Example Use Case

Alice is a 16-year-old (public) high school student who works at Bob's Grocery. She lives with her mom. Alice went to the movies with Catherine and Catherine's friend Heather on Saturday, worked Sunday, developed symptoms Monday after going to Summer School, and was diagnosed with COVID-19 Tuesday.

```
NOTES:
     <PREFIXES> =   PREFIX : <undefined>...
                    PREFIX path: <http://www.semanticweb.org/ContactTracingOntology#>
                    PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
                    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
                    PREFIX owl: <http://www.w3.org/2002/07/owl#>
                    PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

          Public Health Server: 192.168.100.4
          All External Servers (Job/Movies/School): 192.168.100.5

          Content added during rewrite
          Primary location where return result is set
          #Comments (can understand the query by looking for and reading the comments)
```

Query 1: Household Members - Before Rewriting

```
<PREFIXES>
SELECT DISTINCT ?property ?Info
WHERE{
  #Selects all Persons who live at the samw Place as Person1a (Alice)
   path:Person1a path:livesAt ?Place.
   ?Person path:livesAt ?Place.

  #Eliminates Person1a from Results
   FILTER(?Person != path:Person1a)

  #Selects all Personal Information Properties OR Datatype
  #properties that contain personal information
   {?property rdfs:subPropertyOf* path:personalInformation.}
   UNION {?property rdf:type owl:DatatypeProperty.}
```

```
#Selects the most specific version
#of each property within the system
#that still has some sort of value
FILTER NOT EXISTS{
       ?sub rdfs:subPropertyOf ?property.
             FILTER(?sub != ?property)
             ?Person ?sub ?currRange.
          }

#Returns all of the selected properties
#that store information about the
#?Person as well as what info is stored
?Person ?property ?Info.
}
```

Query 1: Household Members - After Rewriting

```
<PREFIXES>
SELECT DISTINCT ?property ?Info
WHERE{
  path:Person1a path:livesAt ?Place.
  ?Person path:livesAt ?Place.

  FILTER(?Person != path:Person1a)

  #Specifies the User must be a Contact Tracer
  path:UserC path:hasRole path:Contact_Tracer
  #Specifies who the User must be assigned to
  {path:UserC path:assigned ?Person.}
  UNION {path:UserC path:assigned path:Person1a.}

  {?property rdfs:subPropertyOf* path:personalInformation.
    #Specifies the range of values that the Personal Information
    #properties may take based on the disease status of ?Person
     {?range rdfs:subClassOf* path:Open_Information.}
     UNION {path:UserC path:assigned ?Person.
     ?Person path:isCaseFor ?disease.
     ?range rdfs:subClassOf* path:Protected_Information.}
      ?property rdfs:range ?range.}
  UNION {?property rdf:type owl:DatatypeProperty.}

FILTER NOT EXISTS{
      ?sub rdfs:subPropertyOf ?property.
            FILTER(?sub != ?property)
            ?Person ?sub ?currRange.
         }

?Person ?property ?Info.
}
```

Query 1: Household Members - Sample Results

	property	Info
1	path:primaryPhoneNumber	path:123-123-1234
2	path:dateOfContact	"2021-07-07T00:00:00"^^xsd:dateTime
3	path:dateOfClear	"2021-07-29T00:00:00"^^xsd:dateTime
4	path:hasFirstName	"Samantha"
5	path:hasLastName	"Smith"
6	path:canQuarantine	"false"^^xsd:boolean

Query 2: Job Query - Before Rewrite

```
<PREFIXES>
SELECT DISTINCT ?firstName ?lastName ?contactInformation
?localJob ?description
WHERE{
  #Collects enough information to identify the PUI
  path:Person1a path:hasFirstName ?fname;
                path:hasLastName  ?lname.
  ?contact rdfs:subPropertyOf* path:hasContactInformation.
  path:Person1a ?contact ?contactInfo.

  #Collects enough information to identify Place of work
  path:Person1a path:hasJob ?job.
  ?job path:hasEmployer ?employer.
  {?employer rdf:type path:Place.
   ?employer path:hasAddress ?identifier}
    UNION{?employer rdf:type path:Organization.
    ?employer path:description ?identifier.}

  #Queries from the Grocery Store
  SERVICE <http://192.168.100.5:3030/ds>{
    #Makes sure Person1a and PUI are the same Person
    ?PUI path:hasFirstName ?fname;
         path:hasLastName ?lname;
         ?contact ?contactInfo.

    #Ensures that Users are allowed access to the
    #Visit we want to look at with our PUI
    ?someone path:allowAccess ?Visit.
    ?Visit path:visitedBy ?PUI, ?Person.

    #Make sure the place of work matches (so
    #we querying the right place)
    ?PUI path:hasJob ?PUIJob.
    ?PUIJob path:hasEmployer ?localEmployer.
   {?localEmployer rdf:type path:Place.
   ?localEmployer path:hasAddress ?identifier}
   UNION{?localEmployer rdf:type path:Organization.
   ?localEmployer path:description ?identifier.}

    #Returns info about employees Alice exposed
    OPTIONAL{
      ?Person path:hasFirstName ?firstName;
              path:hasLastName ?lastName;
              ?contact ?contactInformation;}
    OPTIONAL{?Person path:hasJob ?localJob.}
    OPTIONAL{?Person path:description ?description.
}}}
```

Query 2: Job Query - After Rewrite

```
<PREFIXES>
SELECT DISTINCT ?firstName ?lastName ?contactInformation
?localJob ?description
WHERE{
  path:Person1a path:hasFirstName ?fname;
                path:hasLastName  ?lname.
  ?contact rdfs:subPropertyOf* path:hasContactInformation.
  path:Person1a ?contact ?contactInfo.

  path:Person1a path:hasJob ?job.
  ?job path:hasEmployer ?employer.
  {?employer rdf:type path:Place.
   ?employer path:hasAddress ?identifier}
    UNION{?employer rdf:type path:Organization.
    ?employer path:description ?identifier.}

 #Ensures Users  are authorized to see this information through
 #their role & assignment
  {path:UserC path:hasRole path:Contact_Tracer;
              path:assigned path:Person1a.
   path:Person1a path:isCaseFor ?disease.}
  UNION {path:UserC path:hasRole path:Outbreak_Investigator;
                path:assigned ?employer.}

  SERVICE <http://192.168.100.5:3030/ds>{
   ?PUI path:hasFirstName ?fname;
        path:hasLastName ?lname;
        ?contact ?contactInfo.

   ?someone path:allowAccess ?Visit.
   ?Visit path:visitedBy ?PUI, ?Person.

   ?PUI path:hasJob ?PUIJob.
   ?PUIJob path:hasEmployer ?localEmployer.
  {?localEmployer rdf:type path:Place.
  ?localEmployer path:hasAddress ?identifier}
  UNION{?localEmployer rdf:type path:Organization.
  ?localEmployer path:description ?identifier.}

    OPTIONAL{
      ?Person path:hasFirstName ?firstName;
              path:hasLastName ?lastName;
              ?contact ?contactInformation;}
    OPTIONAL{?Person path:hasJob ?localJob.
    #Only allows access to job info if the visit is jobRelated
    ?Visit path:isJobRelated "true"^^xsd:boolean.}
    OPTIONAL{?Person path:description ?description.
  }
}
```

Query 2: Job Query - Sample Results

	firstName	lastName	contactInformation	localJob	description
1	"Ryan"	"Ryanson"	path:123-777-7878	path:Bob's_Cashier	"Out sick 07/01-07/07 2021"
2	"Quinn"	"Quinson"	path:123-246-2468	path:Bob's_Cashier	
3	"Martha"	"Marks"	path:123-333-3333	path:Bob's_Manager	
4	"Leighton"	"Lee"	path:123-444-5555	path:Bob's_Stocker	
5	"Alice"	"Smith"	path:123-456-7890	path:Bob's_Cashier	

Query 2: Job Query - Sample Results

	firstName	lastName	contactInformation	localJob	description
1	"Ryan"	"Ryanson"	path:123-777-7878	path:Bob's_Cashier	"Out sick 07/01-07/07 2021"
2	"Quinn"	"Quinson"	path:123-246-2468	path:Bob's_Cashier	
3	"Martha"	"Marks"	path:123-333-3333	path:Bob's_Manager	
4	"Leighton"	"Lee"	path:123-444-5555	path:Bob's_Stocker	
5	"Alice"	"Smith"	path:123-456-7890	path:Bob's_Cashier	

Query 3: School Query - Overview

- Original Query: UserC requests access the name/contact information of all individuals exposed to Alice in her class.
    - Constraints: Information solicited from public schools is regulated by FERPA, so the User needs explicit permission to access each record
    - Constraints: Only Contact Tracers/Outbreak Investigators can solicit external sources for visit information, and only if assigned to Alice/the Visit or the Visit's Location respectively
    - **Note**: The class is represented as a regular visit in the ontology, since regular visits correspond to recurring meetings with the same group of people
- Rewritten Query: UserC, a Contact Tracer assigned to Alice or an Outbreak Investigator assigned to her Visit/school, can access names and contact information for students in Alice's class with explicit canAccess permission from the school

Query 3: School Query - Before Rewrite

```
<PREFIXES>
SELECT DISTINCT ?firstName ?lastName ?contactInformation
?description ?data
WHERE{
  #Get  enough information to identify Alice
  path:Person1a path:hasFirstName ?fname;
                path:hasLastName  ?lname.
  ?contact rdfs:subPropertyOf* path:hasContactInformation.
  path:Person1a ?contact ?contactInfo.

  #Get the address of the visited place
  ?Visit path:visitedBy path:Person1a;
         path:visitedPlace ?Place.

  #Get all addresses associated with that place (if you visit a class,
  #the room may not have an address but the school will)
  {?Place path:hasAddress ?address.}
  UNION{
  {?Place path:belongsTo ?entity.}
  UNION {?Place path:partOf ?entity.}
  ?entity path:hasAddress ?address.}

  #Queries data from the school directly
  SERVICE <http://192.168.100.5:3030/ds>{
```

```
  #Identify the PUI as Alice
  ?PUI path:hasFirstName ?fname;
       path:hasLastName ?lname;
       ?contact ?contactInfo.

  #Collects all the people at the visit with Alice and where they were
  ?Visit2 path:visitedBy ?PUI, ?Person;
          path:visitedPlace ?Place2.

  #Ensures that the visits correspond to the same place
  {?Place2 path:hasAddress ?address}
   UNION {
  {?Place2 path:belongsTo ?entity2.}
   UNION {?Place2 path:partOf ?entity2.}
   ?entity2 path:hasAddress ?address.}

  #Gets requested information about the
  #visit/students
  ?Person path:hasFirstName ?firstName;
          path:hasLastName ?lastName.
  OPTIONAL{?Person ?contact ?contactInformation.}
  OPTIONAL{?Person path:description ?description.}
  OPTIONAL{?Visit2 path:visitData ?data.
           ?data path:dataType path:Attendance.})}
```

Query 3 - School Query - After Rewrite

```
<PREFIXES>
SELECT DISTINCT ?firstName ?lastName ?contactInformation
?description ?data
WHERE{
  path:Person1a path:hasFirstName ?fname;
                path:hasLastName  ?lname.
  ?contact rdfs:subPropertyOf* path:hasContactInformation.
  path:Person1a ?contact ?contactInfo.

  ?Visit path:visitedBy path:Person1a;
         path:visitedPlace ?Place.

  {?Place path:hasAddress ?address.}
  UNION{
  {?Place path:belongsTo ?entity.}
  UNION {?Place path:partOf ?entity.}
  ?entity path:hasAddress ?address.}

  #Confirms User's role & assignment to make sure they are allowed the info
  {path:UserC path:hasRole path:Contact_Tracer
  {path:UserC path:assigned path:Person1a.
  path:Person1a path:isCaseFor ?disease.}
  UNION{path:UserC path:assigned ?Visit.}}
  UNION {path:UserC path:hasRole path:Outbreak_Investigator.
  {path:UserC path:assigned ?Visit.}
  UNION {{path:UserC path:assigned ?Place.}
  UNION{path:UserC path:assigned ?entity.}}}
```

```
  #Gets UserID to compare access in the  the school's local server
  path:UserC path:userID ?ID.

SERVICE <http://192.168.100.5:3030/ds>{

  #Gets UserC's  local identifier at the school & checks they can explicitly
  #access Visit2
  ?UserC path:userID ?ID.
  ?UserC path:canAccess ?Visit2.
  ?PUI path:hasFirstName ?fname;
       path:hasLastName ?lname;
       ?contact ?contactInfo.

  ?Visit2 path:visitedBy ?PUI, ?Person;
          path:visitedPlace ?Place2.

  {?Place2 path:hasAddress ?address}
   UNION {
  {?Place2 path:belongsTo ?entity2.}
   UNION {?Place2 path:partOf ?entity2.}
  ?entity2 path:hasAddress ?address.}

  ?Person path:hasFirstName ?firstName;
          path:hasLastName ?lastName.
  OPTIONAL{?Person ?contact ?contactInformation.}
  OPTIONAL{?Person path:description ?description.}
  OPTIONAL{?Visit2 path:visitData ?data.
           ?data path:dataType path:Attendance.
           #Checks UserC can explicitly access the Roster
           ?UserC path:canAccess ?data.}}}
```

Query 3: School Query - Sample Results

	firstName	lastName	contactInformation	description	data
1	"David"	"Dooberman"	path:123-456-7893		path:Mr._Green's_Attendance_Record
2	"Carmen"	"Carrera"	path:123-456-7892		path:Mr._Green's_Attendance_Record
3	"Robert"	"Brown"	path:123-456-7891		path:Mr._Green's_Attendance_Record
4	"Alice"	"Smith"	path:123-456-7890		path:Mr._Green's_Attendance_Record

Query 4: Movie Theater Query - Before Rewrite

```
<PREFIXES>
SELECT DISTINCT ?Person ?property ?Info
WHERE{
  #Isolates the visits made by the infected Person1a (Alice)
  #Then gets the identifying information of that Visit
  ?Visit path:visitedBy path:Person1a;
  path:visitedPlace ?Place1;
  path:visitStartTime ?time;
  path:description ?description.

  #Gets the address of the Place or the place containing place, in order to make sure
  #the right visit is queried at the movie theater
  {?Place1 path:hasAddress ?address.}
  UNION{{?Place1 path:partOf ?z.} UNION {?z path:partOf ?Place1.}
  ?z path:hasAddress ?address.}

  #Isolates the appropriate properties containing personal information
  {?property rdfs:subPropertyOf* path:personalInformation.}
  UNION{?property rdf:type owl:DatatypeProperty.}

  #Filters out repetitive properties with a more specific subproperty
  FILTER NOT EXISTS{
      ?sub rdfs:subPropertyOf ?property.
           FILTER(?sub != ?property)
      ?currDomain ?sub ?currRange.
  }
```

```
#Queries the movie theater data
SERVICE <http://192.168.100.5:3030/ds>{
  #Gets identifying visit information,
  #isolating the correct visit with the
  #place address and the start time
  ?Visit2 path:visitStartTime ?time;
  path:description ?description;
  path:visitedPlace ?Place;
  path:visitedBy ?Person. #Gets persons at the visi

  #Makes sure we are referring to the same place by compa
  #associated addresses
  {?Place2 path:hasAddress ?address}
  UNION{{?Place2 path:partOf ?b.}
  UNION {?b path:partOf ?Place2.}
  ?b path:hasAddress ?address.}

   #Gets the personal information
  ?Person ?property ?Info.}}
```

Query 4: Movie Theater Query - After Rewrite

```
<PREFIXES>
SELECT DISTINCT ?Person ?property ?Info
WHERE{
  ?Visit path:visitedBy path:Person1a;
  path:visitedPlace ?Place1;
  path:visitStartTime ?time;
  path:description ?description.

  {?Place1 path:hasAddress ?address.}
  UNION{{?Place1 path:partOf ?z.} UNION {?z path:partOf ?Place1.}
  ?z path:hasAddress ?address.}

 #Checks the User has appropriate Role & is assigned appropriate entities
 {path:UserC path:hasRole path:Contact_Tracer.
    {path:UserC path:assigned path:Person1a.}
    UNION {path:UserC path:assigned ?Visit.}}
  UNION{path:UserC path:hasRole path:Outbreak_Investigator.
    {path:UserC path:assigned ?Visit.}
    UNION{{path:UserC path:assigned ?Place1.}
      UNION {path:UserC path:assigned ?z}}}

   #Restricts properties returned to those that reveal Open Information only
   #Not protected or closed information
  {?property rdfs:subPropertyOf* path:personalInformation.
   ?range rdfs:subClassOf* path:Open_Information.
   ?property rdfs:range ?range.}
  UNION{?property rdf:type owl:DatatypeProperty.}

FILTER NOT EXISTS{
    ?sub rdfs:subPropertyOf ?property.
         FILTER(?sub != ?property)
    ?currDomain ?sub ?currRange.
}

SERVICE <http://192.168.100.5:3030/ds>{
  ?Visit2 path:visitStartTime ?time;
  path:description ?description;
  path:visitedPlace ?Place;
  path:visitedBy ?Person.

  #Ensures that this Visit is available to solicit
  #information from
  ?someone path:allowAccess ?Visit2

  {?Place2 path:hasAddress ?address}
  UNION{{?Place2 path:partOf ?b.}
  UNION {?b path:partOf ?Place2.}
  ?b path:hasAddress ?address.}

  ?Person ?property ?Info.}}
```

Query 4: Movie Theater Query - Sample Results

	Person	property	Info
1	path:000015	path:hasEmail	path:currerBell@domain.org
2	path:000015	path:hasLastName	"Bronte"
3	path:000015	path:hasFirstName	"Charlotte"
4	path:000015	path:description	"reserved 3 tickets online"
5	path:000013	path:hasEmail	path:lsmith@domain.org
6	path:000013	path:hasFirstName	"Lewis"
7	path:000013	path:hasLastName	"Smith"
8	path:000013	path:description	"reserved 1 ticket online"
9	path:000010	path:hasEmail	path:vfd@domain.org
10	path:000010	path:hasFirstName	"Bauldelaire"
11	path:000010	path:hasFirstName	"Violet"
12	path:000010	path:description	"reserved 3 tickets"
13	path:000008	path:hasEmail	path:aliceanderson@domain.org
14	path:000008	path:hasFirstName	"Alice"

Query 4: Movie Theater Query - Sample Results (cont)

15	path:000008	path:hasLastName	"Anderson"
16	path:000008	path:description	"reserved 1 ticket online"
17	path:000003	path:hasEmail	path:hwright@domain.org
18	path:000003	path:description	"Bought 1 ticket online"
19	path:000003	path:hasFirstName	"Heather"
20	path:000003	path:hasLastName	"Wright"
21	path:000002	path:hasEmail	path:ccole@domain.org
22	path:000002	path:description	"Bought 1 ticket online"
23	path:000002	path:hasFirstName	"Catherine"
24	path:000002	path:hasLastName	"Cole"
25	path:000001	path:hasEmail	path:alicesmith@domain.org
26	path:000001	path:hasFirstName	"Alice"
27	path:000001	path:description	"Bought 1 ticket online"
28	path:000001	path:hasLastName	"Smith"

Query 5: Backtalk Query - Before Rewrite

```
<PREFIXES>
SELECT DISTINCT ?Place2 ?property ?info ?disease
WHERE{
  #Gets the place the user represents and an address to compare to
  path:User1 path:represents ?Place.
  {?Place path:hasAddress ?address.}
  UNION {
    #Gets the places  associated with where the user represents
    {?Place path:partOf ?tempPlace1.}
    UNION{?tempPlace1 path:belongsTo ?Place.}
    UNION{?tempPlace1 path:partOf ?Place.}
    ?tempPlace1 path:hasAddress ?address.}

  #Ports to Public Health database
  SERVICE <http://192.168.100.4:3030/ds>{
    #Compares addresses to ensure these arethe same place
    {?Place2 path:hasAddress ?address.}
    UNION {
      {?Place2 path:partOf ?tempPlace2.}
      UNION {?tempPlace2 path:partOf ?Place2.}
      UNION {?tempPlace1 path:belongsTo ?Place2.}
      ?tempPlace2 path:hasAddress ?address.}
```

```
  #Checks to see if a diseased person has visited any of the Places of interes
  ?Visit path:visitedPlace ?Place2;
                 path:visitedBy ?Person.
  ?Person path:isCaseFor ?disease.

  #Gets place properties to with information about the places the user
  #represent (useful for identifying places without addresses, like theaters)
  {?property rdfs:subPropertyOf* path:placeProperty.}
  UNION{?property rdf:type owl:DatatypeProperty.
  ?property rdfs:domain/(owl:unionOf/rdf:rest*/rdf:first)* path:Place.}

  #Eliminates repetitive properties for which a more specific property exists
  FILTER NOT EXISTS{?sub rdfs:subPropertyOf ?property.
  FILTER(?sub != ?property)
    {?Place2 ?sub ?currRange.}
     UNION {?currDomain ?sub ?Place2.}}

   #Returns the place information
   {?Place2 ?property ?info.}
   UNION {?info ?property ?Place2.}}}
```

Query 5: Backtalk Query - After Rewrite

```
<PREFIXES>
SELECT DISTINCT ?Place2 ?property ?info ?disease
WHERE{
  path:User1 path:represents ?Place.

  {?Place path:hasAddress ?address.}
  UNION {
    {?Place path:partOf ?tempPlace1.}
    UNION{?tempPlace1 path:belongsTo ?Place.}
    UNION{?tempPlace1 path:partOf ?Place.}
    ?tempPlace1 path:hasAddress ?address.}

  #Confirms that User1 has the appropriate role to access the data
  path:User1 path:hasRole path:Business_Representative.

  SERVICE <http://192.168.100.4:3030/ds>{
    {?Place2 path:hasAddress ?address.}
    UNION {
      {?Place2 path:partOf ?tempPlace2.}
      UNION {?tempPlace2 path:partOf ?Place2.}
      UNION {?tempPlace1 path:belongsTo ?Place2.}
      ?tempPlace2 path:hasAddress ?address.}
    ?Visit path:visitedPlace ?Place2;
           path:visitedBy ?Person.
    ?Person path:isCaseFor ?disease.

    #Checks that an external user is allowed to view the visit's information
    ?someone path:allowAccess ?Visit.

    {?property rdfs:subPropertyOf* path:placeProperty.
    #Excludes the personrelated property livesAt from the results
    FILTER (?property != path:livesAt)}
    UNION{?property rdf:type owl:DatatypeProperty.
    ?property rdfs:domain/(owl:unionOf/rdf:rest*/rdf:first)* path:Place.}
    FILTER NOT EXISTS{
      ?sub rdfs:subPropertyOf ?property.
           FILTER(?sub != ?property)
       {?Place2 ?sub ?currRange.}
        UNION {?currDomain ?sub ?Place2.}}

    {?Place2 ?property ?info.}
     UNION {?info ?property ?Place2.}}}
```

Query 5: Backtalk Query - Sample Results

	Place2	property	info	disease
1	path:Place4a	path:partOf	path:Place3a	path:COVID-19
2	path:Place4a	path:description	"Theatre 2"	path:COVID-19
3	path:Place4a	path:description	"Theatre 2 in AMC Theatre, Screened Black Widow at 8pm 7/3/21"	path:COVID-19