Note: All OCL Expressions are bound to a UML Model. We choose to bind the OCL expressions to the domain model diagram.

1. "Offerings are unique. In other words, multiple offerings on the same day and time slot must be offered at a different location."
    a. **Explanation**: every offering in OfferingsDB has a unique combination of startTime, endTime, city and location. We treat location and city differently as we can consider the same location name in different cities (ex. there can be two "Hall buildings" in two different cities). Also, startTime and endTime are DateTime objects, so include the date as well as the time.

```
context OfferingsDB
        inv: self.offering → forAll(o1, o2: Offering | o1 <> o2 implies
                                    (o1.startTime <> o2.startTime or
                                     o1.endTime <> o2.endTime or
                                     o1.city <> o2.city or
                                     o1.location <> o2.location))
```

2. "Any client who is underage must necessarily be accompanied by an adult who acts as their guardian."
    a. **Explanation**: We have treated Guardians + Minors as separate from Clients. This means, a Guardian is an Actor, and they "guard" Minors. Minors are created upon creating a new Guardian, and only Guardians can make Bookings for their Minors. Making sure Clients and Guardians are above the age of 18 assures that no one below the age of 18 can make a booking. Also, making sure every Minor has a non-null Guardian will ensure that each Minor is "accompanied by an adult who acts as their guardian". Keep in mind that GuardiansDB is the composite of the Guardian class, meaning all Guardians are necessarily a part of GuardiansDB. Thus, if a Minor has a non-null Guardian, then that Guardian necessarily exists in GuardiansDB.
    b. This design leads to a more complex implementation, but a simple OCL expression.

```
context Client
        inv: self.age >= 18

context Guardian
        inv: self.age >= 18

context Minor
        inv: self.guardian <> null
```

3. “The city associated with an offering must be one the city’s that the instructor has indicated in their availabilities.”
   a. **Explanation**: If the Offering has an instructor, there exists a city in the Instructor’s cities which is the Offering’s city. An Instructor’s cities attribute is an array of strings.

      ```
      context Offering
              inv: self.instructor <> null implies (self.instructor.cities →
                                                    includes(self.city))
      ```

   b. Note: We did not find many examples of how to deal with a collection of primitives in OCL. This is our best guess in how to convey the requirement.

4. “A client does not have multiple bookings on the same day and time slot.”
   a. Explanation: For all Bookings of a client, the combination of startTime and endTime of each associated Offering is unique. The same applies to Minors.

      ```
      context Client
              inv: self.bookings → forAll(b1, b2 | b1 <> b2 implies
                                   b1.offering.startTime <> b2.offering.startTime or
                                   b1.offering.endTime <> b2.offering.endTime)

      context Minor
              inv: self.minorBookings → forAll(b1, b2 | b1 <> b2 implies
                                   b1.offering.startTime <> b2.offering.startTime or
                                   b1.offering.endTime <> b2.offering.endTime)
      ```