# CSE 414 Section 6

November 2, 2017

Consider a graph of colored vertices and undirected edges where the vertices can be red, green, blue. In particular, you have the relations

```
Vertex(x, color)
Edge(x, y)
```

The Edge relation is symmetric in the if (x, y) is in Edge, then (y, x) is in Edge. Your goal is to write a datalog program to answer each of the following questions.

1. Find all green vertices.

   ```
   GreenV(x) :- Vertex(x, 'green')
   ```

2. Find all pairs of blue vertices connected by one edge.

   ```
   BluePairs(x) :- Vertex(x, 'blue'), Vertex(y, 'blue'), Edge(x, y)
   ```

3. Find all triangles where all the vertices are the same color. Output the three vertices and their shared color.

   ```
   Triangle(x, y, z, a) :- Vertex(x, a), Vertex(y, a), Vertex(z, a),
                           Edge(x, y), Edge(y, z), Edge(z, x)
   ```

4. Find all vertices that don't have any neighbors.

   WRONG ANSWER (UNSAFE)

   ```
   LonleyV(x) :- not Edge(x,_)
   ```

   WRONG ANSWER (UNSAFE)

   ```
   LonleyV(x) :- Vertex(x,_), not Edge(x,_)
   ```

   RIGHT ANSWER (SAFE)

   ```
   OnlyX(x) :- Edge(x, _)
   LonleyV(x) :- Vertex(x,_), not OnlyX(x)
   ```

5. Find all vertices such that they only have red neighbors.

   ```
   BlueV(x) :- Vertex(x,_), Edge(x, y), Vertex(y, 'blue')
   GreenV(x) :- Vertex(x,_), Edge(x, y), Vertex(y, 'green')
   RedV(x) :- Vertex(x,_), not BlueV(x), not GreenV(x)
   ```

6. Find all vertices such that they only have neighbors with the same color. Return the vertex and color.

   ```
   SameColor(x, y, a) :- Vertex(x, a), Vertex(y, a)
   NotSameNeigh(x) :- Vertex(x,_), Edge(x, y), Edge(x, z), not SameColor(y, z)
   OnlySameNeigh(x, a) :- Vertex(x, a), not NotSameNeigh(x)
   ```

   OR

   ```
   Neigh(x, y, a) :- Edge(x, y), Vertex(y, a)
   DifferentNeigh(x) :- Neigh(x, y, a), Neigh(x, z, b), a != b
   OnlySameNeigh(x, a) :- Vertex(x, a), not DifferentNeigh(x)
   ```

7. (Optional) For some vertex $v$, find all vertexes connected to $v$ by blue vertexes (this one require recursion).

   ```
   ConnectedTo(x) :- Vertex(x,'blue'), Edge(x, v)
   ConnectedTo(x) :- Vertex(x,'blue'), Edge(x, y), ConnectedTo(y)
   ```