



CERTIFICATE

This is to certify that Mr / Ms WAQQA S AKHTER
has satisfactorily completed the course of experiments in practical
of DBMS Lab prescribed by
the Department of ISE,
Ramaiah Institute of Technology for the year 2020 - 2021.

Component	Max. Marks	Marks Obtained
Record		
Lab Test		
Viva Voce		
CIE Marks		

Signature of the Student with Date

Name and Signature of the Faculty with Date

Signature of Head of the Department

RAMAIAH INSTITUTE OF TECHNOLOGY

(Autonomous Institute, Affiliated to VTU)

MSR Nagar, Bangalore - 560054

INDEX

1. Consider an Employee with a social security number (SSN) working on multiple projects with definite hours for each. Each Employee belongs to a Department. Each project is associated with some domain areas such as Database, Cloud and so on. Each Employee will be assigned to some project. Assume the attributes for Employee and Project Relations.

a) Mention the constraints neatly.

Employee : Primary Key - { SSN }

Project : Primary Key - { Pro }

Works : Primary Key - { SSN, Pro }

Foreign Key - { SSN, Pro }

c) Create the tables, insert suitable tuples and perform the following operations in SQL

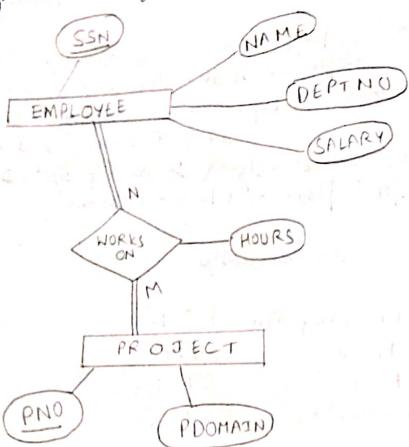
~~1. Obtain the~~

```
create table employee (
    SSN  varchar(6),
    name  varchar(10),
    Deptno number(2),
    Salary  number(6),
    primary key (SSN)
);
```

table created

```
Create table project(
    Pro  varchar(6),
    pdomain  varchar(10),
```

(b) Design the ER diagram



c) State the schema diagram for the ER diagram

EMPLOYEE	SSN	NAME	DEPTNO	SALARY
PROJECT	PNO	PDODMAIN		

WORKSON

SSN	PNO	HOURS
-----	-----	-------

Experiment No.: 1

Date: 24/7/2020

primary key (pno)
);

Table Created

```
create table workson(  
    SSN varchar(6),  
    pno varchar(6),  
    hours number(2),  
    primary key (SSN, pno),  
    foreign key (SSN) references employee (SSN),  
    foreign key (pno) references project (pno)
```

);

Table Created

```
insert into employee values ('E1', 'Sam', 2, 1000);  
insert into employee values ('E2', 'Rick', 3, 5000);  
insert into employee values ('E3', 'Don', 2, 1500);
```

~~1 row updated~~

```
insert into project values ('P1', 'cloud');  
insert into project values ('P2', 'database');
```

~~2 rows updated~~

```
insert into workson values ('E1', 'P1', 1);  
insert into workson values ('E2', 'P2', 3);  
insert into workson values ('E3', 'P1', 2);
```

I/O SQL

SSN

E2

2. count (*)

1

Deptno

2

3

Experiment No.: 1

Date: 24/12/2020

1. Obtain the SSN of employees assigned to database project

```
select SSN, pno  
from works_on, employee, project  
where employee.SSN = works_on.SSN  
AND works_on.pno = project.pno  
AND project.pdomain = 'database';
```

2. Find the number of employees working in each department

```
select count(*), Deptno  
from employee  
group by Deptno;
```

3. Update the Project details of Employee bearing SSN
• # SSN to ProjectNo = # ProjectNo

```
update works_on  
set pno = 'P2'  
where SSN = 'E1';
```

1 row updated

e) Create the table, insert suitable tuples and perform the following operations using MongoDB

```
db.createCollection("employee")
```

```
db.employee.insert({ "SSN": "E1", "name": "Sam", "Dept": "Dept1",  
"pno": "P1" })
```

Q1 MongoDB

1. {
 "id": ObjectId("5e0d89cfad11053b"),
 "SSN": "E2",
 "name": "rick",
 "Dept": "Dept2",
 "pro": "P2"

2. {

"id": ObjectId("5e0d89cfad11053b"),
 "SSN": "E1",
 "name": "Sam",
 "Dept": "Dept1",
 "pro": "P1"

PL/SQL

Q1

No. of employees whose salary was incremented = 2

Experiment No.: 1 Date: 24/12/2020

db.employee.insert({ "SSN": "E2", "name": "rick", "Dept": "Dept2", "pro": "P2" })

1. List all the employees of the Department named #Deptname

db.employee.find({ "Dept": "Dept2" }).pretty()

2. Name the employees working on project number #Projectno

db.employee.find({ "pro": "P1" }).pretty()

3. Write a program that gives all employees in Department #number a 15% pay increase. Display a message displaying how many employees were awarded the increase.

BEGIN

update employee

set salary = salary + (0.15 * salary)

where Deptno = 2;

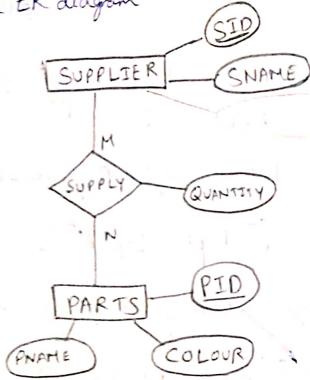
dbms_output.put_line("No of employees whose salary was incremented:" || SQL%ROWCOUNT);

END;

/

-X-

b) Design the ER diagram



c) State the schema diagram for the ER diagram

SUPPLIER	
SID	SNAME
PARTS	
PID	PNAME
COLOUR	
SUPPLY	
SID	PID
QUANTITY	

Experiment No.: 2

Date: 24/12/2020

2. Consider the relations: DART, SUPPLIER and SUPPLY. The Supplier relation holds information about suppliers. The SID attribute identifies the supplier, while the other attributes each hold one piece of information about the supplier. The Part relation holds information about parts. Likewise, the PID attribute identifies the part, while the other attributes hold information about the part. The Shipment relation holds information about shipments. The SID and PID attributes identify the supplier of the shipment and the part shipped, respectively. The remaining attribute indicates how many parts were shipped.

a) Mention the constraints neatly

Parts : Primary Key - { PID }

Supplier: Primary Key - { SID }

Supply: Primary Key - { PID, SID }

Foreign Key - { PID, SID }

d) Create the above tables, insert suitable tuples and perform following operation in Oracle SQL

```
create table supplier(  
    SID varchar(3),  
    Sname varchar(10),  
    primary key(SID)  
)
```

Table created

create table parts(

PID nvarchar(3),
 pname nvarchar(10),
 colour nvarchar(6),
 primary key (PID)

);

Table Created

create table supply(

SID nvarchar(3),
 PID nvarchar(3),
 quantity number(3),
 primary key (SID, PID),
 foreign key (SID) references supplier(SID) on delete cascade
 foreign key (PID) references parts(PID) on delete cascade

);

Table Created

insert into supplier values ('S1', 'tata');

insert into supplier values ('S2', 'mi');

insert into supplier values ('S3', 'oppo');

insert into parts values ('P1', 'cord', 'red');

insert into parts values ('P2', 'wire', 'black');

insert into parts values ('P3', 'nut', 'grey');

insert into supply values ('S1', 'P1', 5);

insert into supply values ('S2', 'P2', 20);

SQL

PID

P1

2. sname
mi

Experiment No. 2

Date : 24/12/2020

insert into supply values ('S3', 'P3', 10);

1. Obtain the part identifiers of parts supplied by supplier # supplier name

select PID

from supply, supplier
where supply.SID = supplier.SID
AND sname = 'tata';

2. Obtain the names of suppliers who supply # part name

select sname
from supplier, parts, supply
where supplier.SID = supply.SID
AND supply.PID = parts.PID
AND parts.pname = 'wire';

3. Delete the parts which are in # colour

delete from parts
where colour = 'grey';

1 row deleted

e) Create the table, insert suitable tuples and perform the following operations using mongo db

db.createCollection("warehouse")

IP MongoDB

2. {
 " _id": ObjectId("5a0d8f5bd110536"),
 "PID": "P1",
 "Name": "wave",
 "SID": "S1",
 "Name": "mi"

Experiment No.: 2

Date: 24/12/2020

```
db.warehouse.insert({ "PID": "P1", "Name": "wire", "SID": "S1",  
                       "Name": "tata" })  
db.warehouse.insert({ "PID": "P2", "Name": "nut", "SID": "S2",  
                       "Name": "mi" })
```

1. Update the part identifier

```
db.warehouse.update({ "PID": "P1" },  
                       { $set: { "PID": "P3" } },  
                       { multi: true })
```

2. Display all suppliers who supply the part with part identifier
#Part_No.

```
db.warehouse.find({ "PID": "P2" }).pretty()
```

3. Write a PL/SQL program to display the contents of the above tables and then update the quantity of parts shipped by 5%.

```
set serveroutput on
```

BEGIN

```
for cursor1 in (select * from supplier
```

BEGIN

```
dbms_output.put_line('Table Supply :');
```

```
dbms_output.put_line('SID PID quantity');
```

```
for i in (select * from supply) loop
```

```
dbms_output.putline(i.SID||' '||i.PID||' '||i.quantity);
```

```
end loop;
```

I/O PLSQL

SID	PI	Quantity
S1	P1	5
S2	P2	20
S3	P3	10

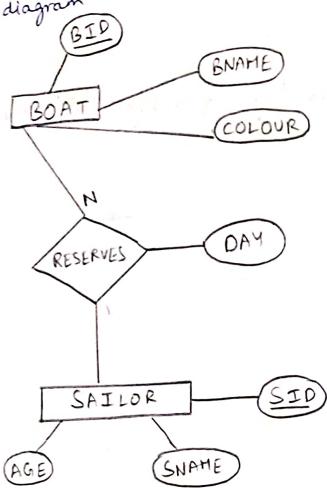
1 (EW(S) updated)

Experiment No. 2

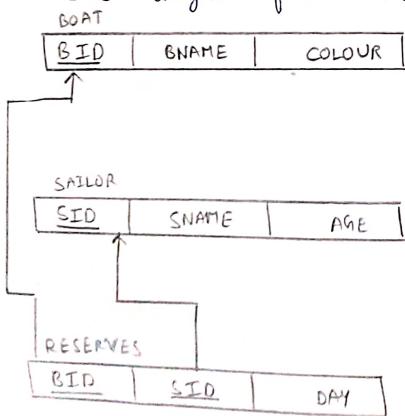
Date 24/12/20

update supply
set quantity = quantity + (0.05 * quantity);
END;
\

b) Design me ER diagram



c) State the schema diagram for the ER diagram



Experiment No.: 3

Date: 25/12/2020

3 Consider the relations BOAT, SAILOR and RESERVES. The relation BOAT identifies the features of a boat such as unique identifier, color and a name. The list of sailors with attributes such as sailorID, name, age, etc. are stored in the relation Sailor. The sailors are allowed to reserve any number of boats on any day of the week.

a) Mention the constraints neatly

Boat : Primary Key - {BID}

Sailor : Primary Key - {SID}

Reserves : Primary Key - {BID, SID}

Foreign Key - {BID, SID}

d) Create the tables, insert suitable tuples and perform the following operations in SQL

create table boat(

BID varchar(3),

bname varchar(10),

colour varchar(10),

primary key (BID)

);

Table created

create table sailor(

SID varchar(3),

sname varchar(10),

age number (2),

primary key (SID)
);

Table created

Create Table reserves(
BID varchar(3),
SID varchar(3),
day varchar(10),
primary key (BID, SID))

foreign key (BID) references boat(BID),
foreign key (SID) references sailor(SID)

);

Table created

insert into boat values ('B1', 'abc', 'red');

insert into boat values ('B2', 'xyz', 'yellow');

insert into boat values ('B3', 'pqr', 'white');

insert into sailor values ('S1', 'sam', 40);

insert into sailor values ('S2', 'rick', 30);

insert into reserves values ('B1', 'S1', 'sunday');

insert into reserves values ('B2', 'S2', 'friday');

insert into reserves values ('B3', 'S1', 'tuesday');

insert into reserves values ('B2', 'S1', 'saturday');

1. Obtain the BID of the boats reserved by : # Sailor_Name

select BID

I/O SQL

BID

B1
B3

2.

BID
B2

3

Count(BID)
3
1

SID

S1
S2

Experiment No.: 3

Date: 25/12/2020

from reserves
where SID in (select SID
from sailor
where sname = 'Sam');

2. Retrieve the bid of the boats reserved by all the sailors.

Select B.BID
from boat B
where NOT EXISTS (select S.SID
from sailor S
where NOT EXISTS (select R.SID
from reserves R
where R.BID = B.BID
AND R.SID = S.SID));

3. Find the number of boats reserved by each other sailor

Select count(BID), SID
from reserves
group by SID;

e) Create the table, insert suitable tuples and perform the following operations using MongoDB

db.createCollection("port")

db port.insert({ "BID": "B1", "bname": "abc", "colour": "black", "sname": "John", "SID": "S1", "Day": "friday"})

I/O Mongo DB

1.

2. {
 "id": ObjectId('5a0d946fd11053b180.fbd5'),
 "BID": 'B1'
 "brane": 'abc'
 "Colour": 'black'
 "Name": 'John'
 "SID": 'S1'
 "day": 'friday'
}

I/O PL/SQL

Enter the value of n: 5

Prime

Enter the value of n: 10

Not Prime

Experiment No.: 3

Date: 25/12/2020

db.createCollection("boats");
db.boats.insert([{"BID": "B1", "brane": "abc", "Colour": "red", "Name": "sam", "SID": "S1", "day": "sunday"}])

1. Obtain the number of boats obtained by sailor # Sailor_name

db.boats.find({"Name": "sam"}).count()

2. Retrieve the boats of color: # color

db.boats.find({"Colour": "black"}).pretty()

3. Write a PL/SQL program to check whether a given number is prime or not

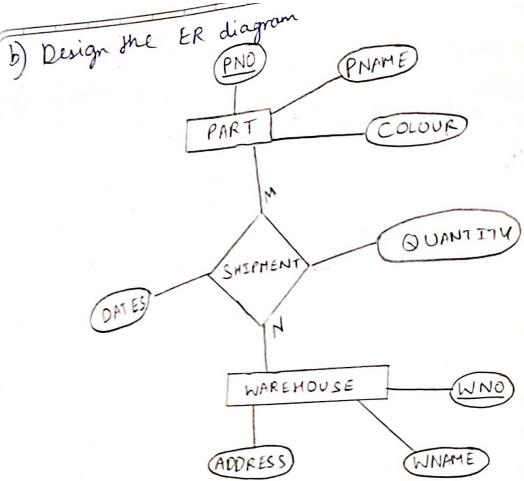
```
declare
  flag number := 0;
  n number;
  i number;
begin
  n:=< n >;
  for i in 2..n/2 loop
    if mod(n,i)=0 then
      flag := 1;
      exit;
    end if;
  end loop;
  if flag = 1 then
    dbms_output.put_line("Not Prime");
  else
    dbms_output.put_line("Prime");
  end if;
end;
```

Experiment No. : 3

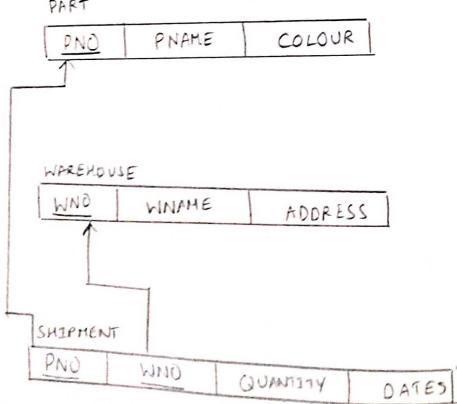
Date : 25/12/2020

```
else  
    dbms_output.put_line("Prime");  
end if;  
end;  
/  
*  
-----  
-----
```

b) Design the ER diagram



c) State the schema diagram for the ER diagram



Experiment No.: 4

Date: 25/12/2020

4. Consider the Shipment - Warehouse case study. Different parts like nut, bolts etc. are being shipped to different warehouses. The address of the warehouse also stored for shipment purposes. The quantity & date are recorded within warehouse.

a) Mention the constraints neatly.

Part : Primary key - { pno }

Warehouse : Primary key - { wno }

Shipment : Primary key - { pno, wno }

Foreign key - { pno, wno }

d) Create the above tables, insert suitable tuples and perform the following operations in SQL.

```

create table part (
    pno varchar(3),
    pname varchar(10),
    colour varchar(10),
    primary key(pno)
);
  
```

Table created

```

create table warehouse (
    wno varchar(3),
    wname varchar(10),
    address varchar(25),
    primary key(wno)
);
  
```

Table created

Create Table Shipment(

```

    pno nvarchar(3),
    wno nvarchar(3),
    quantity number(2),
    dates date,
    primary key(pno, wno),
    foreign key(pno) references part(pno),
    .foreign key(wno) references warehouse(wno)
);

```

Table created

insert into part values('P1', 'nut', 'red');

insert into part values('P2', 'wire', 'grey');

insert into part values('P3', 'screw', 'green');

insert into warehouse values('W1', 'abc', '1234Kolkata');

insert into warehouse values('W2', 'xyz', '007Mumbai');

insert into shipment values('P1', 'W1', 10, '7-DEC-2018');

insert into shipment values('P2', 'W2', 7, '31-JAN-2019');

insert into shipment values('P3', 'W1', 23, '25-MAY-2020');

insert into shipment values('P1', 'W2', 2, '8-AUG-2021');

1. Obtain the names of warehouses which have shipped red colour parts.

select whname

from warehouse

I10 SQL

1.

wname
abc

2.

P.pno
P1

3.

count(pno)

2

2

Wno

W1

W2

4.

w.wno
W1

w.name
abc

w.address
1234 Kolkata

Experiment No.: 4

Date: 25/11/2020

where wno in(select wno

from shipment

where pno = (select pno

from part

where colour = 'red'));

2. Retrieve the PNO of the parts shipped by all the warehouses

select P.pno

from part P

where NOT EXISTS (

Select W.wno

from warehouse W

where NOT EXISTS (

select S.wno

from shipment S

where S.pno = P.pno

AND S.wno = W.wno));

3. Find the number of parts supplied by each warehouse

select count(pno), wno

from shipment

group by wno;

4. list the warehouse details which ships maximum quantity of parts.

select w.wno, w.name, w.address

from warehouse w join shipment s

I/O Mongo DB

1. {
 "_id": ObjectID("5a0d9876d1153810fb0d9"),
 "pno": "P1",
 "pname": "wire",
 "wno": "W1",
 "uname": "abc",
 "quantity": 10
}
2. {
 "_id": "abc", "total": 10 }
 {
 "_id": "xyz", "total": 12 }

I/O PL/SQL

1 row(s) inserted

Part:

pno	pname	colour
P1	nut	red
P2	wire	grey

Part 1:

pno	pname	colour
P1	nut	red
P2	wire	grey

Experiment No. 9

Date: 25/12/2020

on w.no = s.no
where quantity in (select max(quantity)
from shipment);

- c) Create the table, insert suitable tuples and perform the following operations using MongoDB

db.createCollection("shipment")

db.shipment.insert({ "pno": "P1", "pname": "wire", "wno": "W1",
 "uname": "abc", "quantity": 10 })
db.shipment.insert({ "pno": "P2", "pname": "nut", "wno": "W2",
 "uname": "xyz", "quantity": 12 })

1. Find the parts shipped from warehouse: Wname

db.shipment.find({ "wname": "abc" }).pretty()

2. List the total quantity supplied from each warehouse

db.shipment.aggregate([{\$group:{_id:"\$wname", total:
 {\$sum:"\$quantity"} }}])

- (f) Using cursors demonstrate the process of copying the contents of one table to a new table

DECLARE

cursor c1 is
select *
from ~~shipment~~ part;

v_part part%rowtype;

BEGIN

open cl;

loop

fetch cl into v_part.pno, v_part.pname, v_part.colour;

exit when cl % NOTFOUND;

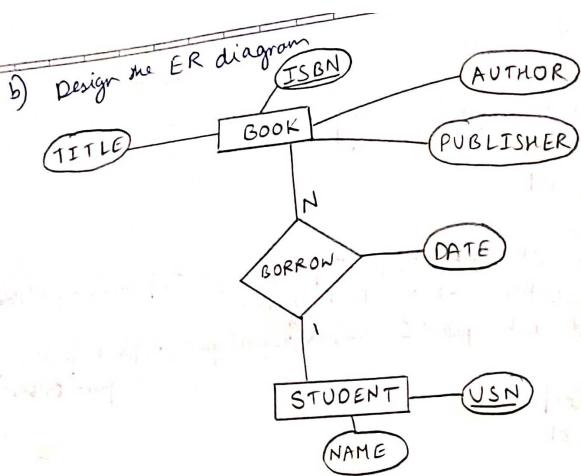
insert into part1 values(v_part.pno, v_part.pname,
v_part.colour);

end loop;

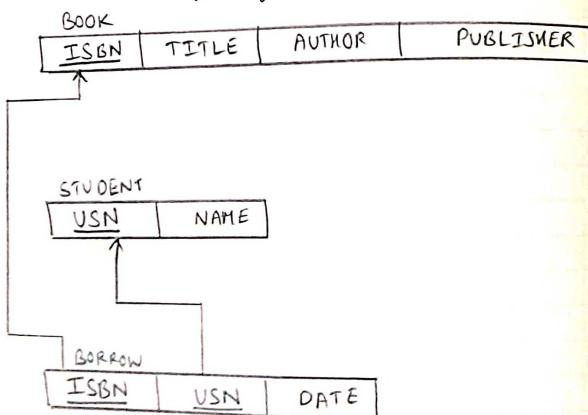
close cl;

end;

/



c) State the schema diagram for the ER diagram



Experiment No. : 5

Date : 25/12/2020

5. Consider the book lending system from the library. The students are allowed to borrow any number of books on a given date from the library. The details of the book should include ISBN, Title of the Book, author and publisher. All student need not compulsorily borrow books.

a) Mention the constraints neatly.

Book : Primary Key - {ISBN}

Student : Primary Key - {USN}

Borrow : Primary Key - {ISBN, USN}

Foreign Key - {ISBN, USN}

d) Create the above tables, insert suitable tuples and perform the following operations in SQL

```

create table book(
    ISBN varchar(10),
    title nvarchar(10),
    author nvarchar(10),
    publisher nvarchar(10),
    primary key (ISBN)
);
  
```

Table created.

```

create table student(
    USN varchar(10),
    name nvarchar(10),
    primary key (USN)
);
  
```

);

Table created

```
create table borrow(
    ISBN  varchar(10),
    USN  varchar(10),
    dates  date,
    primary key (ISBN, USN),
    foreign key (ISBN) references book (ISBN),
    foreign key (USN) references student (USN)
);
```

Table created.

```
insert into book values('B11', 'food', 'sam', 'swan');
insert into book values('B12', 'sports', 'rick', 'dhyani');
insert into book values('B13', 'dbw', 'jay', 'sun');
```

```
insert into student values('IS127', 'saif');
insert into student values('IS130', 'yash');
```

```
insert into borrow values('B11', 'IS127', '12-7-1998');
insert into borrow values('B12', 'IS130', '1-1-2020');
insert into borrow values('B13', 'IS127', '12-31-2015');
```

1. Obtain the name of the Student who has borrowed the book bearing ISBN 'B12'.

```
select name
```

I10 SQL

1.

name
yash

2.

name
saif

3.

count (ISBN)
2
1

USN
IS127
IS130

Experiment No.: S

Date : 25/12/2020

from student
where USN = (select USN
from borrow
where ISBN = 'B12');

2. Obtain the names of the student who have borrowed data base books

select name
from student
where USN = (select USN
from borrow
where ISBN = (select ISBN
from book
where title = 'db'));

3. Find the number of books borrowed by each student.

select count (ISBN), USN
from borrow
group by USN;

e) Create the table, insert suitable tuples and perform the following operations using MongoDB

db.createCollection("library")

db.library.insert([{"USN": "IS127", "name": "saif", "ISBN": "BH",
"title": "Cars"}])

I/O Mongo db

1. `{ "name": "sail" }`
2. `{ "name": "yash" }`

I/O PL/SQL

Fibonacci series is :

0
1
1
2
3
5
8
13

Experiment No. 5

Date: 25/12/2020

```
db.library.insert({ "USN": "ISB0", "name": "yash", "ISBN": "B1",  
"title": "db1" })
```

1. Obtain the name of the student who has borrowed the book bearing ISBN 'B1'

```
db.library.find({ "ISBN": "B1" }, { "name": 1, _id: 0 }).pretty()
```

2. Obtain the names of the student who have borrowed data base book.

```
db.library.find({ "title": "db1" }, { "name": 1, _id: 0 }).pretty()
```

3. Write a PL/SQL procedure to print the first 8 fibonacci numbers and a program to call the same.

DECLARE

a number := 0;

b number := 1;

temp number;

i number;

PROCEDURE fib(n IN number) IS

BEGIN

```
dbms_output.put_line('Fibonacci Series is :');
```

```
dbms_output.put_line(a);
```

```
dbms_output.put_line(b);
```

```
for i in 3..n loop
```

```
temp := a + b;
```

```
a := b;
```

```
b := temp;
```

Experiment No. : 5

Date : 25/12/2020

dbms_output.put_line (temp);
end loop;
end;
BEGIN
fib(8);
END;
/

→