

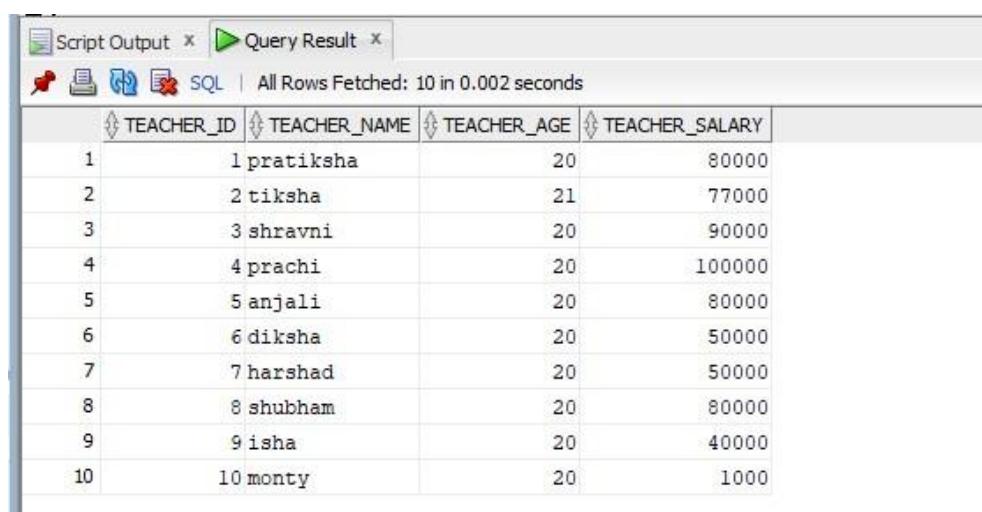
Experiment no – 1

1. Practice DDL & DML commands execute basic utilities used to interact with Oracle DBMS /mysql

Program:-

```
CREATE table person(person_id int primary key,FirstName varchar(50),LastName varchar(50));  
ALTER table person ADD Email varchar(60); ALTER table person  
ADD PhoneNumber int; select * from person;  
ALTER TABLE person DROP COLUMN PhoneNumber; ALTER TABLE person  
MODIFY Lastname varchar(150); rename person to staff;  
ALTER TABLE STAFF DROP COLUMN Email; SELECT * FROM  
staff; truncate table staff;  
SELECT * FROM staff;create table Teacher(teacher_id int primary key,teacher_name varchar(50),teacher_age  
int,teacher_salary int);  
INSERT into Teacher values(1,'pratiksha',20,80000);  
INSERT into Teacher values(2,'tiksha',21,77000);  
INSERT into Teacher values(3,'shravni',20,90000);  
INSERT into Teacher values(4,'prachi',20,100000);  
INSERT into Teacher values(5,'anjali',20,80000);  
INSERT into Teacher values(6,'diksha',20,50000);  
INSERT into Teacher values(7,'harshad',20,50000);  
INSERT into Teacher values(8,'shubham',20,80000);  
INSERT into Teacher values(9,'isha',20,40000);  
INSERT into Teacher values(10,'monty',20,1000);  
SELECT * FROM Teacher;
```

Output:-



TEACHER_ID	TEACHER_NAME	TEACHER_AGE	TEACHER_SALARY
1	1 pratiksha	20	80000
2	2 tiksha	21	77000
3	3 shravni	20	90000
4	4 prachi	20	100000
5	5 anjali	20	80000
6	6 diksha	20	50000
7	7 harshad	20	50000
8	8 shubham	20	80000
9	9 isha	20	40000
10	10 monty	20	1000

Experiment No. :-2

Title :- Design and Implement the Fragmentation schema and the Replication scheme for the social networking websites /online e-shopping/e-learning websites .

SQL Queries :-

```
CREATE TABLE BOB (Acc_ID int,Acc_hold varchar(100),Branch varchar(100),Mob_no INT,DOP_acc DATE,Balance int);

CREATE TABLE Alice (Acc_ID int,Acc_hold varchar(100),Branch varchar(100),Mob_no INT,DOP_acc DATE,Balance int);

insert INTO Alice VALUES (1,'shubham','kolhapur',98453,'1-5-2023',5000);

insert INTO Alice VALUES (2,'isha','kundal',98843,'1-5-2023',4000);

insert INTO Alice VALUES (3,'vishal','kolhapur',98940,'1-5-2023',15000);

insert INTO Alice VALUES (4,'shinchan','kavathe',98745,'1-5-2023',54000);

insert INTO Alice VALUES (5,'soha','kur',98738,'1-5-2023',50000);

insert INTO Alice VALUES (6,'soham','tamgoan',98023,'1-5-2023',25000);

insert INTO Alice VALUES (7,'harshd','kolhapur',95283,'1-5-2023',50745);

insert INTO Alice VALUES (8,'anjali','kundal',90213,'1-5-2023',94500);

insert INTO Alice VALUES (9,'shravni','sangali',98323,'1-5-2023',74300);

insert INTO Alice VALUES (10,'Rn','majgoan',983453,'1-5-2023',6500);

select *from Alice;

create table Alice2 as select Acc_ID,Acc_hold,Branch,Mob_no from BOB ;

select *from Alice2;

create table Alice3 as select Acc_ID,Acc_hold,Branch from BOB ;

select *from Alice3;

create table Alice4 as select*from Alice where Branch='kolhapur';

select *from Alice4;

create table Alice5 as select*from Alice where Balance>=15000;

select *from Alice5;
```

Output :-

ACC_ID	ACC_HOLD	BRANCH	MOB_NO	DOP_ACC	BALANCE
1	1pratiksha	kolhapur	98453	01-05-23	5000
2	2isha	kundal	98843	01-05-23	4000
3	3ayusha	kolhapur	98940	01-05-23	15000
4	4shinchan	kavathe	98745	01-05-23	54000
5	5soha	kur	98738	01-05-23	50000
6	6prachi	tamgoan	98023	01-05-23	25000
7	7harshd	kolhapur	95283	01-05-23	50745
8	8anjali	kundal	90213	01-05-23	94500
9	9shravni	sangali	98323	01-05-23	74300
10	10Rn	majgoan	983453	01-05-23	6500

1. Vertical Fragmentation :-

Query Result x | Query Result 1 x | Query Result 2 x | Query Result 3 x | Query Result 4 x
SQL | All Rows Fetched: 40 in 0.004 seconds

ACC_ID	ACC_HOLD	BRANCH	MOB_NO
1	1 johny	kolhapur	98453
2	2 piyush	kundal	98843
3	3 aishu	kolhapur	98940
4	4 mayuri	kavathe	98745
5	5 sanika	kur	98738
6	6 prachi	tamgoan	98023
7	7 harshda	kolhapur	95283
8	8 omkya	kundal	90213
9	9 jayesh	sangali	98323
10	10 omkar	majgoan	983453

Query Result x | Query Result 1 x | Query Result 2 x | Query Result 3 x | Query Result 4 x
SQL | All Rows Fetched: 40 in 0.006 seconds

ACC_ID	ACC_HOLD	BRANCH
1	1 johny	kolhapur
2	2 piyush	kundal
3	3 aishu	kolhapur
4	4 mayuri	kavathe
5	5 sanika	kur
6	6 prachi	tamgoan
7	7 harshda	kolhapur
8	8 omkya	kundal
9	9 jayesh	sangali
10	10 omkar	majgoan

2. Horizontal Fragmentation :-

Query Result x | Query Result 1 x | Query Result 2 x | Query Result 3 x | Query Result 4 x
SQL | All Rows Fetched: 12 in 0.006 seconds

ACC_ID	ACC_HOLD	BRANCH	MOB_NO	DOP_ACC	BALANCE
1	1 johny	kolhapur	98453	01-05-23	5000
2	3 aishu	kolhapur	98940	01-05-23	15000
3	7 harshda	kolhapur	95283	01-05-23	50745

Query Result x | Query Result 1 x | Query Result 2 x | Query Result 3 x | Query Result 4 x
SQL | All Rows Fetched: 12 in 0.006 seconds

ACC_ID	ACC_HOLD	BRANCH	MOB_NO	DOP_ACC	BALANCE
1	1 johny	kolhapur	98453	01-05-23	5000
2	3 aishu	kolhapur	98940	01-05-23	15000
3	7 harshda	kolhapur	95283	01-05-23	50745
4	1 johny	kolhapur	98453	01-05-23	5000
5	3 aishu	kolhapur	98940	01-05-23	15000
6	7 harshda	kolhapur	95283	01-05-23	50745
7	1 pratiksha	kolhapur	98453	01-05-23	5000
8	3 ayusha	kolhapur	98940	01-05-23	15000
9	7 harshd	kolhapur	95283	01-05-23	50745
10	1 pratiksha	kolhapur	98453	01-05-23	5000

Experiment 3

Title: Implementation of 2 Phase Commit protocol for distributed databases.

Server:

```
//Two Phase Commit Protocol SERVER package twophase;

import java.io.*; import java.net.*; import
java.util.*;

public class Server {
    boolean closed = false, inputFromAll = false; List<ClientThread>
    thread;
    List<String> data;
    Server() {
        thread = new ArrayList<ClientThread>(); data = new
        ArrayList<String>();
    }

    public static void main(String args[]) { Socket clientSocket
        = null; ServerSocket serverSocket = null; int
        port_number = 1111;
        Server server = new Server(); try {
            serverSocket = new ServerSocket(port_number);
        } catch (IOException e) { System.out.println(e);
        }

        while (!server.closed) { try {
            clientSocket = serverSocket.accept();
            ClientThread clientThread = new ClientThread(server, clientSocket); (server.thread).add(clientThread);
            System.out.println("\nNow Total clients are : " + (server.thread).size()); (server.data).add("NOT_SENT");
            clientThread.start();
        } catch (IOException e) {
        }
    }
    try {
        serverSocket.close();
    } catch (Exception e1) {
    }
}
}

class ClientThread extends Thread { DataInputStream is = null;
String line;
String destClient = ""; String name;
PrintStream os = null; Socket clientSocket =
null; String clientIdentity; Server server;

public ClientThread(Server server, Socket clientSocket) { this.clientSocket =
    clientSocket;
    this.server = server;
}

@SuppressWarnings("deprecation") public void run() {
try {
    is = new DataInputStream(clientSocket.getInputStream()); os = new
    PrintStream(clientSocket.getOutputStream()); os.println("Enter your name.");
    name = is.readLine(); clientIdentity = name;
    os.println("Welcome " + name + " to this 2 Phase Application.\nYou will receive a vote Request now..."); os.println("VOTE_REQUEST\nPlease enter COMMIT or ABORT to proceed : "); for (int i = 0; i <
    (server.thread).size(); i++) {
```

```

        if ((server.thread).get(i) != this) {
            ((server.thread).get(i)).os.println("---A new user " + name + " entered the Application---");
        }
    }

    while (true) {
        line = is.readLine();
        if (line.equalsIgnoreCase("ABORT")) { System.out.println("\nFrom " +
            clientIdentity
            + " : ABORT\n\nSince aborted we will not wait for inputs from other clients.");
            System.out.println("\nAborted.... ");

        for (int i = 0; i < (server.thread).size(); i++) {
            ((server.thread).get(i)).os.println("GLOBAL_ABORT"); ((server.thread).get(i)).os.close();
            ((server.thread).get(i)).is.close();
        }
        break;
    }

    if (line.equalsIgnoreCase("COMMIT")) { System.out.println("\nFrom " + clientIdentity
        + " : COMMIT"); if ((server.thread).contains(this)) {
            (server.data).set((server.thread).indexOf(this), "COMMIT"); for (int j = 0; j <
            (server.data).size(); j++) {
                if (!((server.data).get(j)).equalsIgnoreCase("NOT_SENT")) { server.inputFromAll = true;
                    continue;
                } else {
                    server.inputFromAll = false;
                    System.out.println("\nWaiting for inputs from other clients."); break;
                }
            }
        }

        if (server.inputFromAll) { System.out.println("\n\nCommitted
            ..... ");
            for (int i = 0; i < (server.thread).size(); i++) {
                ((server.thread).get(i)).os.println("GLOBAL_COMMIT"); ((server.thread).get(i)).os.close();
                ((server.thread).get(i)).is.close();
            }
            break;
        }

        } // if thread.contains
    } // commit
} // while server.closed = true;
clientSocket.close();
} catch (IOException e) {
}
}

} // end class thread

/*
*
*
* Coordinator Cohorts
* QUERY TO COMMIT
* ----->-----
* VOTE YES/NO prepare/abort
* < -----
* commit/abort COMMIT/ROLLBACK
* ----->-----
* ACKNOWLEDGMENT commit/abort

```

```

* < -----
* end
*
*
* Two Phases :
*
* 1.Prepare and Vote Phase
* 2. Commit or Abort Phase
*
* "Either All Commit Or All RollBack."
*
*
*/

```

```

Now Total clients are : 1
Now Total clients are : 2
Now Total clients are : 3
| From 'Client 1' : COMMIT
Waiting for inputs from other clients.
From 'Client 2' : COMMIT
Waiting for inputs from other clients.
From 'client 3' : ABORT  []
Since aborted we will not wait for inputs from other clients.
Aborted....

```

Client.java

```

//Two Phase Commit Protocol CLIENT package twophase;

import java.io.*; import java.net.*;

public class Client implements Runnable { static Socket
clientSocket = null;
static PrintStream os = null; static DataInputStream is
= null;
static BufferedReader inputLine = null; static boolean closed =
false;

public static void main(String[] args) { int port_number =
1111;
String host = "localhost"; try {
clientSocket = new Socket(host, port_number);
inputLine = new BufferedReader(new InputStreamReader(System.in)); os = new
PrintStream(clientSocket.getOutputStream());
is = new DataInputStream(clientSocket.getInputStream());
} catch (Exception e) {
System.out.println("Exception occurred : " + e.getMessage());
}
}

if (clientSocket != null && os != null && is != null) { try {
new Thread(new Client()).start(); while (!closed) {
os.println(inputLine.readLine());
}
os.close();
is.close(); clientSocket.close();
} catch (IOException e) { System.err.println("IOException: " + e);
}
}

```

```
    }
}
}

@SuppressWarnings("deprecation") public void run() {
    String responseLine; try {
        while ((responseLine = is.readLine()) != null) { System.out.println("\n" +
            responseLine);
            if (responseLine.equalsIgnoreCase("GLOBAL_COMMIT") == true
                || responseLine.equalsIgnoreCase("GLOBAL_ABORT") == true) { break;
        }
    }
    closed = true;
} catch (IOException e) { System.err.println("IOException: " + e);
}
}
} // end client
```

```
Enter your name.
client 3

Welcome client 3 to this 2 Phase Application.

You will receive a vote Request now...

VOTE_REQUEST

Please enter COMMIT or ABORT to proceed :
ABORT

GLOBAL_ABORT
```

Experiment No – 4

Partition by range:

```
create table product_range(p_id number,p_name varchar(30),p_quantity number,p_price number) partition by range(p_price)
(partition p1 values less than(20001), partition p2 values less
than(50001), partition p3 values less than(100000)
);
```

```
insert into product_range values(1,'Computer',50,50000); insert into
product_range values(2,'Laptop',40,45000); insert into product_range
values(3,'Mobile',30,15000); insert into product_range
values(4,'Printer',20,70000); insert into product_range
values(5,'Tv',10,85000); insert into product_range
values(6,'Fridge',30,18000); insert into product_range
values(7,'Oven',15,35000);
```

```
select * from product_range;
```

P_ID	P_NAME	P_QUANTITY	P_PRICE
1	3 Mobile	30	15000
2	6 Fridge	30	18000
3	1 Computer	50	50000
4	2 Laptop	40	45000
5	7 Oven	15	35000
6	4 Printer	20	70000
7	5 Tv	10	85000

```
select * from product_range partition(p1);
```

P_ID	P_NAME	P_QUANTITY	P_PRICE
1	3 Mobile	30	15000
2	6 Fridge	30	18000

```
select * from product_range partition(p2);
```

P_ID	P_NAME	P_QUANTITY	P_PRICE
1	1 Computer	50	50000
2	2 Laptop	40	45000
3	7 Oven	15	35000

```
select * from product_range partition(p3);
```

	P_ID	P_NAME	P_QUANTITY	P_PRICE
1	4	Printer	20	70000
2	5	Tv	10	85000

Partition by list:

```
create table student_list(Roll_no number,Name varchar(30),city varchar(20))
```

```
partition by list(city) (partition l1
```

```
values('Kolhapur'), partition l2 values('Pune'),
```

```
partition l3 values('Mumbai'));
```

```
insert into student_list values(1,'Diksha','Pune'); insert into student_list
```

```
values(2,'sushmita','Kolhapur'); insert into student_list
```

```
values(3,'Poonam','Kolhapur'); insert into student_list
```

```
values(4,'Shruti','Mumbai'); insert into student_list
```

```
values(5,'Vaishnavi','Kolhapur'); insert into student_list
```

```
values(6,'Shivani','Pune'); insert into student_list
```

```
values(7,'Sai','Mumbai'); insert into student_list
```

```
values(8,'Shradhha','Pune'); select * from student_list;
```

	ROLL_NO	NAME	CITY
1	2	sushmita	Kolhapur
2	3	Poonam	Kolhapur
3	5	Vaishnavi	Kolhapur
4	1	Diksha	Pune
5	6	Shivani	Pune
6	8	Shradhha	Pune
7	4	Shruti	Mumbai
8	7	Sai	Mumbai

```
select * from student_list partition(l1);
```

	ROLL_NO	NAME	CITY
1	2	sushmita	Kolhapur
2	3	Poonam	Kolhapur
3	5	Vaishnavi	Kolhapur

```
select * from student_list partition(l2);
```

	ROLL_NO	NAME	CITY
1	1	Diksha	Pune
2	6	Shivani	Pune
3	8	Shradhha	Pune

```
select * from student_list partition(l3);
```

	ROLL_NO	NAME	CITY
1	4	Shruti	Mumbai
2	7	Sai	Mumbai

Partition by hash:

```
create table emp_hash(emp_id number,name varchar(20)) partition by
hash(emp_id) partitions 4; insert into emp_hash values(10,'Vikas'); insert
into emp_hash values(20,'Rohan'); insert into emp_hash
values(30,'Shubham'); insert into emp_hash values(40,'Omkar'); insert into
emp_hash values(50,'Sanskar'); insert into emp_hash values(60,'Nitin');
insert into emp_hash values(70,'Jay'); insert into emp_hash
values(80,'John'); select * from emp_hash;
```

	EMP_ID	NAME
1	30	Shubham
2	40	Omkar
3	50	Sanskar
4	80	John
5	10	Vikas
6	70	Jay
7	20	Rohan
8	60	Nitin

```
select * from emp_hash partition(SYS_P948);
```

	EMP_ID	NAME
1	30	Shubham
2	40	Omkar
3	50	Sanskar
4	80	John

```
select * from emp_hash partition(SYS_P949);
```

	EMP_ID	NAME
1	10	Vikas
2	70	Jay

```
select * from emp_hash partition(SYS_P950);
```

	EMP_ID	NAME
1	20	Rohan

```
select * from emp_hash partition(SYS_P951);
```

	EMP_ID	NAME
1	60	Nitin

Experiment No.5

Q. Implementation of Oracle Synonyms and Sequence. create SEQUENCE

```
seq_student
```

```
start with 1 increment by 2
```

```
nocycle nocache;
```

```
Sequence SEQ_STUDENT created.
```

```
create table student_info(id number, name varchar(50))
```

```
Table STUDENT_INFO created.
```

```
insert into student_info Values(seq_student.nextval,'ABC');
```

```
1 row inserted.
```

```
SELECT * FROM student_info;
```

The screenshot shows the Oracle SQL Developer interface. At the top, there are tabs for 'Script Output' and 'Query Result'. Below the tabs, there are icons for 'Run', 'Stop', 'Reset', and 'SQL'. A status bar at the bottom indicates 'All Rows Fetched: 1 in 0.078 seconds'. The main area displays the following SQL code and its execution results:

ID	NAME
1	ABC

```
insert into student_info Values(seq_student.nextval,'DEF'); insert
```

```
into student_info Values(seq_student.nextval,'GHI'); insert into
```

```
student_info Values(seq_student.nextval,'JKL'); insert into
```

```
student_info Values(seq_student.nextval,'MNO'); insert into
```

```
student_info Values(seq_student.nextval,'PQR');
```

```
SELECT * FROM student_info;
```

The screenshot shows the Oracle SQL Developer interface. At the top, there are tabs for 'Script Output' and 'Query Result'. Below the tabs, there are icons for 'Run', 'Stop', 'Reset', and 'SQL'. A status bar at the bottom indicates 'All Rows Fetched: 6 in 0.003 seconds'. The main area displays the following SQL code and its execution results:

ID	NAME
1	ABC
2	DEF
3	GHI
4	JKL
5	MNO
6	PQR

```
create SEQUENCE seq_value start
```

```
with 1 minvalue 1 maxvalue 5
```

```
increment by 1
```

```
cycle cache 4;
```

```
Sequence SEQ_VALUE created.
```

```
insert into student_info Values(seq_value.nextval,'abc');
```

```
1 row inserted.
```

```

insert into student_info Values(seq_value.nextval,'abc'); insert
into student_info Values(seq_value.nextval,'xyz'); insert into
student_info Values(seq_value.nextval,'def'); insert into
student_info Values(seq_value.nextval,'ghi'); insert into
student_info Values(seq_value.nextval,'jkl');

```

	ID	NAME
4	7	JKL
5	9	MNO
6	11	PQR
7	13	abc
8	15	xyz
9	17	def
10	19	ghi
11	21	jk1
12	1	abc
13	2	xyz
14	3	def
15	4	ghi
16	5	jk1

```

insert into student_info Values(seq_value.currvl,'abcd');

```

	ID	NAME
6	11	PQR
7	13	abc
8	15	xyz
9	17	def
10	19	ghi
11	21	jk1
12	1	abc
13	2	xyz
14	3	def
15	4	ghi
16	5	jk1
17	5	abcd

```

insert into student_info Values(seq_value.nextval,'jklm');

```

	ID	NAME
11	21	jk1
12	1	abc
13	2	xyz
14	3	def
15	4	ghi
16	5	jk1
17	5	abcd
18	1	jklm

```

CREATE SYNONYM student_id_seq FOR seq_student;

```

Synonym STUDENT_ID_SEQ created.

```

INSERT INTO student_info (id, name) VALUES (student_id_seq.nextval, 'John Doe');

```

```

SELECT * FROM student_info;

```

	ID	NAME
11	21	jk1
12	1	abc
13	2	xyz
14	3	def
15	4	ghi
16	5	jk1
17	5	abcd
18	1	jklm
19	23	John Doe

Experiment No – 6

1. Functions :

```
create or replace function square_area(side number)
```

```
return number is
```

```
pi constant number(7,3) := 3.141;
```

```
area number(7,3);
```

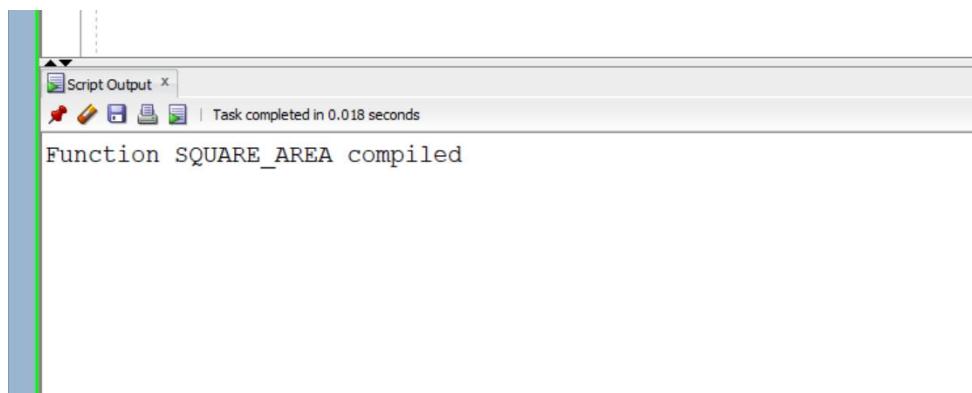
```
begin
```

```
area := side*side;
```

```
return area;
```

```
end;
```

```
/
```



The screenshot shows the Oracle SQL Developer interface. A script editor window is open with the following PL/SQL code:

```
create or replace function square_area(side number)
return number is
pi constant number(7,3) := 3.141;
area number(7,3);
begin
area := side*side;
return area;
end;
/
```

The output window below shows the result of the compilation:

```
Function SQUARE_AREA compiled
```

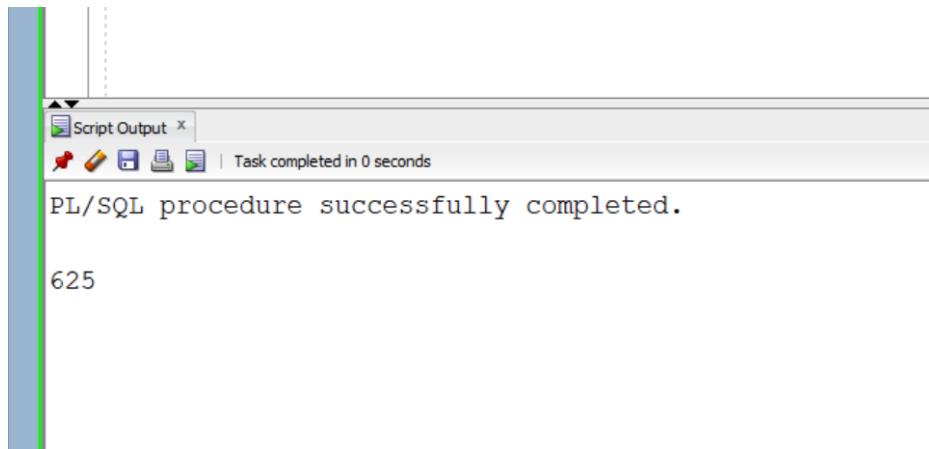
```
set SERVEROUTPUT ON;
```

```
begin
```

```
DBMS_OUTPUT.PUT_LINE(SQUARE_AREA(25));
```

```
end;
```

```
/
```



The screenshot shows the Oracle SQL Developer interface. A script editor window is open with the following PL/SQL code:

```
set SERVEROUTPUT ON;
begin
DBMS_OUTPUT.PUT_LINE(SQUARE_AREA(25));
end;
/
```

The output window below shows the result of the execution:

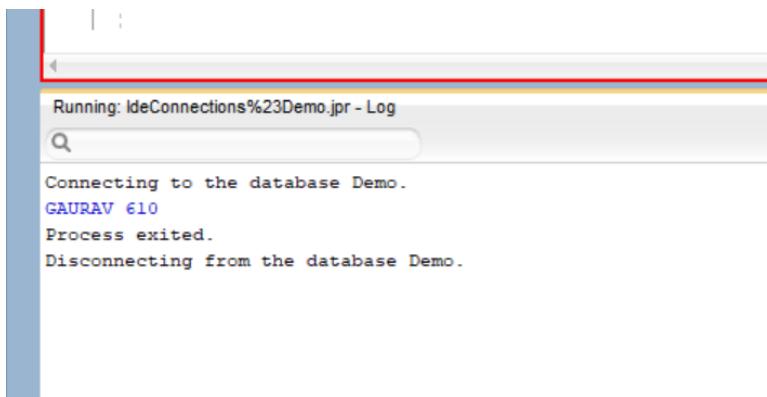
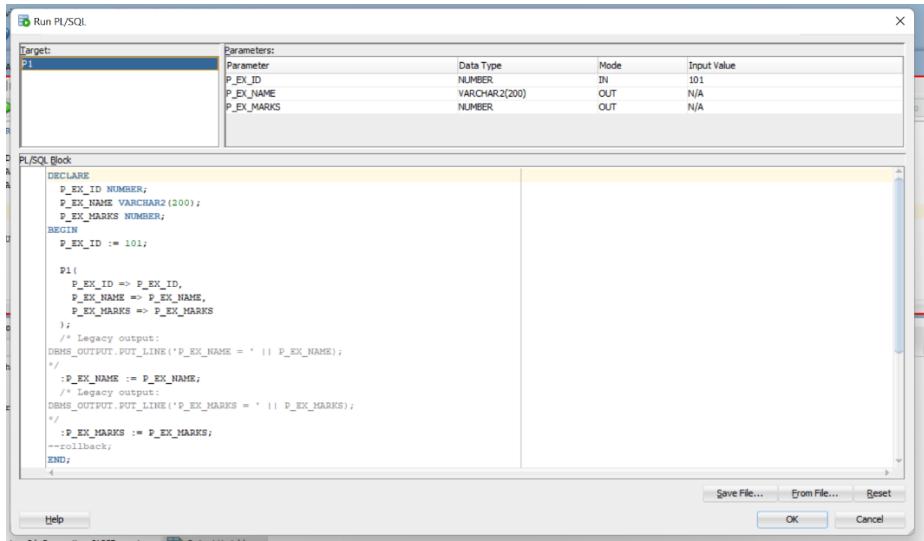
```
PL/SQL procedure successfully completed.
```

```
625
```

Procedures :

```
CREATE OR REPLACE PROCEDURE P1
```

```
(P_EX_ID IN NUMBER , P_EX_NAME OUT VARCHAR2 , P_EX_MARKS OUT NUMBER ) AS
BEGIN
SELECT EX_NAME,EX_MARKS INTO P_EX_NAME,P_EX_MARKS FROM EXAM WHERE EX_ID = P_EX_ID;
DBMS_OUTPUT.PUT_LINE(P_EX_NAME || ' ' || P_EX_MARKS);
END P1;
```



3. Cursors :

```
set SERVEROUTPUT ON;
DECLARE
v_name varchar(30);
cursor emp_cursor is
select ex_name from exam
where ex_id<110;
begin
open emp_cursor;
loop
fetch emp_cursor into v_name;
DBMS_OUTPUT.PUT_LINE(v_name);
```

```
exit when emp_cursor%notfound;  
end loop;  
close emp_cursor;  
end;
```

```
IBPO  
SSC  
RBI  
CDS  
RRB  
SBI  
NDA  
LIC  
LIC
```

4. Triggers :

```
CREATE OR REPLACE TRIGGER MY_TRIGGER
```

```
BEFORE UPDATE ON EXAM
```

```
BEGIN
```

```
DBMS_OUTPUT.PUT_LINE('RECORD UPDATED');
```

```
END;
```

```
Trigger MY_TRIGGER compiled
```

```
UPDATE EXAM SET EX_MARKS = '340' WHERE EX_ID = '101';
```

```
Running: IdeConnections%23Demo.jpr - Log  
Connecting to the database Demo.  
RECORD UPDATED  
Process exited.  
Disconnecting from the database Demo.
```

```
select * from exam;
```

The screenshot shows the 'Script Output' tab in Oracle SQL Developer. The output window title is 'Script Output X'. Below the title bar, there are several icons and the text 'Task completed in 0 seconds'. The main content area displays the results of a query:

EX_ID	EX_NAME	EX_MARKS
101	GAURAV	340
102	VIRAT	920
103	CRISTIANO	700

5. Views :

a. Creating views :

```
create view exam_test as select ex_id,ex_name,ex_marks from exam where ex_id = 109 with  
check option constraint exam_test_cnst;
```

The screenshot shows the 'Script Output' tab in Oracle SQL Developer. The output window title is 'Script Output X'. Below the title bar, there are several icons and the text 'Task completed in 0.215 seconds'. The main content area displays the message: 'View EXAM_TEST created.'

```
select * from exam_test;
```

The screenshot shows the 'Script Output' tab in Oracle SQL Developer. The output window title is 'Script Output X'. Below the title bar, there are several icons and the text 'Task completed in 0.031 seconds'. The main content area displays the results of a query:

EX_ID	EX_NAME	EX_MARKS
109	RRB	650

```
insert into exam_test(ex_id,ex_name,ex_marks) values(109,'wt',610);
```

```
Script Output x | Task completed in 0.015 seconds
```

```
1 row inserted.
```

```
select * from exam_test;
```

```
Script Output x | Task completed in 0.015 seconds
```

```
1 row inserted.
```

EX_ID	EX_NAME	EX_MARKS
109	RRB	650
109	wt	610
109	wt	610

b. Updating views :

```
update exam_test set ex_name ='st' where Ex_marks = 610;
```

```
select * from EXAM_TEST;
```

```
Script Output x | Task completed in 0.015 seconds
```

```
2 rows updated.
```

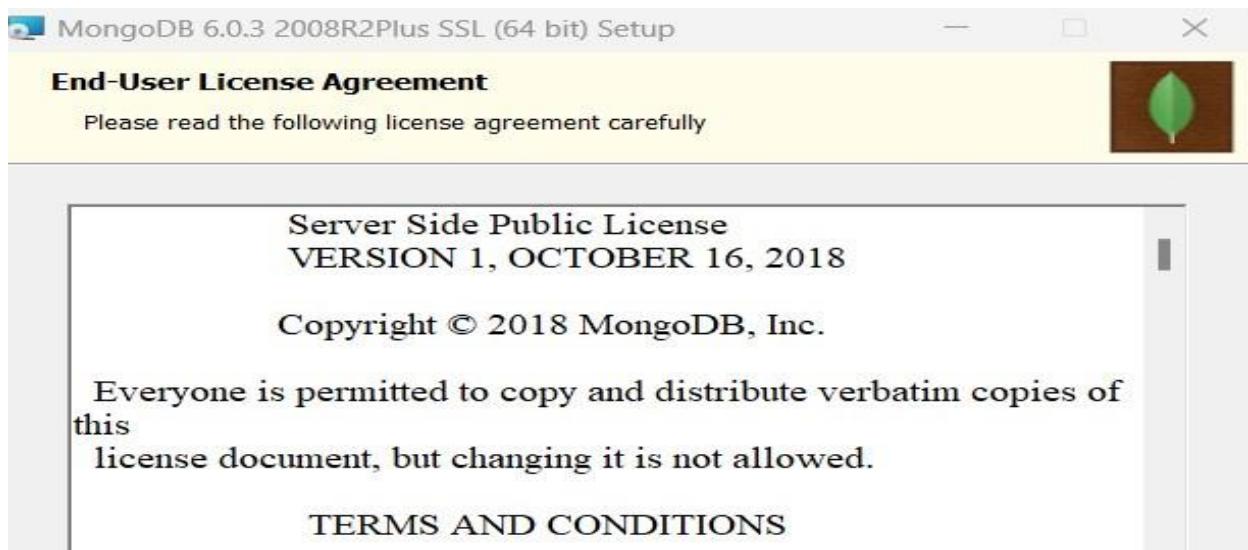
EX_ID	EX_NAME	EX_MARKS
109	RRB	650
109	st	610
109	st	610

Experiment no.7

MongoDB Installation:

The screenshot shows the MongoDB website's product selection interface. On the left, a sidebar lists various products: MongoDB Atlas, MongoDB Enterprise Advanced, MongoDB Community Edition, MongoDB Community Server (which is selected and highlighted in green), MongoDB Community Kubernetes Operator, Tools, and Mobile & Edge. The main content area is titled "MONGODB COMMUNITY SERVER" and "MongoDB Community Server Download". It describes the Community version as a flexible document database with support for ad-hoc queries, secondary indexing, and real-time aggregations. It also mentions MongoDB Atlas as a managed service. Below this, there's a terminal window showing command-line instructions: "\$ brew install mongodb-atlas" and "\$ atlas setup". Further down, dropdown menus allow selecting the "Version" (set to 6.0.3 (current)), "Platform" (set to Windows), and "Package" (set to msi). A prominent green "Download" button is at the bottom.





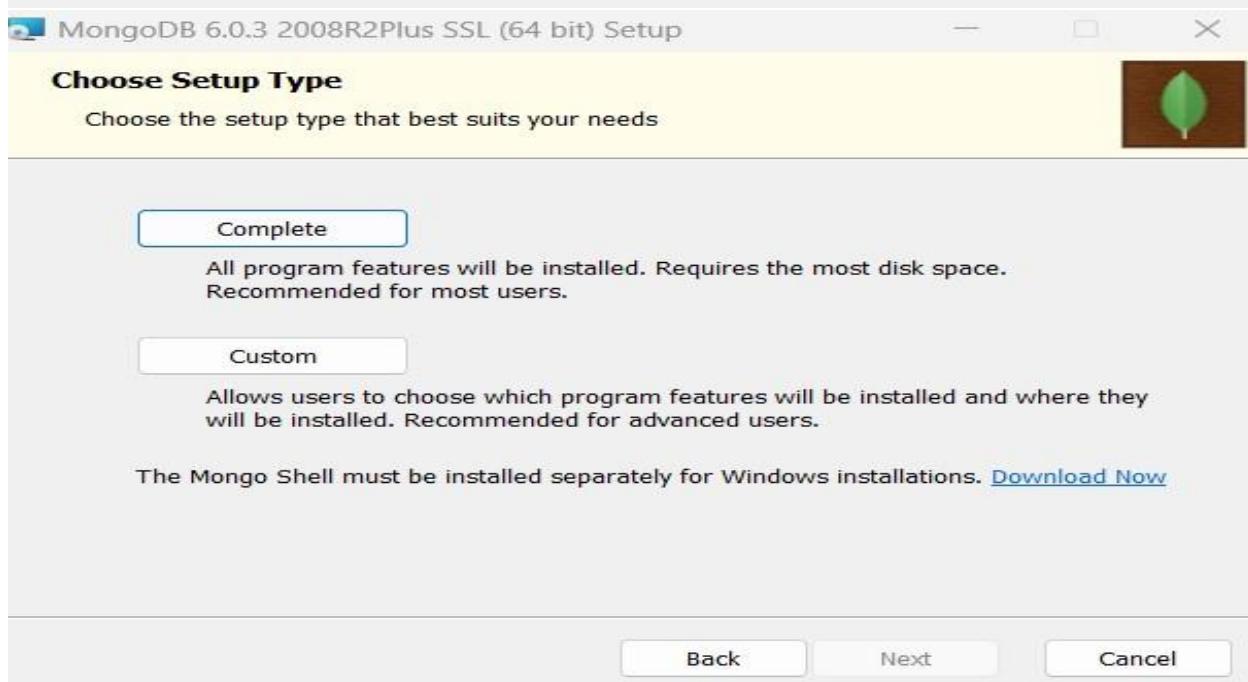
I accept the terms in the License Agreement

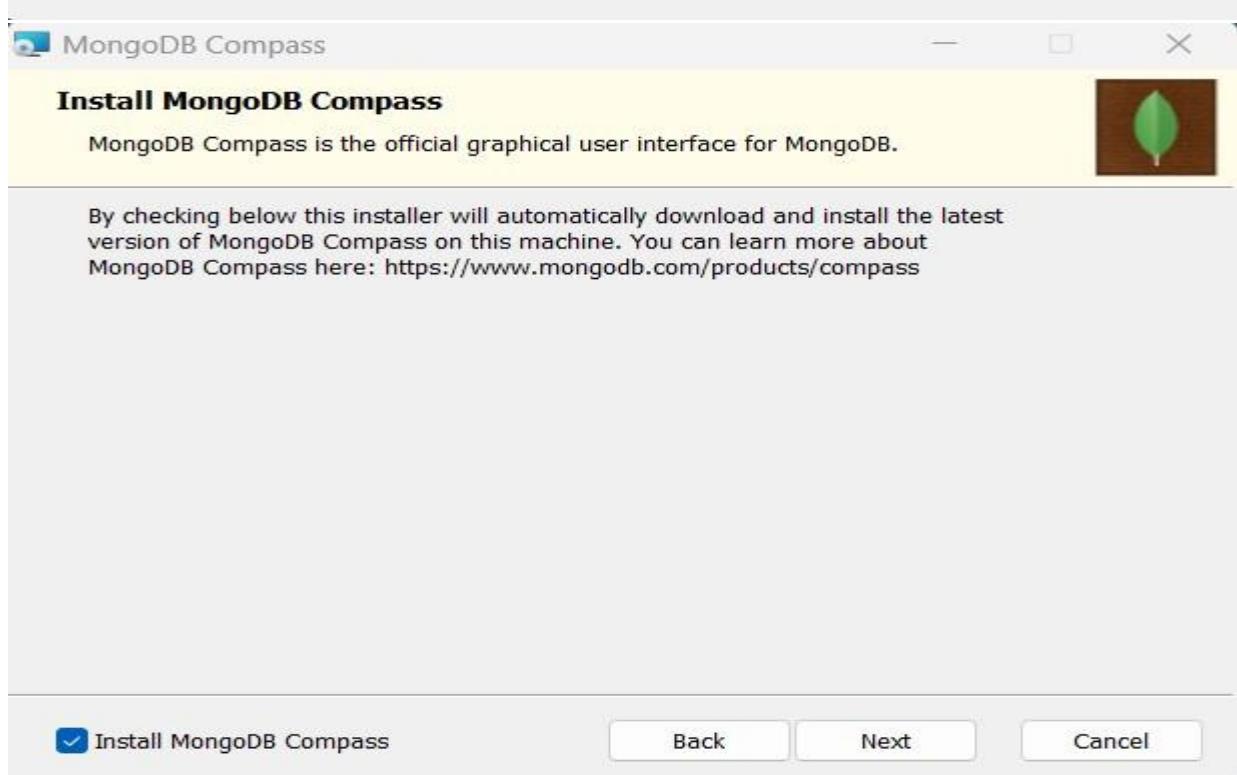
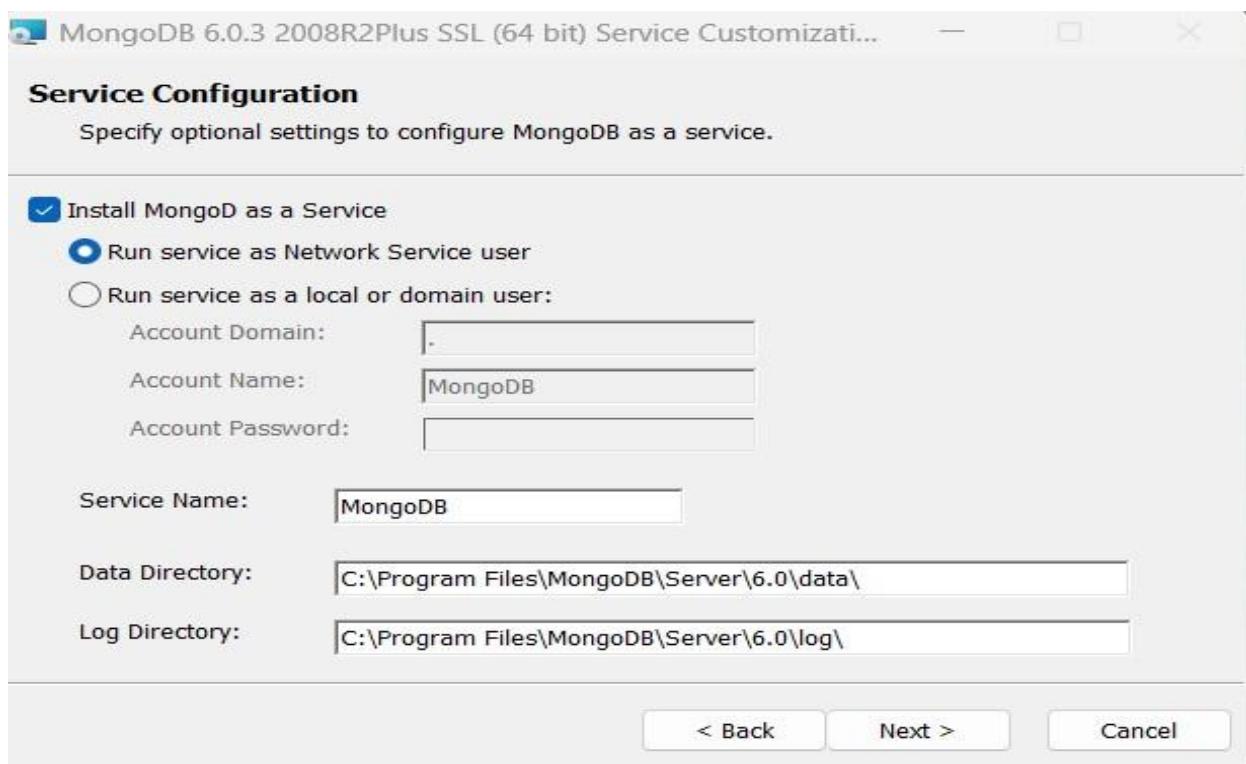
Print

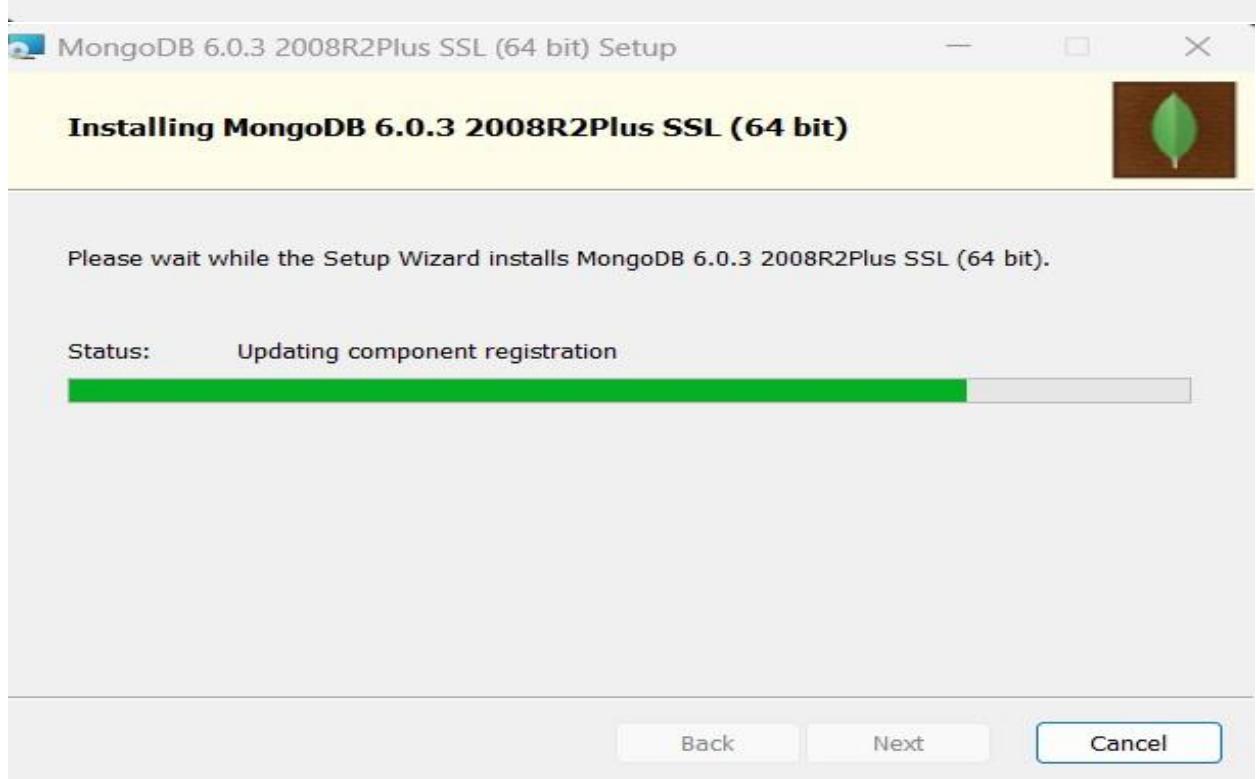
Back

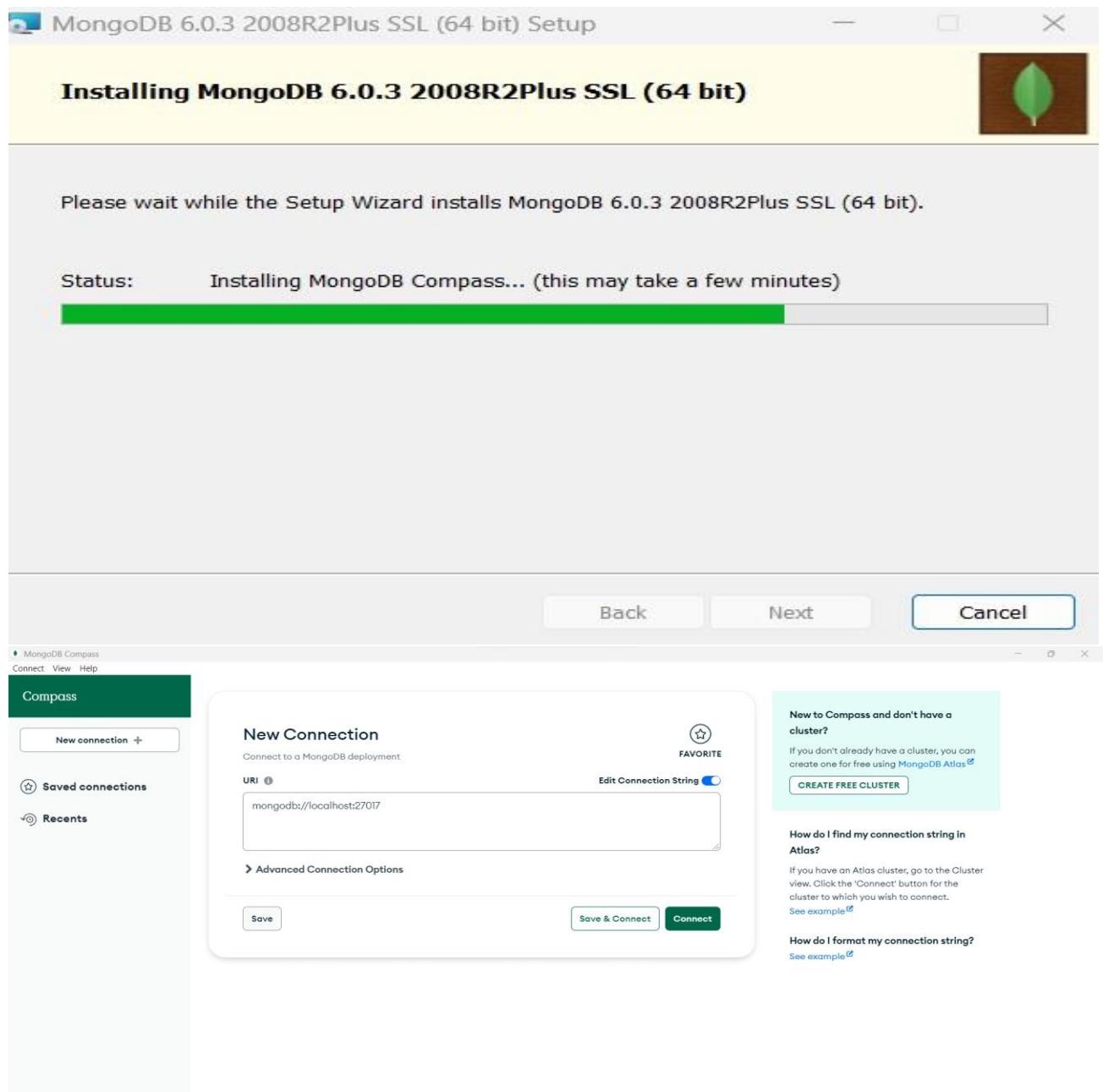
Next

Cancel









Connect View Help

localhost:27017 ...

My Queries Databases Performance

Create database View :: Sort by Database Name ↴ ↵

admin

Storage size: 20.48 kB Collections: 1 Indexes: 1

collection

Storage size: 20.48 kB Collections: 1 Indexes: 1

config

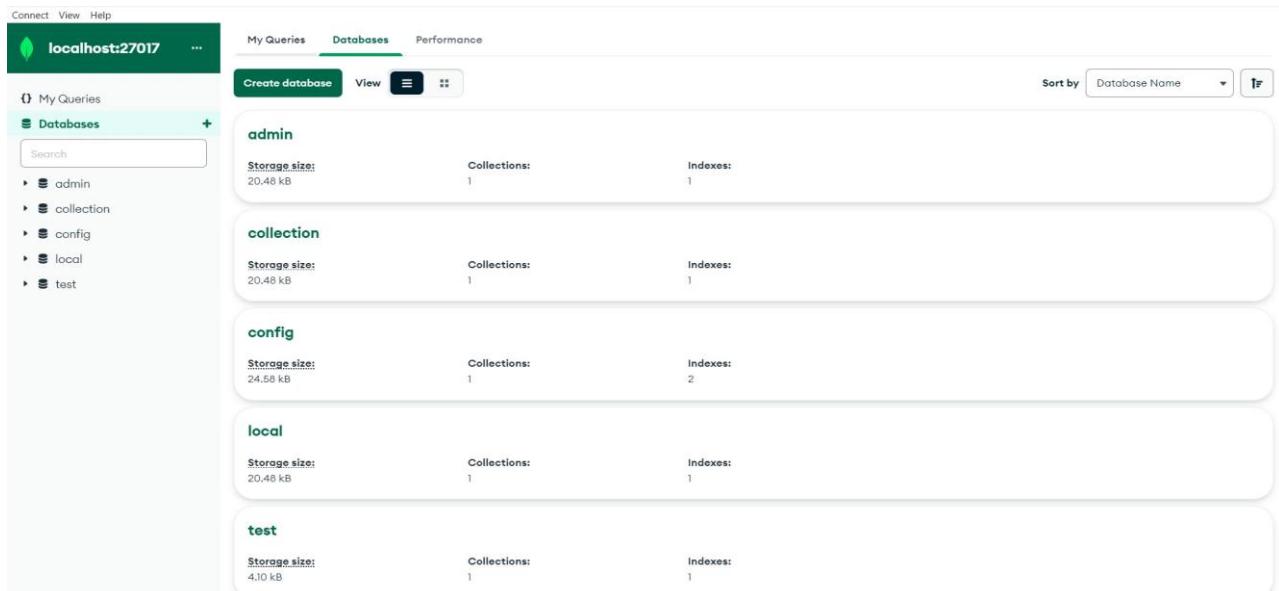
Storage size: 24.68 kB Collections: 1 Indexes: 2

local

Storage size: 20.48 kB Collections: 1 Indexes: 1

test

Storage size: 4.10 kB Collections: 1 Indexes: 1



Create Database

Database Name

First_Database

Collection Name

First_Collection

➤ Advanced Collection Options (e.g. Time-Series, Capped, Clustered collections)

Cancel

Create Database

MongoDB Compass - localhost:27017/First_Database

Connect View Help

localhost:27017 ...

Collections Create collection View :: Sort by Collection Name ↴ ↵

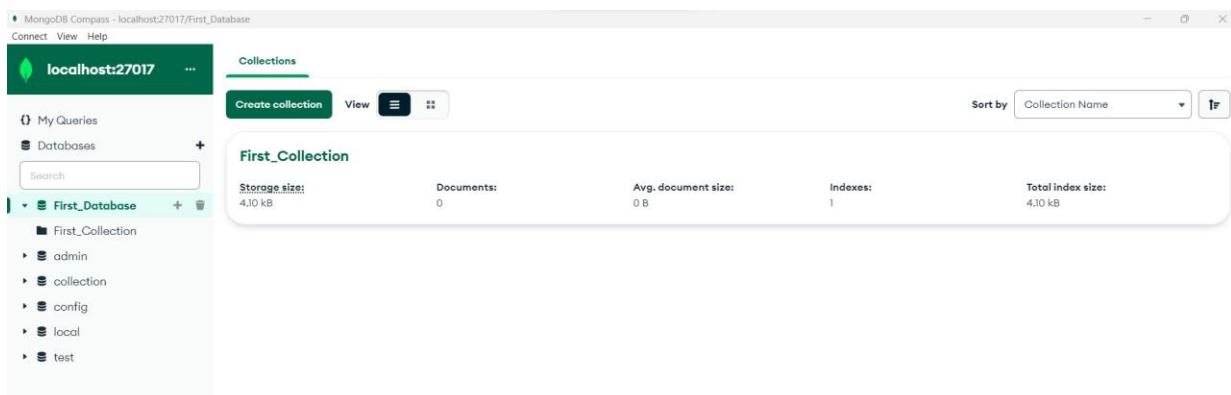
First_Collection

Storage size: 4.10 kB Documents: 0 Avg. document size: 0 B Indexes: 1 Total index size: 4.10 kB

My Queries Databases

First_Database +

- First_Collection
- admin
- collection
- config
- local
- test



Insert to Collection local.startup_log

VIEW

{}

☰

```
1 *//**
2  * Paste one or more documents here
3 */
4 {
5   "_id": {
6     "$oid": "6391cff42c06e99b0e425f90"
7   },
8   "title" : "RSA..post",
9   "content" : "My first post content.."
10 }
```

Cancel

Insert

MongoDB Compass - localhost:27017/local.startup_log

Connect View Collection Help

localhost:27017 ... Documents local.startup_log +

0 My Queries 2 DOCUMENTS 1 INDEXES

■ Databases

Search

▼ First_Database

 ■ First_Collection

▶ admin

▶ collection

▶ config

▼ local

 ■ startup_log ...

 ▶ test

local.startup_log

Documents Aggregations Schema Explain Plan Indexes Validation

Filter ⓘ Type a query: { field: 'value' }

Reset Find More Options ↗

1-2 of 2 ⏪ ⏩ ⏴ ⏵ ⏷ ⏸

ADD DATA EXPORT COLLECTION

Import file L670494331357 RSA

Insert document 2022-12-08T10:12:11.000+00:00 local: "Thu Dec 8 15:42:11.357"

 > cmdLine: Object
 pid: 103636
 > buildinfo: Object

_id: ObjectId('6391cff42c06e99b0e425f90')
title: "RSA..post"
content: "My first post content.."

Insert to Collection

First_Database.First_Collection

VIEW

{}

≡

```
1 /**
2  * Paste one or more documents here
3  */
4 {
5   "_id": {
6     "$oid": "6391d3a72c06e99b0e425f94"
7   },
8   "title" : "Inserting First Document",
9   "content" : "Content of First Document"
10 }
```

Cancel

Insert

MongoDB Compass - localhost:27017/First_Database.First_Collection

Connect View Collection Help

localhost:27017 ...

Documents First_Database.Fir...

My Queries

Databases

Search

First_Database

First_Collection

admin

collection

config

local

startup.log

test

First_Database.First_Collection

Documents Aggregations Schema Explain Plan Indexes Validation

Filter Type a query: { field: 'value' }

1 DOCUMENTS 1 INDEXES

1-1 of 1

ADD DATA EXPORT COLLECTION

_id: ObjectId('6391d3a72c06e99b0e425f94')
title: "Inserting First Document"
content: "Content of First Document"

The screenshot shows the MongoDB Compass interface. At the top, there's a header bar with 'MongoDB Compass - localhost:27017/First_Database.First_Collection'. Below it is a navigation bar with 'Connect', 'View', 'Collection', and 'Help' buttons. The main area has tabs for 'Documents' (which is selected), 'Aggregations', 'Schema', 'Explain Plan', 'Indexes', and 'Validation'. On the left, there's a sidebar with 'My Queries' and 'Databases' sections, and a search bar. The 'First_Database' section is expanded, showing 'First_Collection' as the selected collection. The 'First_Collection' page displays a single document with the '_id' field set to an ObjectId ('6391d3a72c06e99b0e425f94'), 'title' field set to 'Inserting First Document', and 'content' field set to 'Content of First Document'. There are buttons for 'Cancel' and 'Insert' at the bottom of the document editor.

EXPERIMENT NO. 9

CRUD in MongoDB:

Create Operations

createCollection():

```
db.createCollection('MarkTech')
```

```
> db.createCollection('MarkTech')
< { ok: 1 }
> show collections
< MarkTech
```

insertOne():

```
db.MarkTech.insertOne({ 'name' : 'Harry', 'lang' : 'Java', 'member_since' : 5 })
```

```
> db.MarkTech.insertOne({ 'name' : 'Harry', 'lang' : 'Java', 'member_since' : 5 })
{ acknowledged: true,
  insertedId: ObjectId("6392f660f6509c003116b5d8") }
```

insertMany():

```
db.MarkTech.insertMany([
  {name:'Vikas','lang':'JavaScript','member_since':1},
  {name:'Rohan','lang':'Python','member_since':6},
  {name:'Karan','lang':'Java','Role':'Java Developer'}
])
```

```
{ acknowledged: true,
  insertedIds:
    { '0': ObjectId("6392f884f6509c003116b5dc"),
      '1': ObjectId("6392f884f6509c003116b5dd"),
      '2': ObjectId("6392f884f6509c003116b5de") } }
```

Read Operations

find()

```
db.MarkTech.find()
```

```
db.MarkTech.find()
{ _id: ObjectId("6392f5eff6509c003116b5d7") }
{ _id: ObjectId("6392f660f6509c003116b5d8"),
  name: 'Harry',
  lang: 'Java',
  member_since: 5 }
{ _id: ObjectId("6392f884f6509c003116b5dc"),
  name: 'Vikas',
  lang: 'JavaScript',
  member_since: 1 }
{ _id: ObjectId("6392f884f6509c003116b5dd"),
  name: 'Rohan',
  lang: 'Python',
  member_since: 6 }
{ _id: ObjectId("6392f884f6509c003116b5de"),
  name: 'Karan',
  lang: 'Java',
  Role: 'Java Developer' }
```

```
db.MarkTech.find({lang:'Java'})
```

```
db.MarkTech.find({lang:'Java'})
{ _id: ObjectId("6392f660f6509c003116b5d8"),
  name: 'Harry',
  lang: 'Java',
  member_since: 5 }
{ _id: ObjectId("6392f884f6509c003116b5de"),
  name: 'Karan',
  lang: 'Java',
  Role: 'Java Developer' }
```

findOne()

```
db.MarkTech.findOne({lang:'Java'})
```

```
db.MarkTech.findOne({lang:'Java'})
{ _id: ObjectId("6392f660f6509c003116b5d8"),
  name: 'Harry',
  lang: 'Java',
  member_since: 5 }
```

Update Operations

updateOne()

```
db.MarkTech.updateOne({name:'Karan'},{$set: {'name':'Karan', 'lang':'Java','Role':'Software Developer'}})
```

```
db.MarkTech.updateOne({name:'Karan'},{$set: {'name':'Karan', 'lang':'Java','Role':'Software Developer'}})
{ acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
```

updateMany()

```
db.MarkTech.updateMany({lang:'Java'},{$set:{'name':'Bhushan'}})
```

```
db.MarkTech.updateMany({lang: 'Java'}, {$set: { 'name' : 'Bhushan' } })
{ acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0 }
```

replaceOne()

```
db.MarkTech.replaceOne({name:'Vikas'},{'name':'Harry'})
```

```
db.MarkTech.replaceOne({name: 'Vikas'}, {name: 'Harry'})
{ acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
```

Delete Operations

deleteOne()

```
db.MarkTech.deleteOne({name:'Harry'})
```

```
db.MarkTech.deleteOne ({name:'Harry'})
{ acknowledged: true, deletedCount: 1 }
```

deleteMany()

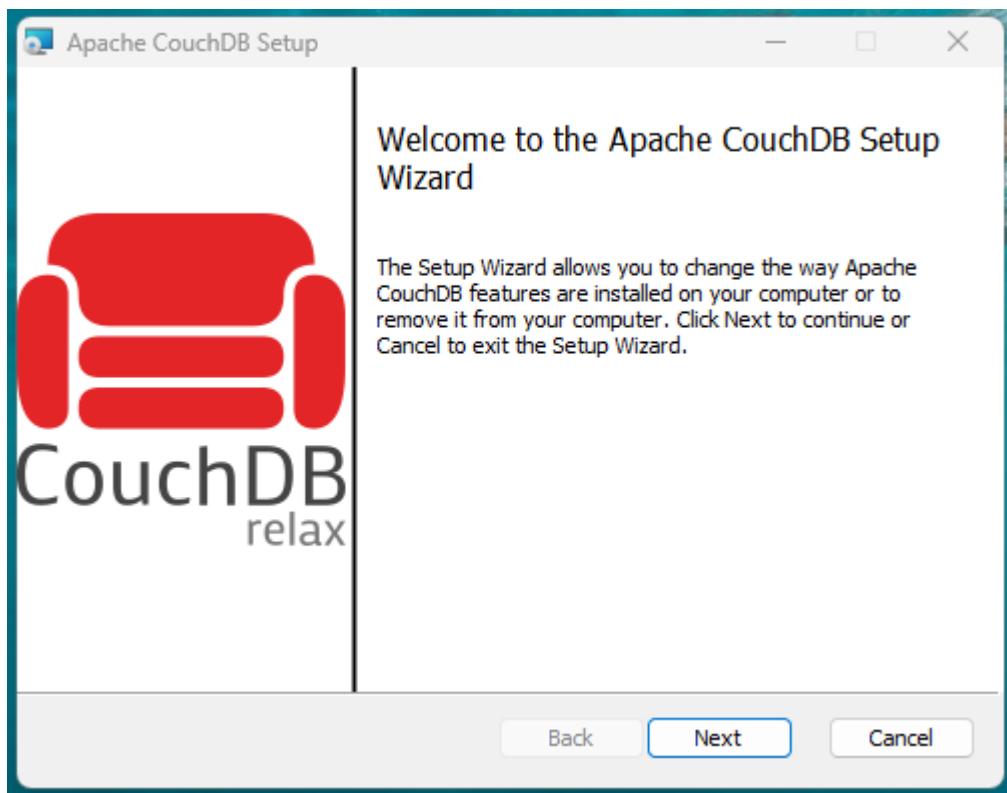
```
db.MarkTech.deleteMany({lang:'Java'})
```

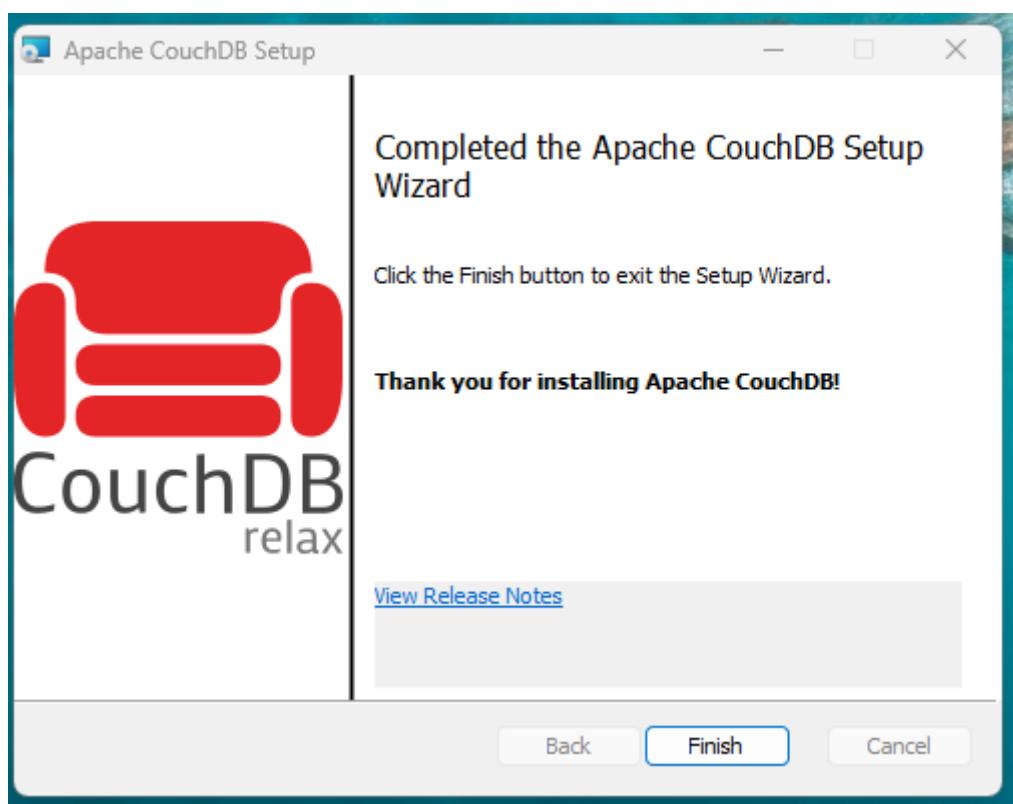
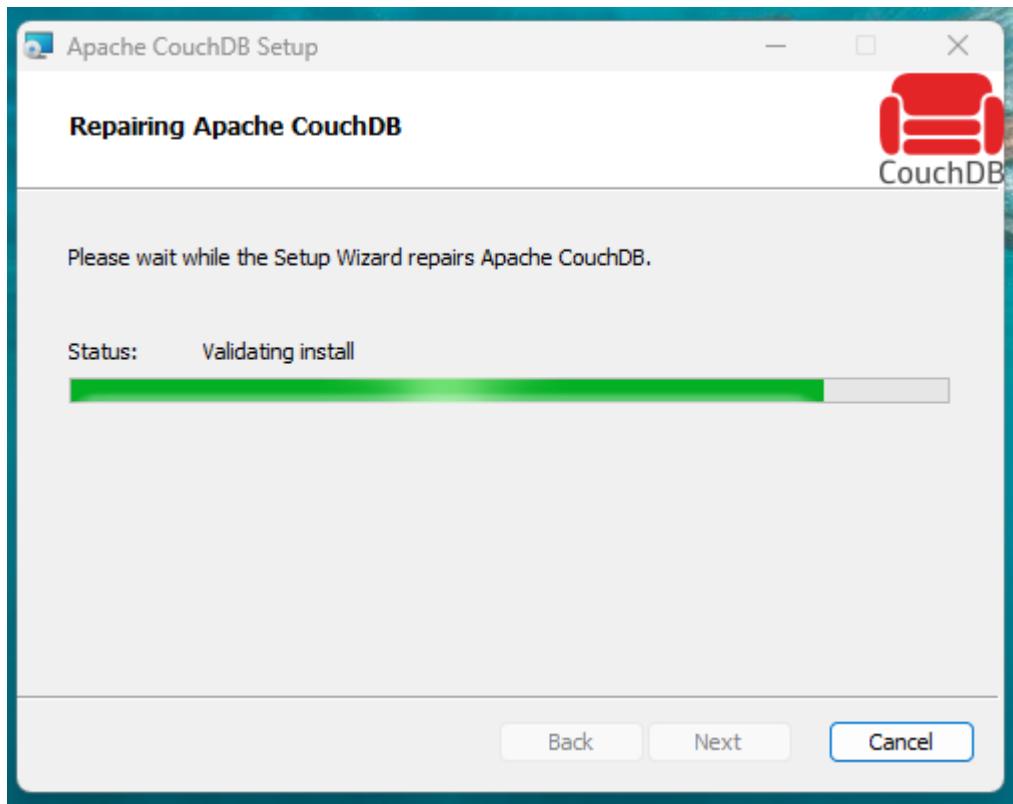
```
db.MarkTech.deleteMany ({lang:'Java'})
{ acknowledged: true, deletedCount: 2 }
```

EXPERIMENT NO. 9

CouchDB

Installation:





Create Database:

The screenshot shows the Fauxton interface for managing databases. On the left is a sidebar with various icons for database management. The main area displays a table for 'Databases' with columns: Name, Size, # of Docs, and Partitioned. A modal window titled 'Create Database' is open on the right, prompting for a 'Database name' (set to 'demo') and 'Partitioning' (unchecked). Below the modal are links for 'Cancel' and 'Create'. The URL in the browser is `localhost:5984/_utils/#/_all_dbs`.

The screenshot shows the Fauxton interface for the 'demo' database. The left sidebar includes options like 'All Documents', 'Run A Query with Mango', 'Permissions', 'Changes', and 'Design Documents'. The main content area features a large icon of a sofa and the text 'No Documents Found'. At the bottom, it says 'Showing 0 documents. Documents per page: 20'. The URL in the browser is `localhost:5984/_utils/#database/demo/_all_docs`.

Create Document:

The screenshot shows the Fauxton interface for Apache CouchDB. On the left is a sidebar with icons for database management. The main area shows a list of documents under the database 'demo'. A context menu is open over a specific document, showing options like 'Add New', 'New Doc', 'New View', and 'Mango Indexes'. The document details are displayed in a JSON editor. At the bottom right, there are navigation links and a status message.

localhost:5984/_utils/#database/demo/_all_docs

All Documents

Run A Query with Ma

Add New

Permissions

Changes

Design Documents

Mango Indexes

Document ID

Options

{ } JSON

Create Document

id "5a860bb979b31bb6c8a67ce2770038f8"

```
{
  "_id": "5a860bb979b31bb6c8a67ce2770038f8",
  "key": "5a860bb979b31bb6c8a67ce2770038f8",
  "value": {
    "rev": "1-6b8ef96f17832bf512d13c742802e3fc"
  },
  "doc": {
    "_id": "5a860bb979b31bb6c8a67ce2770038f8",
    "_rev": "1-6b8ef96f17832bf512d13c742802e3fc",
    "name": "Robert Trump",
    "age": "50"
  }
}
```

Fauxton on Apache CouchDB v. 3.2.2

Log Out

Showing document 1 - 1. Documents per page: 20

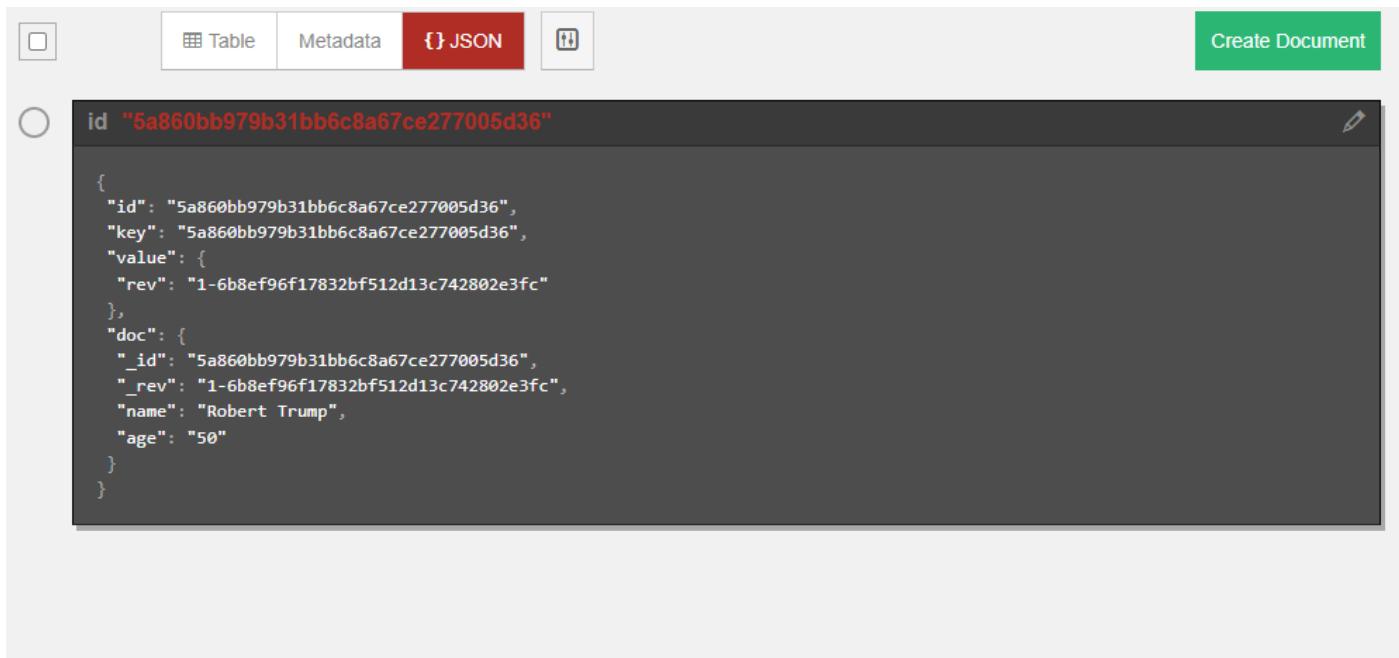
The screenshot shows the Fauxton interface for creating a new document. The title bar says 'demo > New Document'. A green button labeled 'Create Document' with a checkmark is prominent. Below it is a code editor where a JSON document is being typed. The document contains fields for '_id', 'name', and 'age'.

demo > New Document

Create Document

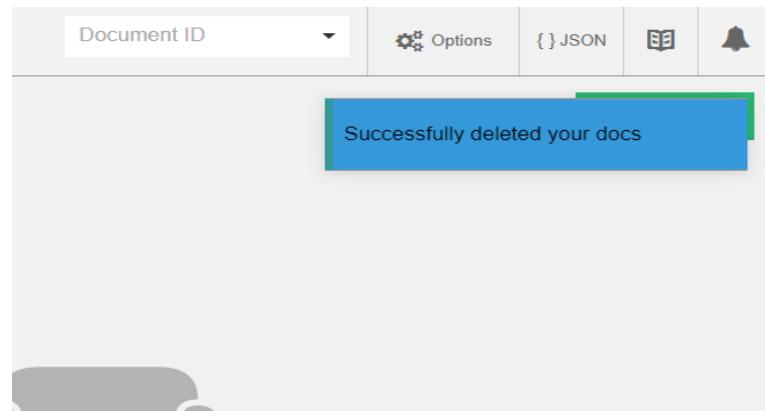
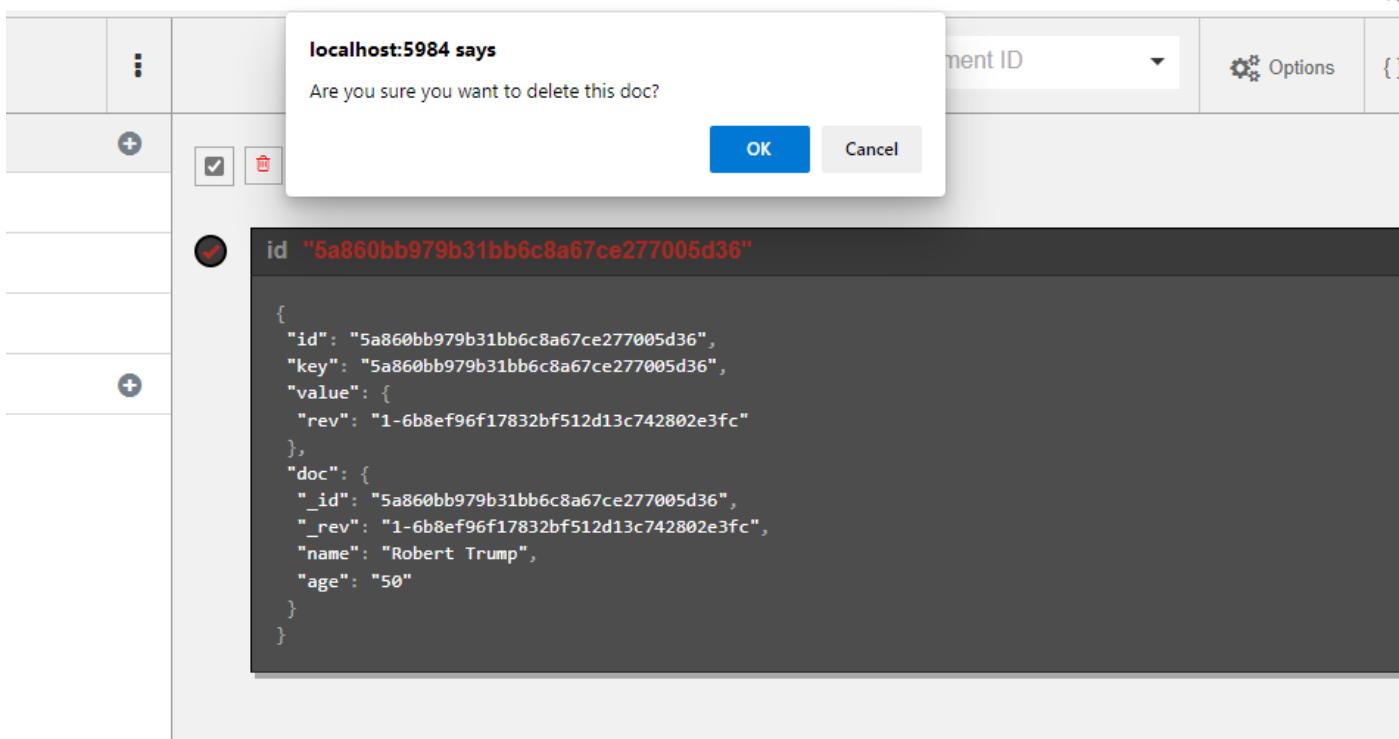
Cancel

```
1 {  
2   "_id": "5a860bb979b31bb6c8a67ce277005d36",  
3   "name": "Robert Trump",  
4   "age": "50"  
5 }
```



```
{  
  "id": "5a860bb979b31bb6c8a67ce277005d36",  
  "key": "5a860bb979b31bb6c8a67ce277005d36",  
  "value": {  
    "rev": "1-6b8ef96f17832bf512d13c742802e3fc"  
  },  
  "doc": {  
    "_id": "5a860bb979b31bb6c8a67ce277005d36",  
    "_rev": "1-6b8ef96f17832bf512d13c742802e3fc",  
    "name": "Robert Trump",  
    "age": "50"  
  }  
}
```

Delete Document:



Delete Database:

Databases

Name	Size	# of Docs	Partitioned	Actions
demo	0 bytes	0	No	

Fusion on Apache CouchDB v3.2.2

Log Out

Showing 1–1 of 1 databases. Databases per page: 20 < 1 >

Databases

Confirm Deletion

Warning: This action will permanently delete `demo`. To confirm the deletion of the database and all of the database's documents, you must enter the database's name.

Cancel

Fusion on Apache CouchDB v3.2.2

Log Out

Showing 1–1 of 1 databases. Databases per page: 20 < 1 >

Database name

Create Database { } JSON

Partitioned

The database demo has been deleted.

Experiment No . 10

Title : Case study of Oracle Database Administration, Security & study of database administrator's responsibilities.

--to create user :

```
CREATE USER c##intro_user IDENTIFIED BY mypassword;
```

--to grant connect privilege to user :

```
GRANT CONNECT TO c##intro_user;
```

--to grant session to user and any kind of privilege:

```
GRANT CREATE SESSION , GRANT ANY PRIVILEGE TO c##intro_user;
```

2

```
GRANT UNLIMITED TABLESPACE TO c##intro user;
```

--to grant create table privilege to user:

```
GRANT CREATE TABLE TO c##intro_user;
```

--To see all the users : -- SELECT * FROM all_users;

--to revoke permission of other user by intro_user :

```
REVOKE ALL on table_name from user_name;
```

Output:

SQL*Plus: Release 12.2.0.1.0 Production on Tue Oct 14 12:51:58 2025

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Enter user-name: c##raj

Enter password:

Connected to:

Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> create table student12(int ID ,varchar(55) Name)

SQL> insert into student12(int ID ,varchar(55) Name) values (1, 'Alice'), (2, 'Bob'),(3, 'Charlie');

insert into student12(int ID ,varchar(55) Name) values (1, 'Alice'),

SQL> insert into student12(int ID ,varchar(55) Name) values (1, 'Alice')

SQL> insert into student12(int ID ,varchar(55) Name) values (2, 'BOb')

SQL> insert into student12(int ID ,varchar(55) Name) values (3, 'Marie')

SQL> Select * from student12;

SQL> CREATE TABLE student12 (ID INT, Name VARCHAR(55));

Table created.

SQL> INSERT INTO student12 (ID, Name) VALUES (1, 'Alice');

1 row created.

SQL> INSERT INTO student12 (ID, Name) VALUES (2, 'Bob');

1 row created.

SQL> INSERT INTO student12 (ID, Name) VALUES (3, 'Charlie');

1 row created.

SQL> SELECT * FROM student12;

ID NAME

1 Alice

2 Bob

3 Charlie

Experiment No: 11

Title: Study of CASE concept and tools.

Computer aided software engineering (CASE) is the implementation of computer facilitated tools and methods in software development. CASE is used to ensure a high-quality and defect-free software. CASE ensures a check-pointed and disciplined approach and helps designers, developers, testers, managers and others to see the project milestones during development.

CASE can also help as a warehouse for documents related to projects, like business plans, requirements and design specifications. One of the major advantages of using CASE is the delivery of the final product, which is more likely to meet real-world requirements as it ensures that customers remain part of the process.

CASE illustrates a wide set of labor-saving tools that are used in software development. It generates a framework for organizing projects and to be helpful in enhancing productivity. There was more interest in the concept of CASE tools years ago, but less so today, as the tools have morphed into different functions, often in reaction to software developer needs. The concept of CASE also received a heavy dose of criticism after its release.

CASE Tools:

The essential idea of CASE tools is that in-built programs can help to analyse developing systems in order to enhance quality and provide better outcomes. Throughout the 1990, CASE tool became part of the software lexicon, and big companies like IBM were using these kinds of tools to help create software.

Various tools are incorporated in CASE and are called CASE tools, which are used to support different stages and milestones in a software development life cycle.

Types of CASE Tools:

1. Diagramming Tools:

It helps in diagrammatic and graphical representations of the data and system processes. It represents system elements, control flow and data flow among different software components and system structure in a pictorial form.

For example, Flow Chart Maker tool for making state-of-the-art flowcharts.

2. Computer Display and Report Generators:

It helps in understanding the data requirements and the relationships involved.

3. Analysis Tools:

It focuses on inconsistent, incorrect specifications involved in the diagram and data flow. It helps in collecting requirements, automatically check for any irregularity, imprecision in the diagrams, data redundancies or erroneous omissions.

For example,

- (i) Accept 360, Accompa, CaseComplete for requirement analysis.
- (ii) Visible Analyst for total analysis.

4. Central Repository:

It provides the single point of storage for data diagrams, reports and documents related to project management.

5. Documentation Generators:

It helps in generating user and technical documentation as per standards. It creates documents for technical users and end users.

For example, Doxygen, DrExplain, Adobe RoboHelp for documentation.

6. **Code Generators:**

It aids in the auto generation of code, including definitions, with the help of the designs, documents and diagrams.

Uses in Databases

CASE tools can serve many functions in database design, including:

- Collecting and analyzing data.
- Designing a data model.
- Feasibility analysis.
- Requirements definition.
- Implementing the database.
- Prototyping.
- Data conversion.
- Generating application code.
- Generating reports.
- Programming and testing.
- Maintenance.

Case Tool

Microsoft Visio is a diagramming tool to create simple as well as complex diagrams and vector graphics. It enables you to create detailed organization charts, floor plans and pivot diagrams according to your business needs. It comes with inbuilt templates which helps you virtually create diagrams of any complexity like shape-based artwork or complex drawing.

Features of Microsoft Visio:

- VISO provides diagramming capabilities to prepare various business engineering Software, and Database Architecture easily.
- Prepare BPMN Models, Maps, and Diagrams. Prepare Flow Charts, Capture Brain Storming discussions, etc.
- Visio Standard Online and Visio Professional both are downloadable products. You can use these products as long as they're compatible with your device.
- The tool allows you to capture the information in ways that are valuable for you and your business.
- Provide support for creating varieties of diagramming shapes.
- Visio Pro offers features like automatic updates and multiple installations.

Visio File Types

Now in this MS Visio tutorial, we will learn about different Visio File Types. Following are important file types available with Visio software:

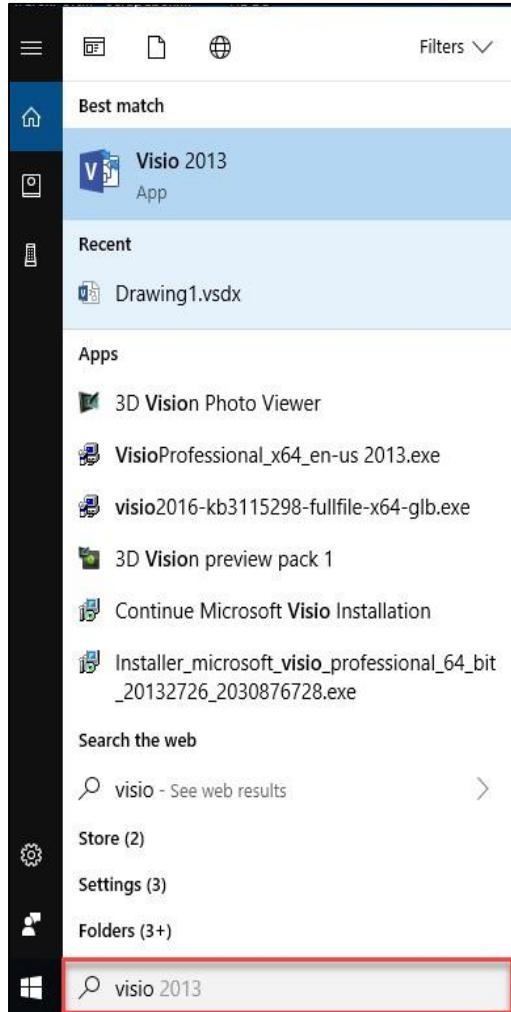
- **VSD (Visio Drawing):** This file extension is associated with Visio binary file format. It is used for storing flowcharts and diagram document files.
- **VSS (Visio Stencil):** This file extension is associated with Microsoft Visio. The file contains smart shapes (stencils). The file is in binary Visio document format.
- **VST (Visio Template):** The VST file extension is associated with Microsoft Visio templates.

- **VDW (Visio Web drawing):** The VDW extension is associated with Microsoft Office Visio. It contains web drawing which is created in Microsoft Office Visio.

Download and Install Visio

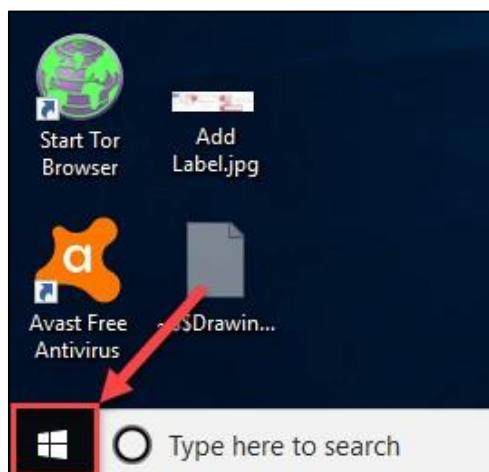
Notes: You require to have an active Visio license to download Visio. You need to run the “Click-toRun” version of Microsoft Office before installing Visio. The Click-to-Run version is installed from <https://www.office.com>.

How to start Visio

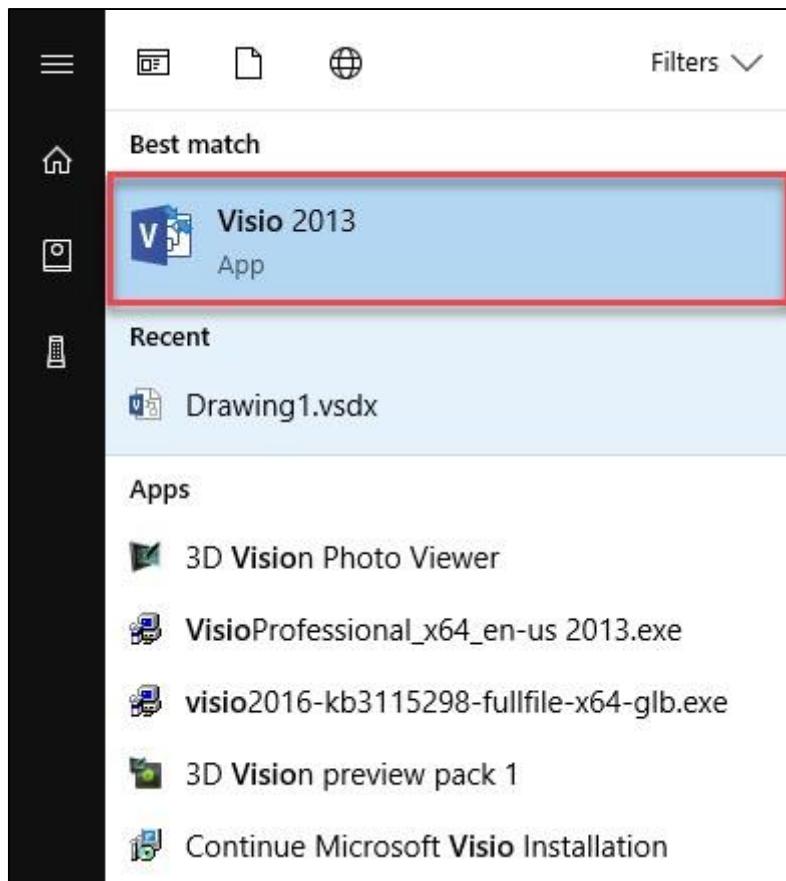


2. Click on “Visio 2013” menu.

Step 1) Click on “start” button.



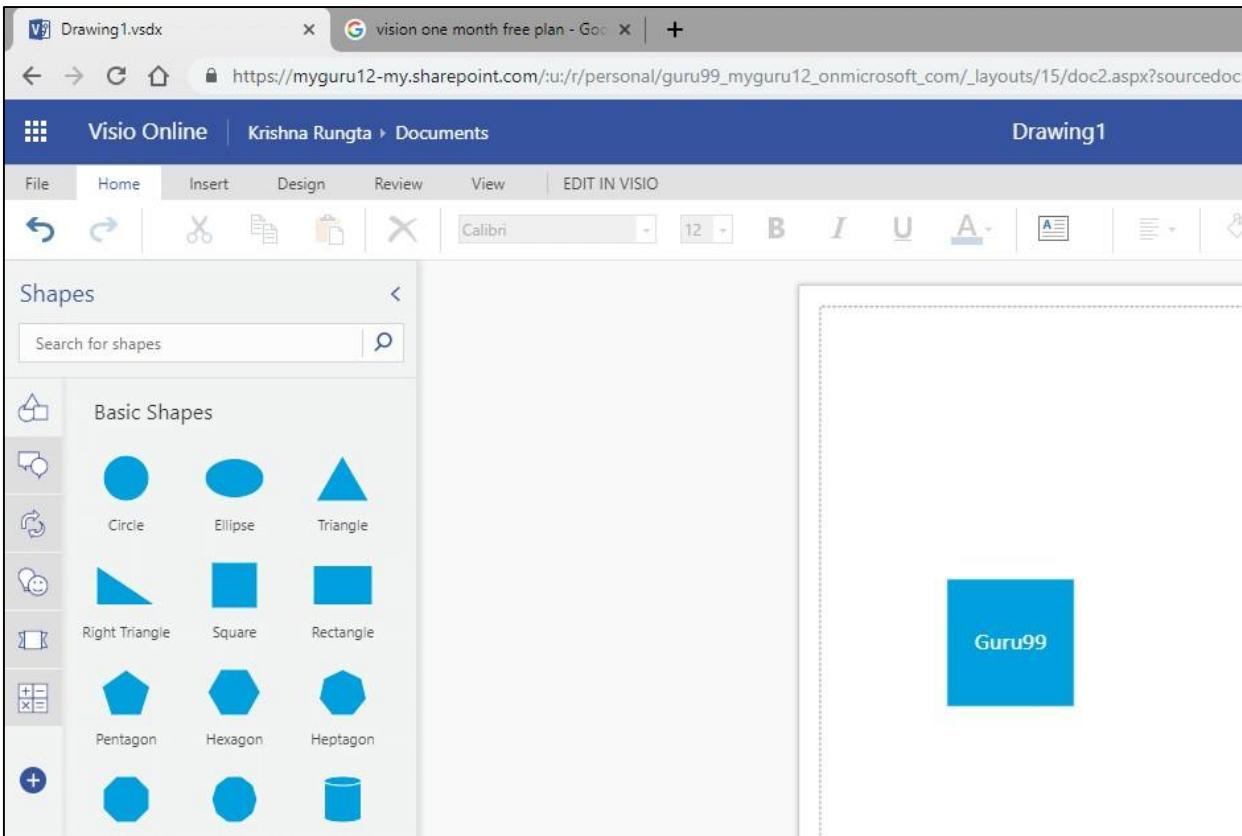
1. Type Visio in a “search” box.



You can also register with Office 365 and access Microsoft Visio Online.

The screenshot shows the Microsoft Visio Online interface. At the top, it displays a browser window with the URL <https://office.live.com/start/Visio.aspx?auth=2&nf=1>. The main content area is titled 'Visio' and features a 'Gallery' tab selected. Under 'Popular Diagrams', there are three cards: 'Basic Diagram', 'Basic Flowchart', and 'Cross Functional Flowchart'. Each card has a description, a preview image, and a 'Create' button. Below this, under 'Basic Diagram', there are five cards: 'Basic Diagram', 'Marketing Mix', 'Financial Ratios', 'Hub and Spoke Model', and 'Product Development'. Each card has a similar structure with a description, preview image, and 'Create' button. Navigation arrows are present on both sides of the diagram cards. At the bottom, there are links for 'Change language', 'Accessibility', 'Privacy and Cookies', 'Legal', 'Trademarks', and '© 2019 Microsoft'.

And create diagrams



How to use Microsoft Visio

Now in this Visio tutorials guide, we will learn how to use Microsoft Visio:

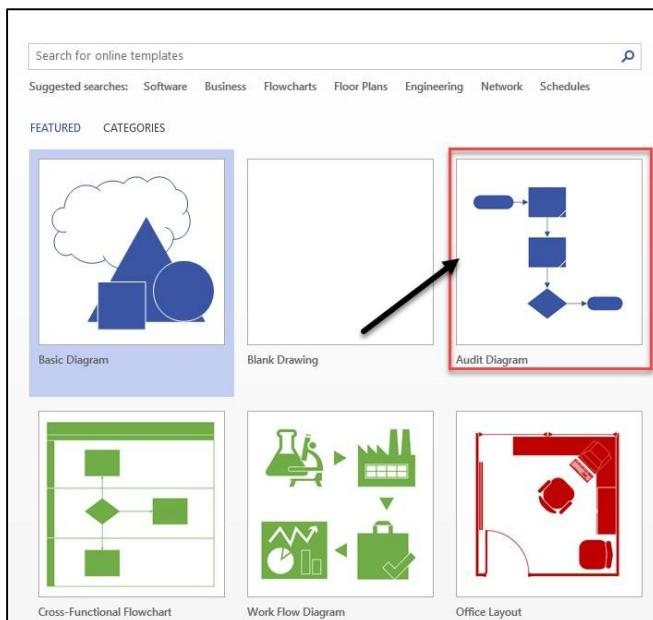
Here, are steps to plan and draw a basic Visio chart:

Step 1) Gather the suitable detail.

Decide what you're trying to accomplish and gather the suitable detail requirements to draw your diagram or chart.

Step 2) Determine chart type.

Determine which chart type is best suited to illustrate the information you want to convey. In our case, we will create a login flowchart.



Step 3) Open the template.

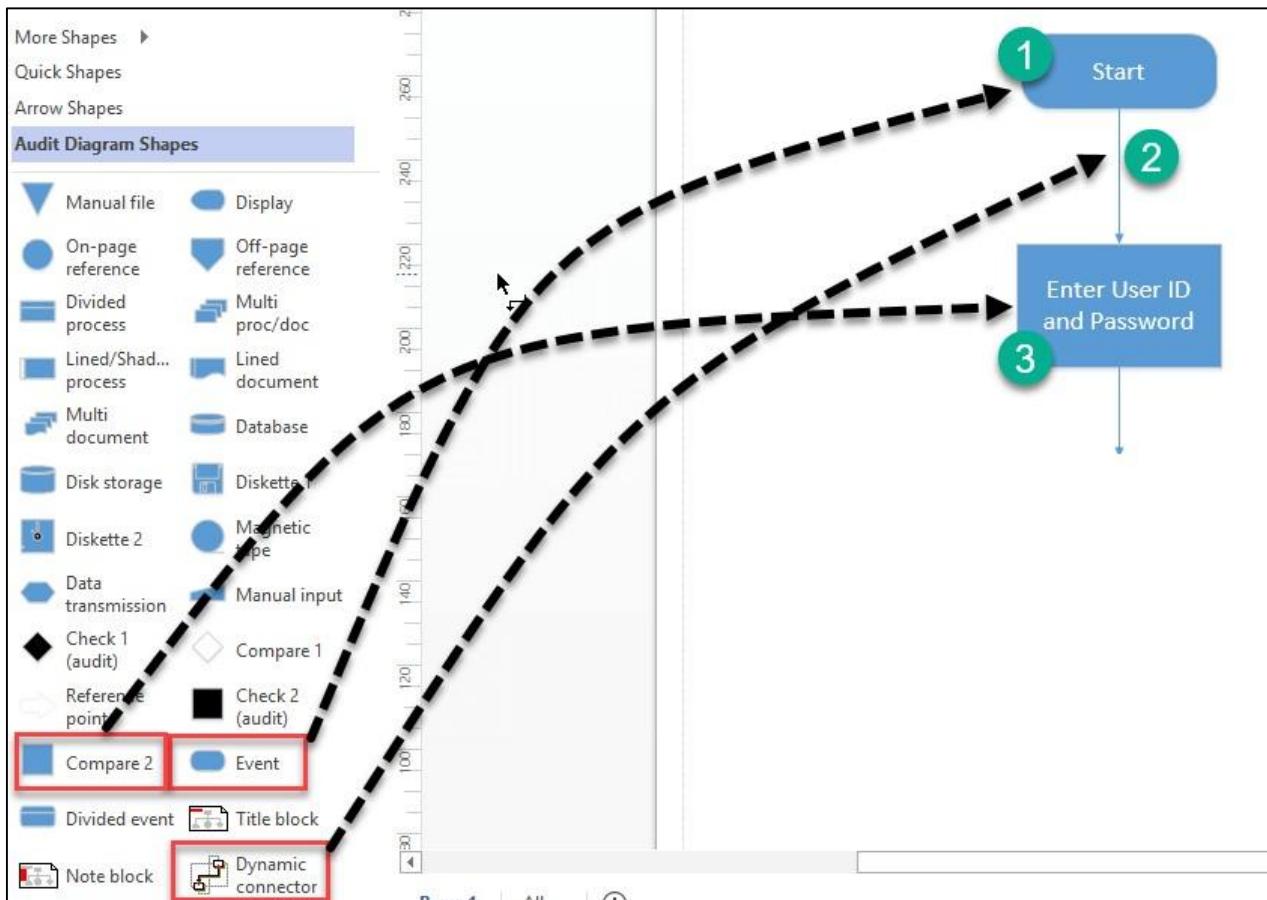
In Visio, select and open the template according to your chart type.

Each shape represents different things, which ranges from simple rectangles, squares, ovals, and arrows to hundreds of highly specialized shapes and symbols.

Step 4) Connect shapes.

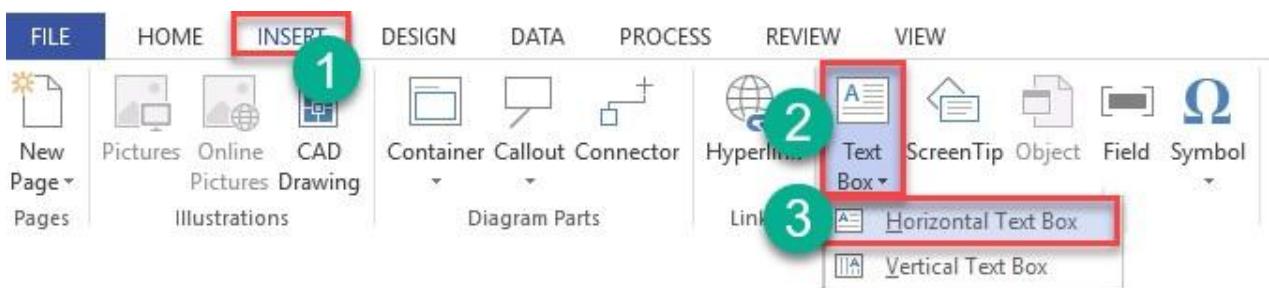
Drag and connect shapes or symbols to illustrate the items and their flow.

1. Drag terminator shape to drawing area.
2. Drag Compare2 shape.
3. Drag dynamic connector to drawing area and connect these two shapes.

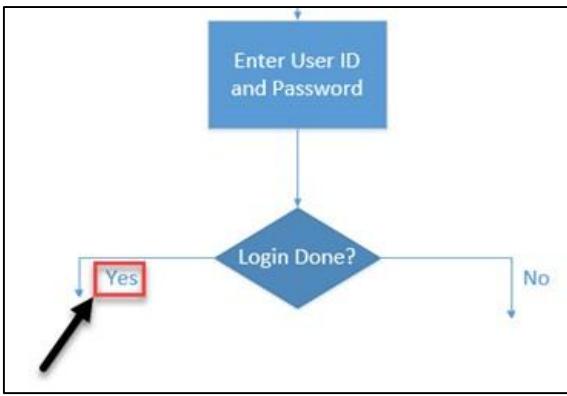


Step 5) Add a label.

To add a label, go to Insert tab then click on “Text Box” and select horizontal Text Box.



Click on drawing area to add label.



Step 6) Review your diagram.

Lastly, review your diagram and compare with your rough design.

Below is an example of a flowchart showing a login process?

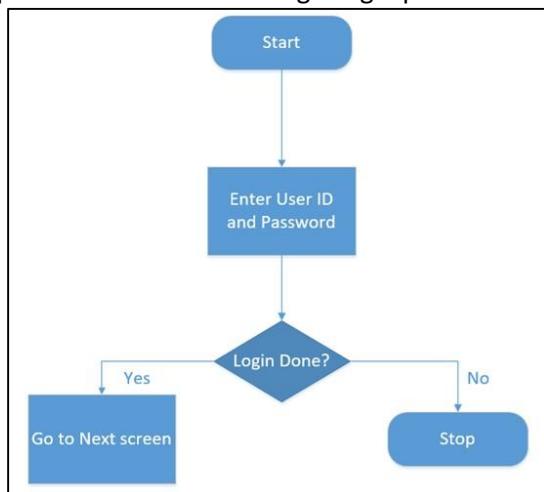


FIGURE 15.7 An example of a CASE tool: Visio Professional

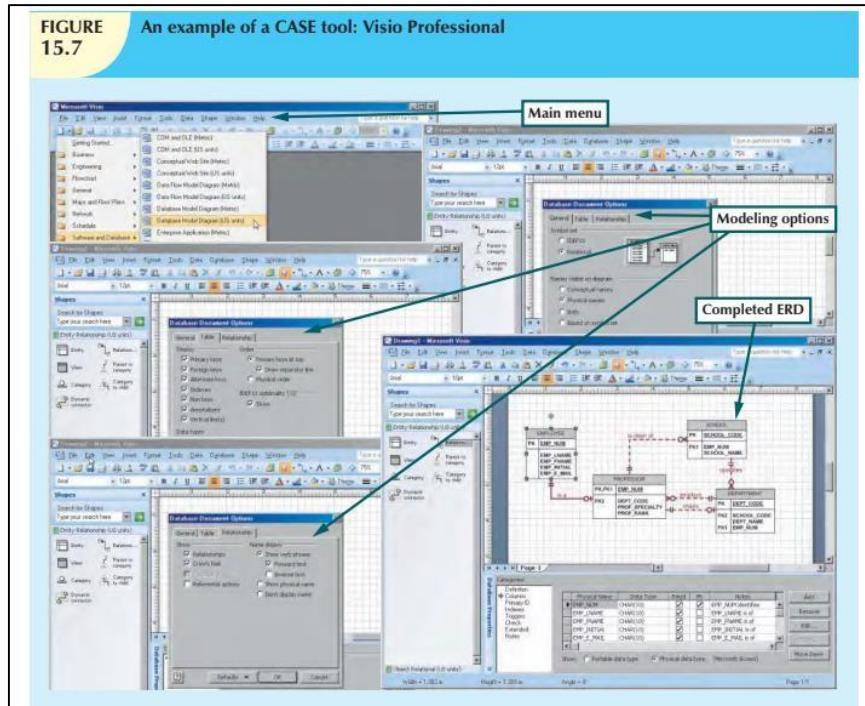


Figure: illustrates how Microsoft Visio Professional can be used to produce an ER diagram.

Shows a short list of the many available CASE tool vendors

TABLE
15.5**CASE Tools**

COMPANY	PRODUCT	WEB SITE
Casewise	Corporate Modeler Suite	www.casewise.com
Computer Associates	ERwin	www.ca.com/us/it-management-products.aspx
Embarcadero Technologies	ER/Studio	www.embarcadero.com/products/er_studio
Microsoft	Visio	office.microsoft.com/en-us/visio
Oracle	Designer	www.oracle.com/technology/products/designer
IBM	System Architect	www.telelogic.com/Products/systemarchitect/
Sybase	Power Designer	www.sybase.com/products/modelingdevelopment/powerdesigner
Visible	Visible Analyst	www.visible.com/Products/Analyst

Applications of Microsoft Visio

Here, are essential applications and functions of Microsoft Visio:

Usage	Description
Business Process Diagrams	It is one of the most popular used of Visio. It helps you to illustrate business processes.
Organization Charts	Organization charts are frequently used in the diagram for your business.
Brainstorming Diagram	A brainstorming diagram allows you to record and develop any set of related ideas or information like new strategies for business, book outlines, travel notes, meeting minutes, etc.
Building Plan	You can also build the most accurate building plans in Visio. This diagram also provides an overhead view of walls, doors, windows, and furniture.
Flowchart	A flowchart can be used to document and analyze a process; standardize a process for efficiency and quality. It also uses for training or understanding by other parts of the organization.
Analysis	Visio offers focus approach to create professional Data flow diagrams (DFDs) for all types of projects.
In business	You can draw Business Process Modeling and Notation shortly known as (BPMN). It is targeted at participants and other stakeholders in a business process.
In Chemical Engineering	Visio allows you to draw a Process Flow Diagram (PFD), which is a type of flowchart that explains the relationships between various components at an industrial plant.

EXPERIMENT NO. 12

```
create table sales(  
    continent varchar(20),  
    country varchar(20),  
    city varchar(20),  
    units_sold int  
);  
  
insert into sales values('North America','Canada','Toronto',10000);  
insert into sales values('North America','Canada','Montreal',5000);  
insert into sales values('North America','Canada','Vancouver',15000);  
insert into sales values('Asia','China','Hong Kong',7000);  
insert into sales values('Asia','China','Shanghai',3000);  
insert into sales values('Asia','Japan','Tokyo',5000);  
insert into sales values('Europe','UK','London',6000);  
insert into sales values('Europe','UK','Manchester',12000);  
insert into sales values('Europe','France','Paris',5000);  
  
select * from sales  
order by continent, country, city;  
  
//select continent, sum(units_sold)  
//from sales  
//group by continent;  
  
//select country, sum(units_sold)  
//from sales  
//group by country;  
  
//select city, sum(units_sold)  
//from sales  
//group by city;
```

-- drill down :

```
select continent, country, city, sum(units_sold)
from sales
group by grouping sets(continent, country, city);
```

-- RollUP :

```
select continent, country, city, sum(units_sold)
from sales
group by rollup(continent, country, city);
```

-- cube :

```
select continent, country, city, sum(units_sold)
from sales
group by cube(continent, country, city);
```

	CONTINENT	COUNTRY	CITY	TOTAL_UNITS
1	(null)	(null)	(null)	136000
2	(null)	(null)	Paris	10000
3	(null)	(null)	Tokyo	10000
4	(null)	(null)	London	12000
5	(null)	(null)	Motreal	10000
6	(null)	(null)	Toronto	20000
7	(null)	(null)	Shanghai	6000
8	(null)	(null)	Hong Kong	14000

	CONTINENT	COUNTRY	CITY	UNITS SOLD
1	Asia	China	Hong Kong	7000
2	Asia	China	Hong Kong	7000
3	Asia	China	Shanghai	3000
4	Asia	China	Shanghai	3000
5	Asia	Japan	Tokyo	5000
6	Asia	Japan	Tokyo	5000
7	Europe	France	Paris	5000
8	Europe	France	Paris	5000

	CONTINENT	COUNTRY	CITY	TOTAL_UNITS
2	North America	Canada	Motreal	10000
3	North America	Canada	Vancouver	30000
4	Asia	China	Hong Kong	14000
5	Asia	China	Shanghai	6000
6	Asia	Japan	Tokyo	10000
7	Europe	UK	London	12000
8	Europe	UK	Manchester	24000
9	Europe	France	Paris	10000

	CONTINENT	COUNTRY	CITY	TOTAL_UNITS
1	North America	Canada	Toronto	20000
2	North America	Canada	Motreal	10000
3	North America	Canada	Vancouver	30000
4	Asia	China	Hong Kong	14000
5	Asia	China	Shanghai	6000
6	Asia	Japan	Tokyo	10000
7	Europe	UK	London	12000
8	Europe	UK	Manchester	24000

	CONTINENT	COUNTRY	CITY	TOTAL_UNITS
1	(null)	(null)	(null)	136000
2	(null)	(null)	Paris	10000
3	(null)	(null)	Tokyo	10000
4	(null)	(null)	London	12000
5	(null)	(null)	Motreal	10000
6	(null)	(null)	Toronto	20000
7	(null)	(null)	Shanghai	6000
8	(null)	(null)	Hong Kong	14000

