Design Document Database

Document Database

- Documents encapsulate and encode data (or information) in some standard format or encoding
- Encodings include XML, JSON, binary forms like PDF and Microsoft Office documents (Word, Excel, and so on)
- They are not required to adhere to a standard schema
- Documents in a document store are roughly equivalent to the programming concept of an object
- Allow different types of documents in a single store
- Allow documents to be encoded using different encoding systems
- Collection of documents

A document, encoded in JSON

```
{
    "FirstName": "Bob",
    "Address": "5 Oak St.",
    "Hobby": "sailing"
}
```

A second document might be encoded in XML

```
<contact>
  <firstname>Bob</firstname>
  <lastname>Smith</lastname>
  <phone type="Cell">(123) 555-0178</phone>
  <phone type="Work">(890) 555-0133</phone>
  <address>
    <type>Home</type>
    <street1>123 Back St.</street1>
    <city>Boys</city>
    <state>AR</state>
    <zip>32225</zip>
    <country>US</country>
  </address>
 </contact>
```

Design Document Database

- Document database modelers depend more on heuristics, or rules of thumb, when designing databases
- The rules are not formal
- Must consider how users will query the database, how much inserting will be done, and how often and in what ways documents will be updated

Design Document Database (continued)

- Data redundancy, or denormalization, is commonly used in document database modeling to minimize the need for joins and improve query performance
- Use queries as a guide to determine the appropriate data redundancy
- Developers will write more code to avoid data anomalies
- Indexes can significantly improve query performance, but could slow down the insert, delete and update operations because of index maintenance