

```
# Lab Assignment #2 Database Design and Implementation
```

```
# if no module found, install using this command: !pip install networkx
import networkx as nx
```

```
# if no module found, install using this command: !pip install matplotlib
import matplotlib.pyplot as plt
```

```
# create graph to represent the social network of students and their connection
G = nx.Graph()
```

```
# student list
students= ["Alice", "Bob", "Charlie", "David", "Eve", "Frank", "Grace"]
```

```
# add students as nodes to the graph
G.add_nodes_from(students)
```

```
print(students)
```

```
['Alice', 'Bob', 'Charlie', 'David', 'Eve', 'Frank', 'Grace']
```

```
# list of connections between students, represents a connection between two students
```

```
connections = [
    ("Alice", "Bob"),
    ("Alice", "Charlie"),
    ("Bob", "Charlie"),
    ("Bob", "David"),
    ("Charlie", "David"),
    ("Charlie", "Eve"),
    ("David", "Eve"),
    ("Eve", "Frank"),
    ("Frank", "Grace"),
    ("Grace", "Eve")
]
```

```
# add connections as edges to the graph
G.add_edges_from(connections)
```

```
print(connections)
```

```
[('Alice', 'Bob'), ('Alice', 'Charlie'), ('Bob', 'Charlie'), ('Bob', 'David'), ('Charlie', 'David'), ('Charlie', 'Eve'), ('David', 'Eve')]
```

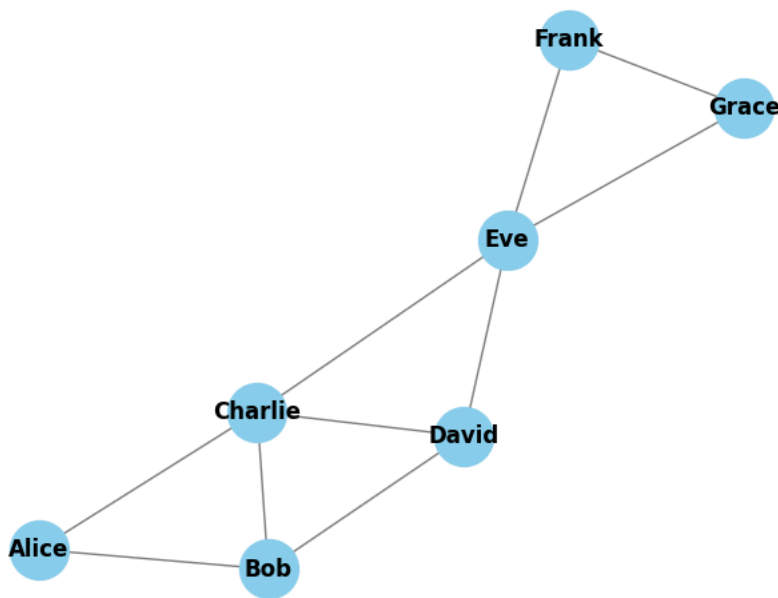
```
# print basic information about the graph
print("Nodes of the graph:", G.nodes())
print("Edges of the graph:", G.edges())
print("Number of nodes:", G.number_of_nodes())
print("Number of edges:", G.number_of_edges())
```

```
Nodes of the graph: ['Alice', 'Bob', 'Charlie', 'David', 'Eve', 'Frank', 'Grace']
Edges of the graph: [('Alice', 'Bob'), ('Alice', 'Charlie'), ('Bob', 'Charlie'), ('Bob', 'David'), ('Charlie', 'David'), ('Charlie', 'Eve'), ('David', 'Eve'), ('Eve', 'Frank'), ('Frank', 'Grace'), ('Grace', 'Eve')]
Number of nodes: 7
Number of edges: 10
```

```
# visualize network
```

```
nx.draw(G, with_labels=True, font_weight='bold', node_color='skyblue', node_size=1000, edge_color='gray')
plt.title("Social Network Graph Model")
plt.show()
```

Social Network Graph Model



```

# centrality means a network is directly connected to many others (degree centrality)
degree centrality = nx.degree_centrality(G)
print("\nDegree Centrality:")
for student, centrality in degree_centrality.items():
    print(f"{student}: {centrality:.2f}")

```

```

Degree Centrality:
Alice: 0.33
Bob: 0.50
Charlie: 0.67
David: 0.50
Eve: 0.67
Frank: 0.33
Grace: 0.33

```

```

# serve as a key broker between many other nodes (betweenness centrality)
betweenness centrality = nx.betweenness_centrality(G)
print("\nBetweenness Centrality:")
for student, centrality in betweenness_centrality.items():
    print(f"{student}: {centrality:.2f}")

```

```

Betweenness Centrality:
Alice: 0.00
Bob: 0.03
Charlie: 0.33
David: 0.10
Eve: 0.53
Frank: 0.00
Grace: 0.00

```

```

# close to many others indirectly (closeness centrality)
closeness centrality = nx.closeness_centrality(G)
print("\nCloseness Centrality:")
for student, centrality in closeness_centrality.items():
    print(f"{student}: {centrality:.2f}")

```

```

Closeness Centrality:
Alice: 0.50
Bob: 0.55
Charlie: 0.75
David: 0.67
Eve: 0.75
Frank: 0.50
Grace: 0.50

```

