**CSI 300**

**Database Management Systems**

**Lucien Aibel**

You've been hired by the Queen of Dragons to build her a database to keep track of her subjects. It gets hard when they're flying around all the time.

* Dragons have only one name, but often multiple titles after it; for instance, hers are "Sapphire the Magnificent, the Wisest, the Most Forward-Thinking."
* By ancient law, dragons always hold exactly one territory, and do not share territories.
* She explains that dragons are very proud, and no dragon is willing to have any entry that doesn't mention both their territory and at least three titles (this will break 1NF).

Sketch a table to satisfy her.

Explain how 1NF could benefit the database.

She seems moved by your arguments, particularly upon realizing she can finally take a fourth title, "the Excellently Data-Based". And anyway, the other titles will be right there. Normalize your database to 1NF in UML.

Identify an update, deletion, and insertion anomaly possible if you stay in 1NF.

She's still worried about separating dragons from their sources of pride. Split off only one table to achieve 2NF.

Why would 3NF be an improvement on 2NF?

Finally getting into the idea of data relationships, she points out that many dragons have titles based on their children's titles ("Cerulean, father of Nagaman the Destroyer"). Figure out how to track a dragon's children (who are themselves dragons), then finish normalizing to 3NF.

What would be the query to get the first title of the first child of dragon #3?