The transform function is designed to convert a DetailedPerson record into a DetailedStudent record. This transformation involves mapping the fields of the DetailedPerson to the corresponding fields of the DetailedStudent while handling nested records and arrays.

Transform function is,

function transform(DetailedPerson person) returns DetailedStudent => {};

Input Record (DetailedPerson person):

* id (int): Unique identifier for the person.
* firstName (string): First name of the person.
* lastName (string): Last name of the person.
* age (int): Age of the person.
* visaType (string): Visa type of the person.
* weight (string): Weight of the person.
* salary (string): Salary of the person.
* totalCredits (int): Total academic credits.
* isRegistered (boolean): Registration status.
* personalDetails (PersonalDetails): Nested record containing homeAddress and courseDetails.
* emergencyContacts (DetailedPerson[]): Array of emergency contacts.

Nested Record: personalDetails in person

* + homeAddress (Address): Nested homeAddress record containing city, zip code, and country.

homeAddress

* + 1. city (string): City of residence.
    2. zipCode (string): ZIP code of residence.
    3. country (string): Country of residence.
  + courseDetails (CourseDetails[]) as follows

A screenshot of a course details

Description automatically generated

Output Record (DetailedStudent):

* id (string): Unique identifier for the student.
* name (string): Full name of the student.
* age (int): Age of the student.
* weight (float): Weight of the student.
* courses (Course[]) as follows like courseDetails in personalDetails in person

A screenshot of a course

Description automatically generated

* totalCredits (float): Total academic credits.
* isRegistered (string): Registration status.
* employment (EmploymentDetails): Nested employment details containing job title and salary.
* address (Address): Nested address record containing city, zip code, and country.
* contacts (int): Number of emergency contacts.