

A Practical Introduction to Value-Entity-Association (VEA) Modeling

Using Java and PostgreSQL with the `chenup` framework

Overview

This paper provides a rapid introduction to the Value-Entity-Association (VEA) model as well as implementation in Java and PostgreSQL, in particular through the `chenup` Java framework.

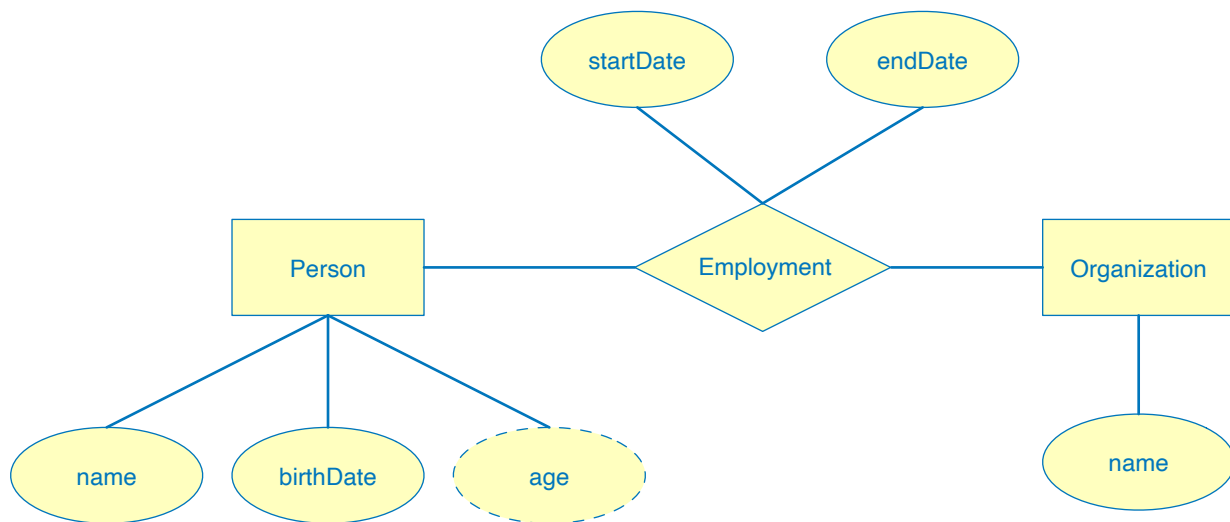
A word here about definitions.

- Values have no identity and are immutable. I avoid the term ‘value object’ (Evans 2003) since it seems to exist mainly to preserve the ‘everything is an object’ catchphrase, whereas if we define an object as something with identity, state, and behavior, ‘value object’ is an oxymoron.
- An Entity is ‘[s]omething that has real existence; a thing or being, as opposed to an attribute, quality, relation, etc.’ (Oxford English 2025)
- I use the term ‘association’ throughout this paper, rather than relationship, except when directly referring to Chen’s Entity-Relationship model, to avoid confusing ‘relationship’ with ‘relation’.¹

Tutorial introduction

We begin with a simple ER diagram in Chen notation.

¹ For a recent but hardly unique example of this confusion, see <https://cs50.harvard.edu/extension/sql/2024/fall/notes/1/> which incorrectly states ‘These tables have some relationships between them, and hence we call the database a **relational database**.’



For those unfamiliar with Chen's ER notation, rectangles represent entities, diamonds represent relationships, and ovals represent attributes. Dashed ovals represent computed attributes, that is, attributes that can be derived from other properties.

Values are represented in Java as record types, and in Postgres

An example is a `PersonName`, which we will for simplicity suppose is composed of

Value types are defined in Java using records:

Entities are represented as Java interfaces, with the `@Entity` annotation.

Computed values

Implementation within the chenup framework

This paper is primarily focused on VEA modeling and so this will be an incomplete introduction to chenup and provides only the minimum information needed to create the above model in Java and PostgreSQL and execute it within a main.

Chenup is available from maven central. Version 0.1 is preparing for release, but currently a snapshot version is available by using the

```
<repositories>
  <repository>
    <id>central-portal-snapshots</id>
    <url>https://central.sonatype.com/repository/maven-
snapshots/</url>
    <releases>
      <enabled>false</enabled>
    </releases>
    <snapshots>
      <enabled>true</enabled>
    </snapshots>
  </repository>
</repositories>
```

And then the dependency

```
<dependencies>
  <dependency>
    <groupId>com.ahimsasystems</groupId>
    <artifactId>chenup</artifactId>
    <version>0.1.0-SNAPSHOT</version>
  </dependency>
</dependencies>
```

Implementation choices and limitations

Background

The E-R model was introduced by Chen (1976).

Smalltalk popularized the ‘everything is an object’ viewpoint.

References

- Chen, P. P.-S. (1976). "The entity-relationship model—toward a unified view of data." ACM Trans. Database Syst. **1**(1): 9–36.
- Evans, E. (2003). Domain-Driven Design: Tackling Complexity In the Heart of Software, Addison-Wesley Longman Publishing Co., Inc. Evans.
- Oxford English, D. (2025). entity, n., sense 2, Oxford University Press.