 1 Page No. 2

Nested Table Representation

```
create type PURCHASE_ORDER_TYPE as object (
        PO varchar2(25),
        orderdate DATE,
        shipdate DATE );

create type PURCHASE_ORDERS1 as table of PURCHASE_ORDER_TYPE;
/
Create table CUSTOMER1(Name Varchar(25),
        Orders PURCHASE_ORDERS)
nested table ORDERS store as ORDERS TAB;
/
insert into CUSTOMER values ('A', PURCHASE_ORDERS
(PURCHASE_ORDER_TYPE ('1', '1-DEC-16', '1-JAN-17'),
PURCHASE_ORDER_TYPE ('2', '1-DEC-17', '1-JAN-18'),
PURCHASE_ORDER_TYPE ('3', '1-DEC-18', '1-JAN-19')) );

select Name, PO, OrderDate, ShipDate from CUSTOMER table(
Customer.Orders);

update TABLE (select orders from CUSTOMER where Name='A')
set OrderDate = '01-NOV-16' where PO='1';
```

Teacher's Signature : _____

Date 4/11/2018

Expt. No. 1 Page No. 3

delete TABLE (select orders from CUSTOMER where Name='A')
where PO='1';

28/12/19

E=08
V=02
10

10/10

Teacher's Signature : _____

Table Created

Rows Updated

Date 13/11/2018

Expt. No. 2 Page No. 4

2 Design and develop a suitable student database application by considering appropriate attributes. Couples of attributes to be maintained is the attendance of a student in each subject for which he/she has enrolled and Internal assessment using TRIGGERS, write active rules to do the following

a. Whenever the attendance is updated, check if the attendance is less than 85%, if so, notify the Head of the Department concerned.

b. Whenever the marks in an Internal assessment Test are entered, check if the mark are less than 40%, if so notify the Head of the department concerned.

```
create table STUDENTS(
regno varchar(10) primary key,
name varchar2(10) not null,
major varchar2(10),
mark number(4),
attendance number(4));

insert into students values ('001', 'pinky', 'cse', 85, 70);
insert into students values('002', 'john', 'cse', 30, 70);
insert into students values('003', 'arun', 'cse', 90, 95);

create TRIGGER trig_att_mark after update on
STUDENTS for each row
begin
```

Teacher's Signature :

TRIGGER Created

Pinky bearing REG NO. 001 has attendance less than 85
1 row updated.

Select * from Students [after updating]

REG.NO	Name	Major	Mark	Attendance
001	Pinky	CSE	85	25
002	John	CSE	30	70
003	Arun	CSE	90	95

Date .13/11/2018

Expt. No. ..... 2 .....

Page No. ..... 5 .....

```
if (:new.attendance <85) then  dbms_output.put_line (':new.name ||
' bearing  REG.NO.' || :new.regno || ' has attendance less than 85');
end if;

if (:new.mark <40) then dbms_output.line (:new.name || ' bearing
REG.NO.' || :new.regno || 'got mark less than 40').
end if;
end;
/

set serveroutput on
update students set attendance = 25 where regno = '001';
```

C = 08
V = 02
10

10/10

Teacher's Signature : _____

Date 23/12/2018

Expt. No. 3 Page No. 6

3. Develop a database application to demonstrate storing and retrieving of BLOB and CLOB objects.

a. Write a binary large object (BLOB) to a database as either binary or character (CLOB) data, depending on the type of the field in your data source. To write a BLOB value to the database, use the appropriate INSERT or UPDATE statement and pass the BLOB value as an input parameter. If your BLOB is stored as text such as a SQL server text field, pass the BLOB as a string parameter. If the BLOB is stored in binary format such as a SQL Server image field, pass an array of type byte as a binary parameter.

b. Once storing of BLOB and CLOB objects is done, retrieve them and display the results accordingly.

db.txt

```
create database CSE;
use database CSE;
create table blobz (
id int auto_increment not null,
title varchar(50),
type varchar(50),
data longblob,
primary key(id),
unique id(id)
);
```

Teacher's Signature : _____

Date .....

Expt. No. 3 Page No. 7

dbb.php

```
<?php
$conn = mysqli_connect('localhost', 'root', '', 'cse');
if(!$conn){
        die("Could not connect to database");
    }
// $db = mysqli_select_db("cse", $conn);
// if(!$db) {
    //     die("Could not Connect to database $db database");
    // }
$query = "SELECT id, title, type FROM blobs ORDER BY title asc";
$result = mysqli_query($conn, $query);
while($row = mysqli_fetch_array($result))
{
    echo "<a href = show.php?id=".$row['id'].">".$row['id']."
        </a>", $row['title'], "<br/>";
}
echo "<a href = new.html>NEW</a> <br/>";
?>
```

Teacher's Signature : _____

Date .....

Expt. No. 3 Page No. 8

New html

```
<html>
<head>
<title> UPLOAD FILE AS BLOB </title>
</head>
<form enctype = "multipart/form-data" action = "new.php" method = "post">
<P>
  TITLE: <input type = "text" name = "title" size = "30" maxlength = "30">
UPLOAD FILE : <input type = "file" name = "sys" size = "30" >
</P>
  input type "submit" value: "upload" name: "cmd submit"></br>
</form>
</body>
</html>
```

Teacher's Signature : .....

Date .....

Expt. No. 3 Page No. 9

New.php

```
$name = ($_FILES['file']['name']);
$size = ($_FILES['file']['size']);
$type = ($_FILES['file']['type']);
$file = ($_FILES['file']['tmp_name']);
$title = ($_POST['title']);

if (empty($title) || $file == "none"){
        die("you should have both title and file.");
}
//$conn= mysqli_connect("localhost", "root", "");
$conn = new mysqli("localhost", "root", "", "cse");
if (!$conn){
    die ("my sql connection failed.");
}
// $db= mysqli_select_db("cse", $conn);
//if (!$db){
        //die ("mysqli selection failed");
//}
$handle = fopen ($file, "r");
$content = fread ($handle, $size);
$content = addslashes ($content, $size);
$query = "INSERT INTO blobs values(0, '$title', '$type', '$content')";
$result = mysqli_query ($conn, $query);
header ("location: http://localhost/blob.php");
exit;
?>
```

Teacher's Signature : _____

'ocalhost/new.html

TITLE [ ] UPLOAD FILE : choose file No file choosen

upload

test.txt

Hi, Hello BMSIT

//new.html

TITLE [Test] upload file : test.txt

upload

local host / blob.php

1. Test

NEW

local host / show.php

text

type text

data: Hi, Hello BMSIT

Date .....

Expt. No. 3 Page No. 10

Show.php

```
<?php
$id= $_GET["id"];
if(!is_numeric($id)){
    die("Invalid blob ID specified");
}
$conn= mysqli_connect("localhost","root","","cse");
if(!$conn){
    die("mysqli_connect failed");
//$db= mysqli_select_db("cse",$conn);
//if(!$db){
    die("mysqli_select_db failed");
//}
$query= "SELECT type, data FROM blobs WHERE id= $id";
$result= mysqli_query($conn, $query) or die("mysqli_query failed");
while($row= mysqli_fetch_array($result MYSQLI_ASSOC)){
  echo "type: ".$row['type']."<br>", "data: ".$row['data']."<br>";
}
//if(mysqli_num_rows($result)==1){
//$type= @mysqli_result($result, 0, "type");
//$data= @mysqli_result($result, 0, "data");
//header("Content-type: $type");
//echo $data;
//}
//else{
    //echo "Record doubt exist.") //}
    echo "Successfull";
?>
```

10/10

16/1/19

E=08
V=02
10

Teacher's Signature : _____

Date .....

Expt. No. 4 .....

Page No. 11 .....

4. Design, develop and execute a program to implement specific Apriori Algorithm for mining association rules. Run the program against any large database available in the public domain and discuss the results.

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

import weka.associations.Apriori;
import weka.core.Instances;

public class Weka_Apriori {
    public static void main (String[] args){
        Instances data = null;
        try {
            BufferReader reader = new BufferedReader (new FileReader
                ("C:\\Program files\\Weka-3-8\\data\\supermarket.arff"))
            data = new Instances (reader);
            reader.close();
            data.setClassIndex(data.numAttributes()-1);
        }
        catch (IOException e){
            e.printStackTrace();
        }
        double deltaValue = 0.05;
        double lowerBoundMinSupportValue = 0.1;
        double minMetricValue = 0.5;
```

Teacher's Signature : _____

```
Apriori
=======

Minimum support: 0.4 (1851 instances)
Minimum metric <confidence>: 0.5
Number of cycles performed: 12

Generated sets of large itemsets:

Size of set of large itemsets L(1): 18

Size of set of large itemsets L(2): 16

Best rules found:

 1. biscuits=t 2685 ==> bread and cake=t 2083    <conf:(0.8)> lift:(1.11)
lev:(0.04) [288] conv:(1.4)
 2. milk-cream=t 2939 ==> bread and cake=t 2337    <conf:(0.8)> lift:(1.1)
lev:(0.05) [221] conv:(1.37)
 3. fruit=t 2962 ==> bread and cake=t 2325    <conf:(0.78)> lift:(1.09)
lev:(0.04) [193] conv:(1.3)
 4. baking needs=t 2795 ==> bread and cake=t 2191    <conf:(0.78)> lift:(1.09)
lev:(0.04) [179] conv:(1.29)
 5. frozen foods=t 2717 ==> bread and cake=t 2129    <conf:(0.78)> lift:(1.09)
lev:(0.04) [173] conv:(1.29)
 6. vegetables=t 2961 ==> bread and cake=t 2298    <conf:(0.78)> lift:(1.08)
lev:(0.04) [167] conv:(1.25)
 7. juice-sat-cord-ms=t 2463 ==> bread and cake=t 1869    <conf:(0.76)>
lift:(1.05) lev:(0.02) [96] conv:(1.16)
 8. vegetables=t 2961 ==> fruit=t 2207    <conf:(0.75)> lift:(1.16) lev:(0.07)
[311] conv:(1.41)
 9. fruit=t 2962 ==> vegetables=t 2207    <conf:(0.75)> lift:(1.16) lev:(0.07)
[311] conv:(1.41)
10. bread and cake=t 3330 ==> milk-cream=t 2337    <conf:(0.7)> lift:(1.1)
lev:(0.05) [221] conv:(1.22)
11. bread and cake=t 3330 ==> fruit=t 2325    <conf:(0.7)> lift:(1.09)
lev:(0.04) [193] conv:(1.19)
12. baking needs=t 2795 ==> vegetables=t 1949    <conf:(0.7)> lift:(1.09)
lev:(0.03) [160] conv:(1.19)
13. milk-cream=t 2939 ==> fruit=t 2038    <conf:(0.69)> lift:(1.08) lev:(0.03)
[156] conv:(1.17)
14. frozen foods=t 2717 ==> vegetables=t 1882    <conf:(0.69)> lift:(1.08)
lev:(0.03) [143] conv:(1.17)
15. bread and cake=t 3330 ==> vegetables=t 2298    <conf:(0.69)> lift:(1.08)
lev:(0.04) [167] conv:(1.16)
16. milk-cream=t 2939 ==> vegetables=t 2025    <conf:(0.69)> lift:(1.08)
lev:(0.03) [144] conv:(1.16)
17. fruit=t 2962 ==> milk-cream=t 2038    <conf:(0.69)> lift:(1.08) lev:(0.03)
[156] conv:(1.17)
18. frozen foods=t 2717 ==> fruit=t 1861    <conf:(0.68)> lift:(1.07) lev:(0.03)
[121] conv:(1.14)
```


Date .....

Expt. No. ..... Page No. ...12.....

```
int numRulesValue = 20;
double upperBoundMinSupportValue = 1.0,
String resultApriori;
Apriori apriori = new Apriori();
apriori.setDelta(deltaValue);
apriori.setLowerBoundMinSupport(lowerBoundMinSupportValue);
apriori.setNumRules(numRulesValue);
apriori.setUpperBoundMinSupport(upperBoundMinSupportValue);
apriori.setMinMetric(minMetricValue);
try{
    apriori.buildAssociations(data);
}catch(Exception e){
    e.printStackTrace();
}
resultApriori = apriori.toString();
System.out.println(resultApriori);
}
```

Teacher's Signature : _____

Output Serial Monitor

Hello World
Hello World
Hello World

Date 5/11/18

Expt. No. 1

Page No. 14

Transmit a string using UART

```
void setup(){
  //Initialize serial communication at 9600 bits per second,
  Serial.begin(9600);
}
// The loop routine runs over and over again forever;
void loop(){
  Serial.println("Hello World");
  delay(1000); // delay in between reads for stability
}
```

Teacher's Signature :

Date .. 14/11/18

Expt. No. ..... 2 .....

Page No. ....15.....

Peer to Peer Communication of two Nodes over the Radio Frequency

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
const char* ssid = "mtech_ap";
const char* password = "";

ESP8266WebServer server(80);
void handleRoot(){
    String s = "Hello World";
    server.send(200, "text/html", s);
}

void setup(){
  Serial.begin(9600);
  serial println("");
  WiFi.mode(WIFI_AP);
  WiFi.softAP(ssid, password);

  IPAddress myIP = WiFi.softAPIP();
  Serial.print("HotSpot IP:");
  Serial.println(myIP);

  server.on("/", handleRoot);

  server.begin();
  Serial.println("HTTP server started");
}
```

Teacher's Signature : _____

Output

http://192.168.4.1

Hello World

hotspot IP : 192.168.4.1

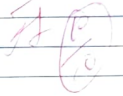
HTTP Server started

Date : 14/12/18

Expt. No. 2

Page No. 15

```
void loop()
  server.handleClient();
}
```



Teacher's Signature : _____

Date 21/11/18

Expt. No. 3 Page No. 17

Multi-point to single point communication of Motor over the radio frequency. CAN (Submitting)

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

const char * ssid = "EMSIT";
const char * password = "";

ESP8266WebServer server(80);
void handleRoot(){
  String s = "Hello World";
  server.send(200, "text/html", s);
}
void setup(){
  Serial.begin(9600); WiFi.mode(WIFI_STA); WiFi.disconnect();
  WiFi.begin(ssid, password);
  Serial.println("");

  While (WiFi.status() != WL_CONNECTED){
    delay(500);
    Serial.print(".");

  Serial.println("");
  Serial.print("Connected to");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
```

Teacher's Signature :

Output

Connected to BNSIT
IP Address 192.168.43.116
HTTP server started

From different mobile connect to IP: 192.168.43.116

Hello World

Date 9/12/18

Expt. No. 3

Page No. 13

```
server.on("/", handleRoot);
server.begin();
Serial.println("HTTP server started");
}
void loop(){
server.handleClient();
}
```

Teacher's Signature :

Output Serial Monitor

Humidity: 57.68 Temperature: 26.37
Humidity: 57.57 Temperature: 26.37
Humidity: 57.50 Temperature: 25.36

Date 11.11.19

Expt. No. 4 Page No. 19

I2C protocol study, Reading temperature and Relative Humidity value from the sensor.

```
#include "Adafruit_Si7021.h"
Adafruit_Si7021 sensor = Adafruit_Si7021();

Void setup(){
  Serial.begin(9600);
  Serial.println("Si7021 test!");
  sensor.begin();
}

Void loop(){
  Serial.println("Humidity : ");
  Serial.println(sensor.readHumidity(),2);
  Serial.print("Temperature: ");
  Serial.println(sensor.readTemperature(),2);
  delay(1000);
}
```

10/10

Background knowledge = 2
Select Appropriate Equipment = 2
Design = 2M
Logic construction = 2M
Documentation = 2M

10

Teacher's Signature : 11/11/19