

ASSIGNMENT 2

Solution:

```
import java.util.*;

class Employee {

    private String employeeId;

    private String department;

    private String gender;

    public Employee(String employeeId, String department, String gender) {

        this.employeeId = employeeId;

        this.department = department;

        this.gender = gender;

    }

    public String getEmployeeId() {

        return employeeId;

    }

    public String getDepartment() {

        return department;

    }

    public String getGender() {

        return gender;

    }

    public void setDepartment(String department) {

        this.department = department;

    }

    public void setGender(String gender) {

        this.gender = gender;

    }

    @Override

    public String toString() {

        return "{EmployeeID=" + employeeId + ", Department=" + department + ", Gender=" +
```

```

gender + "}}";
}
}

public class EmployeeCollectionManager {
    private Map<String, List<Employee>> collections = new HashMap<>();

    public void createCollection(String collectionName) {
        collections.putIfAbsent(collectionName, new ArrayList<>());
        System.out.println("Created collection: " + collectionName);
    }

    public void indexData(String collectionName, String excludeColumn) {
        if (collections.containsKey(collectionName)) {
            for (Employee employee : collections.get(collectionName)) {
                if ("Department".equalsIgnoreCase(excludeColumn)) {
                    employee.setDepartment(null);
                } else if ("Gender".equalsIgnoreCase(excludeColumn)) {
                    employee.setGender(null);
                }
            }
            System.out.println("Indexed data in " + collectionName + ", excluding column: " +
                excludeColumn);
        } else {
            System.out.println("Collection not found.");
        }
    }

    public List<Employee> searchByColumn(String collectionName, String columnName, String
        columnValue) {
        List<Employee> result = new ArrayList<>();
        if (collections.containsKey(collectionName)) {
            for (Employee employee : collections.get(collectionName)) {
                if ("Department".equalsIgnoreCase(columnName) &&
                    columnValue.equals(employee.getDepartment())) {

```

```

result.add(employee);
} else if ("Gender".equalsIgnoreCase(columnName) &&
columnValue.equals(employee.getGender())) {
result.add(employee);
}
}
}

System.out.println("Search results for column " + columnName + " with value " +
columnValue + ": " + result);

return result;
}

public int getEmpCount(String collectionName) {
int count = collections.getDefault(collectionName, Collections.emptyList()).size();
System.out.println("Employee count in " + collectionName + ": " + count);
return count;
}

public void delEmpById(String collectionName, String employeeId) {
if (collections.containsKey(collectionName)) {
collections.get(collectionName).removeIf(employee ->
employeeId.equals(employee.getEmployeeId()));
System.out.println("Deleted employee with ID " + employeeId + " from " + collectionName);
}
}

public Map<String, Long> getDepFacet(String collectionName) {
Map<String, Long> departmentCount = new HashMap<>();
if (collections.containsKey(collectionName)) {
for (Employee employee : collections.get(collectionName)) {
if (employee.getDepartment() != null) {
departmentCount.merge(employee.getDepartment(), 1L, Long::sum);
}
}
}
}

```

```

}

System.out.println("Department facet in " + collectionName + ": " + departmentCount);

return departmentCount;

}

public static void main(String[] args) {

EmployeeCollectionManager manager = new EmployeeCollectionManager();

String v_nameCollection = "Vikas";

String v_phoneCollection = "Vikas_8643";

manager.createCollection(v_nameCollection);

manager.createCollection(v_phoneCollection);

Employee emp1 = new Employee("E02001", "Non-IT", "Male");

Employee emp2 = new Employee("E02002", "UI/UX", "Female");

Employee emp3 = new Employee("E02003", "IT", "Male");

manager.collections.get(v_nameCollection).add(emp1);

manager.collections.get(v_nameCollection).add(emp2);

manager.collections.get(v_nameCollection).add(emp3);

manager.getEmpCount(v_nameCollection);

manager.indexData(v_nameCollection, "Department");

manager.indexData(v_phoneCollection, "Gender");


manager.delEmpById(v_nameCollection, "E02003");

manager.getEmpCount(v_nameCollection);

manager.searchByColumn(v_nameCollection, "Department", "IT");

manager.searchByColumn(v_nameCollection, "Gender", "Male");

manager.searchByColumn(v_phoneCollection, "Department", "IT");

manager.getDepFacet(v_nameCollection);

manager.getDepFacet(v_phoneCollection);

}

}

```