Scenario 1: You have a list of dictionaries containing information about employees in a company. Each dictionary represents one employee, and has keys for 'name', 'age', 'salary', and 'department'. Create a data frame from this data, and set the employee names as the index.

Question: How can you create a Pandas data frame from the list of employee dictionaries, and set the index to be the employee names?

import pandas as pd

# List of employee dictionaries

employees = [

{'name': 'Alice', 'age': 25, 'salary': 50000, 'department': 'Marketing'},

{'name': 'Bob', 'age': 30, 'salary': 60000, 'department': 'Sales'},

{'name': 'Charlie', 'age': 35, 'salary': 70000, 'department': 'Engineering'},

{'name': 'David', 'age': 40, 'salary': 80000, 'department': 'Finance'}

]

# Create data frame and set index

df = pd.DataFrame(employees)

df.set\_index('name', inplace=True)

Scenario 2: You have a data frame with information about customers of an online store. The data frame has columns for 'customer\_id', 'name', 'age', 'city', and 'total\_spent'. You want to create a new data frame that only includes the customers from a specific city, and select only the 'name' and 'total\_spent' columns.

Question: How can you create a new data frame that only includes customers from a specific city, and select only the 'name' and 'total\_spent' columns?

# Original data frame

df = pd.DataFrame({

'customer\_id': [1, 2, 3, 4],

'name': ['Alice', 'Bob', 'Charlie', 'David'],

'age': [25, 30, 35, 40],

'city': ['New York', 'Los Angeles', 'Chicago', 'New York'],

'total\_spent': [100, 200, 300, 400]

})

# Create new data frame for customers in New York, with only 'name' and 'total\_spent' columns

ny\_customers = df.loc[df['city'] == 'New York', ['name', 'total\_spent']]

Scenario 3: You have a data frame with information about students in a class. The data frame has columns for 'student\_id', 'name', 'age', 'major', and 'gpa'. You want to select a subset of the data frame using integer-based indexing, including rows 2 through 4 and columns 1 through 3.

Question: How can you select a subset of the data frame using integer-based indexing, including rows 2 through 4 and columns 1 through 3?

# Original data frame

df = pd.DataFrame({

'student\_id': [1, 2, 3, 4, 5],

'name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve'],

'age': [18, 19, 20, 21, 22],

'major': ['Biology', 'Chemistry', 'Physics', 'Math', 'Computer Science'],

'gpa': [3.5, 3.2, 3.7, 3.9, 3.8]

})

# Select rows 2 through 4 and columns

# Select rows 2 through 4 and columns 1 through 3 using iloc

subset = df.iloc[1:4, 1:4]

Scenario 4: You have a data frame with information about books in a library. The data frame has columns for 'book\_id', 'title', 'author', 'publisher', and 'year'. You want to select a subset of the data frame using label-based indexing, including all rows with 'author' values in a given list and columns 'title' and 'year'.

Question: How can you select a subset of the data frame using label-based indexing, including all rows with 'author' values in a given list and columns 'title' and 'year'?

# Original data frame

df = pd.DataFrame({

'book\_id': [1, 2, 3, 4, 5],

'title': ['The Great Gatsby', 'To Kill a Mockingbird', '1984', 'Pride and Prejudice', 'Animal Farm'],

'author': ['F. Scott Fitzgerald', 'Harper Lee', 'George Orwell', 'Jane Austen', 'George Orwell'],

'publisher': ['Scribner', 'J. B. Lippincott & Co.', 'Secker & Warburg', 'T. Egerton, Whitehall', 'Secker & Warburg'],

'year': [1925, 1960, 1949, 1813, 1945]

})

# Select rows with 'author' values in given list and columns 'title' and 'year' using loc

subset = df.loc[df['author'].isin(['George Orwell', 'F. Scott Fitzgerald']), ['title', 'year']]

Scenario 5: You have two data frames, one with information about books and another with information about authors. The book data frame has columns for 'book\_id', 'title', 'author\_id', 'publisher', and 'year', and the author data frame has columns for 'author\_id' and 'author\_name'. You want to merge the two data frames so that you have a single data frame with columns for 'book\_id', 'title', 'author\_name', 'publisher', and 'year'.

Question: How can you merge the two data frames so that you have a single data frame with columns for 'book\_id', 'title', 'author\_name', 'publisher', and 'year'?

# Book data frame

books = pd.DataFrame({

'book\_id': [1, 2, 3, 4, 5],

'title': ['The Great Gatsby', 'To Kill a Mockingbird', '1984', 'Pride and Prejudice', 'Animal Farm'],

'author\_id': [1, 2, 3, 4, 3],

'publisher': ['Scribner', 'J. B. Lippincott & Co.', 'Secker & Warburg', 'T. Egerton, Whitehall', 'Secker & Warburg'],

'year': [1925, 1960, 1949, 1813, 1945]

})

# Author data frame

authors = pd.DataFrame({

'author\_id': [1, 2, 3, 4],

'author\_name': ['F. Scott Fitzgerald', 'Harper Lee', 'George Orwell', 'Jane Austen']

})

# Merge data frames

merged = pd.merge(books, authors, on='author\_id')[['book\_id', 'title', 'author\_name', 'publisher', 'year']]