Name: ____

1. (1 pt.)
   - **Read all material carefully.**
   - You may refer to your books, papers, and notes during this test.
   - No computer or network access of any kind is allowed (or needed).
   - Write, and draw, carefully. Ambiguous or cryptic answers receive zero credit.
   - Use the conventions used in class and the textbook for all material.

   Write your name in the space provided above.

2. (19 pts.) Consider a relation $R(A, B, C, D, E, F, G)$ with the following basis of dependencies:

$$\begin{aligned} AB &\rightarrow C \\ B &\rightarrow D \\ DC &\rightarrow A \\ CEF &\rightarrow AB \\ FG &\rightarrow C \end{aligned}$$

   (a) Provide a smallest (cardinality) instance of R that *violates* the dependency $CEF \rightarrow AB$ without violating any of the other dependencies. Briefly explain why your answer is correct (including why no smaller instance suffices).

(b) Which of the following dependencies are *logically implied* by those in the above basis? *Justify* your answer briefly.

i. $AF \rightarrow B$.

ii. $BCF \rightarrow A$.

iii. $EFG \rightarrow C$.

3. (20 pts.) For the relation R of Question 2 (dependencies repeated here):

$$\begin{aligned} AB &\rightarrow C \\ B &\rightarrow D \\ DC &\rightarrow A \\ CEF &\rightarrow AB \\ FG &\rightarrow C \end{aligned}$$

(a) List **all** keys of R.

(b) Explain your answer, noting why the keys you list are valid and also why there are no other keys.

(c) How many *superkeys* does R have? Explain your answer. (You need not list all superkeys.)

4. (20 pts.) Decompose the schema of Question 2 to BCNF. Show all intermediate steps and details, as in class exercises (keys, projected dependencies, decomposed relations, etc.).

$$
\begin{aligned}
AB &\rightarrow C \\
B &\rightarrow D \\
DC &\rightarrow A \\
CEF &\rightarrow AB \\
FG &\rightarrow C
\end{aligned}
$$

[additional space for answering the earlier question]

5. (10 pts.) Consider an XML database suggested by the following excerpt of a file `ferndb.xml`:

```
<FernDB>
 <Fern>
   <CommonName lang="en">Ostrich Fern</CommonName>
   <BinomialName>
     <Genus>Matteuccia</Genus>
     <Species>struthiopteris</Species>
   </BinomialName>
   <HeightLow units="ft">2</HeightLow>
   <HeightUp units="ft">5</HeightUp>
   <Habitats>
     <Habitat id="woods"/>
   </Habitats>
   <FruitDate>
     <Month lang="en">June</Month>
     <Day>5</Day>
   </FruitDate>
 </Fern>
 <Habitat id="woods">
   Woodland areas.
 </Habitat>
 <Observation>
   <Date format="ISO">2012-06-01</Date>
   <Location>near shed</Location>
   <Fern>Ostrich Fern</Fern>
 </Observation>
</FernDB>
```

Express the following in *XPath*:

(a) A list of all habitat IDs (only).

(b) A list of all observation elements that have at least two fern subelements.

Briefly explain why your answers are correct.

6. (10 pts.) For the XML database of Question 5, express the following in *XQuery*:

   (a) The binomial names of all ferns f such that there is at least one observation that has both f and `Ostrich Fern` as subelements.

   (b) The binomial names of all ferns f such that there every observation that has `Ostrich Fern` as a subelement also has f as a subelement.

   Briefly explain why your answers are correct.

7. $\star$ (20 pts.) Consider a relation `Edge(s,d,c)` that encodes an *undirected* graph with colored edges: a tuple $(s, d, c) \in$ `Edge` represents an undirected edge (s, d) of color c, where s and d are positive integers that identify vertices of the graph, and c is a string representing a color (e.g., "blue").

   (a) Express the following constraints in SQL, using the simplest SQL constraint features that suffice:

      i. For all s, d and c, if $(s, d, c) \in$ `Edge` then there is no c' such that $(s, d, c') \in$ `Edge`.

      ii. For all s and d, if there exists c such that $(s, d, c) \in$ `Edge` then there is no c' such that $(d, s, c') \in$ `Edge`.

   (b) Assuming `Edge` satisfies the above constraints, provide a Datalog query for all vertices that reachable from vertex `7` by following a path composed of edges of colors that alternate between blue and green (starting with either). (Note that each edge between two vertices may be represented in `Edge` in two different ways and your query should work regardless of which way is used for each edge.) Briefly explain why your answer is correct.

[additional space for answering the earlier question]