The ontology has Classes as below:

* Amenity has subclasses BusinessAmenity, Convenience, DiningAmenity, RecreationalAmenity, Relaxation
* Hotel has subclasses BudgetHotel, LuxuryHotel and ResortHotel
* Room has subclasses DoubleRoom, SingleRoom and FamilyRoom
* Quality
* Customer
* Staff has subclasses Manager and Receptionist
* Price
* Booking has subclasses OnlineBooking and WalkInBooking
* Event
* Review
* Location has subclasses Country and City

The ontology has Object Properties as below:

* Room belongsTo Hotel
* Room bookedBy Customer
* Hotel isLocatedAt Location
* Hotel hasAmenity Amenity
* Room hasBooking Booking
* Room hasPrice Price
* Hotel hasQuality Quality
* Hotel hasRoom Room
* Hotel hasStaff Staff
* Hotel hosts Event
* Review isAbout Hotel
* Amenity isAvailableAt Hotel
* Booking isFor Room
* Event isHostedAt Hotel
* Hotel foundIn City
* Price isPriceOf Room
* Review isWrittenBy Customer
* Booking madeBy Customer
* Customer makes Booking
* Manager manages Hotel
* Receptionist serves Customer
* Staff worksAt Hotel

The ontology has Data Properties as below:

* bookingDate for Booking
* amenityType for Amenity
* contactInfo for Staff and Customer
* customerName for Customer
* duration and status for Booking
* eventDate, eventType for Event
* hotelName for Hotel
* location(has subproperty country and city) for Hotel
* price for Room
* priceRange for Hotel
* reviewDate, reviewText and rating for Review
* roomNo for Room
* staffID, staffName and staffRole for Staff
* List all Bookings made by Customer Abraham
* List all Luxury Hotels located in Addis Ababa
* Find all hotel with Swimming pool Amenity and parking Amenity
* List all Rooms which are Booked.
* List all FamilyRooms which are not Booked.
* Show a Room which is Booked by Abraham
* Show a bookings which have confirmed status
* List all hotels located at Adama
* Amenity class has subclasses Business, Convenience, Dining, Recreational, Relaxation
* Hotel class has subclasses BudgetHotel, LuxuryHotel and ResortHotel
* hasAmenity object property has domain Hotel and Range Amenity
* amenityType Data Property has domain Amenity and range xsd:string
* ResortHotel class which is subclass of Hotel class has instance Hotel\_1
* Hotel\_1 has “amenityType” Data Property with value “Swimming\_Pool”, “hotelName” Data Property with value “HaileResort” and “city” Data Property with value “Hawasa”

Based on the above informations give me a query that Finds all hotel with Swimming\_Pool Amenity

**Finds a hotel with Swimming\_Pool Amenity**

**PREFIX ex: <http://www.semanticweb.org/aga1/ontologies/2025/0/HotelOntology#>**

**PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>**

**SELECT ?hotelName**

**WHERE {**

**ex:Hotel\_1 a ex:ResortHotel .**

**ex:Hotel\_1 ex:hasAmenity ?amenity .**

**?amenity ex:amenityName "Swimming\_Pool" .**

**ex:Hotel\_1 ex:hotelName ?hotelName .**

**}**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Find a location of "HaileResort" hotel**

PREFIX ex: <http://www.semanticweb.org/aga1/ontologies/2025/0/HotelOntology#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT?country ?city

WHERE {

?Hotel\_1 a ex:ResortHotel .

?Hotel\_1 ex:hotelName "HaileResort" .

?Hotel\_1 ex:city ?city .

?Hotel\_1 ex:country ?country .

}

**PREFIX hotel: <http://www.semanticweb.org/aga1/ontologies/2025/0/HotelOntology#>**

**SELECT ?amenityName**

**WHERE {**

**?amenity a hotel:Relaxation;**

**hotel:amenityName ?amenityName.**

**}**

**PREFIX ex: <http://example.org/ontology#>**

**PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>**

**SELECT ?name ?age**

**WHERE {**

**?person a ex:Person ; # Find any individual that is a Person**

**ex:hasName ?name ; # Get the person's name**

**ex:hasAge ?age . # Get the person's age**

**}**

**Explanation:**

**1. PREFIX: We define the ex: prefix to use short-hand notation.**

**2. SELECT ?name ?age: We want to retrieve the values of the variables ?name and ?age.**

**3. WHERE { ... }: We specify the patterns that must match:**

**• ?person a ex:Person: Find individuals (?person) that are instances of the ex:Person class.**

**• ?person ex:hasName ?name: Get the name (?name) associated with each person, using ex:hasName property.**

**• ?person ex:hasAge ?age: Get the age (?age) associated with each person, using ex:hasAge property.**