```
create database graph;                              //Creating Database named 'graph'

create table node(
        x int,
        y int,
        PRIMARY KEY CLUSTERED
        (
                x,
                y
        )
);                                                  //Creating table To store tuples

insert into node values(5 , 5);
insert into node values(15 , 5);
insert into node values(5 , 3);
insert into node values(11 , 3);                    // 1(a) . Insert Tuples into node

select * from node;                                 // 1(b) . SQL query that return all the nodes

select * from node where x=y;                       // 1(c) . SQL statement that return all nodes whose x
                                                    and y co-ordinates are same

create table graphedges (
        px int,
        py int,
        cx int,
        cy int,
        weight int DEFAULT 1,
        PRIMARY KEY CLUSTERED
        (
                px,
                py,
                cx,
                cy
        ),
        foreign key (px , py) REFERENCES node(x , y)
        on delete cascade,
        foreign key (cx , cy) REFERENCES node(x , y)
        on delete cascade
);                                                  //Relational schema to store edges between the nodes

insert into graphedges values(5,5,15,5,10);
insert into graphedges values(5,5,5,3,2);
insert into graphedges values(5,5,11,3,3);
insert into graphedges values(11,3,5,3,6);
insert into graphedges values(15,5,5,3,7);          // 2 (a) . Inserting tuples maintaining constraints of 3
```

```
update graphedges
set weight = 10
where (px = 5 and py = 5 and cx = 15 and cy = 5);          //  2 (b) . Update an edge

SELECT * FROM graphedges;                                  // 2 (c) . Show all the edges

DELETE FROM nodes
WHERE x=15 and y =5 ;                                      //Checking node delete maintaining constraints of 3
```