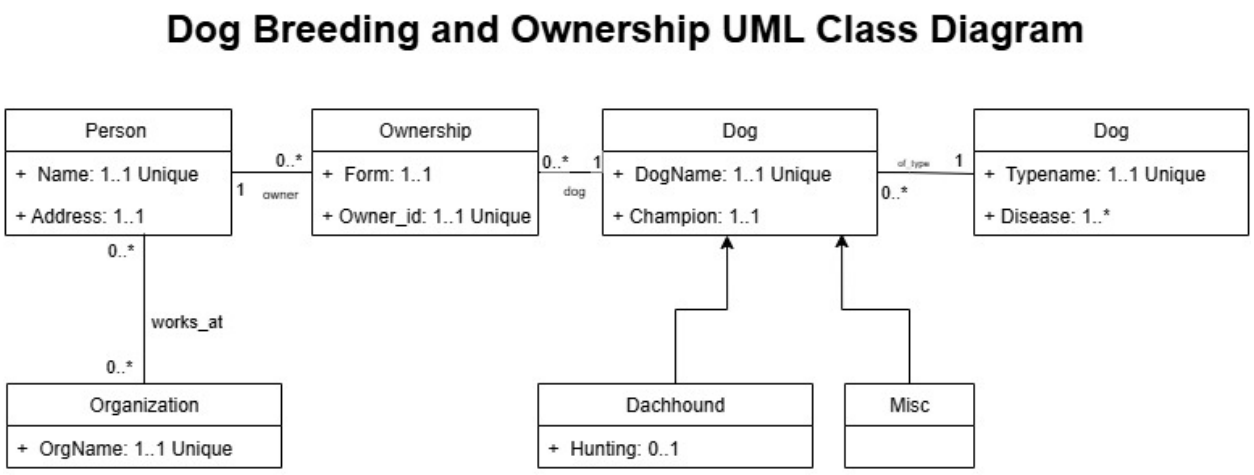
Relational Database Schema Design for Dog Ownership Management

Introduction

This project focuses on designing a dog ownership and management system by first developing a UML class diagram to capture the conceptual model and then creating a corresponding relational database schema. The aim is to demonstrate how conceptual requirements can be translated into a well-structured database design that ensures data integrity, consistency, and usability. Lucid chart and draw.io are the tools used for modelling the UML class diagram and database schema.

Dog Breeding and Ownership UML Class Diagram

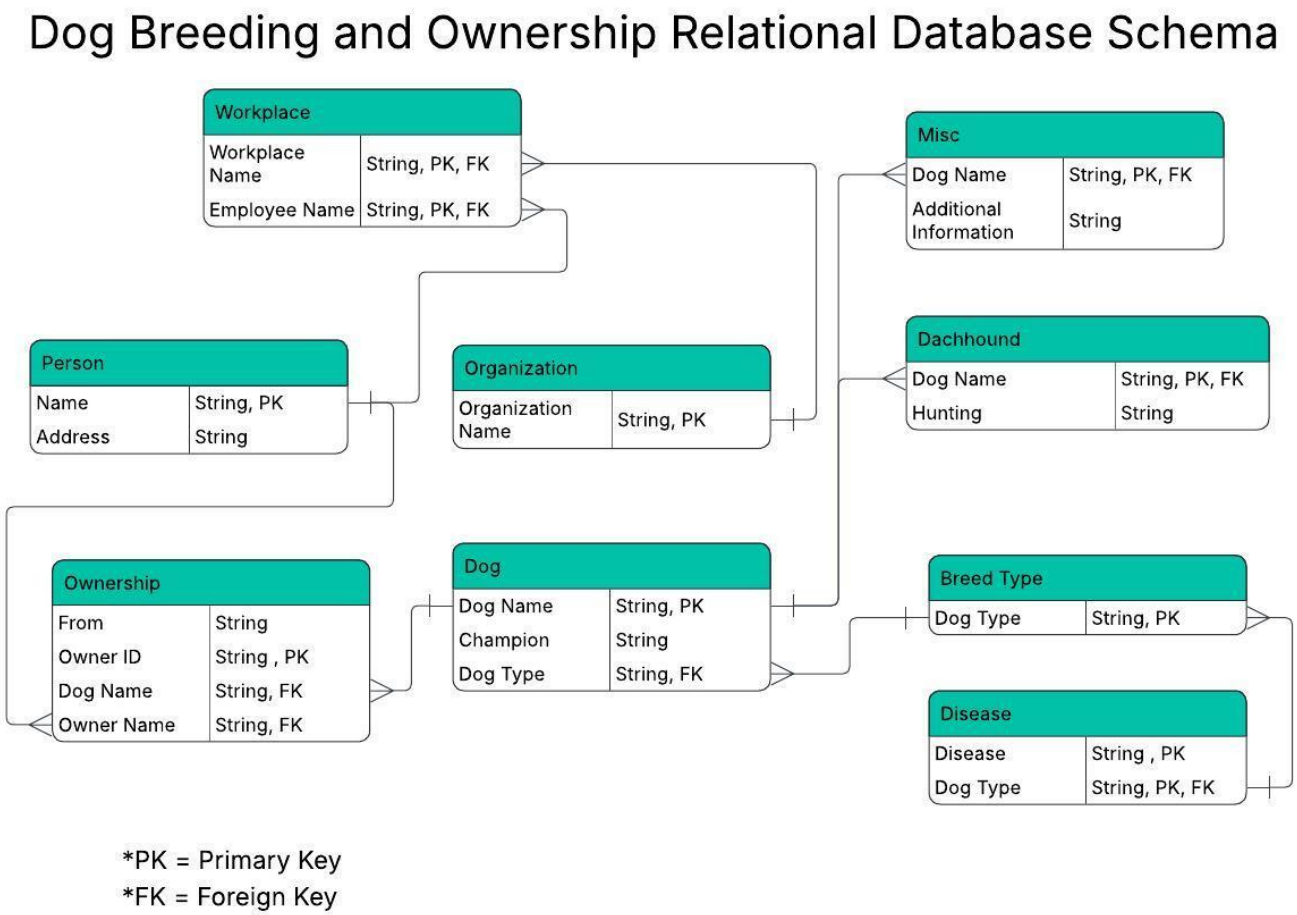
The UML class diagram presents the conceptual design of the dog ownership and management system. It defines the main entities, **Person**, **Organization**, **Ownership**, and **Dog**, together with their attributes and relationships. The diagram also shows specialization in the **Dog** entity, where subclasses such as **Dachhound** and **Misc** extend its properties. Associations are clearly represented, for example, the connection between **Person** and **Organization** through employment, and between **Person**, **Ownership**, and **Dog** to capture ownership details. This diagram provides an abstract, technology-independent view of the system.



Dog Breeding and Ownership Relational Database Schema

The database schema diagram translates the conceptual UML model into a relational database structure that can be implemented in a database system. Each entity is represented as a table with defined primary keys, foreign keys, and attributes. For example, the **Dog** table includes references

to **Breed Type** and supports subclass-specific tables like **Dachhound** and **Misc**. The schema ensures referential integrity by establishing foreign key constraints, such as linking **Ownership** to both **Person** and **Dog**, and associating **Disease** with **Breed Type**. This schema thus forms the logical foundation for storing, managing, and querying the data described in the conceptual model.



Textual Description of Dog Breeding and Ownership Relational Database Schema

The textual representation below provides a structured description of the database schema, listing each table, its attributes, and the relationships or constraints (such as primary keys, foreign keys, and alternate keys) that enforce the integrity of the model.

- Person (Name, Address)
- Organization (Organization Name)
- Workplace (Workplace Name, Employee Name) Workplace.Employee Name is foreign key towards Person.Name . Workplace.Workplace Name is foreign key towards Organization.OrgName .

- Ownership (From, Owner Id, Dog Name, Owner Name), Ownership.Dog Name is foreign key towards Dog.Dog Name. Ownership.Owner Name is foreign key towards Person.Name. Dog Name and Owner Name are alternate keys.
- Dog (Dog Name, Champion, Dog Type), Dog.Dog Type is foreign key towards Breed.Dog Type. Dog Type is alternate key.
- Dachhound (Dog Name, Hunting), Dachhound.Dog Name is foreign key towards Dog.Dog Name.
- Misc (Dog Name, additional information), Misc.Dog Name is foreign key towards Dog.Dog Name.
- Breed (Dog Type)
- Disease (Dog Type, Disease), Disease.Dog Type is foreign key towards Breed.Dog Type.

Conclusion

This project demonstrates the complete process of transforming a conceptual model into a relational database schema for a dog ownership and management system. As a business analyst, the next steps would involve extending this foundation by gathering additional stakeholder requirements, defining business rules in greater detail, and mapping system processes through use cases or workflow diagrams. Further analysis could include creating sample queries and reports to illustrate how the database supports decision-making, as well as developing wireframes or dashboards to show potential user interfaces. These steps would bridge the gap between technical design and business value, ensuring that the system not only reflects the domain accurately but also meets organizational needs effectively.