

DATABASE MANAGEMENT SYSTEMS LAB

Course Code: ISL57	Credits: 0:0:1
Course Coordinator: Sunitha R S	Contact Hours: 14P

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 SSN of employees assigned to database projects.
 2. Find the number of employees working in each department.
 3. Update the Project details of Employee bearing SSN = #SSN to ProjectNo = #Project_No
- e) Create the table, insert suitable tuples and perform the following operations using MongoDB
 1. List all the employees of 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 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.
- 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 part identifiers of parts supplied by supplier #supplier_name.
 2. Obtain the Names of suppliers who supply #part_name.
 3. Delete the parts which are in #colour.
- e) Create the table, insert suitable tuples and perform the following operations using MongoDB
1. Update the parts identifier
 2. Display all suppliers who supply the part with part identifier: #Part_No.
- g) Write a PL/SQL program to display the contents of the above tables and then update the quantity of parts shipped by 5%.

Exercise-III

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.
- 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 bid of the boats reserved by '#Sailor_Name'.
 2. Retrieve the bid of the boats reserved by all the sailors.
 3. Find the number of boats reserved by each sailor
- e) Create the table, insert suitable tuples and perform the following operations using MongoDB.
 1. Obtain the number of boats obtained by sailor :#Sailor_Name
 2. Retrieve boats of color :"#color"
- f) Write a PL/SQL program to check whether a given number is prime or not.

Exercise-IV

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

- 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 SQL:
 1. Obtain the Names of warehouses which have shipped red colored parts.
 2. Retrieve the PNO of the parts shipped by all the warehouses.

3. Find the number of parts supplied by each warehouse.
4. List the warehouse details which ships maximum quantity of parts.
- e) Create the table, insert suitable tuples and perform the following operations using MongoDB
 1. Find the parts shipped from warehouse :Wname”
 2. List the total quantity supplied from each warehouse
- f) Using cursors demonstrate the process of copying the contents of one table to a new table.

Exercise-V

Consider the Book Lending system from the library. The students are allowed 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 students need not compulsorily borrow books.

- 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 SQL:
 1. Obtain the name of the student who has borrowed the book bearing ISBN ‘123’
 2. Obtain the Names of students who have borrowed database books.
 3. Find the number of books borrowed by each student.
- e) Create the table, insert suitable tuples and perform the following operations using MongoDB
 1. Obtain the name of the student who has borrowed the book bearing ISBN ‘123’.
 2. Obtain the Names of students who have borrowed database books.
- f) Write a PL/SQL procedure to print the first 8 Fibonacci numbers and a program to call the same.

Text Book

1. Benjamin Rosenzweig, Elena Silvestrova Rakhimov: Oracle PL/SQL by Example, 4th Edition, 2010.

Course Outcomes:

1.	Transform an information model into an ER diagram and relational database schema by using data definition language and/or utilities to implement the schema using a DBMS. (PO - 1, 2, 3, 10) (PSO - 1,2)
2.	Formulate using SQL/MongoDB solutions to a broad range of query and data update problems. (PO – 2, 5, 12) (PSO - 1,2, 3)
3.	Demonstrate a rudimentary understanding of programmatic components for a database such as control structures, procedures and cursors. (PO – 2,5, 12) (PSO - 1,2, 3)

Solutions for DBMS Lab Programs

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.

Constraints to be written

- b) Design the ER diagram for the problem statement

Draw ER diagram

- c) State the schema diagram for the ER diagram.

Draw the scheme diagram

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

create table e

```
(
  ssn varchar(6),
  name varchar(10),
  deptno int,
  primary key(ssn)
);
```

Table created.

create table p

```
(
  projectno varchar(10),
```

```
projectarea varchar(20),  
primary key(projectno)  
);
```

Table created.

```
create table a  
(  
  usn varchar(6),  
  projectno varchar(10),  
  foreign key(usn)references e(ssn),  
  foreign key(projectno)references p(projectno)  
);
```

Table created.

```
insert into e values('01','abc',10);  
  1 row updated  
insert into e values('02','xyz',20);  
  1 row updated  
insert into e values('03','pqr',30);  
  1 row updated  
insert into e values('04','lmn',40);  
  1 row updated  
insert into p values('100','database');  
  1 row updated  
insert into p values('200','network');  
  1 row updated  
insert into p values('300','android');  
  1 row updated  
insert into a values('01','100');  
  1 row updated  
insert into a values('02','200');  
  1 row updated  
insert into a values('03','300');  
  1 row updated  
insert into a values('01','200');  
  1 row updated
```

```
select * from e;  
SSN NAME DEPTNO
```

01	abc	10
02	xyz	20
03	pqr	30
04	lmn	40

```
select * from p;
```

PROJECTNO	PROJECTAREA
100	database
200	network
300	android

```
select * from a;
```

USN	PROJECTNO
01	100
02	200
03	300
01	200

1. Obtain the SSN of employees assigned to database projects.

```
select ssn from e
where ssn=(select usn
           from a
           where projectno=(select projectno from p
                           where projectarea='database'));
```

SSN
01

2. Find the number of employees working in each department.

```
select count(ssn),deptno from e group by deptno;
```

COUNT(SSN)	DEPTNO
1	30
1	20
1	40

3. Update the Project details of Employee bearing SSN = #SSN to ProjectNo = #Project_No
 update a set projectno='200' where usn='03';
 1 row updated

```
select * from a;
usn    projectno
01      100
02      200
03      200
04      200
```

- e) 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;
/

```

Consider the relations: PART, 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.

Write the constraints

- b. Design the ER diagram for the problem statement

Design the ER diagram

- c. State the schema diagram for the ER diagram.

Write the schema diagram for the ER diagram.

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

create table part

```

(
pno number(10),
pname varchar(20),
colour varchar(20),
primary key(pno)
);

```

Table created.

Create table supplier

```

(
sno number(10),
sname varchar(20),
address varchar(20),
primary key(sno)
);

```

Table created.

create table supply

```

(
pno number(10),
sno number(10),
quantity varchar(20),
primary key(pno,sno),
foreign key(pno) references part(pno) on delete cascade,

```


foreign key(sno) references supplier(sno)on delete cascade
);
Table created.

```
insert into part values(1,'plug','black');
1 row(s) inserted.
insert into part values(2,'bolt','blue');
1 row(s) inserted.
insert into part values(3,'nut','green');
1 row(s) inserted.
insert into supplier values(10,'Anoop','udupi');
1 row(s) inserted.
insert into supplier values(15,'Bharath','mangalore');
1 row(s) inserted.
insert into supplier values(20,'Ram','bangalore');
1 row(s) inserted.
insert into supply values(1,10,50);
1 row(s) inserted.
insert into supply values(2,10,30);
1 row(s) inserted.
insert into supply values(1,15,70);
1 row(s) inserted.
insert into supply values(3,15,40);
1 row(s) inserted.
insert into supply values(1,20,55);
1 row(s) inserted.
insert into supply values(2,20,65);
1 row(s) inserted.
insert into supply values(3,20,75);
1 row(s) inserted.
```

select * from part;

PNO	PNAME	COLOUR
1	plug	black
2	bolt	blue
3	nut	green

select * from supply ;

PNO	SNO	QUANTITY
1	10	50
2	10	30
1	15	70
3	15	40
1	20	55

2	20	65
3	20	75

```
select* from supplier ;
```

SNO	SNAME	ADDRESS
10	Anoop	udupi
15	Bharath	mangalore
20	Ram	Bangalore

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

```
select pno from supply
where sno IN(select sno from
supplier where sname='Ram');
```

PNO
1
2
3

2. Obtain the Names of suppliers who supply #part_name.

```
select sname,pname
from supplier,supply,part
where pname='bolt' AND supply.sno=supplier.sno AND part.pno=supply.pno;
```

SNAME	PNAME
Anoop	bolt
Ram	bolt

3. Delete the parts which are in #colour.

```
delete from part where colour='green';
```

- f) 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;  
/
```

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.

Write the constraints.

b) Design the ER diagram for the problem statement

Design the ER diagram

- c) State the schema diagram for the ER diagram.

Write the schema diagram for the ER diagram.

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

Create table BOAT

```
(  
  BID varchar(6) NOT NULL,  
  BNAME varchar(20),  
  COLOUR varchar(10),  
  PRIMARY KEY(BID)  
);
```

Create table SAILOR

```
(  
  SID varchar(6) NOT NULL,  
  SNAME varchar(20),  
  AGE varchar(3),  
  RATING varchar(2),  
  PRIMARY KEY(SID)  
);
```

Create table RESERVES

```
(  
  BID varchar(6),  
  SID varchar(6),  
  DAY varchar(10),  
  FOREIGN KEY(BID) references BOAT(BID) ON DELETE CASCADE,  
  FOREIGN KEY(SID) references SAILOR(SID) ON DELETE CASCADE  
);
```

```
INSERT INTO BOAT VALUES('01','ABC','RED');  
INSERT INTO BOAT VALUES('02','XYZ','YELLOW');  
INSERT INTO BOAT VALUES('03','PQR','GREEN');  
INSERT INTO BOAT VALUES('04','LMN','BLACK');  
INSERT INTO BOAT VALUES('05','DEF','BLUE');
```

```
INSERT INTO SAILOR VALUES('10','RAM','30','5');  
INSERT INTO SAILOR VALUES('20','RAVI','25','4');  
INSERT INTO SAILOR VALUES('30','MISHRA','22','3');  
INSERT INTO SAILOR VALUES('40','CHANDRA','24','2');
```

```
INSERT INTO SAILOR VALUES('50','SHIVA','36','1');
INSERT INTO SAILOR VALUES('60','KRISHNA','40','6');
```

```
INSERT INTO RESERVES VALUES('01','20','MONDAY');
INSERT INTO RESERVES VALUES('02','30','TUESDAY');
INSERT INTO RESERVES VALUES('03','50','WEDNESDAY');
INSERT INTO RESERVES VALUES('04','10','THURSDAY');
INSERT INTO RESERVES VALUES('05','20','FRIDAY');
```

```
SELECT * FROM RESERVES ;
```

BID	SID	DAY
01	20	MONDAY
02	30	TUESDAY
03	50	WEDNESDAY
04	10	THURSDAY
05	20	FRIDAY

```
SELECT * FROM SAILOR;
```

SID	SNAME	AGE	RATING
10	RAM	30	5
20	RAVI	25	4
30	MISHRA	22	3
40	CHANDRA	24	2
50	SHIVA	36	1
60	KRISHNA	40	6

```
SELECT * FROM BOAT ;
```

BID	BNAME	COLOUR
01	ABC	RED
02	XYZ	YELLOW
03	PQR	GREEN
04	LMN	BLACK
05	DEF	BLUE

4. Obtain the bid of the boats reserved by '#Sailor_Name'.

```
SELECT BID
FROM RESERVES
WHERE SID IN( SELECT SID FROM
              SAILOR WHERE SNAME='RAM');
```

BID

04

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

```
SELECT BID,SNAME
FROM RESERVES r join SAILOR s
on (r.SID=s.SID);
```

BID	SNAME
-----	-------

04	RAM
----	-----

05	RAVI
----	------

01	RAVI
----	------

02	MISHRA
----	--------

03	SHIVA
----	-------

6. Find the number of boats reserved by each sailor

```
SELECT COUNT(BID),SID
FROM RESERVES
GROUP BY SID;
```

COUNT(BID)	SID
------------	-----

1	50
---	----

2	20
---	----

1	10
---	----

1	30
---	----

- e) 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()
```

- f) 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;
/

```

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

g) Mention the constraints neatly.

Write the Constraints

h) Design the ER diagram for the problem statement

Design the ER diagram

i) State the schema diagram for the ER diagram.

Write the schema diagram for the ER diagram

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

```

CREATE TABLE PARTT
(
  PNO VARCHAR(6) NOT NULL,
  PNAME VARCHAR(10) ,
  COLOUR VARCHAR(10),
  PRIMARY KEY(PNO)
);

```

```

CREATE TABLE WAREHOUSE

```

```
(
WNO VARCHAR(6) NOT NULL,
WNAME VARCHAR(10),
CITY VARCHAR(10),
PRIMARY KEY(WNO)
);
```

```
CREATE TABLE SHIPMENT
```

```
(
PNO VARCHAR(6),
WNO VARCHAR(6),
QUANTITY NUMBER,
DATE DATE ,
FOREIGN KEY(PNO) REFERENCES PARTT(PNO) ON DELETE CASCADE,
FOREIGN KEY(WNO) REFERENCES WAREHOUSE(WNO) ON DELETE
CASCADE
);
```

```
INSERT INTO PARTT VALUES('01','ABC','RED');
INSERT INTO PARTT VALUES('02','DEF','BLUE');
INSERT INTO PARTT VALUES('03','LMN','GREEN');
INSERT INTO PARTT VALUES('04','PQR','YELLOW');
INSERT INTO PARTT VALUES('05','XYZ','PINK');
```

```
INSERT INTO WAREHOUSE VALUES('10','AAA','KUMTA');
INSERT INTO WAREHOUSE VALUES('20','BBB','MUMBAI');
INSERT INTO WAREHOUSE VALUES('30','CCC','BANGALORE');
INSERT INTO WAREHOUSE VALUES('40','DDD','UDUPI');
INSERT INTO WAREHOUSE VALUES('50','EEE','KARWAR');
```

```
SELECT * FROM PARTT;
```

PNO	PNAME	COLOUR
01	ABC	RED
02	DEF	BLUE
03	LMN	GREEN
04	PQR	YELLOW
05	XYZ	PINK

SELECT * FROM WAREHOUSE;

WNO	WNAME	CITY
10	AAA	KUMTA
20	BBB	MUMBAI
30	CCC	BANGALORE
40	DDD	UDUPI
50	EEE	KARWAR

INSERT INTO SHIPMENTS VALUES('01','20','300','28-FEB-2013');

INSERT INTO SHIPMENTS VALUES('02','30','400','30-JAN-2013');

INSERT INTO SHIPMENTS VALUES('03','10','00','31-JAN-2013');

INSERT INTO SHIPMENTS VALUES('04','40','600','31-MARCH-2013');

INSERT INTO SHIPMENTS VALUES('05','50','100','31-DEC-2013');

SELECT * FROM SHIPMENTS;

PNO	WNO	QUANTITY	DATEE
01	20	300	28-FEB-13
02	30	400	30-JAN-13
03	10	00	31-JAN-13
04	40	600	31-MAR-13
05	50	100	31-DEC-13

5. Obtain the Names of warehouses which have shipped red colored parts.

SELECT WNAME FROM WAREHOUSE

WHERE WNO IN(SELECT WNO FROM SHIPMENTS

WHERE PNO=(SELECT PNO FROM PARTT WHERE COLOUR='RED'));

WNAME

BBB

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

SELECT PNO,WNAME

FROM SHIPMENTS s join WAREHOUSE w on (s.WNO=w.WNO);

PNO WNAME

03 AAA

01 BBB

02 CCC

04 DDD

05 EEE

7. Find the number of parts supplied by each warehouse.

```
SELECT COUNT(PNO),WNO
FROM SHIPMENTS
GROUP BY WNO;
```

COUNT(PNO)	WNO
1	50
1	20
1	10
1	40
1	30

8. List the warehouse details which ships maximum quantity of parts.

- k) 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'})
```

3. Find the parts shipped from warehouse :Wname"

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

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

4. List the total quantity supplied from each warehouse

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

- l) 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;
/

```

Consider the Book Lending system from the library. The students are allowed 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 students need not compulsorily borrow books.

g) Mention the constraints neatly.

Write the constraints

h) Design the ER diagram for the problem statement

Design the ER diagram

i) State the schema diagram for the ER diagram.

Write the Schema diagram for the ER diagram.

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

```

create table books
(
ISBN varchar(10),
Title varchar(10),
Author varchar(10),
Publisher varchar(10),
primary key(ISBN)
);

```

Insert into books values(

ISBN TITLE AUTHOR PUBLISHER

123	T1	A1	P1
002	DB	A2	P2
003	T3	A3	P3

004	T4	A4	P4
005	T5	A5	P5

```
create table student1
(
  usn varchar(10),
  name varchar(10),
  sem int,
  dept varchar(3),
  primary key(usn)
);
```

Insert into student1 values(

SN	NAME	SEM	DEPT
111	aaa	3	ISE
222	bbb	4	CSE
333	ccc	3	CSE
444	ddd	4	ISE
555	eee	4	ISE

```
create table borrow
(
  ISBN varchar(10),
  usn varchar(10),
  dates varchar(10),
  foreign key(ISBN) references books(ISBN),
  foreign key(usn) references student1(usn)
);
```

insert into borrow values(

ISBN	USN	DATES
123	222	1/2/13
002	333	2/2/13
003	111	3/2/13
005	444	4/2/13
003	555	5/2/13

4. Obtain the name of the student who has borrowed the book bearing ISBN '123'
select NAME from student1

```
where USN=(select USN from borrow where ISBN='123');
```

```
NAME
```

```
bbb
```

5. Obtain the Names of students who have borrowed database books.

```
select NAME from student1
```

```
where USN=(select USN from borrow where ISBN=(select ISBN from books  
where TITLE='DB'));
```

```
NAME
```

```
ccc
```

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

```
select count(ISBN) from borrow
```

```
group by USN;
```

```
COUNT(ISBN)
```

```
1
```

```
1
```

```
1
```

```
1
```

```
1
```

- k) 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":'database',"AUTHOR":'ABC',"PUBLISHER":'selina',"SSN":2015,"date":'2017-05-29'})
```

```
>db.LIBRARY.insert({"ISBN":2233,"TITLE":'database',"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":'database',{ "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;
/
```