

```

# 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'), ('Eve', 'Frank'), ('Frank', 'Grace'), ('Grace', '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: []
Number of nodes: 7
Number of edges: 0

# 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: 0.00
Bob: 0.00
Charlie: 0.00
David: 0.00
Eve: 0.00
Frank: 0.00
Grace: 0.00
```

```
# 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: 0.00
Charlie: 0.00
David: 0.00
Eve: 0.00
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: 0.00
Bob: 0.00
Charlie: 0.00
David: 0.00
Eve: 0.00
Frank: 0.00
Grace: 0.00
```

