```
# Simulating the collection as a dictionary
collections = {}

# Function to create a collection (library)
def createLibrary(p_collection_name):
    collections[p_collection_name] = []  # Initialize an empty list for the collection
    print(f"Created collection '{p_collection_name}'...")

# Function to index data into a collection, excluding a specified column
def indexData(p_collection_name, p_exclude_column):
    # Simulated data (in a real scenario, this would be loaded from a database or file)
    employee_data = [
        {"EmpId": "E02001", "Name": "John", "Department": "HR", "Gender": "Male"},
        {"EmpId": "E02002", "Name": "Alice", "Department": "IT", "Gender": "Female"},
        {"EmpId": "E02003", "Name": "Bob", "Department": "Finance", "Gender": "Male"}
    ]

    for emp in employee_data:
        # Create a shallow copy of the dictionary without the excluded column
        indexed_data = {k: v for k, v in emp.items() if k != p_exclude_column}
        collections[p_collection_name].append(indexed_data)

    print(f"Data indexed into collection '{p_collection_name}', excluding column '{p_exclude_column}'.")

# Function to search within a collection by column and value
def searchByColumn(p_collection_name, p_column_name, p_column_value):
    results = [emp for emp in collections[p_collection_name] if emp.get(p_column_name) == p_column_value]
    print(f"Search results for '{p_column_name}' = '{p_column_value}' in collection '{p_collection_name}':")
    for result in results:
        print(result)

# Function to get the employee count in a collection
def getEmpCount(p_collection_name):
```

```
    count = len(collections[p_collection_name])
    print(f"Employee count in collection '{p_collection_name}': {count}")
    return count

# Function to delete an employee by ID
def delEmpById(p_collection_name, p_employee_id):
    collection = collections[p_collection_name]
    collections[p_collection_name] = [emp for emp in collection if emp.get('EmpId') != p_employee_id]
    print(f"Employee with ID '{p_employee_id}' deleted from collection '{p_collection_name}'.")

# Function to retrieve the count of employees grouped by department
def getDepFacet(p_collection_name):
    dep_facet = {}
    for emp in collections[p_collection_name]:
        department = emp.get('Department', 'Unknown')  # 'Unknown' if Department key is missing
        dep_facet[department] = dep_facet.get(department, 0) + 1

    print(f"Department facet for collection '{p_collection_name}':")
    for department, count in dep_facet.items():
        print(f"{department}: {count}")

# ------------------------------------------------------------
# Example Execution

# Define collection names
v_nameCollection = 'Hash_Anushiya'
v_phoneCollection = 'Hash_9768'

# Create the collections
createLibrary(v_nameCollection)
createLibrary(v_phoneCollection)

# Index data into the collections, excluding specified columns
```

```
indexData(v_nameCollection, 'Department')
indexData(v_phoneCollection, 'Gender')

# Get the employee count for the collections
getEmpCount(v_nameCollection)

# Delete an employee by ID
delEmpById(v_nameCollection, 'E02003')

# Get the updated employee count
getEmpCount(v_nameCollection)

# Search by column values
searchByColumn(v_nameCollection, 'Department', 'IT')
searchByColumn(v_nameCollection, 'Gender', 'Male')
searchByColumn(v_phoneCollection, 'Department', 'IT')

# Get department facets for the collections
getDepFacet(v_nameCollection)
getDepFacet(v_phoneCollection)
```

LINK : [rjroman/Hashagile_task_1 (github.com)]