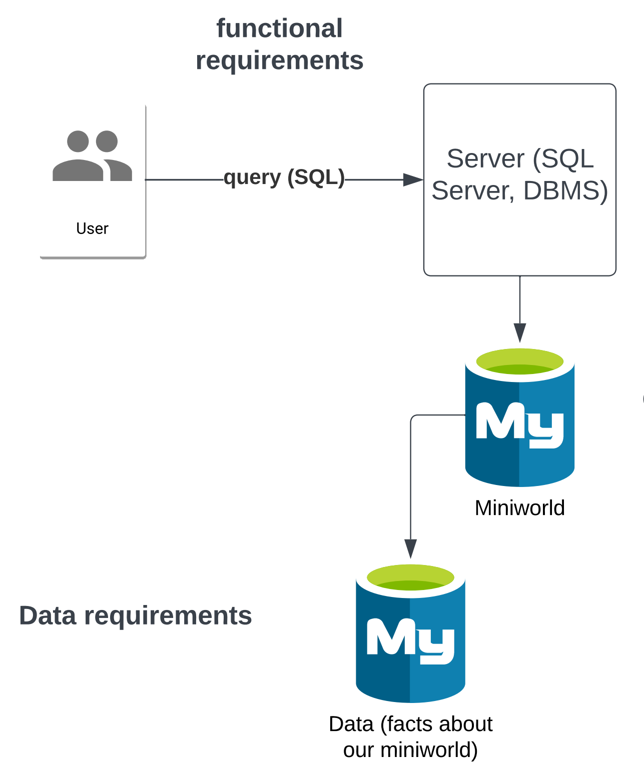


Compiler (guide dog)

MyFile.java

javac MyFile.java
java MyFile

MyFile.sql (Stored Procedure)

Query compiler

Query optimizer

Runtime processor

DBMS (how do we classify them?)

- What data model are we using? (ER Model) - relational models.
  - Other alternatives (Object data model, NOSQL).
- How many users?
  - Single user systems
  - Multi user systems
- Centralized - stored on the same machine.
- Distrubuted - across multiple machines

Homogeneous - same DBMS software
Heterogeneous - different DBMS software

Open sourced (Free)
Proprietary (Expensive)

Data Modeling (ER Model)

Database Design

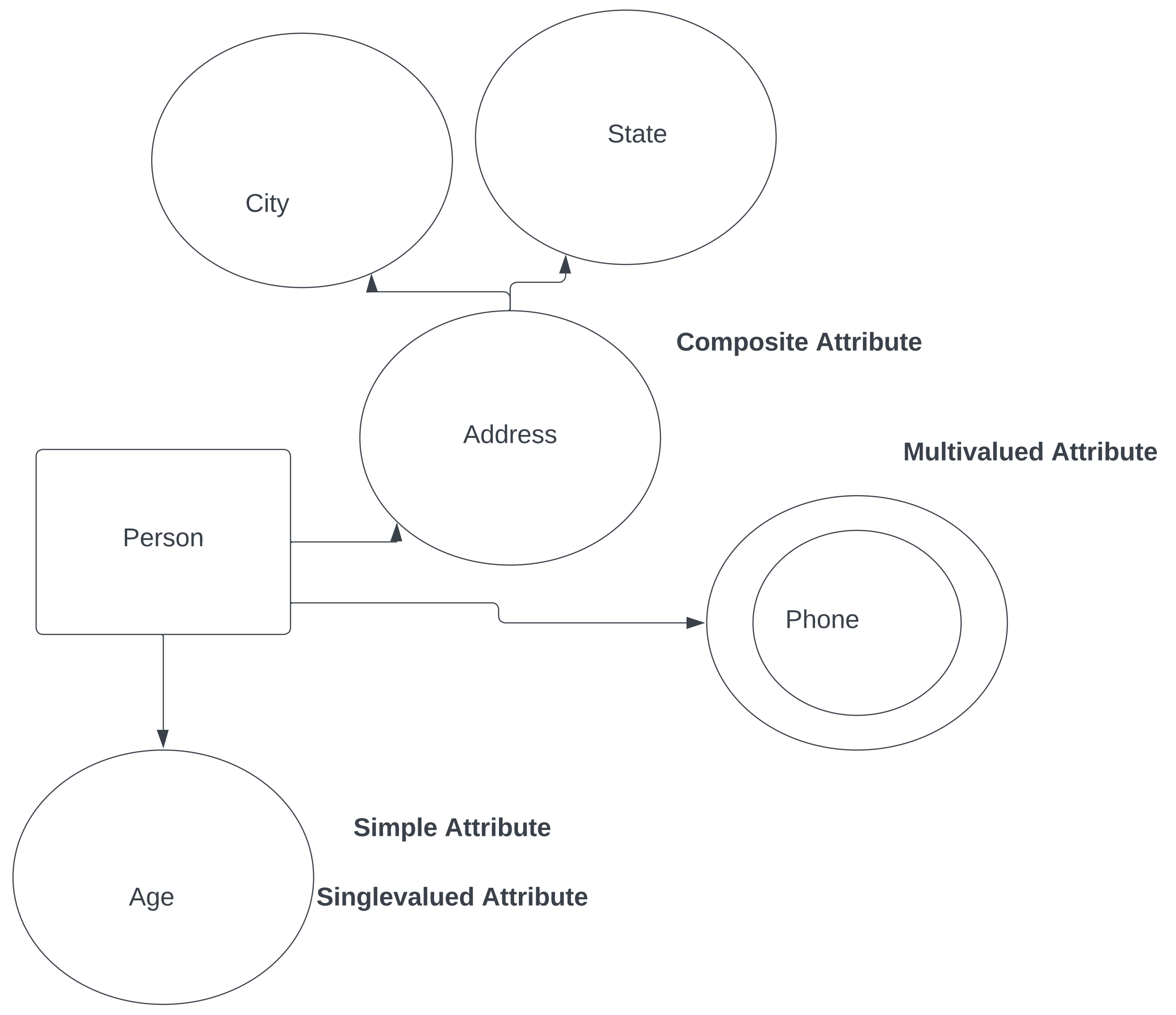
- Requirements gathering and specification
  - Data requirements - concice representation of what we need to store in our miniworld database (Entities, **Attributes**, Relationships)
  - Functional requirements - things the users need to be able to do interacting with our database (Database Applications - any piece of software that allows our users to interface with our database)
- Conceptual design - representation of our logical data requirements and our functional requirements in a way that doesn't require knowledge of the internal schema (generally to orient non technical users). i.e. ER Diagram.
- Logical design (data model mapping) - Implementation schema (Relational Schema). Represents the SQL you would need to write to represent your database.
- Physical design - internal structure, access paths, i.e. indexes.

Entities - object in the real world with independent existence.

- Strong - has a unique identifier)
- Weak - we'll talk about this next time*

- Physical existence - building, car, tree.
- Conceptual existence - class, company, job.
- Attributes - each entity has these, describe the entities.
  - simple vs. composite -
    - simple - attribute that cannot be broken down into parts (represent them with a circle, connected by a single line to the entity).
    - composite - can be broken down further into component parts (we're concerned about accessing component parts directly). Composite attribute is the concatenation of it's parts.
  - single valued vs. multivalued -
    - I would only have a single value for the attribute at one given time (ever)
    - Multiple values at the same time for a given attribute (i.e. Color of your car, Addresses, Phones, etc..)
  - stored vs. derived -
  - NULL values -

Relationships


City
State
Composite Attribute
Address
Multivalued Attribute
Person
Phone
Age
Simple Attribute
Singlevalued Attribute