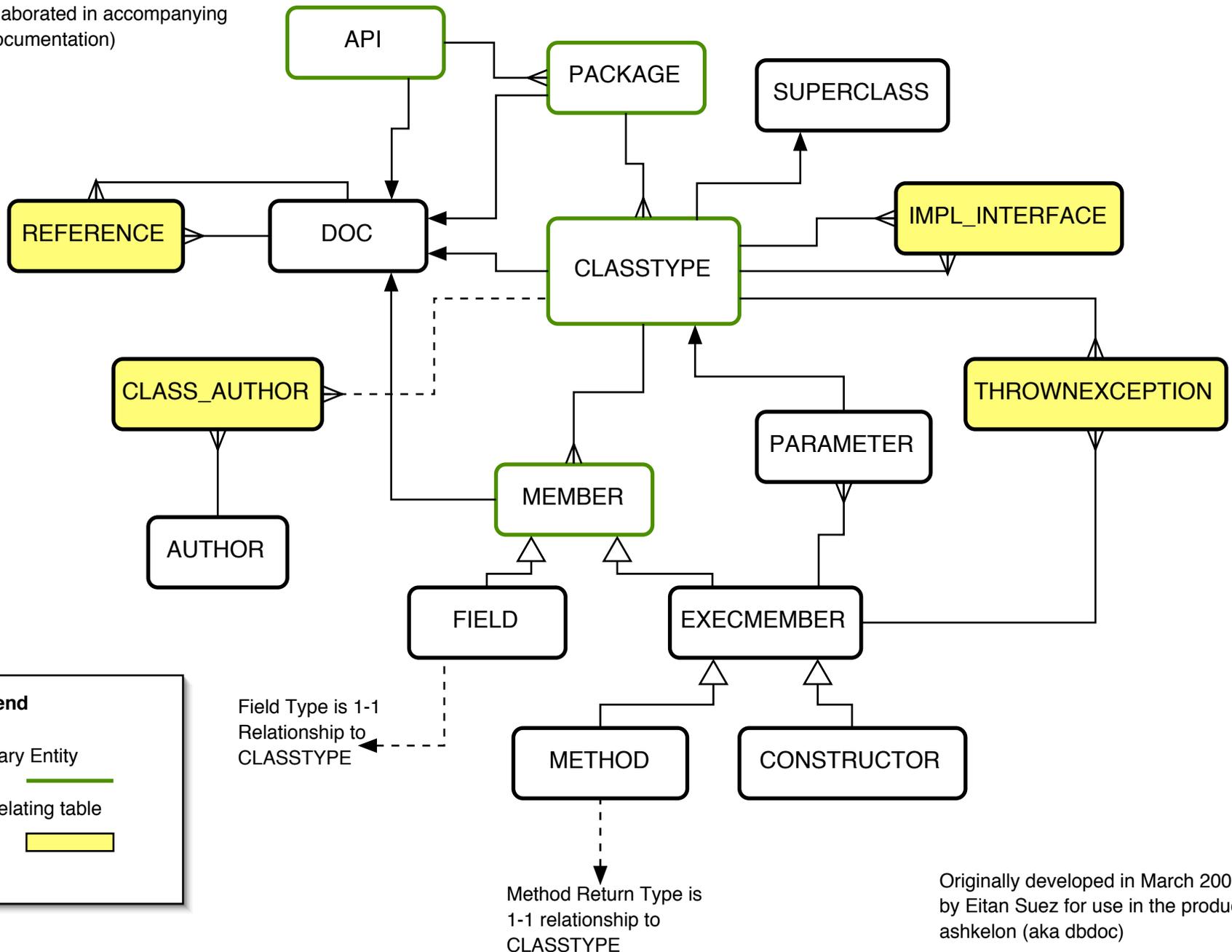


# A Database Schema Modeling Java Programming Language Entities and their Relationships

(Elaborated in accompanying documentation)



## Notes:

1. Table CLASSTYPE represents Java Classes and Interfaces. Name "CLASS" is reserved in some database systems and so is avoided.
2. Table SUPERCLASS is not really necessary. It could have been more simply modeled as a self-referential relationship in CLASSTYPE. However, having the relationship modeled using a separate table did facilitate aspects of the implementation of dbdoc.
3. Model was influenced by the study of the Object Model created by Sun, known as the Doclet API. It would be very beneficial for a reader to be acquainted with the Doclet API model. Many of the terms and concepts used in this schema were taken directly from the Doclet API documentation. If terms like execmember, member, and doc are not clear, please reference the Doclet API.
4. Note the use of inheritance among various types of members (field, method, and constructor).
5. The REFERENCE table is used to define interrelationships between any kind of entity. These are known as 'see also' references. When an author documenting a programming element wants to reference a number of other programming elements, this is modeled via the many-many self-reference among DOC records. For example a specific class can reference a package, a number of methods, and a constructor.