Prolog programs are written using a syntax that is similar to natural language. Here are a couple of examples:

1. **Food and Meals**:

   Consider the following facts and rules related to food and meals:

```
Facts

food(burger).                  % Burger is a food.
food(sandwich).                % Sandwich is a food.
food(pizza).                   % Pizza is a food.
lunch(sandwich).               % Sandwich is a lunch.
dinner(pizza).                 % Pizza is a dinner.

Rules

meal(X) :- food(X).            % Every food is a meal.

Queries

?- food(pizza).                % Is pizza a food?
?- meal(X), lunch(X).           % Which food is both a meal and
                             lunch?
?- dinner(sandwich).           % Is sandwich a dinner?
```

   Explanation:

   - The facts define relationships between foods, lunches, and dinners.
   - The rule states that anything that is a food is also a meal.
   - The queries ask questions about food and meals.

   true.   X = sandwich.   false.

2. **Student-Professor Relations**:

   Let’s look at a student-professor relation example:

```
Facts

studies(charlie, csc135).
studies(olivia, csc135).
studies(jack, csc131).
studies(arthur, csc134).
teaches(kirke, csc135).
teaches(collins, csc131).
teaches(collins, csc171).
teaches(juniper, csc134).
```

```
Rules

professor(X, Y) :- teaches(X, C), studies(Y, C).

Queries

?- studies(charlie, What).     % What does Charlie study?
?- professor(kirke, Students). % Who are the students of
Professor Kirke?
```

Explanation:

- The facts establish student-professor relationships.
- The rule defines that X is a professor of Y if X teaches a course C and Y studies the same course.
- The queries inquire about Charlie’s studies and Professor Kirke’s students.

What = csc135

Students = Charlie ;   Students = olivia.