The database uses a supertype and subtype design for Users, Animals, and Preferences. The User supertype has a common ID, and the subtypes Registered and Guest add extra details when needed. Each subtype has a one-to-one relationship with the supertype, so every user belongs to exactly one subtype and the data stays consistent. This is better than having one table with empty columns for each subtype, which could cause a lot of unused space and weaker rules.

The Animal supertype stores general traits like size, temperament, energy, and care needs. The subtypes Dog, Cat, and Bird store more specific details like breed, coat or wing type, trainability, and talking ability. Each subtype is connected to the supertype with a one-to-one relationship, and deleting an animal also deletes its subtype record. We thought about putting everything in one table, but it would have a lot of empty fields and make the data less reliable.

Preferences also use a supertype/subtype approach. The main Preference table stores common info, while EssentialPreference and PreferredPreference only store the ID and link back to Preference. This way we know what type of preference it is without needing an extra column.

For many-to-many relationships, we use separate tables. Favorite links Registered users to animals, Fulfill links animals to the preferences they meet, and Sets links users to their preferences. Each of these tables uses a combined key to avoid duplicates and keeps the data consistent. Deleting a user, animal, or preference automatically removes related records to prevent orphaned data.

Overall, this design keeps things clear and easy to manage. Using separate tables for subtypes avoids empty columns, enums and booleans make sure the values are valid, and mapping tables handle the many-to-many connections in a simple way.