1

```
CREATE TABLE PROJECT (
    PNO VARCHAR(2) PRIMARY KEY,
    PNAME VARCHAR(10) NOT NULL
);

CREATE TABLE WORKS (
    SSN NUMBER, CREATE TABLE DEPARTMENT (
    DNO NUMBER PRIMARY KEY,
    DNAME VARCHAR(2) NOT NULL
);

CREATE TABLE EMPLOYEE (
    SSN NUMBER PRIMARY KEY,
    NAME VARCHAR(2) NOT NULL,
    DNO NUMBER,
    FOREIGN KEY (DNO) REFERENCES DEPARTMENT(DNO) ON DELETE SET NULL
);



    PNO VARCHAR(2),
    PRIMARY KEY (SSN,PNO),
    FOREIGN KEY (SSN) REFERENCES EMPLOYEE(SSN) ON DELETE CASCADE,
    FOREIGN KEY (PNO) REFERENCES PROJECT(PNO) ON DELETE CASCADE
);

INSERT INTO DEPARTMENT VALUES(101,'D1')
INSERT INTO EMPLOYEE VALUES(1,'E1',101)
INSERT INTO PROJECT VALUES('P1', 'ML')
INSERT INTO WORKS VALUES(1,'P5')


SELECT * FROM EMPLOYEE
WHERE SSN IN (
    SELECT SSN
    FROM WORKS
    WHERE PNO IN (
        SELECT PNO
        FROM PROJECT
        WHERE PNAME='DATABASE'
    )
);
```

```
SELECT E.DNO, COUNT(E.SSN), D.DNAME
FROM EMPLOYEE E, DEPARTMENT D
WHERE E.DNO=D.DNO
GROUP BY E.DNO,D.DNAME;
```

```
UPDATE WORKS
SET PNO='P4'
WHERE SSN=4;
```

```
SELECT *
FROM EMPLOYEE
WHERE SSN NOT IN (
   SELECT SSN
   FROM WORKS
   GROUP BY SSN
   HAVING COUNT(SSN)>=2
);
```

```
db.employee.insert({ssn:1, name:”Aditya”, dept:101, dname:”Database”, pno:”p1”})
db.employee.find({dept:101})
db.employee.find({pno:”p1”},{_id:0, name:true})
```

```
begin
   update EMPLOYEE
   set SALARY= SALARY * 1.15
   where DNO=101;
   dbms_output.put_line('Number of employees awarded hike are :- ' || sql%rowcount);
end;
/
```

2

```
CREATE TABLE PART (
   PID NUMBER PRIMARY KEY,
   PNAME VARCHAR(5) NOT NULL,
   COLOR VARCHAR(10) NOT NULL
);

CREATE TABLE SUPPLIER (
   SID NUMBER PRIMARY KEY,
   SNAME VARCHAR(5) NOT NULL,
   SADDR VARCHAR(10) NOT NULL
);


CREATE TABLE SUPPLY (
   PID NUMBER,
   SID NUMBER,
   QUANTITY NUMBER,
   PRIMARY KEY (PID,SID),
   FOREIGN KEY (SID) REFERENCES SUPPLIER (SID) ON DELETE CASCADE,
   FOREIGN KEY (PID) REFERENCES PART (PID) ON DELETE CASCADE
);



INSERT INTO PART VALUES (1,'P1','RED');
INSERT INTO SUPPLIER VALUES(101,'S1','DELHI');
INSERT INTO SUPPLY VALUES(1,101,3);

SELECT *
FROM PART
WHERE PID IN (
   SELECT PID
   FROM SUPPLY
   WHERE SID IN (
      SELECT SID
      FROM SUPPLIER
      WHERE SNAME='S1'
   )
);
```

```
SELECT SNAME
FROM SUPPLIER
WHERE SID IN (
   SELECT SID
   FROM SUPPLY
   WHERE PID IN(
      SELECT PID
      FROM PART
      WHERE PNAME='P1'
   )
);


DELETE FROM PART
WHERE COLOR LIKE 'GREEN';


SELECT * FROM PART;
SELECT * FROM SUPPLY;


SELECT *
FROM SUPPLIER
WHERE SID IN (
   SELECT SID
   FROM SUPPLY
   GROUP BY SID
   HAVING COUNT(PID)=2
);


CREATE TABLE SHIPMENT AS (
   SELECT *
   FROM SUPPLY
   WHERE 1=2
);


DECLARE
   CURSOR C1 IS SELECT * FROM SUPPLY;
   V_REC SUPPLY%ROWTYPE;
BEGIN
   OPEN C1;
```

```
    LOOP
    FETCH C1 INTO V_REC;
    EXIT WHEN C1%NOTFOUND;
    IF V_REC.PID IN (1,2,3,4,5) THEN
        INSERT INTO SHIPMENT VALUES(V_REC.PID, V_REC.SID, V_REC.QUANTITY);
    END IF;
    END LOOP;
    CLOSE C1;
END;
/

SELECT * FROM SHIPMENT;

db.parts.insert({pno:101, pname:”p1”, color:”red”, sno:1, sname:”s1”, address:”delhi”})
db.parts.find({pno:105})
db.parts.update({pno:101},{$set : {color:"blue"}})
```

3

```
CREATE TABLE BOAT (
   BID NUMBER PRIMARY KEY,
   BNAME VARCHAR(10) NOT NULL,
   COLOR VARCHAR(10)
);

CREATE TABLE SAILOR (
   SID NUMBER PRIMARY KEY,
   SNAME VARCHAR(10) NOT NULL,
   AGE NUMBER
);

CREATE TABLE RESERVES (
   BID NUMBER,
   SID NUMBER,
   DAY VARCHAR(10),
   PRIMARY KEY (BID,SID),
   FOREIGN KEY (BID) REFERENCES BOAT(BID) ON DELETE CASCADE,
   FOREIGN KEY (SID) REFERENCES SAILOR(SID) ON DELETE CASCADE
);


INSERT INTO BOAT VALUES (1,'B1','RED');
INSERT INTO SAILOR VALUES (101,'S1',35);
INSERT INTO RESERVES VALUES (1,101,'MONDAY');

SELECT *
FROM BOAT
WHERE BID IN (
   SELECT BID
   FROM RESERVES
   WHERE SID IN (
      SELECT SID
      FROM SAILOR
      WHERE SNAME = 'S2'
   )
);


SELECT BID,BNAME,COLOR
FROM BOAT B
WHERE NOT EXISTS (
   SELECT SID FROM SAILOR
   MINUS
   SELECT SID FROM RESERVES
   WHERE B.BID=RESERVES.BID
);
```

```
SELECT S.SID, S.SNAME, COUNT(R.BID)
FROM SAILOR S, RESERVES R
WHERE S.SID=R.SID
GROUP BY S.SID,S.SNAME;
```

```
SELECT B.BID,B.BNAME,S.SID,S.SNAME
FROM BOAT B,RESERVES R,SAILOR S
WHERE B.BID=R.BID
AND S.SID=R.SID
AND B.BNAME=S.SNAME;
```

```
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 a 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;
/
```

```
db.boats.insert({bid:1, bname:”b1”, color:”red”, sid:101, sname:”s1”})
db.boats.find({sname:'John'}).count()
db.boats.find({color:"red"},{_id:0, bname:true, bid:true})
```

4

```
CREATE TABLE BRANCH (
   BCODE NUMBER PRIMARY KEY,
   BNAME VARCHAR(5) NOT NULL
);

CREATE TABLE CUSTOMER (
   CCODE NUMBER PRIMARY KEY,
   CNAME VARCHAR(5) NOT NULL
);


CREATE TABLE ACCOUNT (
   BCODE NUMBER,
   CCODE NUMBER,
   ACODE NUMBER PRIMARY KEY,
   TYPE VARCHAR(5),
   FOREIGN KEY (BCODE) REFERENCES BRANCH(BCODE) ON DELETE CASCADE,
   FOREIGN KEY (CCODE) REFERENCES CUSTOMER(CCODE) ON DELETE CASCADE
);

CREATE TABLE TRANSACTION (
   CCODE NUMBER,
   ACODE NUMBER,
   TYPE VARCHAR(5),
   AMOUNT NUMBER,
   FOREIGN KEY(CCODE) REFERENCES CUSTOMER(CCODE) ON DELETE CASCADE,
   FOREIGN KEY(ACODE) REFERENCES ACCOUNT(ACODE) ON DELETE CASCADE
);


INSERT INTO BRANCH VALUES (1,'B1');
INSERT INTO CUSTOMER VALUES(101,'C1');
INSERT INTO ACCOUNT VALUES (1,101,1000, 'C');
INSERT INTO TRANSACTION VALUES (101,1000,'W',100);

SELECT *
FROM CUSTOMER
WHERE CCODE IN (
   SELECT CCODE
   FROM ACCOUNT
   WHERE TYPE = 'C'
```

```
    INTERSECT
    SELECT CCODE
    FROM ACCOUNT
    WHERE TYPE='S'
);


SELECT B.BCODE,B.BNAME, COUNT(A.ACODE) AS NO_OF_ACCOUNTS
FROM BRANCH B,ACCOUNT A
WHERE B.BCODE=A.BCODE
GROUP BY B.BCODE,B.BNAME;


SELECT C.CCODE,C.CNAME
FROM CUSTOMER C,TRANSACTION T
WHERE C.CCODE = T.CCODE
GROUP BY C.CCODE,C.CNAME
HAVING COUNT(T.ACODE)>=3;


SELECT *
FROM BRANCH
WHERE BCODE IN (
    SELECT BCODE
    FROM ACCOUNT
    GROUP BY(BCODE)
    HAVING COUNT(BCODE) < (
        (SELECT COUNT(*) FROM ACCOUNT)
        /
        (SELECT COUNT(*) FROM BRANCH)
    )
);


CREATE TABLE BCOPY AS(
    SELECT * FROM BRANCH WHERE 1=2
);

SELECT * FROM BCOPY;

DECLARE
    CURSOR C1 IS SELECT * FROM BRANCH;
    V_REC BRANCH%ROWTYPE;
BEGIN
```

```
   OPEN C1;
   LOOP
   FETCH C1 INTO V_REC;
   EXIT WHEN C1%NOTFOUND;
   INSERT INTO BCOPY VALUES(V_REC.BCODE,V_REC.BNAME);
   END LOOP;
   CLOSE C1;

END;
/

db.accounts.insert({bcode:101, bname:"b1", ccode:1, cname:"c1", acode:100});
db.accounts.find({bcode:101},{_id:0,bname:true})
db.accounts.aggregate([ {$group : {_id:"$cname", count : {$sum:1}} }])
```

5

```
CREATE TABLE BOOK (
   ISBN NUMBER PRIMARY KEY,
   TITLE VARCHAR(10),
   AUTHOR VARCHAR(10),
   PUBLISHER VARCHAR(10)
);

CREATE TABLE STUDENT (
   USN NUMBER PRIMARY KEY,
   NAME VARCHAR(10),
   SEX VARCHAR(1)
);

CREATE TABLE BORROW (
   ISBN NUMBER,
   USN NUMBER,
   BDATE VARCHAR(10),
   PRIMARY KEY (ISBN,USN, BDATE),
   FOREIGN KEY (ISBN) REFERENCES BOOK(ISBN) ON DELETE CASCADE,
   FOREIGN KEY (USN) REFERENCES STUDENT(USN) ON DELETE CASCADE
);


INSERT INTO BOOK VALUES (121,'T1','A1','P1');
INSERT INTO STUDENT VALUES (1,'S1','M');
INSERT INTO BORROW VALUES (123,1,'1-02-23');

SELECT NAME
FROM STUDENT
WHERE USN IN (
   SELECT USN
   FROM BORROW
   WHERE ISBN IN (123,124)
);

SELECT NAME
FROM STUDENT
WHERE SEX='F'
AND USN IN(
   SELECT USN
   FROM BORROW
```

```
    WHERE ISBN IN (
        SELECT ISBN
        FROM BOOK
        WHERE TITLE IN 'DATABASE'
    )
);
```

```
SELECT S.USN, S.NAME, COUNT(B.ISBN)
FROM BORROW B,STUDENT S
WHERE B.USN=S.USN
GROUP BY S.USN,S.NAME;
```

```
SELECT TITLE
FROM BOOK
WHERE TITLE LIKE 'DA%'
AND ISBN NOT IN (
    SELECT ISBN
    FROM BORROW
);
```

```
db.books.insert({isbn:121, title="database", author:"a1", usn:1, name:"s1"})
db.books.find({author:'a1'})
db.books.find({title:'database'},{_id:0,name:true})
```

```
declare
    n number :=&n;
    a number :=0;
    b number :=1;
    c number;
    i number;

procedure fibo is
    begin
        for i in 2..n loop
            c := a+b;
            dbms_output.put_line(c);
            a :=b;
            b :=c;
        end loop;
    end;
```

```
begin
    dbms_output.put_line(a);
    dbms_output.put_line(b);
    fibo;

end;
/
```