Data Model Design

Effective data models support your application needs. The key consideration for the structure of your documents is the decision to embed or to use references.

Embedded Data Models

With MongoDB, you may embed related data in a single structure or document. These schema are generally known as "denormalized" models, and take advantage of MongoDB's rich documents. Consider the following diagram:

Embedded data models allow applications to store related pieces of information in the same database record. As a result, applications may need to issue fewer queries and updates to complete common operations.

In general, use embedded data models when:

- you have "contains" relationships between entities. See Model One-to-One Relationships with Embedded Documents.
- you have one-to-many relationships between entities. In these relationships the "many" or child documents always appear with or are viewed in the context of the "one" or parent documents. See Model One-to-Many Relationships with Embedded Documents.

In general, embedding provides better performance for read operations, as well as the ability to request and retrieve related data in a single database operation. Embedded data models make it possible to update related data in a single atomic write operation.

To access data within embedded documents, use dot notation to "reach into" the embedded documents. See query for data in arrays and query data in embedded documents for more examples on accessing data in arrays and embedded documents.

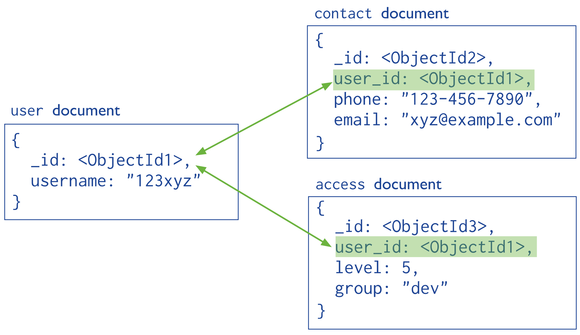
Embedded Data Model and Document Size Limit

Documents in MongoDB must be smaller than the `maximum BSON document size`.

For bulk binary data, consider GridFS.

Normalized Data Models

Normalized data models describe relationships using references between documents.



In general, use normalized data models:

- when embedding would result in duplication of data but would not provide sufficient read performance advantages to outweigh the implications of the duplication.
- to represent more complex many-to-many relationships.
- to model large hierarchical data sets.

To join collections, MongoDB provides the aggregation stages:

- `$lookup` (Available starting in MongoDB 3.2)
- `$graphLookup` (Available starting in MongoDB 3.4)

MongoDB also provides referencing to join data across collections.

For an example of normalized data models, see Model One-to-Many Relationships with Document References.

For examples of various tree models, see Model Tree Structures.

Further Reading

For more information on data modeling with MongoDB, download the MongoDB Application Modernization Guide.

The download includes the following resources:

- Presentation on the methodology of data modeling with MongoDB
- White paper covering best practices and considerations for migrating to MongoDB from an RDBMS data model
- Reference MongoDB schema with its RDBMS equivalent
- Application Modernization scorecard