Spark - Exercises

Exercise #52

- GraphFrame
- Input:
  - The textual file vertexes.csv
    - It contains the vertexes of a graph
  - Each vertex is characterized by
    - id (string): user identifier
    - name (string): user name
    - age (integer): user age

Exercise #52

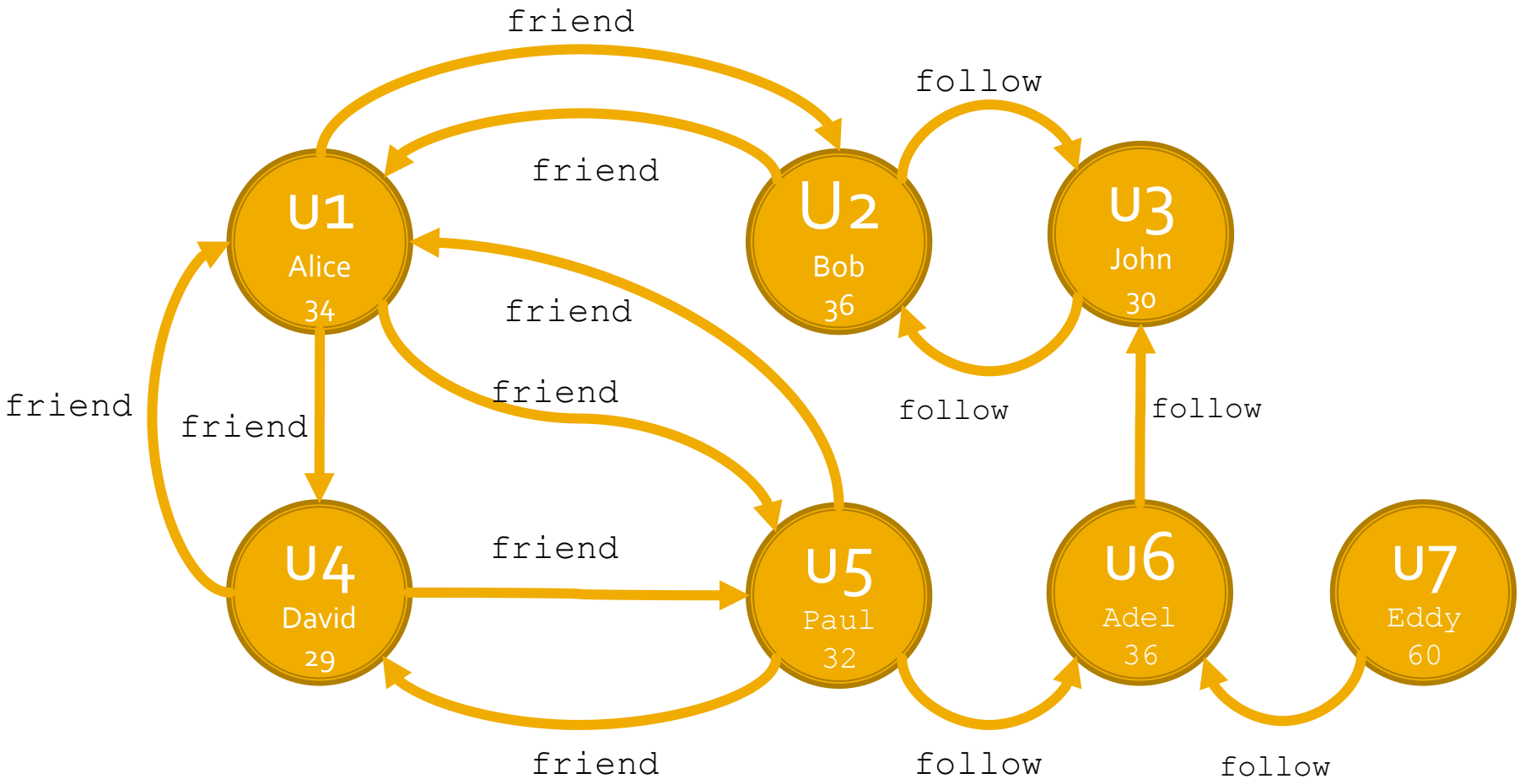
- The textual file edges.csv
  - It contains the edges of a graph
- Each edge is characterized by
  - src (string): source vertex
  - dst (string): destination vertex
  - linktype (string): “follow”or “friend”

Exercise #52

- Output:
  - For each user with at least one follower, store in the output folder the number of followers
    - One user per line
    - Format: user id, number of followers
  - Use the CSV format to store the result

Exercise #52

Input graph example



Exercise #52

- Result

id	numFollowers
u3	2
u6	2
u2	1

Exercise #53

- GraphFrame
- Input:
  - The textual file vertexes.csv
    - It contains the vertexes of a graph
  - Each vertex is characterized by
    - id (string): user identifier
    - name (string): user name
    - age (integer): user age

Exercise #53

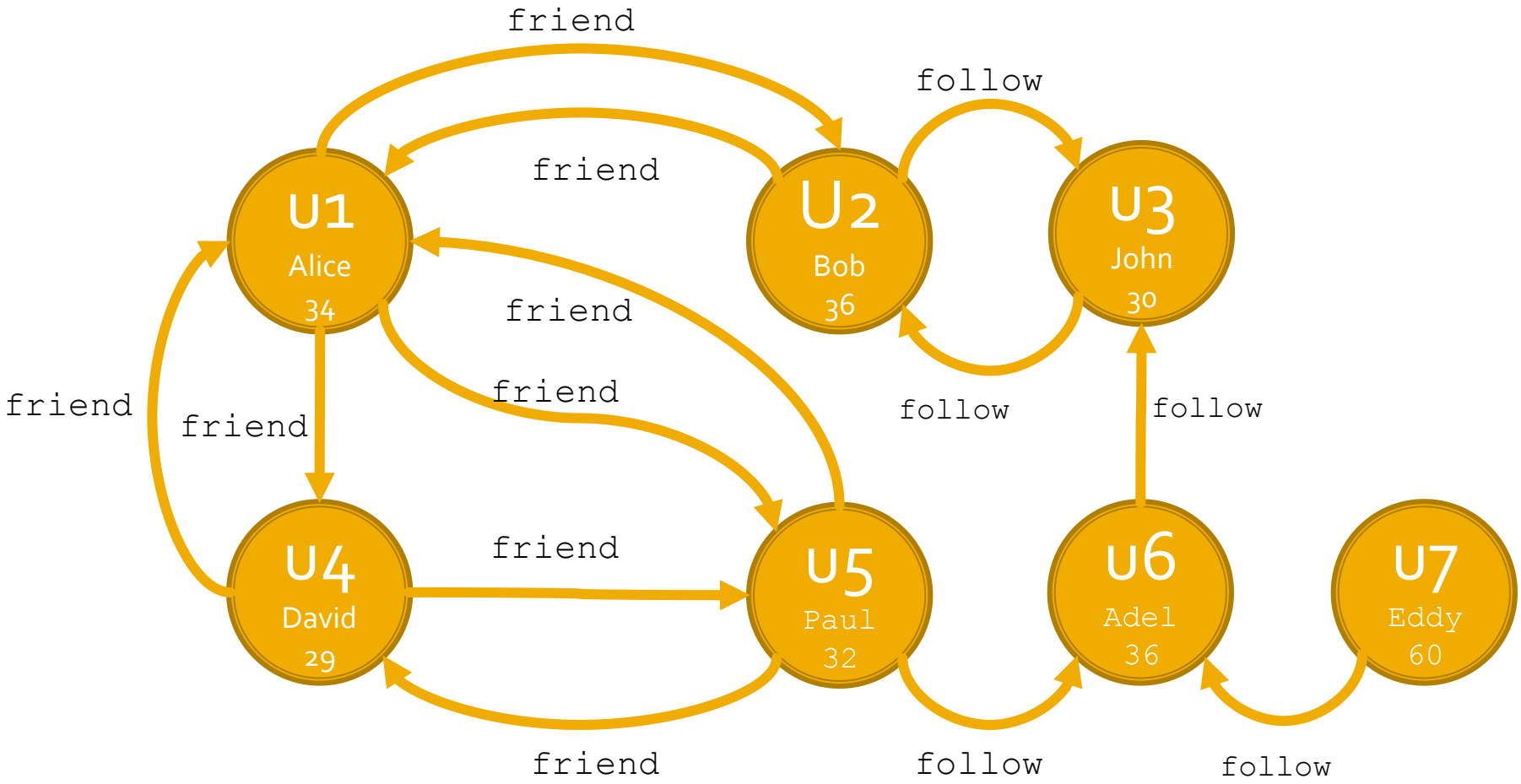
- The textual file edges.csv
  - It contains the edges of a graph
- Each edge is characterized by
  - src (string): source vertex
  - dst (string): destination vertex
  - linktype (string): “follow”or “friend”

Exercise #53

- Output:
  - Consider only the users with at least one follower
  - Store in the output folder the user(s) with the maximum number of followers
    - One user per line
    - Format: user id, number of followers
  - Use the CSV format to store the result

Exercise #53

Input graph example



Exercise #53

- Result

id	numFollowers
u3	2
u6	2

Exercise #54

- GraphFrame
- Input:
  - The textual file vertexes.csv
    - It contains the vertexes of a graph
  - Each vertex is characterized by
    - id (string): user identifier
    - name (string): user name
    - age (integer): user age

Exercise #54

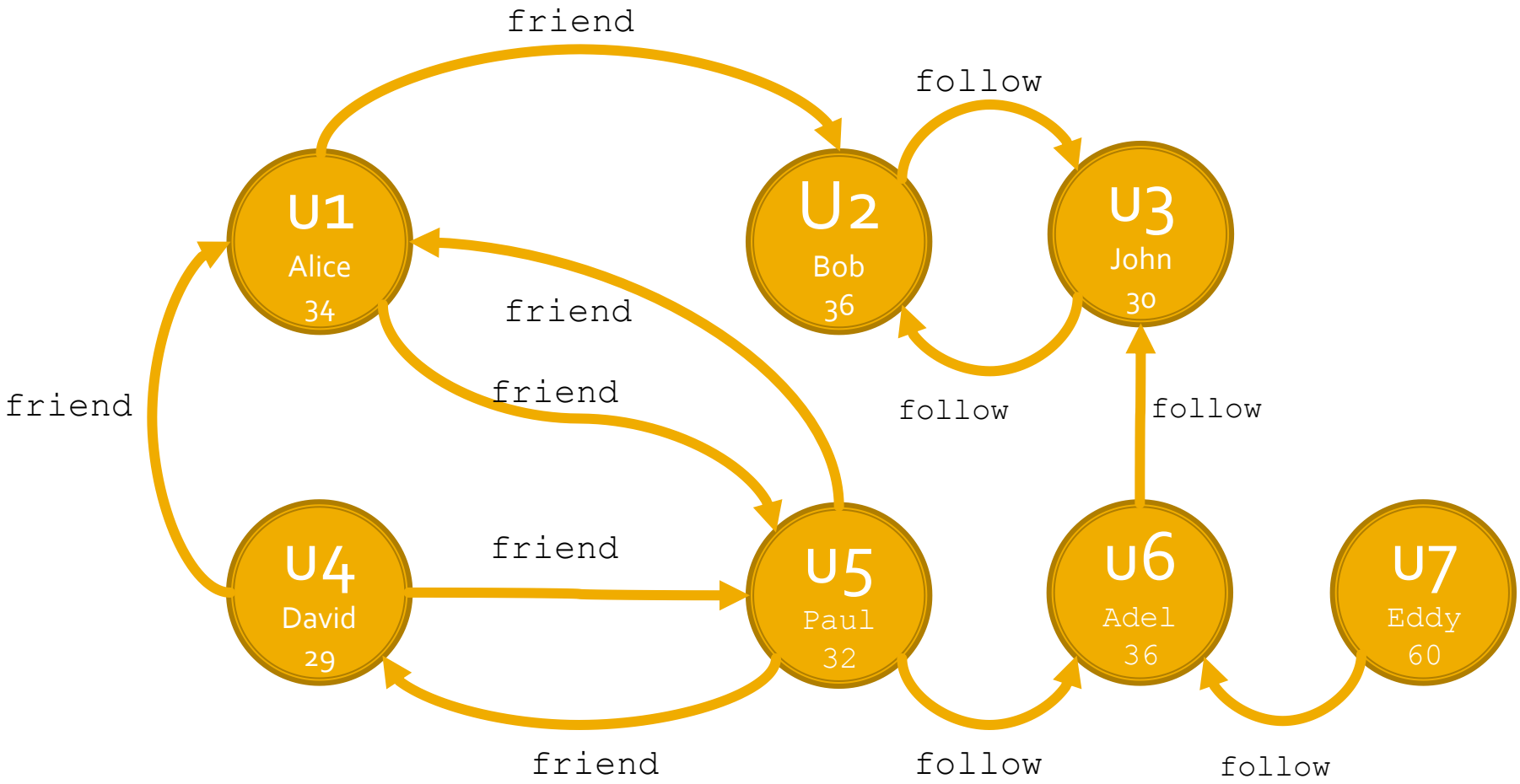
- The textual file edges.csv
  - It contains the edges of a graph
- Each edge is characterized by
  - src (string): source vertex
  - dst (string): destination vertex
  - linktype (string): “follow”or “friend”

Exercise #54

- Output:
  - The pairs of users Ux, Uy such that
    - Ux is a friend of Uy (link "friend" from Ux to Uy)
    - Uy is not a friend of Ux (no link "friend" from Uy to Ux)
  - One pair Ux,Uy per line
  - Format: idUx, idUy
  - Use the CSV format to store the result

Exercise #54

Input graph example



Exercise #54

- Result

IdFriend	IdNotFriend
U4	U1
U1	U2

Exercise #55

- GraphFrame
- Input:
  - The textual file vertexes.csv
    - It contains the vertexes of a graph
  - Each vertex is characterized by
    - id (string): vertex identifier
    - entityType (string): “user” or “topic”
    - name (string): name of the entity

Exercise #55

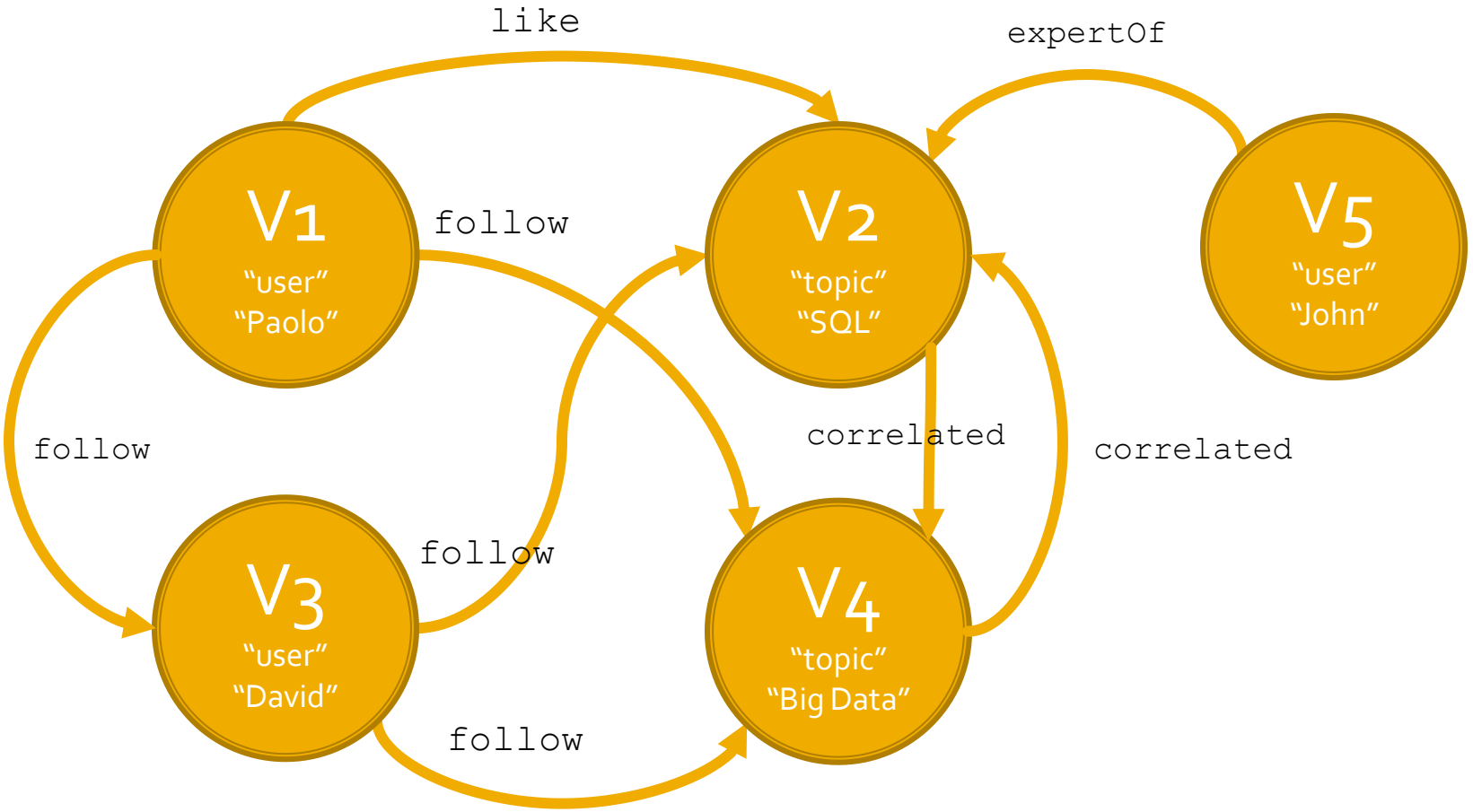
- The textual file edges.csv
  - It contains the edges of a graph
- Each edge is characterized by
  - src (string): source vertex
  - dst (string): destination vertex
  - linktype (string): “expertOf” or “follow” or “correlated”

Exercise #55

- Output:
  - The followed topics for each user
  - One pair (user name, followed topic) per line
  - Format: username, followed topic
  - Use the CSV format to store the result

Exercise #55

Input graph example



Exercise #55

- Result

username	topic
Paolo	Big Data
David	SQL
David	Big Data

Exercise #56

- GraphFrame
- Input:
  - The textual file vertexes.csv
    - It contains the vertexes of a graph
  - Each vertex is characterized by
    - id (string): vertex identifier
    - entityType (string): “user” or “topic”
    - name (string): name of the entity

Exercise #56

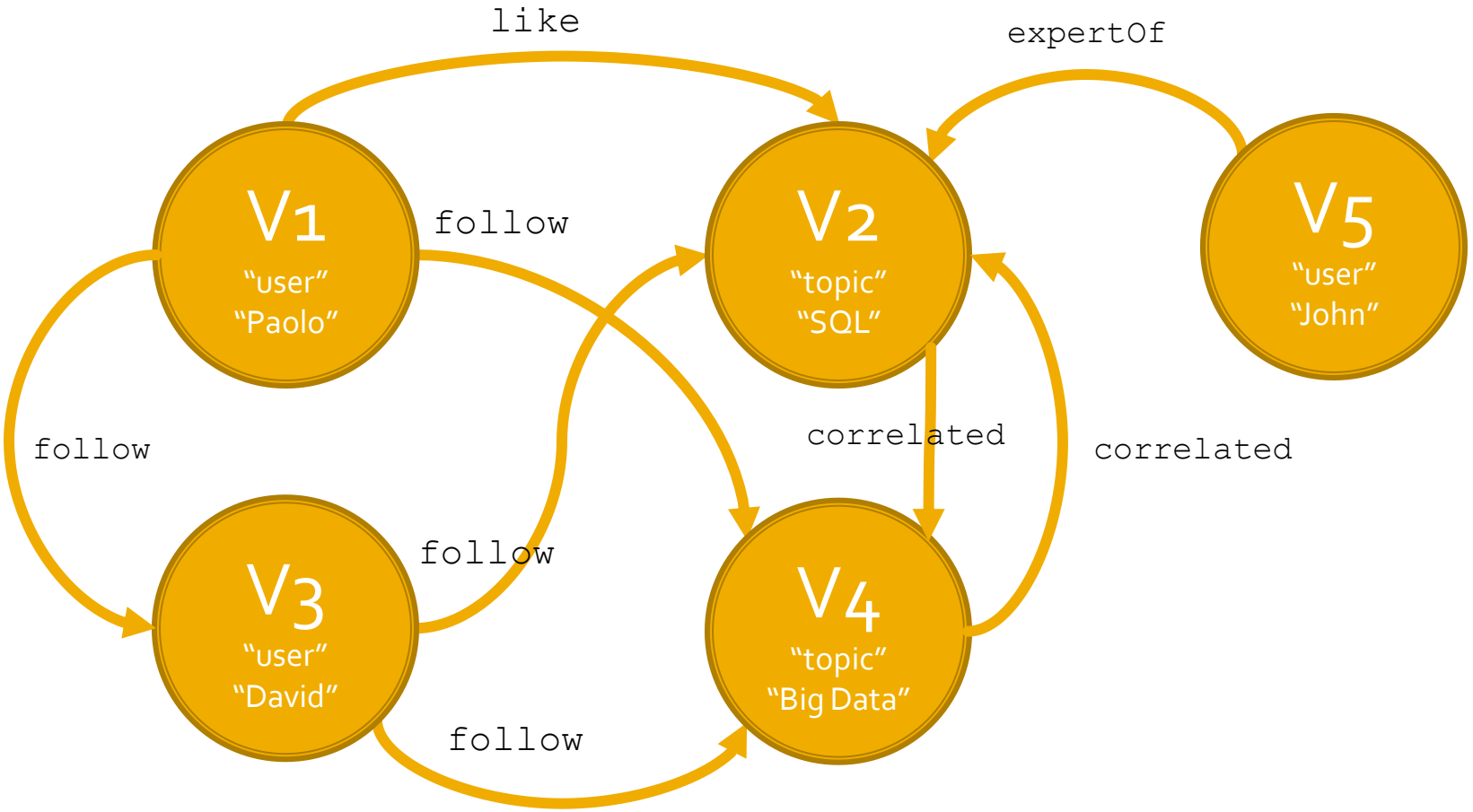
- The textual file edges.csv
  - It contains the edges of a graph
- Each edge is characterized by
  - src (string): source vertex
  - dst (string): destination vertex
  - linktype (string): “expertOf” or “follow” or “correlated”

Exercise #56

- Output:
  - The names of the users who follow a topic correlated to the "Big Data" topic
  - One user name per line
  - Format: username
  - Use the CSV format to store the result

Exercise #56

Input graph example



Exercise #56

- Result

username
David

Exercise #57

- GraphFrame
- Input:
  - The textual file vertexes.csv
    - It contains the vertexes of a graph
  - Each vertex is characterized by
    - id (string): user identifier
    - name (string): user name
    - age (integer): user age

Exercise #57

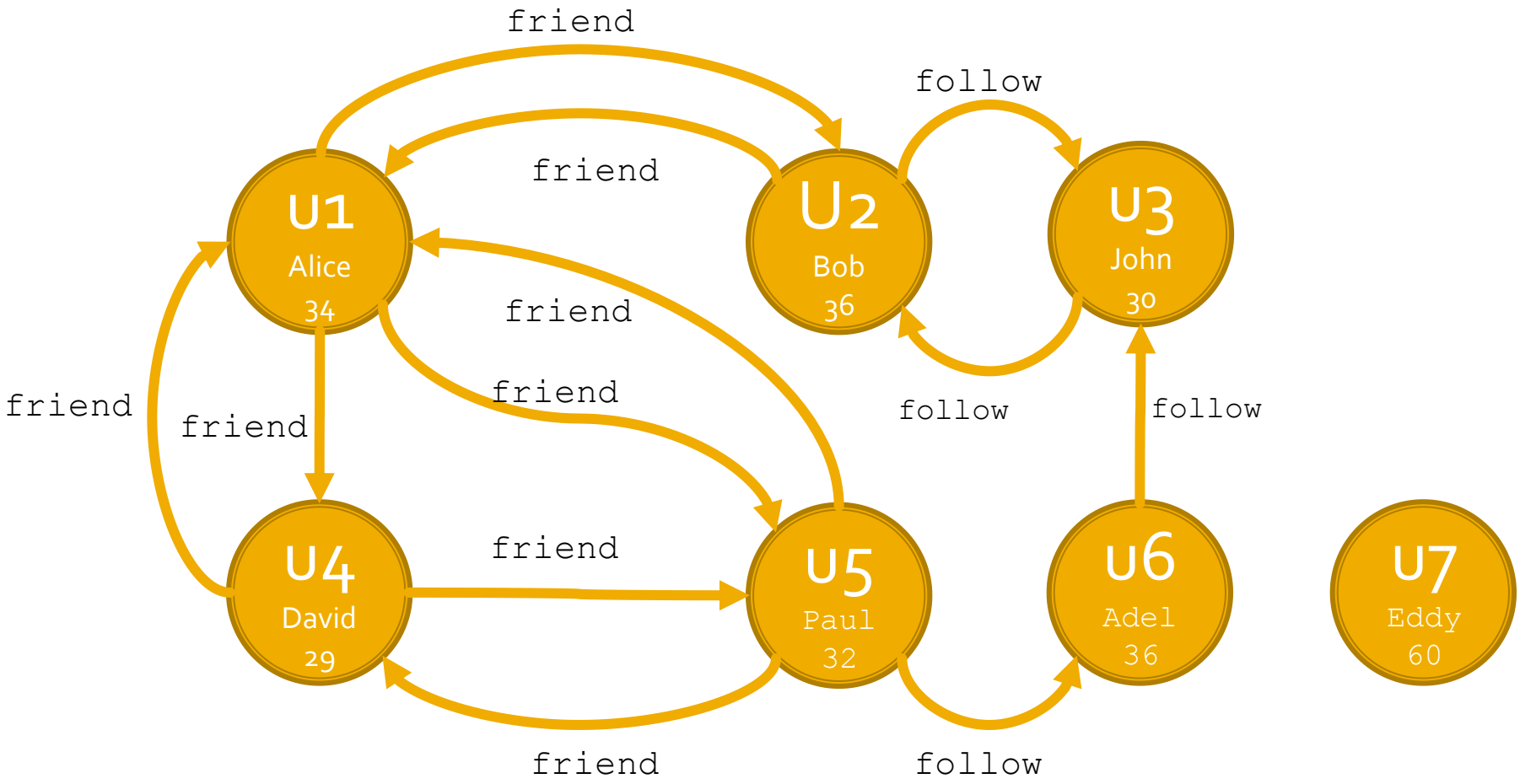
- The textual file edges.csv
  - It contains the edges of a graph
- Each edge is characterized by
  - src (string): source vertex
  - dst (string): destination vertex
  - linktype (string): “follow”or “friend”

Exercise #57

- Output:
  - Select the users who can reach user u1 in less than 3 hops (i.e., at most two edges)
    - Do not consider u1 itself
  - For each of the selected users, store in the output folder his/her name and the minimum number of hops to reach user u1
    - One user per line
    - Format: user name, #hops to user u1
  - Use the CSV format to store the result

Exercise #57

Input graph example



Exercise #57

- Result

name	numHops
Bob	1
John	2
David	1
Paul	1

Exercise #57b

- GraphFrame
- Input:
  - The textual file vertexes.csv
    - It contains the vertexes of a graph
  - Each vertex is characterized by
    - id (string): user identifier
    - name (string): user name
    - age (integer): user age

Exercise #57b

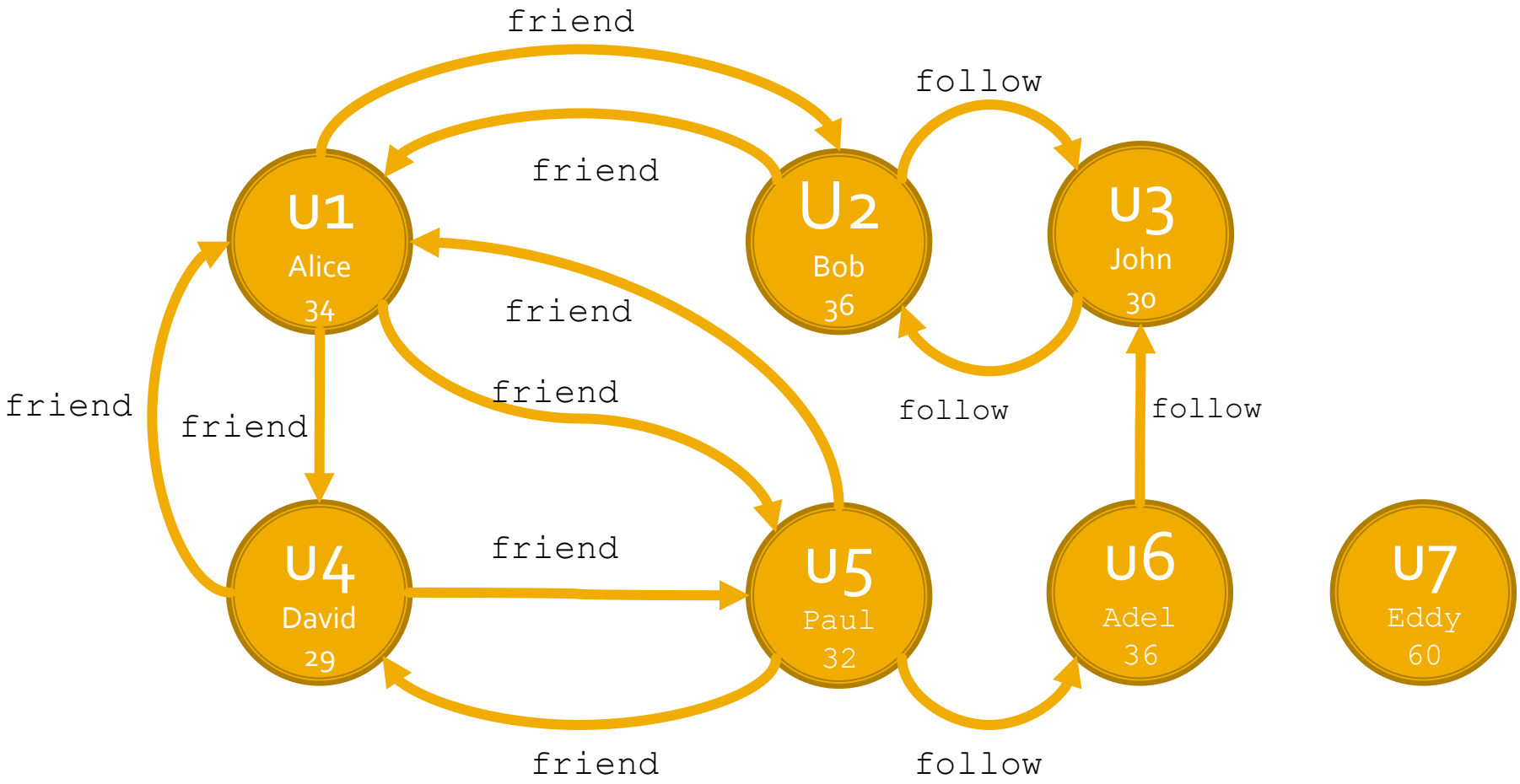
- The textual file edges.csv
  - It contains the edges of a graph
- Each edge is characterized by
  - src (string): source vertex
  - dst (string): destination vertex
  - linktype (string): “follow”or “friend”

Exercise #57b

- Output:
  - Count for each user the number of "neighbors" with ages less than 35
    - User X is a neighbor of User Y if there is a link from User X to User Y
  - For each user with at least one neighbor with ages less than 35, store in the output folder his/her id and the number of neighbors with ages less than 35
  - Use the CSV format to store the result

Exercise #57b

Input graph example



Exercise #57b

- Result

Id	numNeighborsLess35
u1	2
u2	2
u4	2
u5	2
u6	1