Relational v.s. Document Databases

What's going on today?

	Document Databases	Relational Databases
Simpler application code?	a document-like structure with not-too-deep nesting	highly interconnected data, multiple tables involved and JOIN is needed
Schema flexibility	schema-on-read; implicit schema; dynamic type checking; heterogeneous data	schema-on-write
Data locality for queries	the entire document is loaded for access; the size should be small and fairly constant (write-in-palace)	more disk seeks may require for multiple index lookups

- Convergence of both — A hybrid of relational and document models
  - Relational databases (e.g., PostgresSQL) support to save documents (like JSON)
  - Document databases (e.g., RethinkDB) support JOIN and other relational queries

Query Languages for Data

- Declarative query language
  - Specify the pattern of the data you want, not how to achieve the goal
  - Query optimizer chooses indexes, join methods for execution
    - Decouple the user queries from the details of execution
  - SQL, XPath
- Imperative code
  - Tell the computer to perform certain operations in a certain order
  - Map-Reduce programming model