In [ ]:

Q1. Create a dictionary in Python using the following data and perform the following operations:

Prod_Name: OppoA9, Prod_model: 2012, Prod_cost: 25000,
Prod_Year: 2014, Prod_Brand: Oppo
a. Create a dictionary and access the product model and product cost.
b. Find all the keys we use in this dictionary.
c. Change the product cost 25000 Rs to 22000 Rs.
d. Add the items "Prod-weight" in the dictionary.
e. Remove the Prod-Year from the dictionary.

In [1]:
```
# Creating the dictionary
product = {
    "Prod_Name": "OppoA9",
    "Prod_model": 2012,
    "Prod_cost": 25000,
    "Prod_Year": 2014,
    "Prod_Brand": "Oppo"
}

# (a) Accessing the product model and product cost
print("Product Model:", product["Prod_model"])
print("Product Cost:", product["Prod_cost"])

# (b) Finding all keys
print("All keys in the dictionary:", list(product.keys()))

# (c) Changing the product cost
product["Prod_cost"] = 22000
print("Updated Product Cost:", product["Prod_cost"])

# (d) Adding "Prod-weight" to the dictionary
product["Prod-weight"] = "200 grams"
print("Dictionary after adding 'Prod-weight':", product)

# (e) Removing "Prod_Year" from the dictionary
del product["Prod_Year"]
print("Dictionary after removing 'Prod_Year':", product)
```

```
Product Model: 2012
Product Cost: 25000
All keys in the dictionary: ['Prod_Name', 'Prod_model', 'Prod_cost', 'Prod_Y
ear', 'Prod_Brand']
Updated Product Cost: 22000
Dictionary after adding 'Prod-weight': {'Prod_Name': 'OppoA9', 'Prod_model':
2012, 'Prod_cost': 22000, 'Prod_Year': 2014, 'Prod_Brand': 'Oppo', 'Prod-wei
ght': '200 grams'}
Dictionary after removing 'Prod_Year': {'Prod_Name': 'OppoA9', 'Prod_model':
2012, 'Prod_cost': 22000, 'Prod_Brand': 'Oppo', 'Prod-weight': '200 grams'}
```

Q2. Create a list from the given items and perform the following operations:

Football, volleyball, tennis, Basketball, Scouting, Running
a. Create a list and access the first 3 items from the list.
b. Access the element from the [-3] index and return the result.
c. Change the item of index [4] and update the new item as "CDAC".
d. Check and show if the item "tennis" exists in the list or not.
e. Add elements in the list at index [5] with name "Delhi".

In [2]:
```
# Creating the list
items = ["Football", "volleyball", "tennis", "Basketball", "Scouting", "Runr

# (a) Access the first 3 items
print("First 3 items:", items[:3])

# (b) Access the element from the [-3] index
print("Element at [-3] index:", items[-3])

# (c) Change the item at index [4] to 'CDAC'
items[4] = "CDAC"
print("List after updating index [4]:", items)

# (d) Check if 'tennis' exists
print("'tennis' exists in the list:", "tennis" in items)

# (e) Add 'Delhi' at index [5]
items.insert(5, "Delhi")
print("List after adding 'Delhi' at index [5]:", items)
```

```
First 3 items: ['Football', 'volleyball', 'tennis']
Element at [-3] index: Basketball
List after updating index [4]: ['Football', 'volleyball', 'tennis', 'Basketb
all', 'CDAC', 'Running']
'tennis' exists in the list: True
List after adding 'Delhi' at index [5]: ['Football', 'volleyball', 'tennis',
'Basketball', 'CDAC', 'Delhi', 'Running']
```

Q3. Use the data set "Employee.csv" and perform the following operations using pandas library:

a. Select Emp_id, JoiningYear, Age, ExperienceInCurrentDomain, City from the dataset and export the data set in a new file.
b. Find the missing values in 'Age' and fill them by using 'Median' Method.
c. Drop the missing values from the dataset.
d. Find all the employees who belong to 'Bangalore'.

e. From which city maximum employees are working in the company.

In [3]:

```
import pandas as pd

# Load the CSV file into a DataFrame
df = pd.read_csv("Employee.csv")

# (a) Select 'Emp_id', 'JoiningYear', 'Age', 'ExperienceInCurrentDomain', 'C
selected_columns = df[["Emp_id", "JoiningYear", "Age", "ExperienceInCurrentD
selected_columns.to_csv("selected_employee_data.csv", index=False)
print("Selected columns exported to 'selected_employee_data.csv'.")

# (b) Find missing values in 'Age' and fill them using the median method
if df["Age"].isnull().any():
    median_age = df["Age"].median()
    df["Age"] = df["Age"].fillna(median_age)
    print(f"Missing values in 'Age' filled with the median value: {median_ag

# (c) Drop any remaining rows with missing values
df = df.dropna()
print("Data after dropping missing values:")
print(df)

# (d) Find all employees who belong to 'Bangalore'
bangalore_employees = df[df["City"] == "Bangalore"]
print("Employees from Bangalore:")
print(bangalore_employees)

# (e) Find the city with the maximum employees
city_with_max_employees = df["City"].value_counts().idxmax()
print("City with the maximum employees:", city_with_max_employees)
```

```
Selected columns exported to 'selected_employee_data.csv'.
Missing values in 'Age' filled with the median value: 28.0
Data after dropping missing values:
     Emp_id  Education  JoiningYear       City  PaymentTier   Age  Gender  \
0       E01  Bachelors       2017.0  Bangalore          3.0  34.0    Male
1       E02  Bachelors       2013.0       Pune          1.0  28.0  Female
2       E03  Bachelors       2014.0  New Delhi          3.0  38.0  Female
3       E04    Masters       2016.0  Bangalore          3.0  27.0    Male
4       E05    Masters       2017.0       Pune          3.0  24.0    Male
...     ...        ...          ...        ...          ...   ...     ...
4648  E4649  Bachelors       2013.0  Bangalore          3.0  26.0  Female
4649  E4650    Masters       2013.0       Pune          2.0  37.0    Male
4650  E4651    Masters       2018.0  New Delhi          3.0  27.0    Male
4651  E4652  Bachelors       2012.0  Bangalore          3.0  30.0    Male
4652  E4653  Bachelors       2015.0  Bangalore          3.0  33.0    Male

     EverBenched  ExperienceInCurrentDomain  LeaveOrNot
0             No                          0         0.0
1             No                          3         1.0
2             No                          2         0.0
3             No                          5         1.0
4            Yes                          2         1.0
...          ...                        ...         ...
4648          No                          4         0.0
4649          No                          2         1.0
4650          No                          5         1.0
4651         Yes                          2         0.0
4652         Yes                          4         0.0

[4639 rows x 10 columns]
Employees from Bangalore:
     Emp_id  Education  JoiningYear       City  PaymentTier   Age  Gender  \
0       E01  Bachelors       2017.0  Bangalore          3.0  34.0    Male
3       E04    Masters       2016.0  Bangalore          3.0  27.0    Male
5       E06  Bachelors       2016.0  Bangalore          3.0  22.0    Male
7       E08  Bachelors       2016.0  Bangalore          3.0  34.0  Female
10      E11    Masters       2012.0  Bangalore          3.0  27.0    Male
...     ...        ...          ...        ...          ...   ...     ...
4643  E4644  Bachelors       2013.0  Bangalore          3.0  31.0  Female
4646  E4647  Bachelors       2013.0  Bangalore          3.0  25.0  Female
4648  E4649  Bachelors       2013.0  Bangalore          3.0  26.0  Female
4651  E4652  Bachelors       2012.0  Bangalore          3.0  30.0    Male
4652  E4653  Bachelors       2015.0  Bangalore          3.0  33.0    Male

     EverBenched  ExperienceInCurrentDomain  LeaveOrNot
0             No                          0         0.0
3             No                          5         1.0
5             No                          0         0.0
7             No                          2         1.0
10            No                          5         1.0
...          ...                        ...         ...
4643          No                          5         0.0
4646          No                          3         0.0
4648          No                          4         0.0
4651         Yes                          2         0.0
4652         Yes                          4         0.0
```

```
[2220 rows x 10 columns]
City with the maximum employees: Bangalore
```

Q4. Create a SQL data table "Employe" in Python and perform CRUD operations:

a. Use a switch menu to allow users to create, read, update, and delete employees interactively.
b. Connect to a MySQL server with username "root" and password "009483".

In [4]:

```python
import mysql.connector

# Connect to MySQL server
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="009483",
    database="test_db"  # Replace with your database name or create this
)

cursor = conn.cursor()

# Create the table if it doesn't already exist
cursor.execute("""
CREATE TABLE IF NOT EXISTS Employe (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(50),
    Age INT,
    City VARCHAR(50),
    Department VARCHAR(50)
)
""")

# Define a function for each operation
def create_employee():
    name = input("Enter Employee Name: ")
    age = int(input("Enter Employee Age: "))
    city = input("Enter Employee City: ")
    department = input("Enter Employee Department: ")

    query = "INSERT INTO Employe (Name, Age, City, Department) VALUES (%s, %
    values = (name, age, city, department)
    cursor.execute(query, values)
    conn.commit()
    print(f"Employee {name} added successfully.")

def read_employees():
    cursor.execute("SELECT * FROM Employe")
    records = cursor.fetchall()
    print("Employee Records:")
    for record in records:
        print(record)
```

```python
def update_employee():
    emp_id = int(input("Enter Employee ID to Update: "))
    print("What would you like to update?")
    print("1. Name")
    print("2. Age")
    print("3. City")
    print("4. Department")
    choice = int(input("Enter your choice: "))

    if choice == 1:
        new_value = input("Enter New Name: ")
        query = "UPDATE Employe SET Name = %s WHERE ID = %s"
    elif choice == 2:
        new_value = int(input("Enter New Age: "))
        query = "UPDATE Employe SET Age = %s WHERE ID = %s"
    elif choice == 3:
        new_value = input("Enter New City: ")
        query = "UPDATE Employe SET City = %s WHERE ID = %s"
    elif choice == 4:
        new_value = input("Enter New Department: ")
        query = "UPDATE Employe SET Department = %s WHERE ID = %s"
    else:
        print("Invalid choice.")
        return

    cursor.execute(query, (new_value, emp_id))
    conn.commit()
    print(f"Employee ID {emp_id} updated successfully.")

def delete_employee():
    emp_id = int(input("Enter Employee ID to Delete: "))
    query = "DELETE FROM Employe WHERE ID = %s"
    cursor.execute(query, (emp_id,))
    conn.commit()
    print(f"Employee ID {emp_id} deleted successfully.")

# Menu-driven program
while True:
    print("\n--- Employee Management System ---")
    print("1. Create Employee")
    print("2. Read Employees")
    print("3. Update Employee")
    print("4. Delete Employee")
    print("5. Exit")

    choice = int(input("Enter your choice: "))

    if choice == 1:
        create_employee()
    elif choice == 2:
        read_employees()
    elif choice == 3:
        update_employee()
    elif choice == 4:
        delete_employee()
```

```
    elif choice == 5:
        print("Exiting the program.")
        break
    else:
        print("Invalid choice. Please try again.")

# Close the connection
conn.close()
```

```
--- Employee Management System ---
1. Create Employee
2. Read Employees
3. Update Employee
4. Delete Employee
5. Exit
What would you like to update?
1. Name
2. Age
3. City
4. Department
Employee ID 5 updated successfully.

--- Employee Management System ---
1. Create Employee
2. Read Employees
3. Update Employee
4. Delete Employee
5. Exit
Employee Records:
(2, 'Anna', 25, 'Delhi', 'Finance')
(4, 'Anna', 25, 'Delhi', 'Finance')
(5, 'Vansh', 24, 'Delhi', 'RnD')

--- Employee Management System ---
1. Create Employee
2. Read Employees
3. Update Employee
4. Delete Employee
5. Exit
What would you like to update?
1. Name
2. Age
3. City
4. Department
Employee ID 5 updated successfully.

--- Employee Management System ---
1. Create Employee
2. Read Employees
3. Update Employee
4. Delete Employee
5. Exit
```

```
Employee Records:
(2, 'Anna', 25, 'Delhi', 'Finance')
(4, 'Anna', 25, 'Delhi', 'Finance')
(5, 'Vansh', 24, 'New Delhi', 'RnD')

--- Employee Management System ---
1. Create Employee
2. Read Employees
3. Update Employee
4. Delete Employee
5. Exit
Employee Mr Arthor added successfully.

--- Employee Management System ---
1. Create Employee
2. Read Employees
3. Update Employee
4. Delete Employee
5. Exit
Exiting the program.
```

In [ ]:

In [ ]: