20MCA134 – ADVANCED DATABASE MANAGEMENT SYSTEM LAB

Lab Report Submitted By

JUSTIN V KALAPPURA

Reg. No.: AJC21-2068

In Partial fulfillment for the Award of the Degree Of

MASTER OF COMPUTER APPLICATIONS (2 Year)
(MCA)
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY

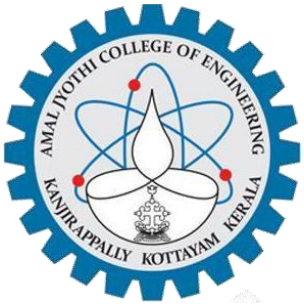
[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2021-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY



CERTIFICATE

This is to certify that the Lab report, **"20MCA134 ADVANCED DATABASE MANAGEMENT SYSTEM LAB"** is the bonafide work of JUSTIN V KALAPPURA**(Reg.No:AJC21-2068)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2021-22.

Grace Joseph
Lab In-Charge

Rev.Fr.Dr.Rubin Thottupuram Jose
Head of the Department

Internal Examiner

External Examiner

CONTENT

Experiment No.: 1

Name: Justin v kalappura

Roll No: 10

Batch: B

Date:08-04-2022

Aim

To study various DDL commands – CREATE, ALTER, DROP, TRUNGATE, RENAME

Questions

1.Create a table emp with attributes empno number (4) as primary key, ename char (10), hiredate, salary, commission

insert 5 rows of data

101	Ramesh	17-Jan 1980	5000	
102	Ajay	05-Jul 1985	5000	500
103	Ravi	12-Aug 1981	1500	
104	Nikesh	03-Mar 1983	3000	700
105	Ravi	05-jul 1985	3000	

2.Modifying the structure of tables

a. Add new columns: sal number(7,2)

b. Dropping a column from a table: sal

c. Modifying existing column :ename varchar2(15)

d. Renaming the tables: emp to emp1

e. truncating the tables:emp1

f. Destroying tables:emp

3.Create a table stud with sname varchar2(20) primary key, rollno number(10) not null,dob date not null

4.Create a table student as regno number (6), mark number (3) check constraint (mark >=0 and mark <=100));

5.Create a table cust with (custid int(6) constraint unique, name char(10)

Procedure

create database EMP;

use EMP;

1.CREATE TABLE emp (EMPNO INT (4) PRIMARY KEY, ENAME CHAR (10), HIREDATE DATE, SALARY INT (5), COMMISSION INT (5));

INSERT INTO emp (EMPNO, ENAME, HIREDATE, SALARY) VALUES (101,'RAMESH','1980-01-17',5000), (102,'AJAY','1985-07-05',5000), (103,'RAVI','1981-08-12',1500), (104,'Nikesh','1983-03-03',3000), (105,'Ravi','1985-07-05',3000);

select * from EMP;

Result Grid | Filter Rows: | Edit:

EMPNO	ENAME	HIREDATE	SALARY	COMMISSION
101	RAMESH	1980-01-17	5000	NULL
102	AJAY	1985-07-05	5000	NULL
103	RAVI	1981-08-12	1500	NULL
104	Nikesh	1983-03-03	3000	NULL
105	Ravi	1985-07-05	3000	NULL
NULL	NULL	NULL	NULL	NULL

EMP 1

2. a. ALTER TABLE emp ADD sal int;

select * from EMP;

Result Grid | Filter Rows: | Edit:

EMPNO	ENAME	HIREDATE	SALARY	COMMISSION	sal
101	RAMESH	1980-01-17	5000	NULL	NULL
102	AJAY	1985-07-05	5000	NULL	NULL
103	RAVI	1981-08-12	1500	NULL	NULL
104	Nikesh	1983-03-03	3000	NULL	NULL
105	Ravi	1985-07-05	3000	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL

EMP 2

b.ALTER TABLE emp DROP COLUMN sal;

select * from EMP;

Result Grid | Filter Rows: | Edit:

EMPNO	ENAME	HIREDATE	SALARY	COMMISSION
101	RAMESH	1980-01-17	5000	NULL
102	AJAY	1985-07-05	5000	NULL
103	RAVI	1981-08-12	1500	NULL
104	Nikesh	1983-03-03	3000	NULL
105	Ravi	1985-07-05	3000	NULL
NULL	NULL	NULL	NULL	NULL

EMP 3

c. ALTER TABLE emp MODIFY ename VARCHAR (15);

select * from EMP;

47 11:11:34 ALTER TABLE emp MODIFY ename VARCHAR(15) 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0

48 11:11:52 select * from EMP LIMIT 0, 1000 5 row(s) returned

Result Grid | Filter Rows: | Edit:

EMPNO	ename	HIREDATE	SALARY	COMMISSION
101	RAMESH	1980-01-17	5000	NULL
102	AJAY	1985-07-05	5000	NULL
103	RAVI	1981-08-12	1500	NULL
104	Nikesh	1983-03-03	3000	NULL
105	Ravi	1985-07-05	3000	NULL
NULL	NULL	NULL	NULL	NULL

EMP 4

d. RENAME table emp to emp1;

select * from EMP1;

49 11:12:49 RENAME table emp to emp1 0 row(s) affected

50 11:13:16 select * from EMP1 LIMIT 0, 1000 5 row(s) returned

Result Grid | Filter Rows: | Edit:

EMPNO	ename	HIREDATE	SALARY	COMMISSION
101	RAMESH	1980-01-17	5000	NULL
102	AJAY	1985-07-05	5000	NULL
103	RAVI	1981-08-12	1500	NULL
104	Nikesh	1983-03-03	3000	NULL
105	Ravi	1985-07-05	3000	NULL
NULL	NULL	NULL	NULL	NULL

EMP1 5

e. TRUNCATE TABLE emp1;

select * from EMP1;

Result Grid | Filter Rows: | Edit:

EMPNO	ename	HIREDATE	SALARY	COMMISSION
NULL	NULL	NULL	NULL	NULL

EMP1 6

f. DROP TABLE emp1;

✓	52	11:14:25	select *from EMP1 LIMIT 0, 1000	0 row(s) returned
✓	53	11:15:03	DROP TABLE emp1	0 row(s) affected

3.CREATE TABLE stud (sname varchar (20) PRIMARY KEY, rollno int NOT NULL, dob date NOT NULL);

SELECT * FROM stud;

Result Grid | Filter Rows:

sname	rollno	dob
NULL	NULL	NULL

stud 7 ×

4.Create table student (regno int, mark int (3) constraint b check (mark >=0 and mark <=100));

SELECT * from STUDENT;

Result Grid | Filter Rows:

regno	mark

STUDENT 8 ×

5.CREATE TABLE cust (custid int (6) UNIQUE, name char(10));

select * from cust;

Result Grid | Filter Rows: | Export:

custid	name

cust 11 ×

Experiment No.: 2

Name: Justin v kalappura

Roll No: 10

Batch: B

Date:25-03-2022

Aim

To familiarize DML commands SELECT, INSERT, UPDATE, DELETE

Questions

Create the following Tables and insert values.

Table 1: DEPOSIT

ACTNO VARCHAR2 (5) PRIMARY KEY, FIRST LETTER MUST START WITH 'D' CNAME VARCHAR2 (15) FOREIGN KEY REFERENCES CUSTOMER
BNAME VARCHAR2 (20) FOREIGN KEY REFERENCES
BRANCH AMOUNT NUMBER (8,2) NOT NULL, CANNOT BE 0
ADATE DATE

Table 2: BRANCH

BNAME VARCHAR2(20) PRIMARY KEY CITY VARCHAR2(30) NOT NULL, any one of NAGPUR, DELHI, BANGALORE, BOMBAY

Table 3: CUSTOMER

CNAME VARCHAR2(15) PRIMARY KEY
CITY VARCHAR (20) NOT NULL,

Table 4: BORROW

LOANNO VARCHAR2(8) PRIMARY KEY / FIRST LETTER MUST START WITH 'L'
CNAME VARCHAR2(15) FOREIGN KEY REFERENCES CUSTOMER
BNAME VARCHAR2(20) FOREIGN KEY REFERENCES BRANCH
AMOUNT NUMBER (8,2) NOT NULL, CANNOT BE 0

1. List all data from table deposite
2. List all data from borrow
3. List all data from customer
4. List all data from branch
5. Give account no and amount of deposite
6. Give customer name and account no of depositors
7. Give name of customers
8. Give name of branches
9. Give name of borrows
10. Give names of customer living in city Nagpur

11. Give names of depositors having amount greater than 4000
12. Give account date of Anil
13. Give name of all branches located in Bombay
14. Give name of borrower having loan number l205
15. Give names of depositors having account at VRCE
16. Give names of all branched located in city Delhi
17. Give name of the customers who opened account date ‘1-12-96’
18. Give account no and deposit amount of customers having account opened between dates ‘1-12-96’ and ‘1-5-96’
19. Give name of the city where branch KAROLBAGH is located
20. Give details of customer ANIL

Procedure

CREATE DATABASE ddl;

DEPOSIT

CREATE TABLE DEPOSIT (ACTNO VARCHAR (5) CHECK (ACTNO LIKE 'D%') PRIMARY KEY, CNAME VARCHAR (15) REFERENCES CUSTOMER(CNAME), BNAME VARCHAR (20) REFERENCES BRANCH(BNAME)AMOUNT FLOAT (8) CHECK(AMOUNT>0) NOT NULL, ADATE DATE);

INSERT INTO DEPOSIT VALUES('D100','ANIL','VRCE',1000.00,'1995-03-01') ('D101', 'SUNIL', 'ANJNI',500.00,'1996-01-04'), ('D102','MEHUL','KAROLBAGH',3500.00,'1995-11-17'), ('D104', 'MADHURI','CHANDNI',1200.00,'1995-10-17'), ('D105','PRAMOD','MG ROAD',3000.00,'1996-03-27'), ('D106','SANDIP','ANDHERI',2000.00,'1996-03-31'), ('D107','SHIVANI','VIRAR',1000.00,'1995-09-05'), ('D108','KRANTI','NEHRU PLACE',5000.00,'1995-07-02'), ('D109', 'MINU',' POWAI', 7000.00,'1995-08-10');

BRANCH

CREATE TABLE BRANCH (BNAME VARCHAR (20) PRIMARY KEY, CITY VARCHAR (30) CHECK (CITY IN('NAGPUR','DELHI','BANGALORE','BOMBAY')) NOT NULL);

INSERT INTO BRANCH VALUES('VRCE','NAGPUR'), ('AJNI','NAGPUR'), ('KAROLBAGH', 'DELHI'), ('CHANDNI','DELHI'), ('DHARAMPETH','NAGPUR'), ('MG ROAD', 'BANGALORE'), ('ANDHERI','BOMBAY'), ('NEHRU PALACE','DELHI'), ('POWAI','BOMBAY');

CUSTOMER

CREATE TABLE CUSTOMER (CNAME VARCHAR (15) PRIMARY KEY, CITY VARCHAR (20) NOT NULL);

INSERT INTO CUSTOMER VALUES ('ANIL','CALCUTTA'), ('SUNIL','DELHI'), ('MEHUL', 'BARODA'), ('MANDAR','PATINA'), ('MADHURI','NAGPUR'), ('PRAMOD','NAGPUR'), ('SANDIP','SURAT'), ('SHIVANI','BOMBAY'), ('KRANIT','BOMBAY'), ('NAREN','BOMBAY');

BORROW

CREATE TABLE BORROW (LOANNO VARCHAR (8) CHECK (LOANNO LIKE 'L%') PRIMARY KEY, CNAME VARCHAR (15) REFERENCES CUSTOMER(CNAME) BNAME VARCHAR (20) REFERENCES BRANCH(BNAME)AMOUNT FLOAT (8) CHECK(AMOUNT>0) NOT NULL);

INSERT INTO BORROW VALUES('L201','ANIL','VRCE',1000.00), ('L206', 'MEHUL', 'AJNI', 5000.00), ('L311','SUNIL','DHARAMPETH',3000.00), ('L321','MADHURI','ANDHERI',2000.00), ('L371','PRAMOD','VIRAR',8000.00), ('L481','KRANTI','NEHRU PLACE',3000.00);

SELECT * FROM CUSTOMER;

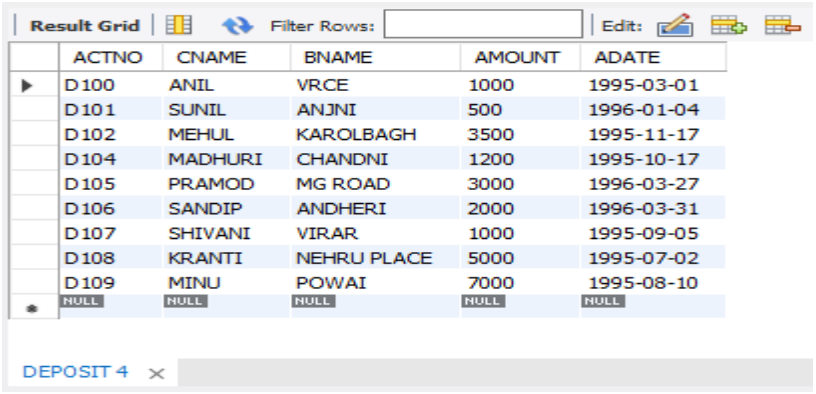
SELECT * FROM BRANCH;

SELECT * FROM DEPOSIT;

SELECT * FROM BORROW;

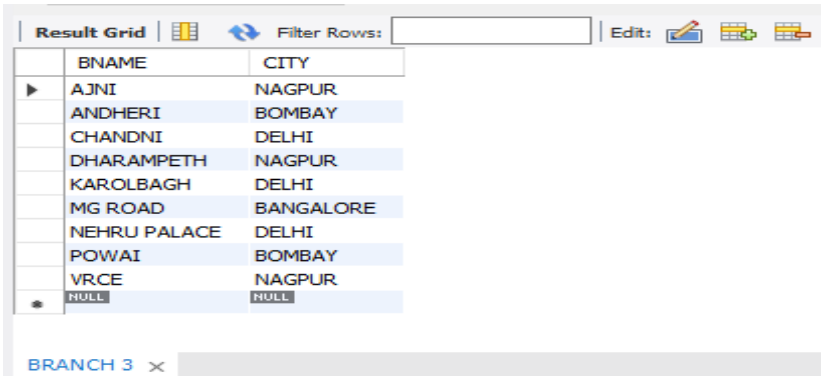
Output Screenshot

DEPOSIT



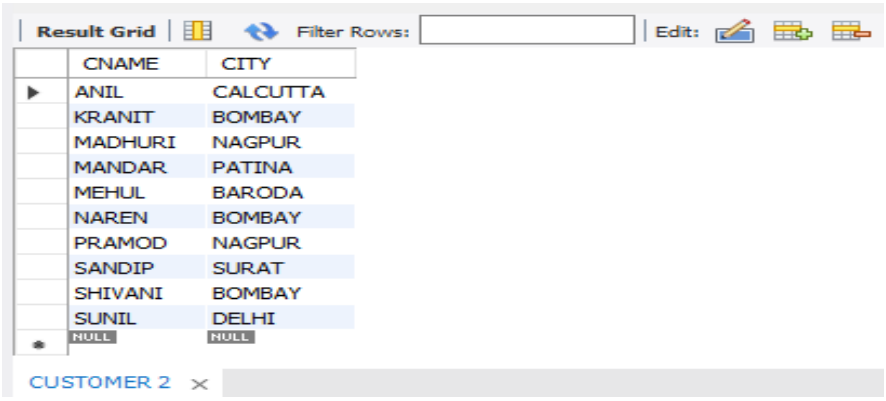
ACTNO	CNAME	BNAME	AMOUNT	ADATE
D100	ANIL	VRCE	1000	1995-03-01
D101	SUNIL	ANJNI	500	1996-01-04
D102	MEHUL	KAROLBAGH	3500	1995-11-17
D104	MADHURI	CHANDNI	1200	1995-10-17
D105	PRAMOD	MG ROAD	3000	1996-03-27
D106	SANDIP	ANDHERI	2000	1996-03-31
D107	SHIVANI	VIRAR	1000	1995-09-05
D108	KRANTI	NEHRU PLACE	5000	1995-07-02
D109	MINU	POWAI	7000	1995-08-10
NULL	NULL	NULL	NULL	NULL

BRANCH



BNAME	CITY
AJNI	NAGPUR
ANDHERI	BOMBAY
CHANDNI	DELHI
DHARAMPETH	NAGPUR
KAROLBAGH	DELHI
MG ROAD	BANGALORE
NEHRU PALACE	DELHI
POWAI	BOMBAY
VRCE	NAGPUR
NULL	NULL

CUSTOMER

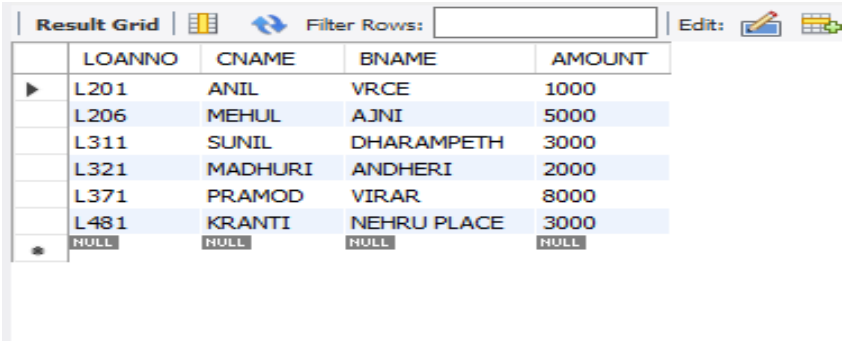


Result Grid | Filter Rows: | Edit:

CNAME	CITY
ANIL	CALCUTTA
KRANIT	BOMBAY
MADHURI	NAGPUR
MANDAR	PATINA
MEHUL	BARODA
NAREN	BOMBAY
PRAMOD	NAGPUR
SANDIP	SURAT
SHIVANI	BOMBAY
SUNIL	DELHI
NULL	NULL

CUSTOMER 2

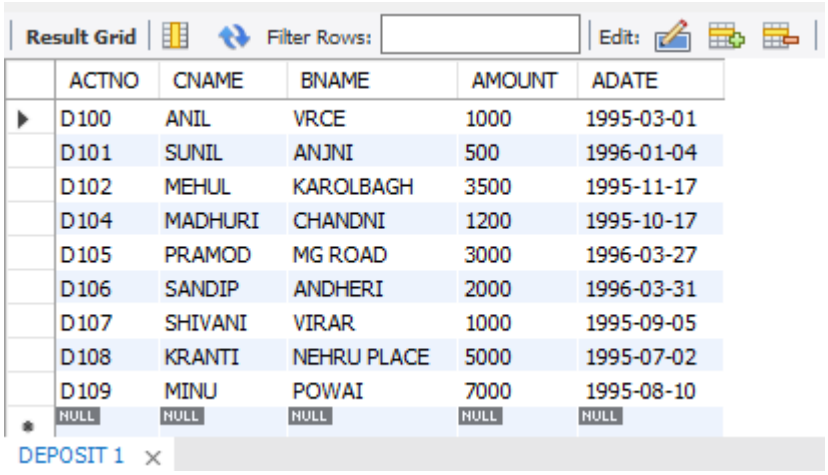
BORROW



Result Grid | Filter Rows: | Edit:

LOANNO	CNAME	BNAME	AMOUNT
L201	ANIL	VRCE	1000
L206	MEHUL	AJNI	5000
L311	SUNIL	DHARAMPETH	3000
L321	MADHURI	ANDHERI	2000
L371	PRAMOD	VIRAR	8000
L481	KRANTI	NEHRU PLACE	3000
NULL	NULL	NULL	NULL

1. select * from DEPOSIT;

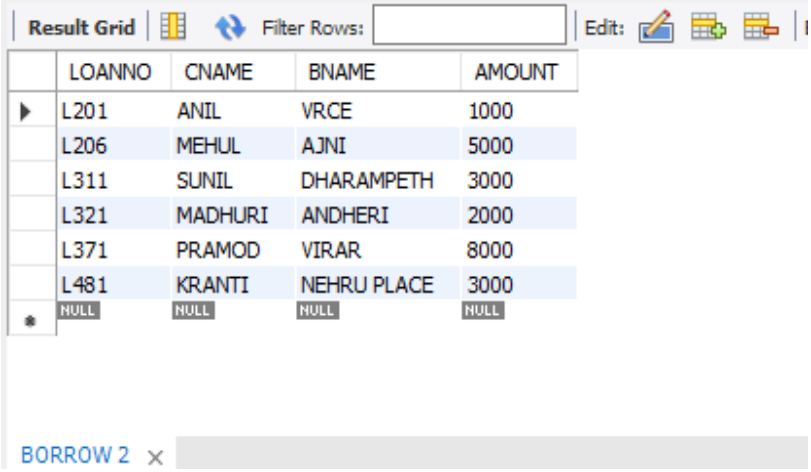


Result Grid | Filter Rows: | Edit:

ACTNO	CNAME	BNAME	AMOUNT	ADATE
D100	ANIL	VRCE	1000	1995-03-01
D101	SUNIL	ANJNI	500	1996-01-04
D102	MEHUL	KAROLBAGH	3500	1995-11-17
D104	MADHURI	CHANDNI	1200	1995-10-17
D105	PRAMOD	MG ROAD	3000	1996-03-27
D106	SANDIP	ANDHERI	2000	1996-03-31
D107	SHIVANI	VIRAR	1000	1995-09-05
D108	KRANTI	NEHRU PLACE	5000	1995-07-02
D109	MINU	POWAI	7000	1995-08-10
NULL	NULL	NULL	NULL	NULL

DEPOSIT 1

2. select * from BORROW;

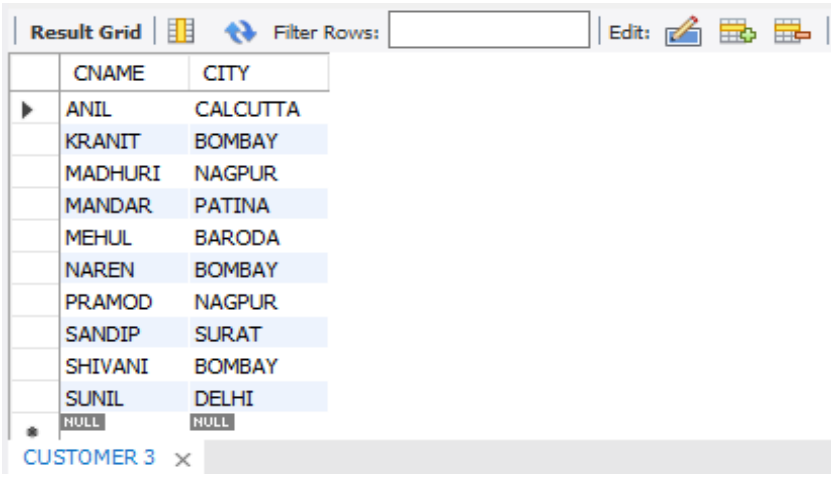


Result Grid | Filter Rows: | Edit:

LOANNO	CNAME	BNAME	AMOUNT
L201	ANIL	VRCE	1000
L206	MEHUL	AJNI	5000
L311	SUNIL	DHARAMPETH	3000
L321	MADHURI	ANDHERI	2000
L371	PRAMOD	VIRAR	8000
L481	KRANTI	NEHRU PLACE	3000
NULL	NULL	NULL	NULL

BORROW 2

3. select * from CUSTOMER;

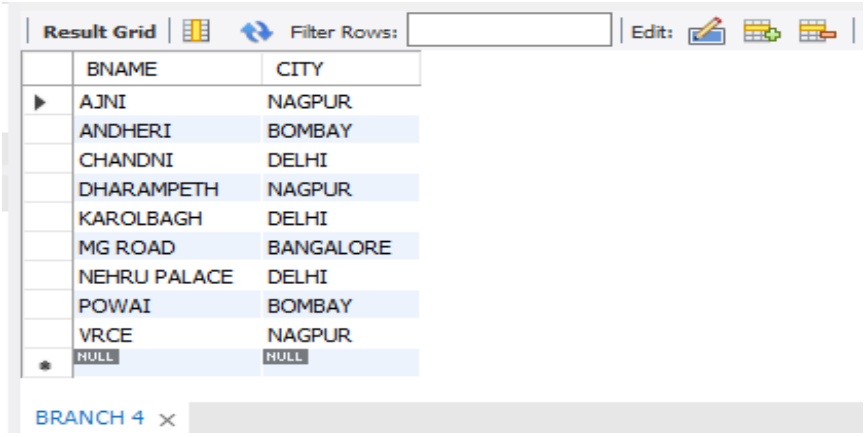


Result Grid | Filter Rows: | Edit:

CNAME	CITY
ANIL	CALCUTTA
KRANIT	BOMBAY
MADHURI	NAGPUR
MANDAR	PATINA
MEHUL	BARODA
NAREN	BOMBAY
PRAMOD	NAGPUR
SANDIP	SURAT
SHIVANI	BOMBAY
SUNIL	DELHI
NULL	NULL

CUSTOMER 3

4. select * from BRANCH;

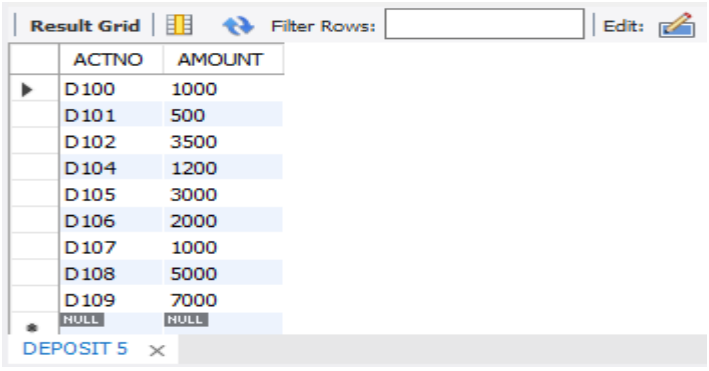


Result Grid | Filter Rows: | Edit:

BNAME	CITY
AJNI	NAGPUR
ANDHERI	BOMBAY
CHANDNI	DELHI
DHARAMPETH	NAGPUR
KAROLBAGH	DELHI
MG ROAD	BANGALORE
NEHRU PALACE	DELHI
POWAI	BOMBAY
VRCE	NAGPUR
NULL	NULL

BRANCH 4

5. select ACTNO, AMOUNT from DEPOSIT;

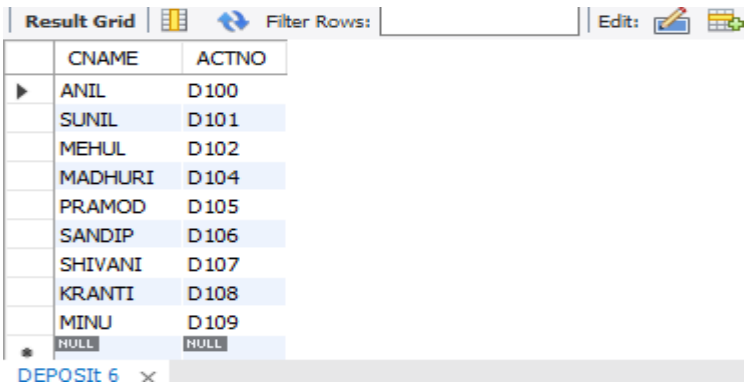


Result Grid | Filter Rows: | Edit:

ACTNO	AMOUNT
D100	1000
D101	500
D102	3500
D104	1200
D105	3000
D106	2000
D107	1000
D108	5000
D109	7000
NULL	NULL

DEPOSIT 5

06. select CNAME, ACTNO from DEPOSIT;

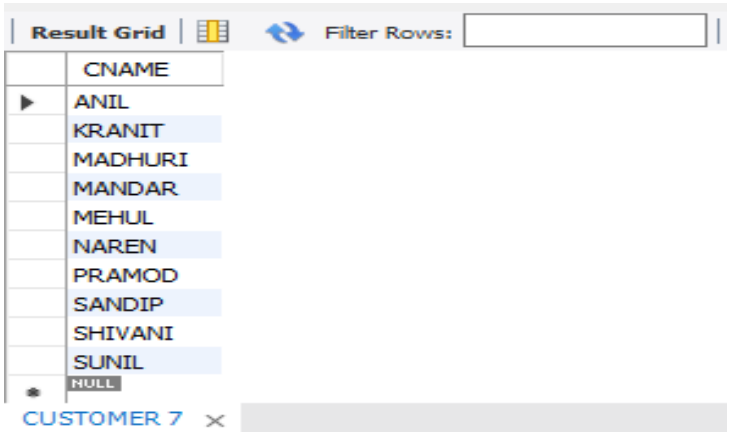


Result Grid | Filter Rows: | Edit:

CNAME	ACTNO
ANIL	D100
SUNIL	D101
MEHUL	D102
MADHURI	D104
PRAMOD	D105
SANDIP	D106
SHIVANI	D107
KRANTI	D108
MINU	D109
NULL	NULL

DEPOSIt 6

7. select CNAME from CUSTOMER;

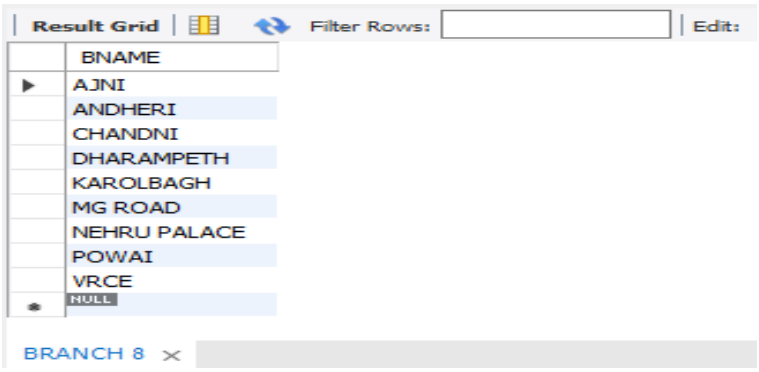


Result Grid | Filter Rows:

CNAME
ANIL
KRANIT
MADHURI
MANDAR
MEHUL
NAREN
PRAMOD
SANDIP
SHIVANI
SUNIL
NULL

CUSTOMER 7

8. select BNAME from BRANCH;

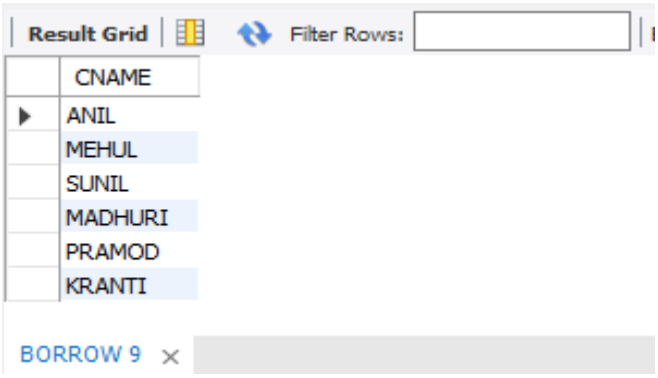


Result Grid | Filter Rows: | Edit:

BNAME
AJNI
ANDHERI
CHANDNI
DHARAMPETH
KAROLBAGH
MG ROAD
NEHRU PALACE
POWAI
VRCE
NULL

BRANCH 8

9. select CNAME from BORROW;

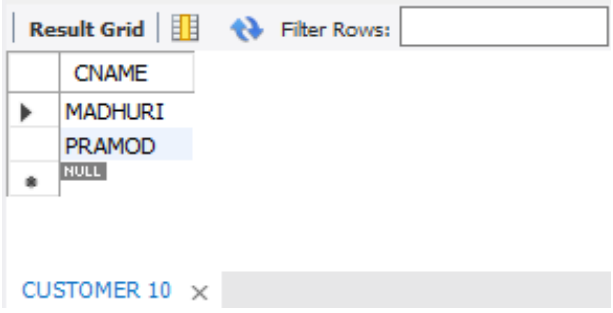


Result Grid | Filter Rows:

CNAME
ANIL
MEHUL
SUNIL
MADHURI
PRAMOD
KRANTI

BORROW 9

10. nselect CNAME from CUSTOMER where CITY='NAGPUR';

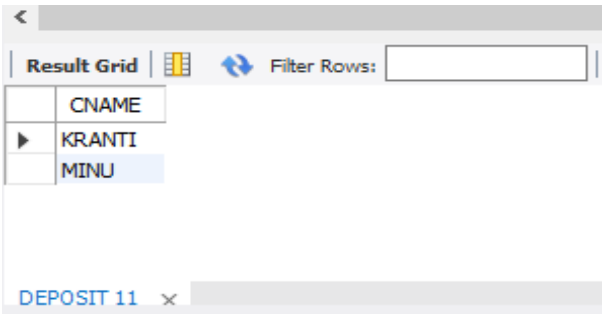


Result Grid | Filter Rows:

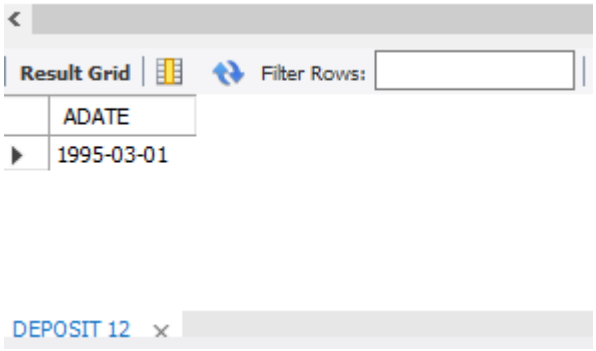
CNAME
MADHURI
PRAMOD
NULL

CUSTOMER 10

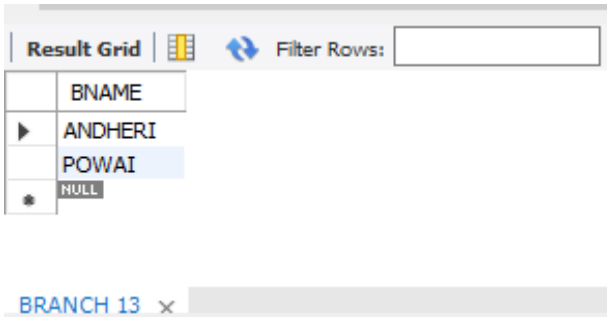
11. select CNAME from DEPOSIT where AMOUNT>4000;



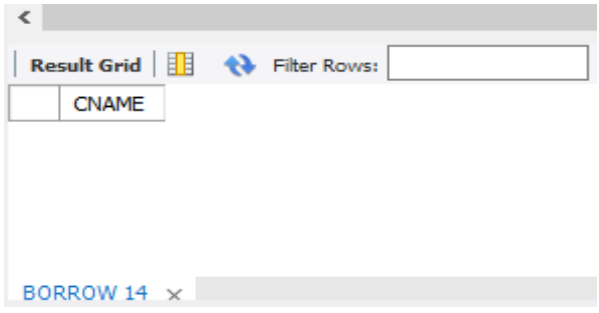
12. select ADATE from DEPOSIT where CNAME='ANIL';



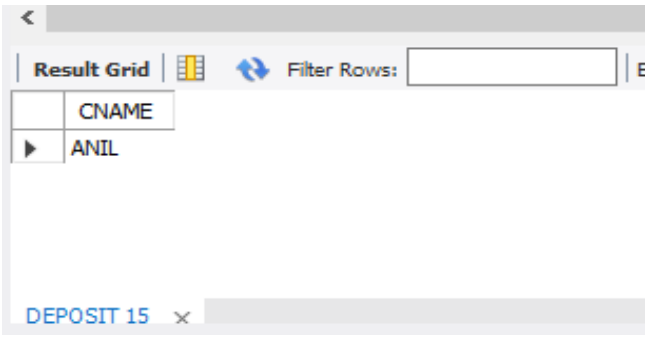
13. SELECT BNAME from BRANCH where CITY='Bombay';



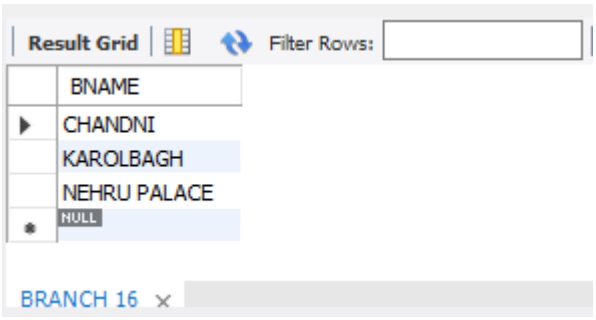
14. SELECT CNAME from BORROW where LOANNO='L205';



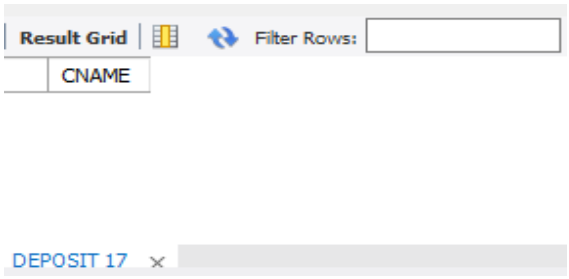
15. SELECT CNAME from DEPOSIT WHERE BNAME='VRCE';



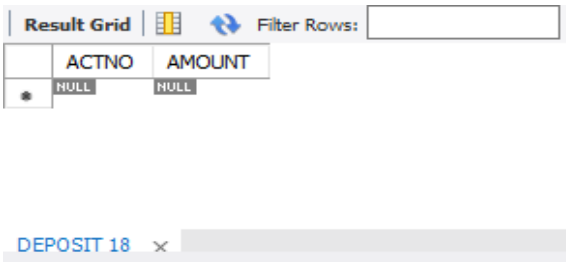
16. SELECT BNAME from BRANCH WHERE CITY='Delhi';



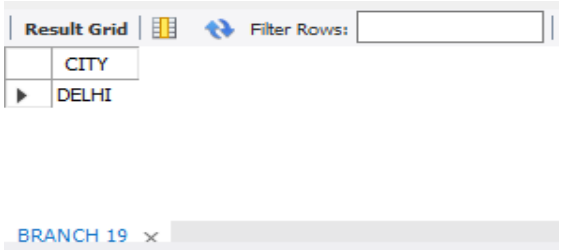
17. SELECT CNAME from DEPOSIT WHERE ADATE='1996-12-1';



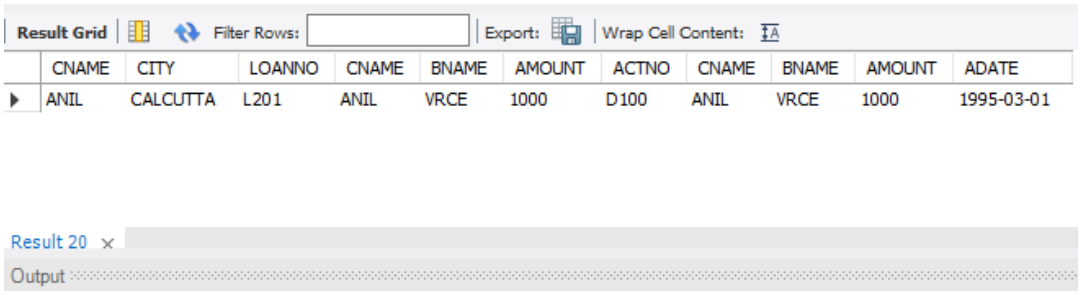
18. SELECT ACTNO, AMOUNT from DEPOSIT WHERE ADATE BETWEEN '1996-12-1' AND '1996-05-1';



19. SELECT CITY from BRANCH WHERE BNAME='KAROLBAGH';



20. SELECT * from CUSTOMER join borrow on CUSTOMER.CNAME=BORROW.CNAME join deposit on DEPOSIT.CNAME=BORROW.CNAME WHERE CUSTOMER.CNAME='ANIL';



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CNAME	CITY	LOANNO	CNAME	BNAME	AMOUNT	ACTNO	CNAME	BNAME	AMOUNT	ADATE
ANIL	CALCUTTA	L201	ANIL	VRCE	1000	D100	ANIL	VRCE	1000	1995-03-01

Result 20 ×

Output

Questions

Name: Justin v kalappura

Roll No: 10

Batch: B

Date: 19-04-2022

1.List total loan

2.List total deposit

3.List total loan taken from KAROLBAGH branch

4.List total deposit of customers having account date later than 1-Jan-96

5.List total deposit of customers living in city NAGPUR

6.List maximum deposit of customer living in Bombay

7.List total deposit of customer having branch in BOMBAY

8.Count total number of branch cities

9.Count total number of customers cities

10.Give branch names and branch wise deposit

11.Give city wise name and branch wise deposit

12.Give the branch wise loan of customer living in NAGPUR

13.Count total number of customers

14.Count total number of depositors branch wise

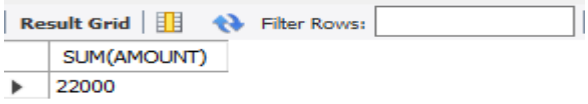
15.Count total number of depositors branch wise

16.Give maximum loan from branch VRCE

17.Give the number of customers who are depositors as well as borrowers

Procedure

1. SELECT SUM(AMOUNT) FROM BORROW;

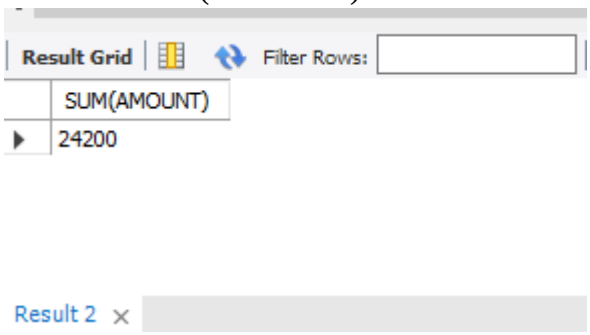


Result Grid | Filter Rows:

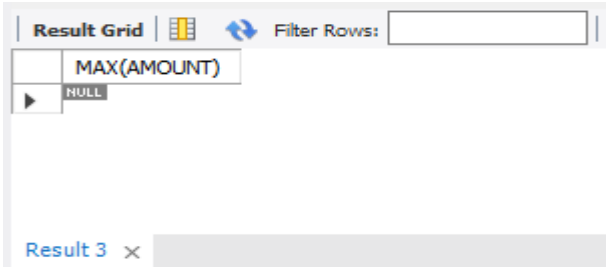
SUM(AMOUNT)
22000

Result 1 ×

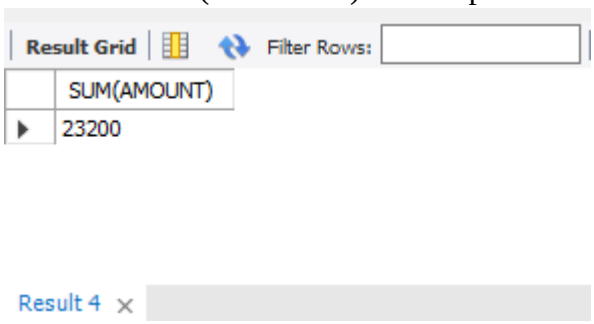
2. SELECT SUM(AMOUNT) FROM DEPOSIT;



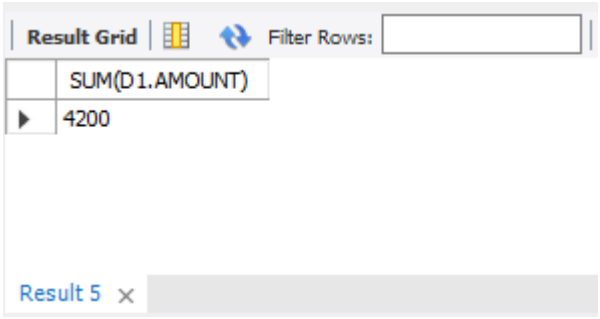
3. SELECT MAX(AMOUNT) FROM BORROW WHERE BNAME ='KAROLBAGH';



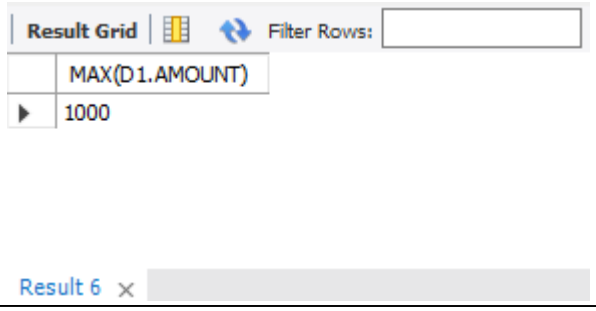
4. SELECT SUM(AMOUNT) from deposit where adate>'1995-03-01';



5. SELECT SUM(D1.AMOUNT) FROM DEPOSIT D1 , CUSTOMER C1 WHERE C1.CITY = 'NAGPUR' AND C1.CNAME = D1.CNAME;



6. SELECT MAX(D1.AMOUNT) FROM DEPOSIT D1 , CUSTOMER C1 WHERE C1.CITY = 'Bombay' AND C1.CNAME = D1.CNAME;



7. SELECT SUM(AMOUNT) from deposit,BRANCH where city='BOMBAY';

Result Grid | Filter Rows:

SUM(AMOUNT)
48400

Result 7

8. SELECT COUNT(DISTINCT(CITY)) FROM BRANCH ;

Result Grid | Filter Rows:

COUNT(DISTINCT(CITY))
4

Result 8

9. SELECT count(city) from CUSTOMER;

Result Grid | Filter Rows:

count(city)
10

Result 9

10. SELECT BNAME , SUM(AMOUNT) FROM DEPOSIT GROUP BY BNAME;

Result Grid | Filter Rows: | Export:

BNAME	SUM(AMOUNT)
VRCE	1000
ANJNI	500
KAROLBAGH	3500
CHANDNI	1200
MG ROAD	3000
ANDHERI	2000
VIRAR	1000
NEHRU PLACE	5000
POWAI	7000

Result 10

11. SELECT C1.CITY , SUM(D1.AMOUNT) FROM CUSTOMER C1 , DEPOSIT D1 WHERE D1.CNAME = C1.CNAME GROUP BY C1.CITY;

Result Grid | Filter Rows: | Export:

CITY	SUM(D1.AMOUNT)
CALCUTTA	1000
DELHI	500
BARODA	3500
NAGPUR	4200
SURAT	2000
BOMBAY	1000

Result 11

12. SELECT BNAME , SUM(AMOUNT) FROM BORROW

Result Grid | Filter Rows: | Export:

BNAME	SUM(AMOUNT)
VRCE	2000
AJNI	10000
DHARAMPETH	6000
ANDHERI	4000
VIRAR	16000
NEHRU PLACE	6000

Result 12

13. OW,CUSTOMER WHERE city ='NAGPUR' GROUP BY BNAME;

Result Grid | Filter Rows:

count(cname)
10

Result 13

14. SELECT count(cname) from CUSTOMER;

Result Grid | Filter Rows: | Export:

BNAME	count(*)
VRCE	1
ANJNI	1
KAROLBAGH	1
CHANDNI	1
MG ROAD	1
ANDHERI	1
VIRAR	1

Result 14

15. SELECT BNAME, count(*) from DEPOSIT, CUSTOMER where deposit.CNAME CUSTOMER.CNAME group by BNAME;

Result Grid | Filter Rows: | Export:

BNAME	count(*)
VRCE	1
ANJNI	1
KAROLBAGH	1
CHANDNI	1
MG ROAD	1
ANDHERI	1
VIRAR	1

Result 15

16. SELECT BNAME, count(*) from DEPOSIT, CUSTOMER where deposit.CNAME = CUSTOMER.CNAME group by BNAME;

Result Grid | Filter Rows:

MAX(AMOUNT)
1000

Result 16

17. select count(customer.CNAME) from customer where customer.CNAME IN (select deposit.cname from deposit) and customer.CNAME IN (select borrow.cname from borrow);

Result Grid | Filter Rows:

count(customer.CNAME)
5

Result 17

Experiment No.: 3

Name: Justin v kalappura

Roll No: 10

Batch: B

Date:19-04-2022

Aim

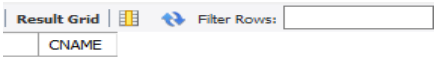
To familiarize with set operations

Questions

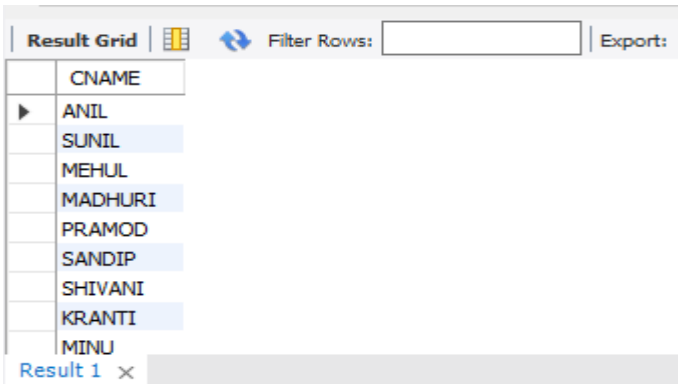
1. List all the customers who are depositors but not borrowers.
2. List all the customers who are both depositors and borrowers
3. List all the depositors having deposit in all the branches where Sunil is having Account
4. List all the customers living in city NAGPUR and having branch city BOMBAY or DELHI
5. List all the depositors living in city NAGPUR
6. List all the depositors living in the city NAGPUR and having branch in city BOMBAY
7. List the branch cities of Anil and Sunil
8. List the customers having deposit greater than 1000 and loan less than 10000.
9. List the cities of depositors having branch VRCE.
10. List the depositors having amount less than 1000 and living in the same city as Anil
11. List all the cities where branches of Anil and Sunil are locate
12. List the amount for the depositors living in the city where Anil is living

Procedure

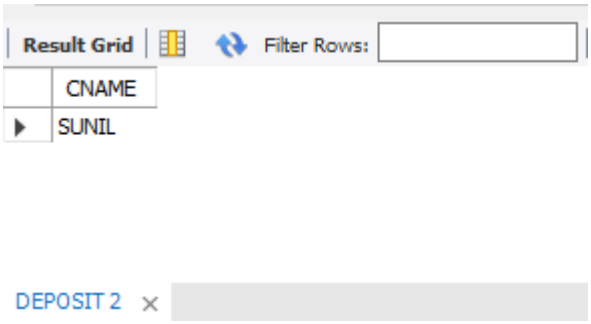
1. SELECT CNAME FROM DEPOSIT WHERE NOT EXISTS (SELECT CNAME FROM BORROW);



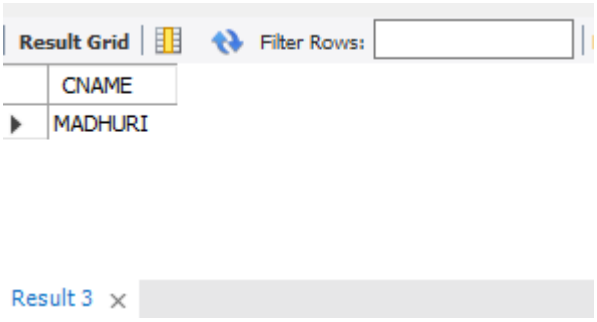
2. SELECT CNAME FROM DEPOSIT UNION (SELECT CNAME FROM BORROW);



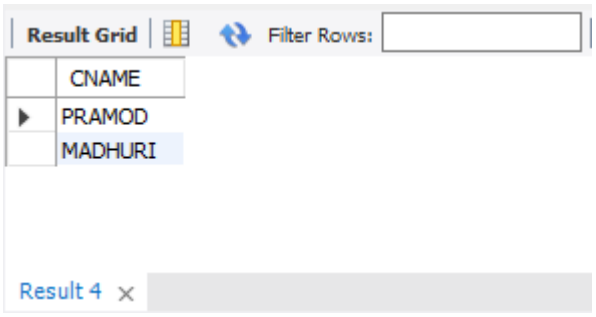
3. SELECT D1.CNAME FROM DEPOSIT D1 WHERE D1.BNAME IN (SELECT D2.BNAME FROM DEPOSIT D2 WHERE D2.CNAME = 'SUNIL');



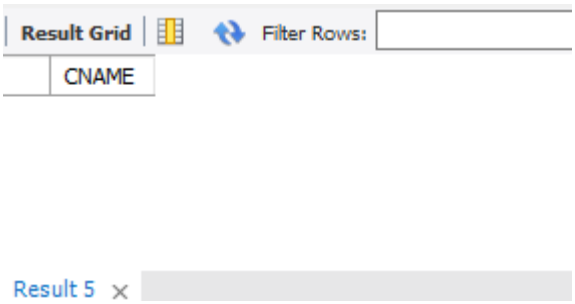
4. SELECT C1.CNAME FROM CUSTOMER C1,DEPOSIT D1, BRANCH B1 WHERE C1.CITY = 'NAGPUR' AND C1.CNAME = D1.CNAME AND D1.BNAME = B1.BNAME AND B1.CITY IN ('BOMBAY','DELHI');



5. SELECT Distinct(CUSTOMER.CNAME) from CUSTOMER,DEPOSIT WHERE City='NAGPUR';



6. SELECT C1.CNAME FROM CUSTOMER C1,DEPOSIT D1, BRANCH B1 WHERE C1.CITY = 'NAGPUR' AND C1.CNAME = D1.CNAME AND D1.BNAME = B1.BNAME AND B1.CITY IN ('BOMBAY');



7. SELECT B1.CITY FROM DEPOSIT D1, BRANCH B1 WHERE D1.BNAME = B1.BNAME AND D1.CNAME IN ('SUNIL' ,'ANIL');

Result Grid | Filter Rows:

CITY
NAGPUR

Result 6

8. SELECT DISTINCT D1.CNAME FROM deposit D1, borrow B1 WHERE D1.AMOUNT>1000 AND B1.AMOUNT<10000;

Result Grid | Filter Rows:

CNAME
MEHUL
MADHURI
PRAMOD
SANDIP
KRANTI
MINU

Result 7

9. SELECT B1.CITY FROM deposit D1, branch B1 WHERE D1.BNAME=B1.BNAME AND B1.BNAME='VRCE';

Result Grid | Filter Rows:

CITY
NAGPUR

Result 8

10. SELECT D1.CNAME FROM deposit D1, customer C1 WHERE AMOUNT<1000 AND C1.CITY=(C1.CNAME='ANIL');

Result Grid | Filter Rows:

CNAME
SUNIL
SUNIL
SUNIL
SUNIL
SUNIL
SUNIL
SUNIL
SUNIL
SUNIL

Result 9

11. SELECT B1.CITY FROM BRANCH B1 WHERE B1.BNAME IN (SELECT D1.BNAME FROM DEPOSIT D1 WHERE D1.CNAME IN ('ANIL','SUNIL'));

Result Grid Filter Rows:

CITY
NAGPUR

BRANCH 10

12. SELECT DISTINCT(D1.CNAME),D1.AMOUNT ,C1.CITY FROM deposit D1, CUSTOMER C1, BRANCH B1 WHERE D1.CNAME=C1.CNAME AND C1.CITY IN(SELECT C2.CITY FROM customer C2 WHERE C2.CNAME='ANIL');

Result Grid Filter Rows:

CNAME	AMOUNT	CITY
ANIL	1000	CALCUTTA

Result 11

Experiment No.: 4

Name: Justin v kalappura

Roll No: 10

Batch: B

Date: 06-05-2022

Aim

To familiarize with join or cartesian product

Questions

1.Give name of customers having living city BOMBAY and branch city NAGPUR

2.Give names of customers having the same living city as their branch city

3.Give names of customers who are borrowers as well as depositors and having city NAGPUR.

4.Give names of borrowers having deposit amount greater than 1000 and loan amount greater than 2000.

5.Give names of depositors having the same branch as the branch of Sunil

6.Give names of borrowers having loan amount greater than the loan amount of Pramod

7.Give the name of the customer living in the city where branch of depositor Sunil is located.

8.Give branch city and living city of Pramod

9.Give branch city of Sunil and branch city of Anil

10.Give the living city of Anil and the living city of Sunil

Procedure

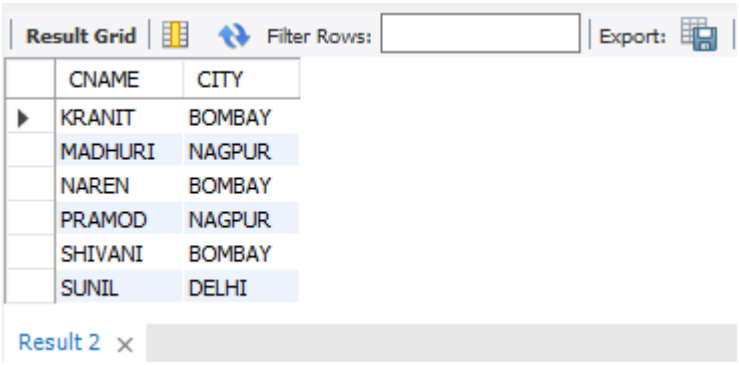
use ddl;

1: SELECT D1.CNAME,D1.BNAME,C1.CNAME,C1.CITY,B1.CITY,B1.BNAME FROM DEPOSIT D1,CUSTOMER C1,BRANCH B1 WHERE C1.CITY = 'BOMBAY' AND B1.CITY = 'NAGPUR' AND D1.CNAME = C1.CNAME AND D1.BNAME = B1.BNAME;

Output Screen

2: SELECT DISTINCT(CUSTOMER.CNAME), BRANCH.CITY FROM BRANCH, CUSTOMER WHERE BRANCH.CITY = CUSTOMER.CITY;

Output Screen

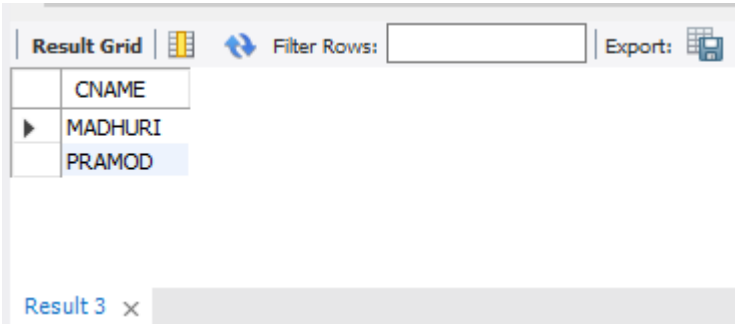

Result Grid | Filter Rows: | Export:

CNAME	CITY
KRANIT	BOMBAY
MADHURI	NAGPUR
NAREN	BOMBAY
PRAMOD	NAGPUR
SHIVANI	BOMBAY
SUNIL	DELHI

Result 2

3: SELECT C1.CNAME FROM CUSTOMER C1,DEPOSIT D1,BORROW B1 WHERE C1.CITY='NAGPUR' AND C1.CNAME=D1.CNAME AND D1.CNAME = B1.CNAME;

Output Screen

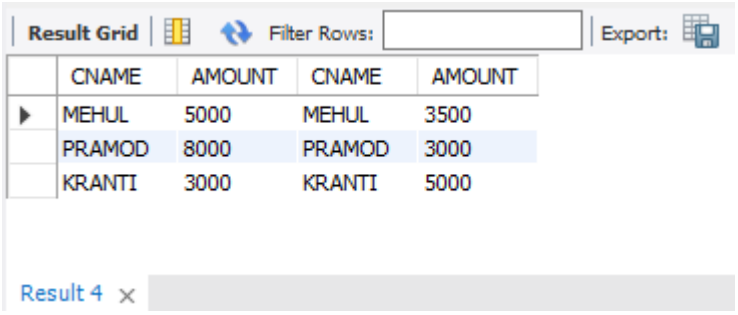

Result Grid | Filter Rows: | Export:

CNAME
MADHURI
PRAMOD

Result 3

4: SELECT BR1.CNAME, BR1.AMOUNT, D1.CNAME, D1.AMOUNT FROM BORROW BR1,DEPOSIT D1 WHERE D1.CNAME = BR1.CNAME AND D1.AMOUNT > 1000 AND BR1.AMOUNT > 2000;

Output Screen

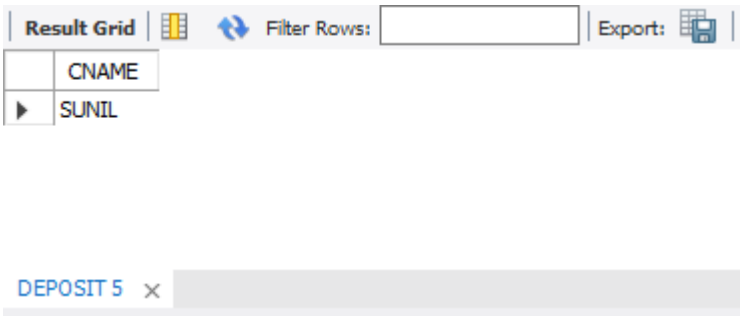

Result Grid | Filter Rows: | Export:

CNAME	AMOUNT	CNAME	AMOUNT
MEHUL	5000	MEHUL	3500
PRAMOD	8000	PRAMOD	3000
KRANTI	3000	KRANTI	5000

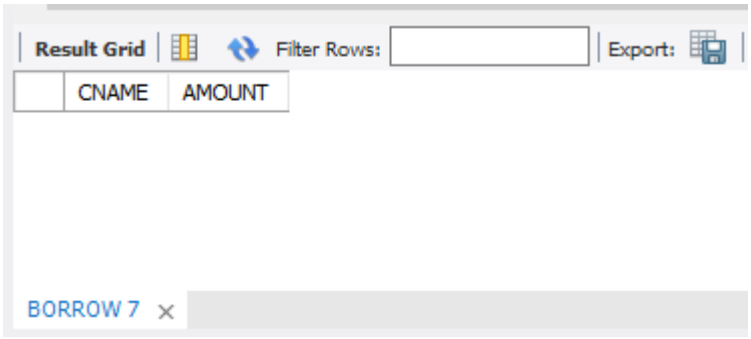
Result 4

5: SELECT D1.CNAME FROM DEPOSIT D1 WHERE D1.BNAME IN (SELECT D2.BNAME FROM DEPOSIT D2 WHERE D2.CNAME = 'SUNIL');

Output Screen

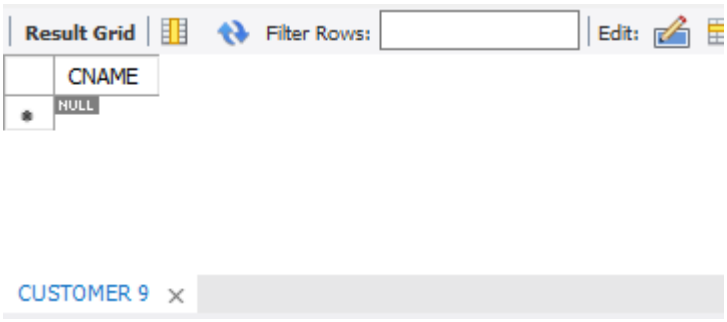


6: SELECT BR1.CNAME, BR1.AMOUNT FROM BORROW BR1 WHERE BR1.AMOUNT > ALL (SELECT BR2.AMOUNT FROM BORROW BR2 WHERE BR2.CNAME = 'PRAMOD');



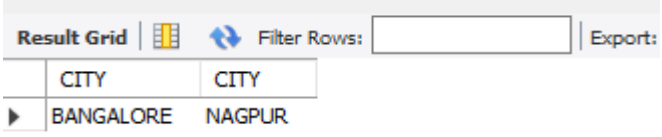
7: SELECT C.CNAME FROM CUSTOMER C WHERE C.CITY IN (SELECT B.CITY FROM BRANCH B WHERE B.BNAME IN (SELECT D.BNAME FROM DEPOSIT D WHERE D.CNAME='SUNIL'));

Output Screen



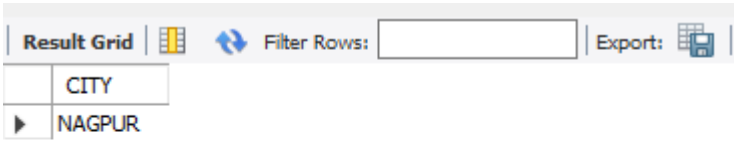
8: SELECT B1.CITY , C1.CITY FROM BRANCH B1,CUSTOMER C1, DEPOSIT D1 WHERE C1.CNAME = 'PRAMOD' AND C1.CNAME = D1.CNAME AND D1.BNAME = B1.BNAME;

Output Screen



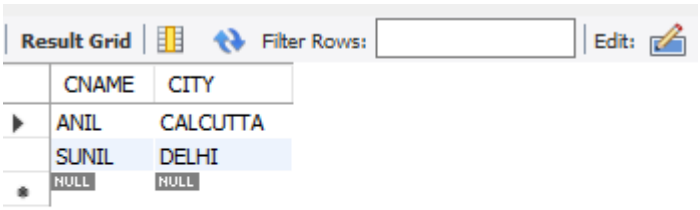
9: SELECT B1.CITY FROM DEPOSIT D1, BRANCH B1 WHERE D1.BNAME = B1.BNAME AND D1.CNAME IN ('SUNIL' ,'ANIL');

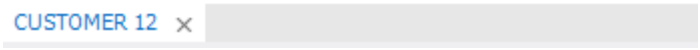
Output Screen



10: SELECT C1.CNAME, C1.CITY FROM CUSTOMER C1 WHERE C1.CNAME = 'ANIL' OR C1.CNAME = 'SUNIL';

Output Screen





Experiment No.: 5

Name: Justin v kalappura

Roll No: 10

Batch: B

Date: 06-05-2022

Aim

To familiarize with Group by and Having clause

Questions

1.List the branches having sum of deposit more than 5000.

2.List the branches having sum of deposit more than 500 and located in city BOMBAY

3.List the names of customers having deposited in the branches where the average deposit is more than 5000.

4.List the names of customers having maximum deposit

5.List the name of branch having highest number of depositors?

6.Count the number of depositors living in NAGPUR.

7.Give names of customers in VRCE branch having more deposite than any other customer in same branch

8.Give the names of branch where number of depositors is more than 5

9.Give the names of cities in which the maximum number of branches are located

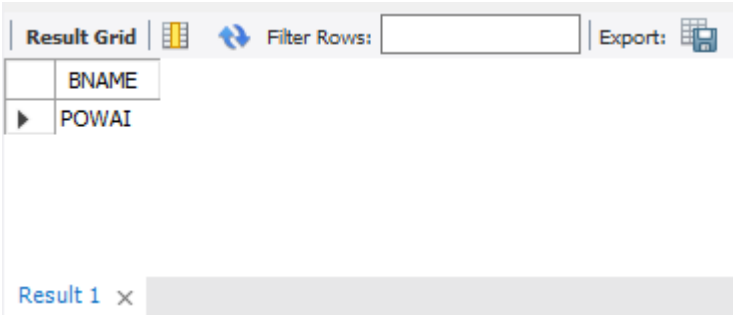
10.Count the number of customers living in the city where branch is located

Procedure

USE DDL;

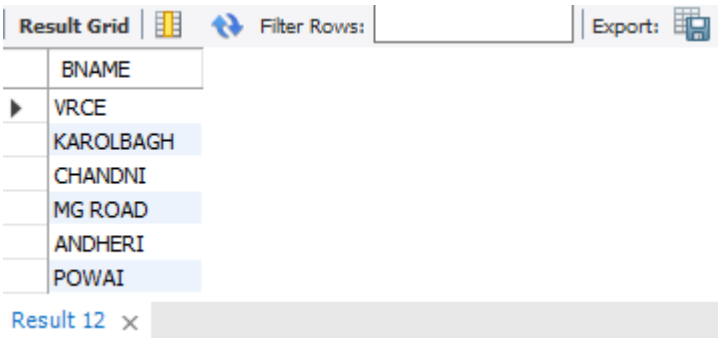
1: SELECT D.BNAME FROM DEPOSIT D, BRANCH B WHERE D.BNAME=B.BNAME AND B.CITY='BOMBAY' GROUP BY D.BNAME HAVING SUM(D.AMOUNT)>5000;

Output Screen



2: SELECT D.BNAME FROM DEPOSIT D, BRANCH B, CUSTOMER C WHERE D.BNAME=B.BNAME AND C.CITY="BOMBAY" GROUP BY D.BNAME HAVING SUM(D.AMOUNT)>500;

Output Screen

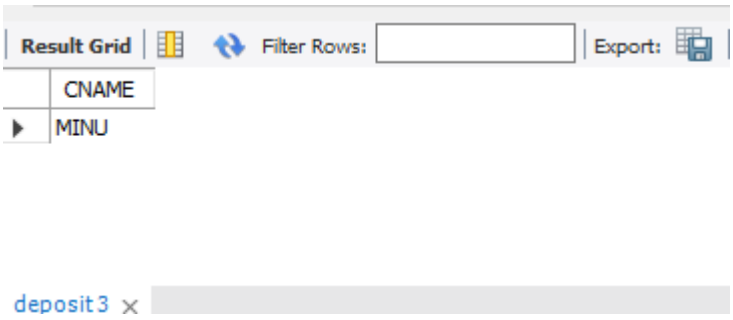

Result Grid | Filter Rows: | Export:

BNAME
VRCE
KAROLBAGH
CHANDNI
MG ROAD
ANDHERI
POWAI

Result 12

3: SELECT CNAME from deposit where AMOUNT=(select AVG(Amount) from DEPOSIT GROUP BY BNAME having AVG(Amount)>5000);

Output Screen

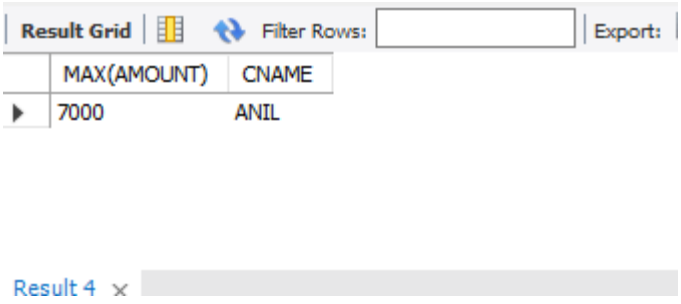

Result Grid | Filter Rows: | Export:

CNAME
MINU

deposit 3

4: SELECT MAX(AMOUNT),CNAME FROM deposit;

Output Screen

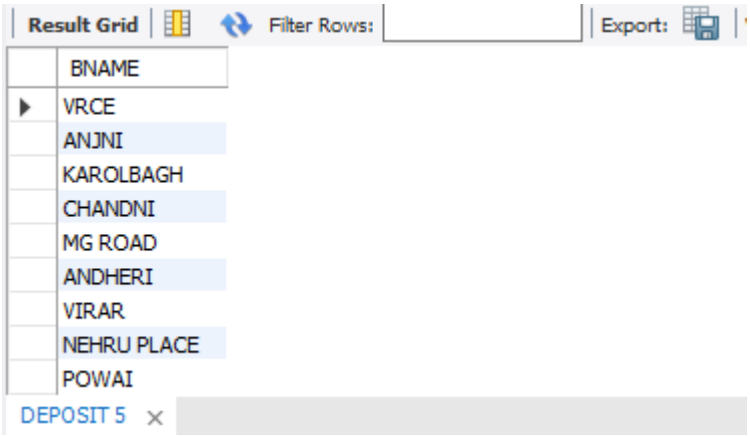

Result Grid | Filter Rows: | Export:

MAX(AMOUNT)	CNAME
7000	ANIL

Result 4

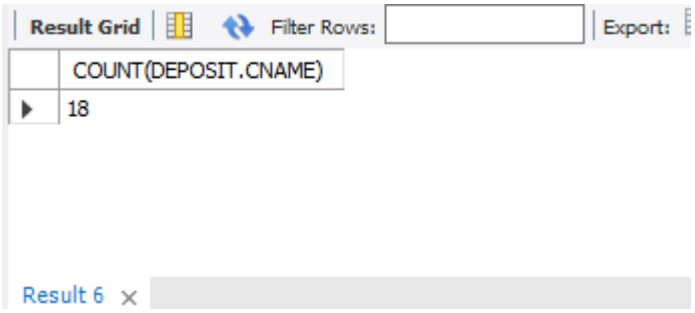
5: SELECT D1.BNAME FROM DEPOSIT D1 GROUP BY D1.BNAME HAVING COUNT(D1.CNAME) >= ALL (SELECT COUNT(D2.CNAME) FROM DEPOSIT D2 GROUP BY D2.BNAME)

Output Screen



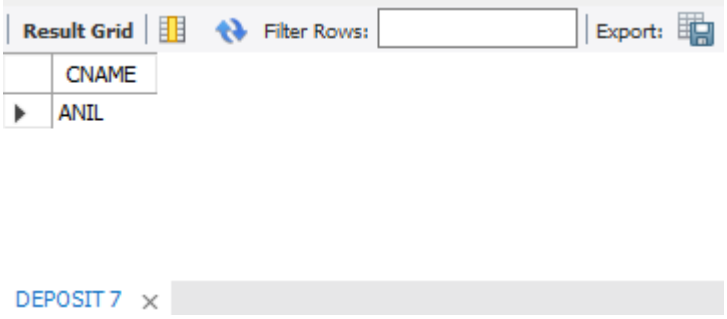
6: SELECT COUNT(DEPOSIT.CNAME)FROM DEPOSIT,CUSTOMER WHERE CUSTOMER.CITY='nagpur';

Output Screen



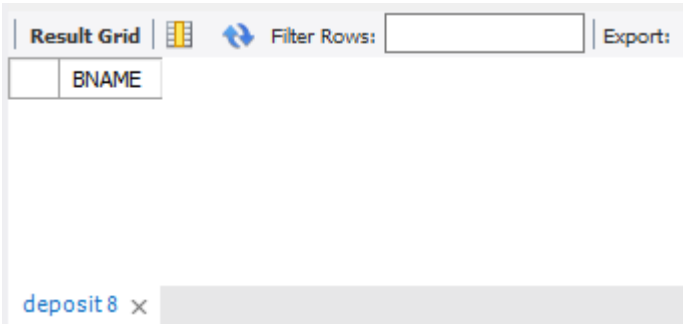
7: SELECT CNAME FROM DEPOSIT WHERE BNAME='VRCE' AND AMOUNT=(SELECT MAX(AMOUNT) FROM DEPOSIT WHERE BNAME='VRCE');

Output Screen



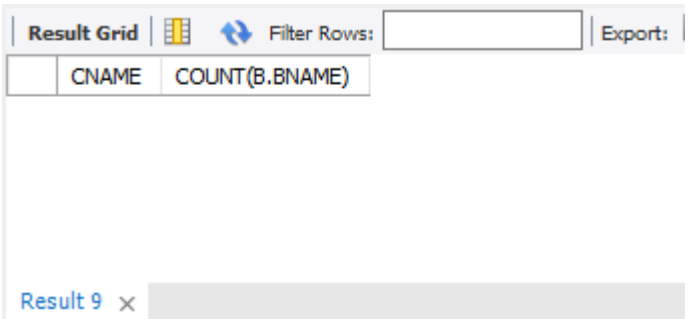
8: SELECT BNAME from deposit GROUP BY BNAME HAVING COUNT(BNAME)>5;

Output Screen



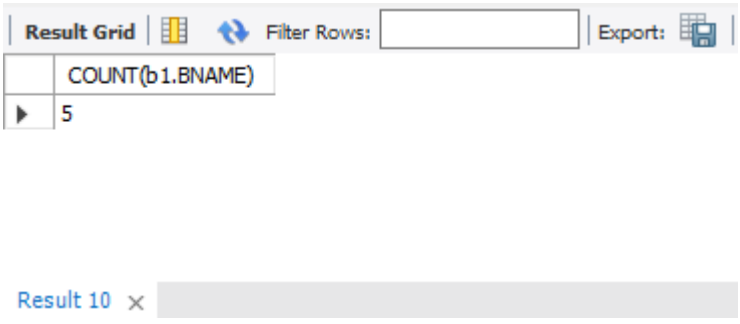
9: SELECT C.CNAME ,COUNT(B.BNAME) FROM CUSTOMER C INNER JOIN BRANCH B ON C.CNAME=B.BNAME GROUP BY C.CNAME ORDER BY COUNT(B.BNAME) DESC;

Output Screen



10: SELECT COUNT(b1.BNAME) FROM DEPOSIT d1 , BORROW b1 , CUSTOMER c1 WHERE c1.CNAME=d1.CNAME AND d1.CNAME=b1.CNAME AND c1.CITY IN (SELECT CITY FROM CUSTOMER);

Output Screen



Experiment No.: 6

Name: Justin v kalappura

Roll No: 10

Batch: B

Date:10-06-2022

Aim

Implementation of triggers

Questions

Create a Trigger for student table that will update another table shows the name, total marks and percentage

Procedure

CREATE TABLE MARKS(MARKID INT PRIMARY KEY AUTO_INCREMENT,NAME VARCHAR(20),TOTAL INT,PER INT);

CREATE TRIGGER MARK_TRIGGER AFTER INSERT ON STUDENTINFO FOR EACH ROW

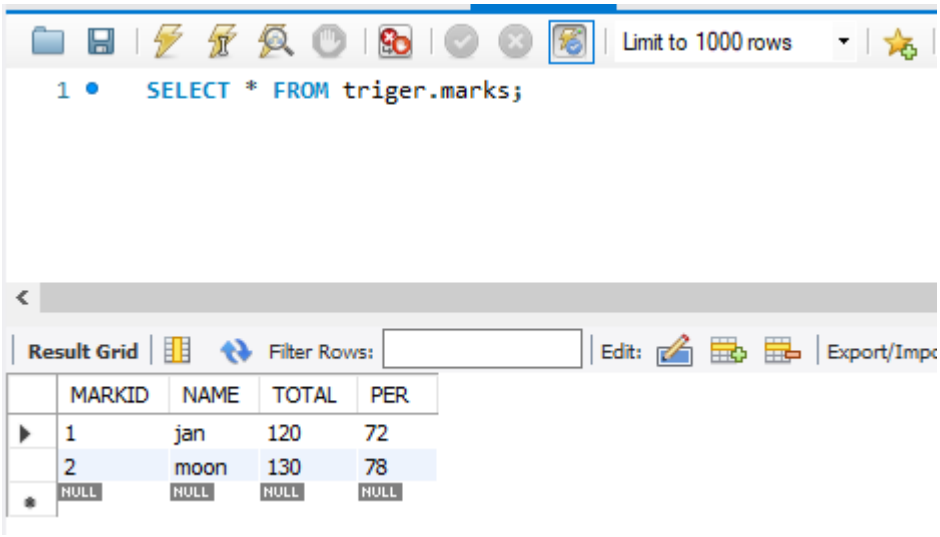
INSERT INTO MARKS(NAME,TOTAL,per) VALUES(new.NAME,new.TOTAL,new.per);

INSERT INTO STUDENTINFO(ID,NAME,SUBJ1,SUBJ2,subj3)

values(1002,'jan',35,42,43),(1003,'moon',44,38,48);

SELECT *FROM MARKS;

Output Screenshot



Question

Create a student table with fields id,name,subject1,subject2,subject3 and total,percentage. For each entry of row, update total marks and percentage using triggers in SQL

Name: Justin v kalappura

Roll No: 10

Batch: B

Date:10-06-2022

Procedure

CREATE DATABASE TRIGER;

USE TRIGER;

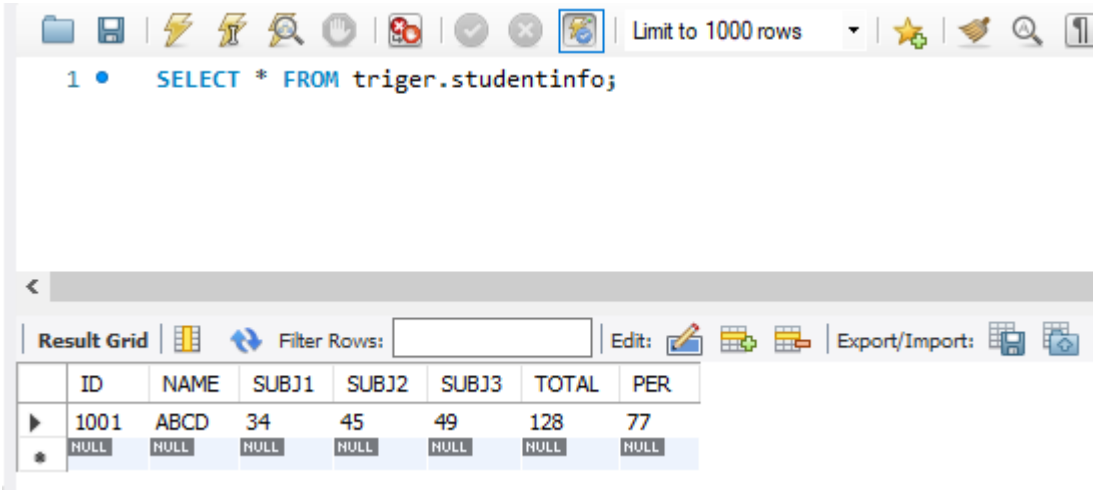
CREATE TABLE STUDENTINFO(ID INT PRIMARY KEY , NAME VARCHAR(20) NOT NULL, SUBJ1 INT,SUBJ2 INT,SUBJ3 INT,TOTAL INT,PER INT);

CREATE TRIGGER MARKCHANGE BEFORE INSERT ON STUDENTINFO FOR EACH ROW SET new.total=new.SUBJ1+new.SUBJ2+NEW.SUBJ3,

NEW.PER=NEW.TOTAL * 60/100;

insert into studentinfo(ID,NAME,SUBJ1,SUBJ2,SUBJ3) VALUES(1001,'ABCD',34,45,49);

Output Screenshot



Experiment No.: 7

Name: Justin v kalappura

Roll No: 10

Batch: MCA -B

Date:24-05-2022

Aim

Installation of MongoDB on Windows

Procedure

The installers for MongoDB are available in both the 32-bit and 64-bit format. The 32-bit installers are good for development and test environments. But for production environments you should use the 64-bit installers. Otherwise, you can be limited to the amount of data that can be stored within MongoDB.

Download & Install MongoDB on Windows

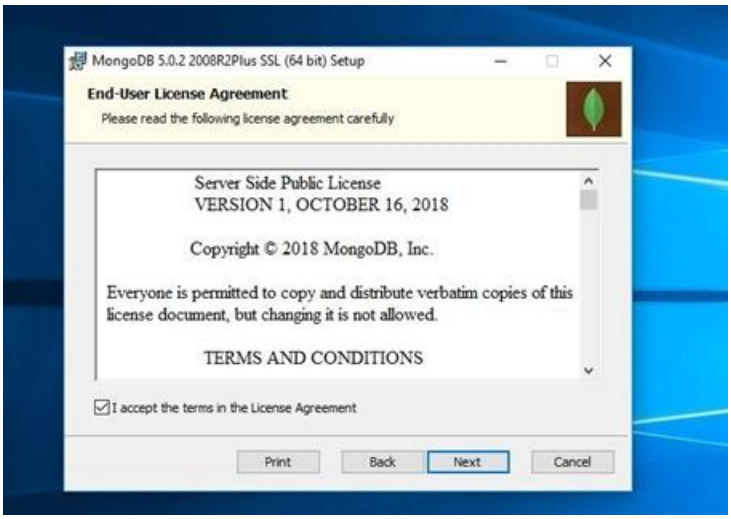
The following steps can be used to install MongoDB on Windows 10:

Step 1:

After downloading MongoDB, open the msi file and click next.

Step 2:

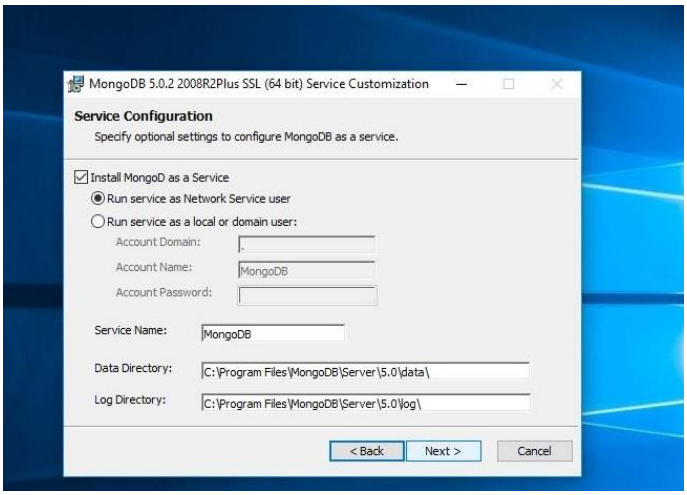
Accept the End-User License Agreement And Click Next



Step 3:

Click on the "complete" button to install all of the components. The custom option can be used to install selective components or if you want to change the location of the installation.

1. Select “Run service as Network Service user”. Make a note of the data directory, we’ll need this later.
2. Click Next

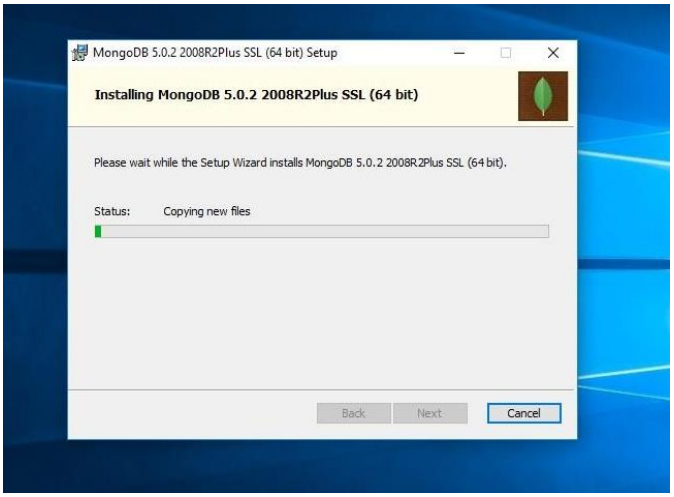


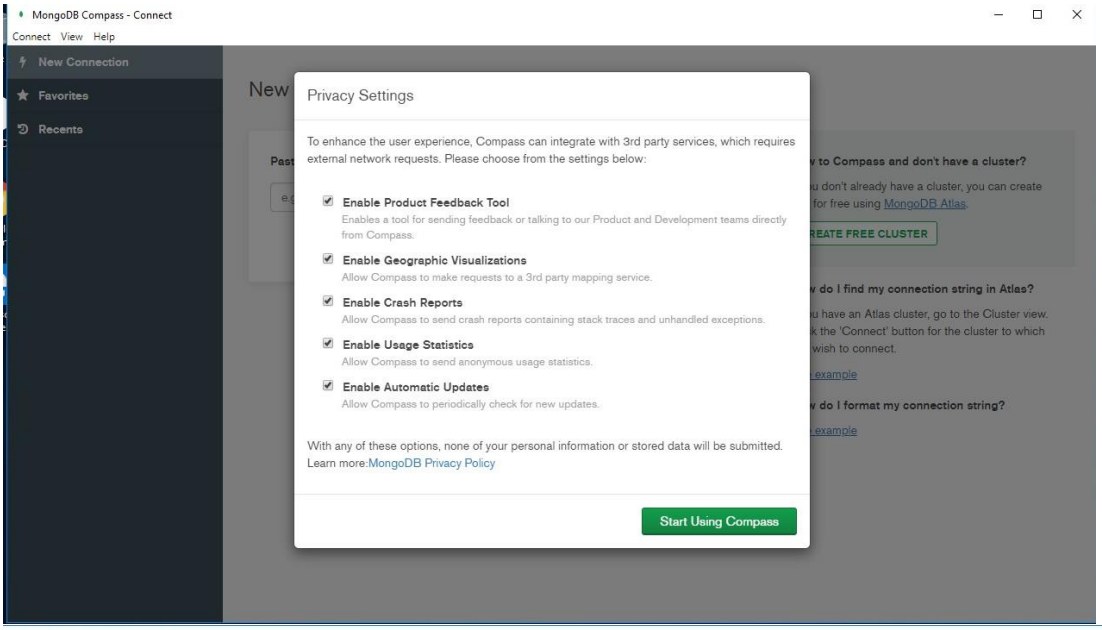
Step 4:

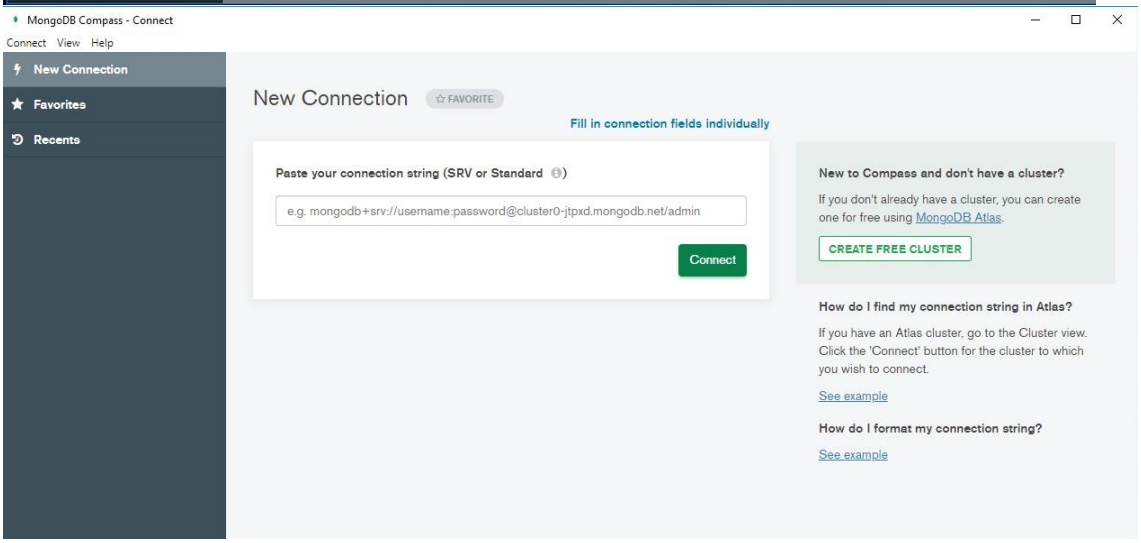
Click on the Install button to start the installation. Installation begins.

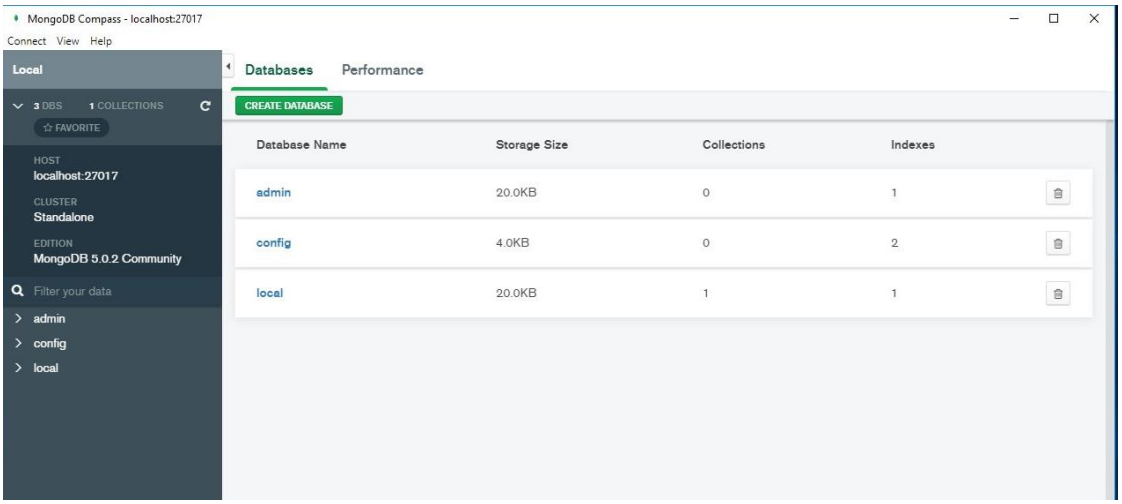
Click next once completed.

Click on the Finish button to complete the installation.




MongoDB Compass - Connect
Connect View Help
New Connection
Favorites
Recents
Privacy Settings
To enhance the user experience, Compass can integrate with 3rd party services, which requires external network requests. Please choose from the settings below:
Enable Product Feedback Tool
Enables a tool for sending feedback or talking to our Product and Development teams directly from Compass.
Enable Geographic Visualizations
Allow Compass to make requests to a 3rd party mapping service.
Enable Crash Reports
Allow Compass to send crash reports containing stack traces and unhandled exceptions.
Enable Usage Statistics
Allow Compass to send anonymous usage statistics.
Enable Automatic Updates
Allow Compass to periodically check for new updates.
With any of these options, none of your personal information or stored data will be submitted.
Learn more:MongoDB Privacy Policy
Start Using Compass


MongoDB Compass - Connect
Connect View Help
New Connection
Favorites
Recents
New Connection
FAVORITE
Fill in connection fields individually
Paste your connection string (SRV or Standard)
e.g. mongodb+srv://username:password@cluster0-jtpxd.mongodb.net/admin
Connect
New to Compass and don't have a cluster?
If you don't already have a cluster, you can create one for free using MongoDB Atlas.
CREATE FREE CLUSTER
How do I find my connection string in Atlas?
If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect.
See example
How do I format my connection string?
See example


MongoDB Compass - localhost:27017
Connect View Help
Local
3 DBS 1 COLLECTIONS
FAVORITE
HOST
localhost:27017
CLUSTER
Standalone
EDITION
MongoDB 5.0.2 Community
Filter your data
admin
config
local
Databases
Performance
CREATE DATABASE
Database Name
Storage Size
Collections
Indexes
admin 20.0KB 0 1
config 4.0KB 0 2
local 20.0KB 1 1

Experiment No.: 8

Name: Justin v kalappura

Roll No: 10

Batch: MCA -B

Date:24-05-2022

Aim

Designing Databases using NoSQL: MongoDB

Procedure

Download correct mongodb driver for PHP, based on your PHP

version, Architecture and whether Thread Safety is enabled. You can get that information from the phpinfo page.

Do the following steps to install and configure MongoDB driver on Windows XAMPP Server.

1. Download the latest stable version of the PHP MongoDB driver from following URL https://pecl.php.net/package/mongodb.

Available Releases			
Version	**State**	**Release Date**	**Downloads**
1.3.2	stable	2017-10-30	mongodb-1.3.2.tgz (904.0kB) DLL
1.3.1	stable	2017-10-16	mongodb-1.3.1.tgz (904.0kB) DLL
1.3.0	stable	2017-09-19	mongodb-1.3.0.tgz (906.1kB) DLL

2. Extract the archive File.
3. Copy the php_mongodb.dll file from the extracted folder to the PHP extension directory. this is usually the "C:\xampp\php\ext" folder in XAMPP Server.
4. Open the php.ini file inside your PHP installation (C:\xampp\php) and add the following line:

extension=php_mongodb.dll

5. Save the file and close it. Restart the Apache web server.

Test MongoDB Connection From PHP Script

Let's write a very simple PHP program that creates a connection to the MongoDB server and dump the connection status.

Add following PHP code to your php script and access from the Web browser. The above PHP example will output something similar to:

```
<?php
$connection = new MongoDB\Driver\Manager("mongodb://localhost:27017");
var_dump($connection);
?>
```

Result

object(MongoDB\Driver\Manager)#1 (2) { ["uri"]=> string(25) "mongodb://localhost:27017" ["cluster"]=> array(1) { [0]=> array(11) { ["host"]=> string(9) "localhost" ["port"]=>

int(27017) ["type"]=> int(0) ["is_primary"]=> bool(false) ["is_secondary"]=> bool(false) ["is_arbiter"]=> bool(false) ["is_hidden"]=> bool(false) ["is_passive"]=> bool(false) ["tags"]=> array(0) { } ["last_is_master"]=> array(0) { } ["round_trip_time"]=> int(-1) } } }

Experiment No.: 9

Name: Justin v kalappura

Roll No: 10

Batch: B

Date:03-06-2022

Aim

Query Processing : Performing CRUD operations with NoSQL database

Procedure

1. Create database and use database
   use crud_operations

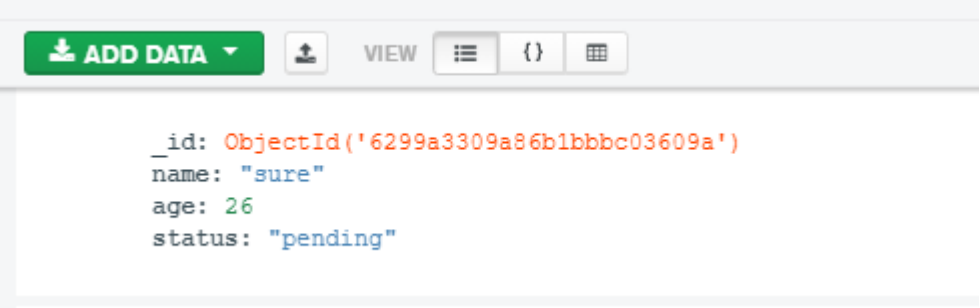
```
> use crud_operations
switched to db crud_operations
```

2. Display all database
   show dbs

```
> show dbs
AJCE             0.000GB
Ajce1            0.000GB
admin            0.000GB
config           0.000GB
crud_operations  0.000GB
local            0.000GB
my_project       0.000GB
```

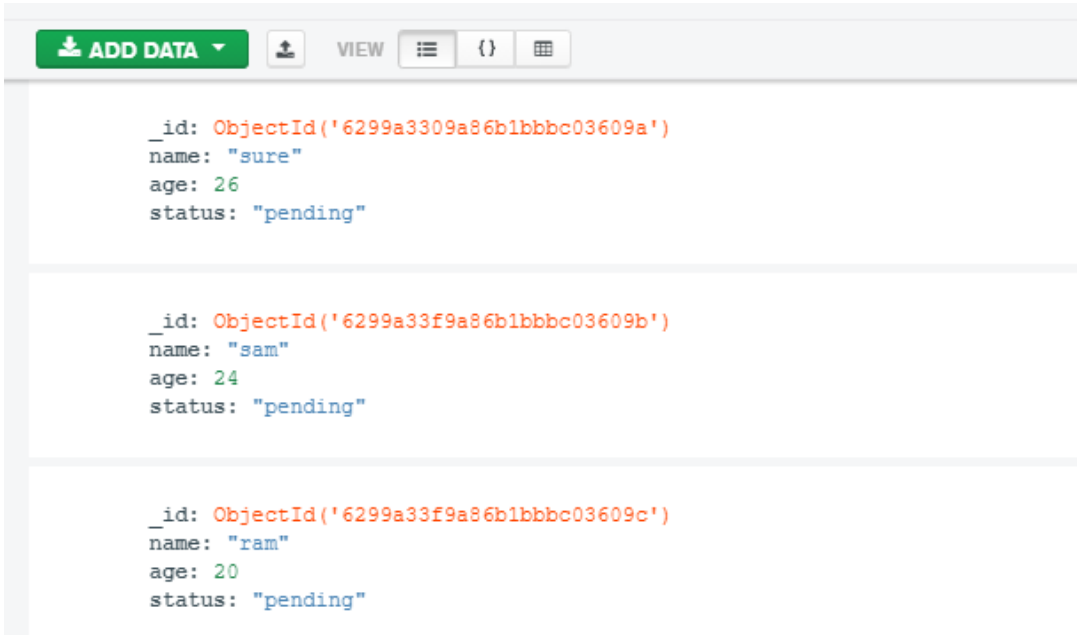
3. Create table and insert value into the table
   db.users.insertOne({ name:"sure",age:26,status:"pending" })

```
> db.users.insertOne(  { name:"sure",age:26,status:"pending" }  )
{
        "acknowledged" : true,
        "insertedId" : ObjectId("6299901b9a86b1bbbc036097")
}
```



4. Insert many values into the table.
db.users.insertMany([{ name:"sam",age:24,status:"pending"},
{name:"ram",age:20,status:"pending"}])

```
> db.users.insertMany( [{ name:"sam",age:24,status:"pending"},
 {name:"ram",age:20,status:"pending"} ])
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("629990f39a86b1bbbc036098"),
                ObjectId("629990f39a86b1bbbc036099")
        ]
}
```



5. Display tables
show collections

```
> show collections
insertmany
insertone
users
```

6. Find particular table Datas.
db.users.find({age:{$gt:18}},

... {name:1,address:1}

...).limit(5)

```
> db.users.find( {age:{$gt:18}},
... {name:1,address:1}
... ).limit(5)
{ "_id" : ObjectId("6299901b9a86b1bbbc036097"), "name" : "sure" }
{ "_id" : ObjectId("629990f39a86b1bbbc036098"), "name" : "sam" }
{ "_id" : ObjectId("629990f39a86b1bbbc036099"), "name" : "ram" }
```

7. Modify existing operation in a collection
db.users.updateMany({age:{$lt:24}}, {$set:{status:"reject"}})

```
> db.users.updateMany( {age:{$lt:24}}, {$set:{status:"reject"}} )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

```
_id: ObjectId('6299901b9a86b1bbbc036097')
name: "sure"
age: 26
status: "pending"

_id: ObjectId('629990f39a86b1bbbc036098')
name: "sam"
age: 24
status: "pending"

_id: ObjectId('629990f39a86b1bbbc036099')
name: "ram"
age: 20
status: "reject"
```

8. Remove documents from a collection
db.uders.deleteMany(

... {status:"reject"}

...)

```
> db.users.deleteMany( {status:"reject"} )
{ "acknowledged" : true, "deletedCount" : 1 }
```

```
_id: ObjectId('6299901b9a86b1bbbc036097')
name: "sure"
age: 26
status: "pending"

_id: ObjectId('629990f39a86b1bbbc036098')
name: "sam"
age: 24
status: "pending"
```

Experiment No.: 10

Name: Justin v kalappura

Roll No: 10

Batch: B

Date:03-06-2022

Aim

NoSQL and Front-End: PHP: Create a PHP form and insert data to mongodb

Procedure

Index.html

```
<html>
<head>
   <title>Document</title>
</head>
<body>
   <h2>insert to mongo</h2>
   <form action="insert.php" method="post">
      <input type="text" name="name" placeholder="name">
      <input type="number" name="rollno" placeholder="rollno">
      <input type="password" name="password" placeholder="password">
      <input type="text" name="firstname" placeholder="firstname">
      <input type="submit" name="submit">
   </form>
</body>
</html>
```

Insert.php

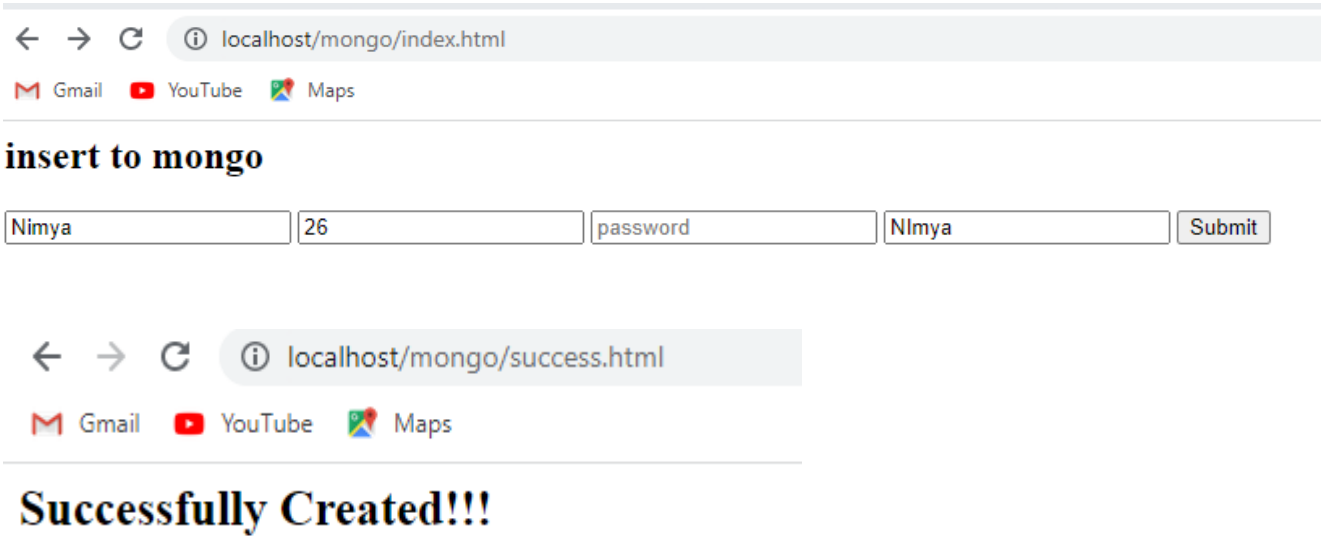
```
<?php
$mongo = new MongoDB\Driver\Manager("mongodb://localhost:27017");
if(isset($_POST["submit"])){
   $name=$_POST["name"];
   $first_name=$_POST["firstname"];
   $rollno=$_POST["rollno"];
```

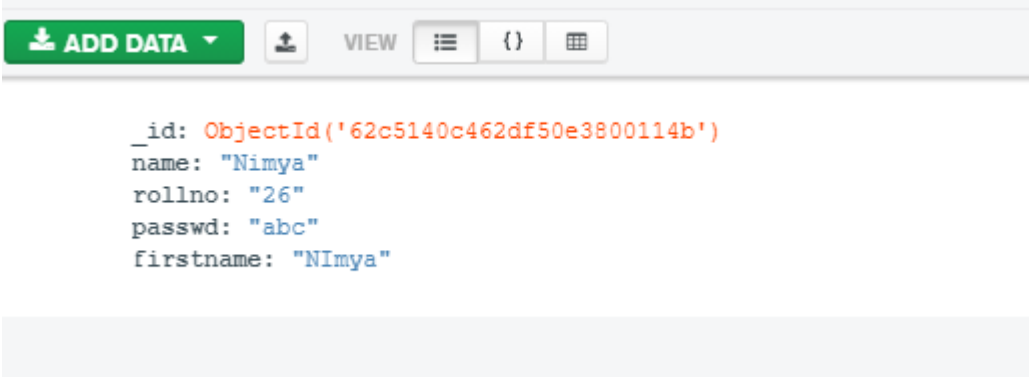
```
    $passwd=$_POST["password"];
    $writer=new MongoDB\Driver\Bulkwrite;
    $writer->insert(["name"=>$name,"rollno"=>$rollno,"passwd"=>$passwd,
    "firstname"=>$first_name]);
    $mongo->executeBulkWrite('form124.insertion',$writer);
    header("Location:success.html");
    die();
}
?>
```

Success.html

```
<html>
    <head>
        <title>Document</title>
    </head>
    <body>
        <h2>Successfully Created!!!</h2>
    </body>
<html>
```

Output Screenshot




ADD DATA
VIEW
{}
_id: ObjectId('62c5140c462df50e3800114b')
name: "Nimya"
rollno: "26"
passwd: "abc"
firstname: "NImya"