History of data models

Repeating the history?

- Hierarchical model
    - IMS (Information Management System) by IBM in the 1970s
    - Tree structure, similar to the JSON model used by document database
    - Worked well for one-to-many (but NOT many-to-many) relationships
        - Not support JOINs
- Better solutions
    - Network model - the CODASYL model
        - Support many-to-many relationships with more than one parent
        - Query through access paths, but very complex for developer to code and change
    - Relational model
        - query optimizer is the key to win in the long run
            - automatically choose ‘access path’ for data operations
            - complicated, but provides a general solution (no programming)
- Document databases use hierarchical model
    - have nested records with their parent record (rather than in a separate table)
    - document reference as a unique identifier, resolved at read time by JOIN or follow-up queries

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