

--QUESTION C)

1) People to whom Brad can reach to either directly or transitively.

The screenshot shows the SQL Developer interface. The 'Worksheet' tab contains the following SQL query:

```
insert into SocialNetwork values ('Quincey', 'James');
select * from SocialNetwork;
WITH DIRECTFRIEND(P,F) AS (
  (SELECT PERSON AS P, FRIEND AS F FROM SocialNetwork)
  UNION ALL
  (SELECT DF1.P,DF2.FRIEND
   FROM DIRECTFRIEND DF1, SocialNetwork DF2
   WHERE DF1.F=DF2.PERSON)
)CYCLE F SET IS_CYCLE TO 1 DEFAULT 0
SELECT DISTINCT F FROM DIRECTFRIEND WHERE P='Brad' AND F!='Brad' AND IS_CYCLE=0;
```

The 'Query Result' tab shows the output of the query, listing 11 names: 1 Leo, 2 Penny, 3 Dave, 4 Gary, 5 Amy, 6 Kate, 7 Melissa, 8 Oliver, 9 Nicole, 10 Hannah, 11 Quincey.

The 'SQL History' tab shows the executed query and its duration (0.207 seconds).

2) People in DePauledIN network to whom Brad cannot reach to.

The screenshot shows the SQL Developer interface. The 'Worksheet' tab contains the following SQL query:

```
--2)People in DePauledIN network to whom Brad cannot reach to.
WITH DIRECTFRIEND(P,F) AS (
  (SELECT PERSON AS P, FRIEND AS F FROM SocialNetwork)
  UNION ALL
  (SELECT DF1.P,DF2.FRIEND
   FROM DIRECTFRIEND DF1, SocialNetwork DF2
   WHERE DF1.F=DF2.PERSON)
)CYCLE F SET IS_CYCLE TO 1 DEFAULT 0
SELECT DISTINCT F FROM DIRECTFRIEND WHERE P!='Brad' AND F!='Brad' AND IS_CYCLE=0;

--3)Only those people who are connected to Christine via transitive relationship i.e not an immediate follower.
WITH DIRECTFRIEND(P,F) AS (
  (SELECT PERSON AS P, FRIEND AS F FROM SocialNetwork)
  UNION ALL
  (SELECT DF1.P,DF2.FRIEND
   FROM DIRECTFRIEND DF1, SocialNetwork DF2
   WHERE DF1.F=DF2.PERSON)
)
```

The 'Query Result' tab shows the output of the query, listing 8 names: 1 Leo, 2 Dave, 3 Penny, 4 Kate, 5 Melissa, 6 Amy, 7 Gary, 8 Oliver.

The 'SQL History' tab shows the executed query and its duration (0.324 seconds).

- 3) Only those people who are connected to Christine via transitive relationship i.e not an immediate follower.

The screenshot shows the SQL Developer interface. The left pane displays the 'Connections' tree with 'Oracle Connections' and 'Database Schema Service Connections'. The main pane shows a SQL worksheet with the following query:

```
WHERE DF1.F=DF2.PERSON)
)CYCLE F SET IS_CYCLE TO 1 DEFAULT 0
SELECT DISTINCT F FROM DIRECTFRIEND WHERE P!= 'Brad' AND F!= 'Brad' AND IS_CYCLE=0;

--3)Only those people who are connected to Christine via transitive relationship i.e not an immediate follower.
WITH DIRECTFRIEND(P,F) AS (
(SELECT PERSON AS P, FRIEND AS F FROM SocialNetwork)
UNION ALL
(SELECT DF1.P,DF2.FRIEND
FROM DIRECTFRIEND DF1, SOCIALNETWORK DF2
WHERE DF1.F=DF2.PERSON)
)CYCLE F SET IS_CYCLE TO 1 DEFAULT 0
SELECT DISTINCT F FROM DIRECTFRIEND WHERE P!= 'Christine' AND F!= 'Christine' AND IS_CYCLE=0;

--4)Find the shortest path to reach from Amy to James.
```

The 'Query Result' pane shows 8 rows of results:

F
1 Leo
2 Penny
3 Dave
4 Gary
5 Kate
6 Melissa
7 Amy
8 Brad

The 'SQL History' pane shows the executed query and its duration (0.162 seconds).

- 4) Find the shortest path to reach from Amy to James.

The screenshot shows the SQL Developer interface. The left pane displays the 'Connections' tree with 'Oracle Connections' and 'Database Schema Service Connections'. The main pane shows a SQL worksheet with the following query:

```
select 'Edgar', 'Kate', 1 from socialnetwork union all
select 'Edgar', 'Melissa', 1 from socialnetwork union all
select 'Edgar', 'Nicole', 1 from socialnetwork union all
select 'Fiona', 'Amy', 1 from socialnetwork union all
select 'Gary', 'Oliver', 1 from socialnetwork union all
select 'Hannah', 'Quincey', 1 from socialnetwork union all
select 'James', 'Quincey', 1 from socialnetwork union all
select 'Melissa', 'Leo', 1 from socialnetwork union all
select 'Oliver', 'Fiona', 1 from socialnetwork union all
select 'Oliver', 'Penny', 1 from socialnetwork union all
select 'Quincey', 'James', 1 from socialnetwork

),
grid as(
select level+1
      friend,
      sys_connect_by_path(person,'/')|| '/'|| friend as PATH,
      '0' || sys_connect_by_path(distance, '+') as DISTANCE
from paths
start with person = 'Amy'
connect by prior friend = person
)
select points,
path,
distance,
EvalMath(distance) as DISTANCE
from grid
where FRIEND = 'James'
```

The 'Query Result' pane shows 0 rows fetched in 0.12 seconds. The 'SQL History' pane shows the executed query and its duration (0.12 seconds).

