CS 420 – Advanced Programming Languages

Assignment 4 – Logical Programming in Prolog – 100 points

In this comprehensive Prolog assignment, you will be provided with a CSV file containing employee information. The file includes columns such as EEID, Full Name, Job Title, Department, Age, Annual Salary, and more.

Your task is to write a code to translate the given CSV file into Prolog knowledge base (FACTS). You will also be writing some simple programs (RULES) and finally testing your code (Queries).

Submissions:

1. Create a folder called Assignment4 on EDORAS
   a. You will turn in all the code in a single file Assign4.pl on EDORAS.
   b. The folder should also contain the provided EmployeeData.csv file
2. You will turn in two files on to canvas.
   1. Assign4.pl file
   2. A simple report including screenshots of test cases. Also indicate progress and provide references.

Task 0: Setting up EDORAS

1. Log into EDORAS and create a directory called Assign4.
2. Use SCP, FileZilla or use VSCODE to move the provided CSV file to the directory created.

Task 1: Import the contents of given CSV file as FACTS

1. You are provided with a CSV file titled "EmployeeData.csv". Write a simple prolog code to read from the file to create FACTS.
2. The FACTS are stored as predicate with name employee with 14 arguments.

   Employee/14
3. You can modify the helper code provided in CANVAS to achieve this.

Task 2: Writing Rules

Complexity: Piece of cake!

1. Rule to determine if an employee is a working in seattle.

   is_seattle_employee(Name)

2. Rule to determine if an employee is a senior manager in the IT department.

   is_senior_manager_in_IT(Name).

3. Rule to determine if an employee is a Director in the Finance department and works in Miami.

   is_director_finance_miami(Name).

4. Rule to determine if an employee is from United States, works in manufacturing, older than 40, Asian and Male.

   is_asian_US_manufacturing_40M(Name, Bussiness_Unit, Gender, Ethinicity, Age).

5. Write a rule to greet an employee.

   greet (EEID).

(The rule should print out Employee Name, Job Title, Department, and Busineess unit)
Sample query: greet(E02387)
Sample output: Hello, Emily Davis, Sr. Manager of IT, Research & Development!

Complexity: In the Goldilocks zone.

The first fact in your prolog dynamic memory would be the first row from CSV file. The row just has all the column names. You might want to remove that fact before proceeding further. You can do that by simply retracting the one fact.

retract(employee('EEID', _, _, _, _, _, _, _, _, _, _, _, _, _)).

6. Rule to compute years until retirement. Assume retirement age is 65.

$$years\ until\ retirement = 65 - age$$

years_until_retirement(Name,Age,Years_to_retire).

7. Rule to determine Research & Development employees with Black Ethnicity, within the age 25-50.

is_rd_black_midAge (Name, Business_unit, Ethnicity, Age).

8. Rule to determine if an employees is from IT or Finance AND from Pheonix or Miami or Austin.

is_ITorFin_PHXorMIAorAUS(Name, Department,City).

9. Rule to determine female employees in senior role (senior job title) (Senior Title begin with Sr.) **Do not hardcode the titles** **(hint: use build-in predicate atom_concat/3)**

is_female_senior_role(Name, JobTitle).

10. Rule to determine if an employee is a highly paid **senior manager**. Highly paid employees make over **$120000**. **(hint: Convert the salary to a number first.)**

is_highly_paid_senior_manager(Name, Salary).

Complexity: Climbing Mount Everest in flip-flops ☺

11. Rule to determine if an employee's age is a prime number.

is_prime_age(Name, Age)

12. Rule to determine average salary for a specified Job title.

$$Average\ Salary\ (director) = \frac{\sum Annal\ Salary(director)}{\sum directors})$$

average_salary(Job_Title, Salary)

13. Rule to compute total salary of person.

$$Total\ salary\ =\ Annual\ salary\ + (Annual\ Salary * \frac{Bonus\ \%}{100})$$

total_salary(Name,Salary)

14. Rule to determine take-home salary after tax.

$$Take\ home\ salary\ =\ Total\ salary -\ (Total\ Salary * \frac{tax\ \%}{100})$$

Assume tax percentage to be

Salary range	**Tax percentage**
< 50000	20%
50001 – 100000	25%
100000 - 200000	30%
>200000	35%

takehome_salary(Name, Job_Title, Take_home_salary)

15. Rule (rules) to determine years of service.

If the employee has exited

$$Total\ years\ =\ Exit\ date - hire\ date$$

If the employee has not exited

$$Total\ years\ =\ Current\ date - hire\ date$$

total_years(Name,Years)

Task 3: Queries (Testing)

Test each rule with at least three queries and add screenshots to your report.

For example:

Testing Rule 1: Check if employee is from Seattle.

Query 1: is_seattle_employee(‘Emily Davis’)

Expected Result: True.

Query 2: is_seattle_employee(‘Theodore Dinh’).
Expected Result: False.

Query 3: is_seattle_employee(X).
Expected Result: X = Emily Davis;
X = Camila Rogers;
X = Isabella Xi;
….
(Continues to print all employees from Seattle).