20: Relational vs. NoSQL

① Recall MongoDB from COMP20
- JSON style syntax; JS
- Databases, collections, docs

② The Bigger Pic: NoSQL DBs
- Performance, speed
- Scalability
- Flexibility
- Ease of use
- Get away from BOFHs (i.e. DB admins)
- Options: Mongo, CouchDB, Hbase, Cassandra, Redis...
- Dynamic data model
  * inserting record into a collection that does not exist will create new collection
  * create docs w/ varying fields
  * last record wins; easy to overwrite records
- RTFM for each database system
- Generally very weak security

③ Where will NoSQL DBs will NOT work?

" " work?

④ The Relational System
- Database → same idea
- Schema: logical def) of a table (i.e., blueprint)
  - Name of table
  - Name + type of each column
- Table: structure w/ rows, each of which has the attributes defined by schema
  - Index: aid lookup
  - Primary key
  - Unique key (multiple in tbl)
  - Foreign key
- Analogy:
  - Collection: table
  - Document: row

⑤ Modeling
- Natural language can only do so much
- Allow engineers to understand, business req. + design
- Identify relationships + constraints + metadata
- Make data model precise w/ SQL tbl defs