Exercise-I

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.

b) Design the ER diagram for the problem statement

c) State the schema diagram for the ER diagram.

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

   1. Obtain the details of employees assigned to “Database” project.
   2. Find the number of employees working in each department with department details.
   3. Update the Project details of Employee bearing SSN = #SSN to ProjectNo = #Project_No and display the same.

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

   1. List all the employees of the Department named #Dept_name.
   2. Name the employees working on Project Number:#Project_No

f) 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.

Exercise-II

Consider the relations: PART, SUPPLIER and SUPPLY. The Supplier relation holds information about suppliers. The attributes SID, SNAME, SADDR describes the supplier. The Part relation holds the attributes such as PID, PNAME and PCOLOR. The Shipment relation holds information about shipments that include SID and PID attributes identifying the supplier of the shipment and the part shipped, respectively. The Shipment relation should contain information on the number of parts shipped.

a) Mention the constraints neatly.

b) Design the ER diagram for the problem statement

c) State the schema diagram for the ER diagram.

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

   1. Obtain the details of parts supplied by supplier #SNAME.
   2. Obtain the Names of suppliers who supply #PNAME.

Delete the parts which are in #PCOLOR

Exercise I Solutions

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

Create the collection:

```
db.createCollection("EMPLOYEE")
```

Inserting the values:

```
>db.EMPLOYEE.insert({"SSN":4567,"Name":'James',"DeptNo":'XYZ',"ProjectNo":101})
>db.EMPLOYEE.insert({"SSN":3256,"Name":'Jack',"DeptNo":'XYZ',"ProjectNo":102})
```

```
>db.EMPLOYEE.find().pretty()
```

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

```
> db.EMPLOYEE.find({"DeptNo":'XYZ'}).pretty()
```

2. Name the employees working on Project Number :#Project_No

```
> db.EMPLOYEE.find({"ProjectNo":104}).pretty()
```

g) 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.

```
set serveroutput on
begin
update employee1
set salary=(1.15*salary) where deptno=10;
dbms_output.put_line('number of rows updated are'||sql%rowcount);
end;
/
```

Exercise II Solutions

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

Create the collection:

```
>db.createCollection("WAREHOUSE")
```

Inserting the values:

```
>db.WAREHOUSE.insert({"PNO":1947,"Pname":'bolts',"Colour":'Black',"SNO":1234,"Sname":'ABC',"Address":'blore'})
>db.WAREHOUSE.insert({"PNO":1950,"Pname":'chain',"Colour":'Blue',"SNO":4567,"Sname":'DEF',"Address":'chen'})
```

Update the parts identifier

```
>db.WAREHOUSE.update({"PNO":1950},{$set:{"PNO":2017}},{multi:true})
>db.WAREHOUSE.find().pretty()
```

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

```
> db.WAREHOUSE.find({"PNO":2017}).pretty()
```

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

```
create table part1(pno int,pname char(20),colour char(20),primary key(pno));
create table copy_part1(pno int,pname char(20),colour char(20),primary key(pno));

insert into part1 values(10,'nuts','black');
insert into part1 values(20,'bolts','grey');
insert into part1 values(30,'screw','green');

set serveroutput on
declare
cursor curr is select *from part1;
counter int;
rows part1%rowtype;
```

```
begin
open curr;
loop
fetch curr into rows ;
exit when curr%notfound;
insert into copy_part1 values(rows.pno,rows.pname,rows.colour);
end loop;
counter := curr%rowcount;
close curr;
dbms_output.put_line(counter||' rows inserted into the table copy_part1 ');
end;
```

Exercise III Solutions

a) Create the table, insert suitable tuples and perform the following operations using MongoDB.
   Create the collection:db.createCollection("BOATRES")

   Insert the values:

   ```
   >db.BOATRES.insert({"BID":9988,"BNAME":'ABC',"COLOUR":'Black',"SNAME":'John',"SID":1234,"DAY":'2017-12-25'})
   >db.BOATRES.insert({"BID":8877,"BNAME":'DEF',"COLOUR":'Black',"SNAME":'Smith',"SID":4567,"DAY":'2017-11-24'})
   ```

   3. Obtain the number of boats obtained by sailor :#Sailor_Name
      `> db.BOATRES.find({"SNAME":'Sucre'}).count()`
   4. Retrieve boats of color :”#color”
      `> db.BOATRES.find({"COLOUR":'Black'}).pretty()`

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

   ```
   SET SERVEROUTPUT ON
   DECLARE
    n number:=&n;
    j number:=2;
    counter number:=0;
   BEGIN
    WHILE(j<=n/2) loop

    if mod(n,j)=0 then
   ```

```
 dbms_output.put_line(n ||' is not prime number');
 counter:=1;
 exit ;
 else
 j:=j+1;
 end if;
 end loop;

 if counter=0 then
 dbms_output.put_line( n || ' is a prime number');
 end if;
end;
/
```

Exercise IV Solutions

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

   Create the collection:

   ```
   >db.createCollection("SHIPMENT")
   ```

   Insert the values:

   ```
   >db.SHIPMENT.insert({"PNO":11,"PNAME":'bolts',"COLOUR":'Black',"WNO":99,
   "WNAME":'ABC',"QUANTITY":45,"DATE":'2017-09-25'})
   >db.SHIPMENT.insert({"PNO":12,"PNAME":'nuts',"COLOUR":'Black',"WNO":99,
   "WNAME":'ABC',"QUANTITY":38,"DATE":'2017-09-28'})
   ```

   1. Find the parts shipped from warehouse :Wname”

      ```
      >db.SHIPMENT.find().pretty()
      >db.SHIPMENT.find({"WNAME":'ABC'}).pretty()
      ```

2. List the total quantity supplied from each warehouse

```
>db.SHIPMENT.aggregate([{ $group:{_id:"$WNAME",total:{$sum:"$QUANTITY"}}}])
```

b) Using cursors demonstrate the process of copying the contents of one table to a new table.

```
create table part1(pno int,pname char(20),colour char(20),primary key(pno));
create table copy_part1(pno int,pname char(20),colour char(20),primary key(pno));

insert into part1 values(10,'nuts','black');
insert into part1 values(20,'bolts','grey');
insert into part1 values(30,'screw','green');

set serveroutput on
declare
cursor curr is select *from part1;
counter int;
rows part1%rowtype;
begin
open curr;
loop
fetch curr into rows ;
exit when curr%notfound;

insert into copy_part1 values(rows.pno,rows.pname,rows.colour);

end loop;
counter := curr%rowcount;
close curr;
dbms_output.put_line(counter||' rows inserted into the table copy_part1 ');
end;
/
```

Exercise 5: Solutions

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

Create the collection:

```
>db.createCollection("LIBRARY")
```

Insert the values:

```
>db.LIBRARY.insert({"ISBN":1122,"TITLE":'datbase',"AUTHOR":'ABC',"PUBLISHER":'selina',"SSN":2015,"date":'2017-05-29'})
>db.LIBRARY.insert({"ISBN":2233,"TITLE":'datbase',"AUTHOR":'DEF',"PUBLISHER":'mcgraw',"SSN":2016,"date":'2017-06-29' })
```

3. Obtain the name of the student who has borrowed the book bearing ISBN ‘123’.
   ```
   >db.LIBRARY.find().pretty() >db.LIBRARY.find({"ISBN":1122}, {"SSN":1,_id:0}).pretty()
   ```
4. Obtain the Names of students who have borrowed database books.
   ```
   >db.LIBRARY.find({"TITLE":'datbase'},{"SSN":1,_id:0}).pretty()
   ```

l) Write a PL/SQL procedure to print the first 8 Fibonacci numbers and a program to call the same.

```
SET SERVEROUTPUT ON
declare
a number;
b number;
c number;
n number;
i number;
begin
n:=8;
a:=0;
b:=1;
dbms_output.put_line(a);
dbms_output.put_line(b);
for i in 1..n-2
loop
c:=a+b;
dbms_output.put_line(c);
a:=b;
b:=c;
end loop;
end;
/
```