Wellness Profiles using Rules:

Note: With regards to N3 syntax, “_:” represents all instances of the subject. In actual implementation, the following format is used:

“:subject_1 ….”
“:subject_2….”
Etc.

Season:

Defines the Season corresponding to the given time comprised of Month and Day.

POSL Syntax:

season(?StartDay, ?StartMonth, ?Season).

N3 Syntax:

```
_:season
          rdf:type          :Season;
          :day              ?StartDay;
          :month            ?StartMonth;
          :value            ?Season.
```

Usage:

StartDay = The day that represents the season.
StartMonth = The month that represents the season.
Season = The season corresponding to the StartTime.

Forecast:

Defines the weather corresponding to the type and start time.

POSL Syntax:

forecast(?Location, dateTime[?Year, ?Month, ?Day, ?Hour, ?Minute, ?Second),?Type,?Weather], ?LowTemp, ?HighTemp, ?AvgTemp, ?Conditions).

N3 Syntax:

```
_:forecast
          rdf:type          :Forecast;
          :location         ?Location
          :dateTime         [rdf:type :DateTime;
                            :year ?Year;
                             :month ?Month;
                             :day ?Day;
                            :hour ?Hour;
                            :minute ?Minute];
          :lowTemp          ?LowTemp;
```

```
        :highTemp       ?HighTemp;
        :avgTemp        ?AvgTemp;
        :conditions     ?Conditions;
```

Usage:

Location: City for the forecast.
Year = The current year that represents the date of the forecast.
Month = The current month that represents the date of the forecast.
Day = The current day that represents the date of the forecast.
Hour = The time of day that represents the time of the forecast.
LowTemp= The lowest temperature during the forecasted day.
HighTemp= The highest temperature during the forecasted day.
AvgTemp= Average temperature for the forecasted day.
Conditions = The value corresponding to the sky at the given dateTime.

Note: Temperature measured in **Celsius**.

Meetup:

Defines the possible locations for individuals to "meet up" for their activities. Each location is tied to its map and activity type.

POSL Syntax:

```
meetup(?MapID,?Activity, ?Ambience,?Location).
```

N3 Syntax:

```
_:meetup
        rdf:type        :Meetup;
        :mapID          ?MapID;
        :activity       ?Activity;
        :inOut          ?Ambience;
        :location       ?Location.
```

Usage:

MapID = The map that the meetup location is assigned to. In order for profiles to share meetup locations, their MapID's must be the same. Correlate to locations from forecast
Activity = The kind of activity being performed.
Ambience = Whether the activity is being performed inside or outside.
Location = The name of the meetup location.

GoodDuration:

This determines whether or not an activity can be carried out based on the event time differences and the duration, via date time functions.

POSL Syntax:

```
goodDuration(dateTime[?DurYear, ?DurMonth, ?DurDay, ?DurHour, ?DurMinute],
dateTime[?StartYear, ?StartMonth, ?StartDay, ?StartHour, ?StartMinute], dateTime[?EndYear,
?EndMonth, ?EndDay, ?EndHour, ?EndMinute]).
```

N3 Syntax:

```
_:goodDuration

        rdf:type        :GoodDuration;

        :duration       [rdf:type :DateTime;
                        :year ?DurYear;
                        :month ?DurMonth;
                        :day ?DurDay;
                        :hour ?DurHour;
                        :minute ?DurMinute];
        :startDateTime  [rdf:type :DateTime;
                        :year ?StartYear;
                         :month ?StartMonth;
                        :day ?StartDay;
                        :hour ?StartHour;
                        :minute ?StartMinute];
        :endDateTime    [rdf:type :DateTime;
                        :year ?EndYear;
                        :month ?EndMonth;
                        :day ?EndDay;
                        :hour ?EndHour;
                        :minute ?EndMinute].
```

Usage:

DurYear = The number of years in the duration of the event.
DurMonth = The number of months in the duration of the event.
DurDay = The number of days of duration in the event….
DurHour = The number of hours of duration in the event.
DurMinutes = The number of minutes of duration in the event.
StartYear = The year for the starting date of the event.
StartMonth = The month for the starting date of the event.
StartDay = The day for the starting date of the event.
StartHour = The hour for starting time of the event.
StartMinute = The minute for starting time of the event.
EndYear = The year for the ending date of the event.
EndMonth = The month for the ending date of the event.
EndDay = The day for the ending date of the event.

EndHour = The hour for ending time of the event.

EndMinute = The minute for ending time of the event.

Yesterday:

The dateTime of the day before the specified dateTime. A yesterday value must be assigned for each event that will use a rule utilizing the yesterday fact.

POSL Syntax:

yesterday(dateTime[?Year, ?Month, ?Day, ?Hour, ?Minute]).

N3 Syntax:

```
_:yesterday
        rdf:type        :Yesterday;
        :dateTime       [rdf:type :DateTime;
                        :year ?Year;
                         :month ?Month;
                         :day ?Day;
                        :hour ?Hour;
                        :minute ?Minute];
```

Usage:

Year = The year for yesterday's date.

Month = The month for yesterday's date.

Day = The day for yesterday's date.

Hour = The hour for yesterday's time.

Minute = The minute for yesterday's time.

Calendar:

Each individual profile has a calendar which will have activities (separated by types) penciled in at times of their choice.

POSL Syntax:

calendar(?ProfileID,?CalendarID).

N3 Syntax:

```
_:calendar
        rdf:type        :Calendar;
        :profileID      ?ProfileID;
        :calendarID     ?CalendarID.
```

Usage:

ProfileID = The profile identification tied to the calendar.

CalendarID = The calendar identification tied to the profile.

Events:

As a user performs events, or plans to perform events, they will be registered in the knowledge base as **Possible, Planned,** (**Performing**), or **Past** events. Possible events are events that do not have a specific activity assigned to them, but the user has indicated that he/she would like to perform an activity between the times registered. Planned refers to an event that has not happened yet, but has a specific activity assigned to it. Past events are events that have already occurred.

<u>POSL Syntax:</u>

event(?CalendarID,?Aspect,?Tense, dateTime[?StartYear,?StartMonth, ?StartDay, ?StartHour, ?StartMinute], dateTime[?EndYear,?EndMonth, ?EndDay, ?EndHour, ?EndMinute]).

<u>N3 Syntax:</u>

```
_:event

        rdf:type          :Event;
        :calendarID       ?CalendarID;
        :aspect           ?Aspect;
        :tense            ?Tense;
        :startDateTime    [rdf:type :DateTime;
                          :year ?StartYear;
                          :month ?StartMonth;
                           :day ?StartDay;
                           :hour ?StartHour;
                          :minute ?StartMinute];
        :endDateTime      [rdf:type :DateTime;
                          :year ?EndYear;
                          :month ?EndMonth;
                           :day ?EndDay;
                           :hour ?EndHour;
                          :minute ?EndMinute];
```

<u>Usage:</u>

CalendarID = The calendar identification tied to the event.
Aspect = The type of event planned. Typically an activity.
Tense = Possible, Planned, or Past. (As described above).
StartYear = The year for the starting date of the event.
StartMonth = The month for the starting date of the event.
StartDay = The day for the starting date of the event.
StartHour = The hour for the starting time of the event.
StartMinute = The minute for the starting time of the event.
EndYear = The year for the ending date of the event.

EndMonth = The month for the ending date of the event.
EndDay= The day for the ending date of the event.
EndHour = The hour for the ending time of the event.
EndMinute = The minute for the ending time of the event.

Map:

Each individual can have multiple maps to use for their meet up locations (which would be based on where they intended to be for the time they have chosen).

POSL Syntax:

```
map(?ProfileID,?MapID).
```

N3 Syntax:

```
_:map
        rdf:type        :Map;
        :profileID      ?ProfileID;
        :mapID          ?MapID.
```

Usage:
ProfileID = The profile identification assigned to the map.
MapID = The map identification assigned to the profile.

Fitness:

A user will define their expected fitness levels (on a scale from 1-10) for each day or time, so that an appropriate activity and location can be chosen.

POSL Syntax:

```
fitness(?ProfileID, dateTime[?Year, ?Month, ?Day, ?Hour, ?Minute,],?ExpectedFitnessLevel).
```

N3 Syntax:

```
_:fitness

        rdf:type                :Fitness;
        :profileID              ?ProfileID;
        :dateTime               [rdf:type :DateTime;
                                :Year ?DurYear;
                                :Month ?DurMonth;
                                :Day ?DurDay;
                                :Hour ?DurHour;
                                :Minute ?DurMinute];
        :expectedFitness        ?ExpectedFitnessLevel.
```

Usage:
ProfileID = The profile identification corresponding to the fitness level.
Year = The year for the date of when the user has the expected fitness level.
Month = The month for the date of when the user has the expected fitness level.
Day = The day for the date of when the user has the expected fitness level.
Hour = The hour for the time of day when the user has the expected fitness level.
Minute = The minute for the time of day when the user has the expected fitness level.
ExpectedFitnessLevel = The user's expected fitness level for that time of day (on a scale from 1-10).

Level:

Each location will have its own defined fitness level set by the user or calculated previously based on the duration of the activity. It will be unique to the individual, activity, and location. Again, duration is stored as date time but is not implemented, so strings must be matched identically.

POSL Syntax:
level(?ProfileID,?Activity,? Ambience,?Location, dateTime[?Years, ?Months, ?Days, ?Hours, ?Minutes], ?FitnessLevel).

N3 Syntax:

```
_:level

        rdf:type          :Level;
        :profileID        ?ProfileID;
        :activity         ?Activity;
        :inOut            ? Ambience;
        :location         ?Location;
        :duration         [rdf:type :DateTime;
                          :year ?DurYear;
                          :month ?DurMonth;
                           :day ?DurDay;
                          :hour ?DurHour;
                          :minute ?DurMinute];
        :fitnessLevel     ?FitnessLevel.
```

Usage:
ProfileID = The profile identification corresponding to the fitness level.
Activity = The type of activity being performed for the fitness level.
Ambience = Whether the activity is indoors or outdoors.
Location = The meet up location where the activity is being performed.

DurYear = The number of years in the duration of the event.
DurMonth = The number of months in the duration of the event.
DurDay = The number of days of duration in the event.
DurHour = The number of hours of duration in the event.
DurMinutes = The number of minutes of duration in the event.
FitnessLevel = The required fitness level required by the user for the corresponding values..

GroupSize:

Each user may define minimum and maximum extremes for the number of participants required in a group to corresponding to a specific activity. When activities are queried, the requested group min and max values are checked against the user's recorded preferences.

POSL Syntax:
groupSize(?ProfileID,?Activity,?Ambience,?Min,?Max).

N3 Syntax:

```
_:groupSize
        rdf:type        :GroupSize;
        :profileID      ?ProfileID;
        :activity       ?Activity;
        :inOut          ?Ambience;
        :min            ?Min;
        :max            ?Max.
```

Usage:
?ProfileID = The profile identification corresponding to the group size requirements.
?Activity = The activity for which the group size requirements are focused on.
?Ambience = Whether the activity is indoors or outdoors.
?Min = The minimum number of participants.
?Max = The maximum number of participants.

MyActivity:

The objective of the previous wellness rules is to accurately suggest a wellness activity to the user according to their profile preferences. Each activity's requirements can be described by the user, adding or removing as many as they like. Although, each activity must have certain base rules in order for them to follow the wellness profile schema.

POSL Syntax:
activity(?ProfileID,?Activity,?Ambience,?MinRSVP,?MaxRSVP, startDateTime(?StartYear, ?StartMonth, ?StartDay, ?StartHour, ?StartMinute), ?EndDateTime(?EndYear, ?EndMonth, ?EndDay, ?EndHour, ?EndMinute) ,?Location, ?Duration, ?FitnessLevel).

N3 Syntax:

```
_: myActivity

        rdf:type        :MyActivity;
        :profileID      ?ProfileID;
        :activity       ?Activity
        :inOut          ? Ambience;
        :minRSVP        ?MinRSVP;
        :maxRSVP        ?MaxRSVP;
        :startDateTime  [:year ?StartYear;
                        :month ?StartMonth;
                         :day ?StartDay;
                         :hour ?StartHour;
                        :minute ?StartMinute];
        :endDateTime    [:year ?EndYear;
                        :month ?EndMonth;
                         :day ?EndDay;
                         :hour ?EndHour;
                        :minute ?EndMinute];
        :location       ? Location;
        :duration       [:year ?DurYear;
                        :month ?DurMonth;
                         :day ?DurDay;
                        :hour ?DurHour;
                        :minute ?DurMinute];
        :fitnessLevel   ?FitnessLevel.
```

Usage:

ProfileID = The profile identification corresponding to the suggested activity.
Activity = The type of suggested activity determined.
Ambience = Whether the suggested activity is indoors or outdoors.
MinRSVP = Submitted minimum number of participants.
MaxRSVP = Submitted maximum number of participants.
StartYear = The year for the starting date of the event.
StartMonth = The month for the starting date of the event.
StartDay = The day for the starting date of the event.
StartHour = The hour for the starting time of the event.
StartMinute = The minute for the starting time of the event.
EndYear = The year for the ending date of the event.
EndMonth = The month for the ending date of the event.
EndDay= The day for the ending date of the event.

EndHour = The hour for the ending time of the event.
EndMinute = The minute for the ending time of the event.
Location = The meet up location for the suggested activity.
Duration = The duration of the suggested activity.
FitnessLevel = The required fitness level of the suggested activity.

Required Rules to Fit Schema:

```
calendar(p0001,?Calendar),
event(?Calendar,Run:Activity,possible,?StartTime,?EndTime),
participation(1:Integer,6:Integer,?MinRSVP,?MaxRSVP),
map(p0001,?Map),
meetup(?Map,run,out,?Place),
level(p0001,run,out,?Place,?Duration,?Level),
greaterThanOrEqual(?ExpectedFitness,?Level),
goodDuration(?Duration,?StartTime,?EndTime).
```

* Many other rules can be implemented such as seasonal requirements, whether requirements, temperature requirements etc… *

Currently Used Activity Types:

Walking
Running
Swimming
Skating
Yoga
Hiking
Baseball

Current Weather Conditions:

Sunny
Partly Sunny
Mostly Sunny
Clear
Cloudy
Partly Cloudy
Mostly Cloudy
Mist
Overcast
Dust
Fog
Smoke
Haze

Rain
Chance of Rain
Showers
Light Rain
Scattered Showers
Freezing Drizzle
Thunderstorm
Chance of TStorm
Scattered Thunderstorms
Snow
Sleet
Light Snow
Rain and Snow
Chance of Snow
Storm
Chance of Storm
Icy