MTU
Ollscoil Teicneolaíochta na Mumhan
Munster Technological University

Project 2

Student Name:	Laiba Asif
Student ID:	R00201303
Module:	NoSQL Data Architectures
Date:	12/02/23

Part 1

What opportunities might exist for denormalizing these relations when defining the physical records forthis database? Under what circumstances would you consider creating such denormalized records?

To enhance query efficiency or make data retrieval easier, denormalization involves consolidating or duplicating data from numerous tables into a single table. Denormalization can be helpful in some circumstances, however normalisation is typically preferable for upholding data integrity and minimising redundancy. Here are some instances of denormalization that might be taken into consideration in the given relationships and circumstances:

1. Combining Store and Employee:

We could denormalize the Store and Employee relations into a single table if there is a regular need to obtain data about the store and the employees who work there simultaneously. Columns from both relations, such as storeID, region, managerID, squareFeet, employeeID, whereWork, employeeName, and employeeAddress, could be included in this denormalized table. Certain queries that require combining the Store and Employee tables would become easier as a result of this denormalization.

Circumstance: Denormalizing the tables can decrease the number of joins and potentially increase performance if there are numerous queries or reports that call for collecting both store and employee details at once.

2. Combining Department and Schedule:

we might denormalize the Department and Schedule relations into a single table if we frequently need to access details about the department and the schedules that go along with it. Columns from both relations, such as departmentID, managerID, salesGoal, employeeID, and date, could be included in this denormalized table. Certain queries that entail combining the Department and Schedule tables would be made easier by this denormalization.

Circumstance: Denormalizing the tables can decrease the number of joins and potentially increase performance if there are numerous queries or reports that call for getting both department and schedule values simultaneously.

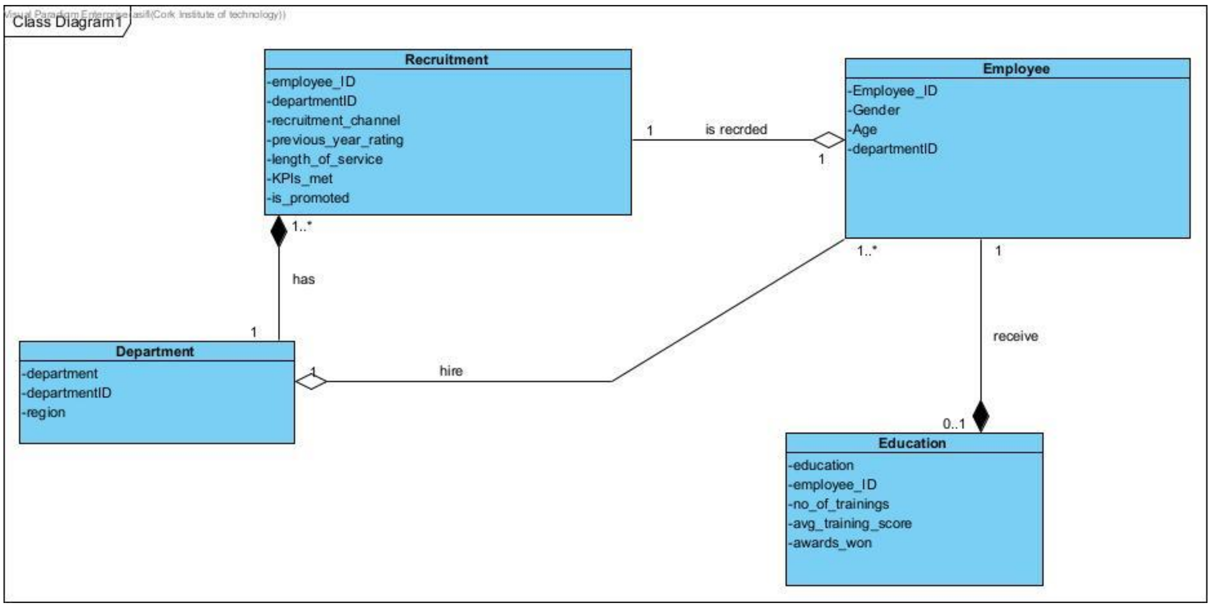
It's crucial to remember that denormalization involves compromises. Increased data redundancy may result, necessitating careful management to maintain data consistency. Denormalized tables could be more difficult to manage and update because several denormalized records may need to be updated when data changes. Denormalization should therefore be carefully addressed in light of the system's unique performance and query optimisation needs.

Part 2

Based on the given ERD, state an aggregate boundary

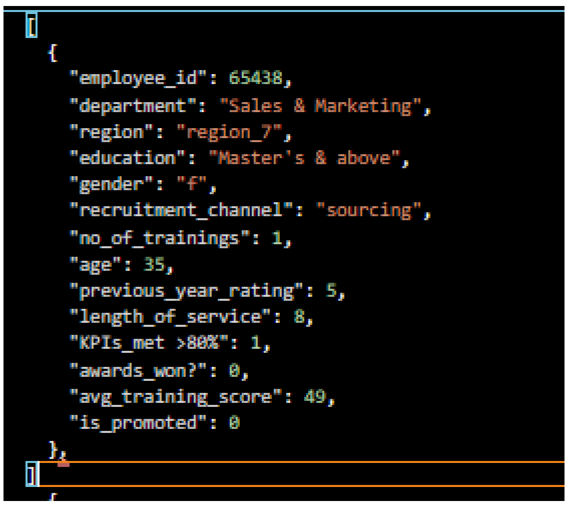
Aggregating based on department: Employee, educational, and recruitment data can all be grouped according to the department they belong to. This would make it possible to examine data and trends at the departmental level.

Draw aggregate data model for the stated aggregate boundary



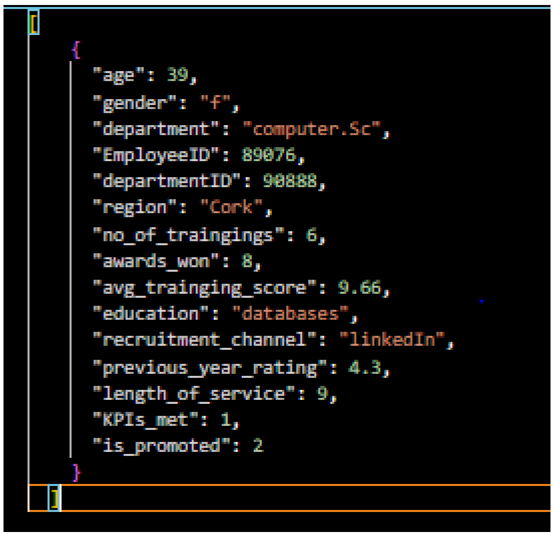
Use the CSV data and the aggregate data model designed in task 2 to create JSON files. Each JSON file only contains one JSON object.

CSV data



```
[
  {
    "employee_id": 65438,
    "department": "Sales & Marketing",
    "region": "region_7",
    "education": "Master's & above",
    "gender": "f",
    "recruitment_channel": "sourcing",
    "no_of_trainings": 1,
    "age": 35,
    "previous_year_rating": 5,
    "length_of_service": 8,
    "KPIs_met >80%": 1,
    "awards_won?": 0,
    "avg_training_score": 49,
    "is_promoted": 0
  },
]
```

Aggregate data model



```
[
    {
      "age": 39,
      "gender": "f",
      "department": "computer.Sc",
      "EmployeeID": 89076,
      "departmentID": 90888,
      "region": "Cork",
      "no_of_traingings": 6,
      "awards_won": 8,
      "avg_trainging_score": 9.66,
      "education": "databases",
      "recruitment_channel": "linkedIn",
      "previous_year_rating": 4.3,
      "length_of_service": 9,
      "KPIs_met": 1,
      "is_promoted": 2
    }
  ]
```