Date _____ 20 ____

Section 2: mapping & indexing data

Mapping = schema definition

- field types
- field index — do you want this field indexed for full text search
  analyzed / not analyzed
- field analyzers
  - define tokenizers → standard/whitespace etc (how to split strings)
  - & token filters → lowercase, uppercase etc

Date _____ 20 ____

Analyzers

character filters
remove html encoding etc

Tokenizer
split strings on whitespace/punctuation/
non letters etc.

Token filter
lowercasing stemming, synonyms, stopwords

choices for Analyzers

standard
split on word boundries, remove punctuation,
lowercases, good choice if language is unknown

simple splits on anything that isnt a letter & lowercase

whitespace splits on whitespace but doesnt lowercase

language accounts for language specific stopwords
& stemming

Sharjeel

Date _____ 20 ____

only need to define mapping if something need to be specifically set to a certain type

or else type inference will be used

Versions

Every document has a version field

Elasticsearch documents are immutable

When you update an existing document:

- new doc created w/ incremented _version
- old doc marked for deletion

Date ____ 20 ____

Dealing with Concurrency

- Optimistic concurrency control
  - sequence number
  - primary shard that owns sequence

use retry_on_conflicts = N to auto retry

Controlling full text search

Using Analyzers

- sometimes text fields should be exact-match
  - use keyword mapping instead of text

- search on analyzed text fields will return anything remotely relevant

  - Depending on the analyzer results will be case insensitive, stemmed, stopwords removed, synonyms applied etc.
  - Searches w/ multiple terms need not match them all

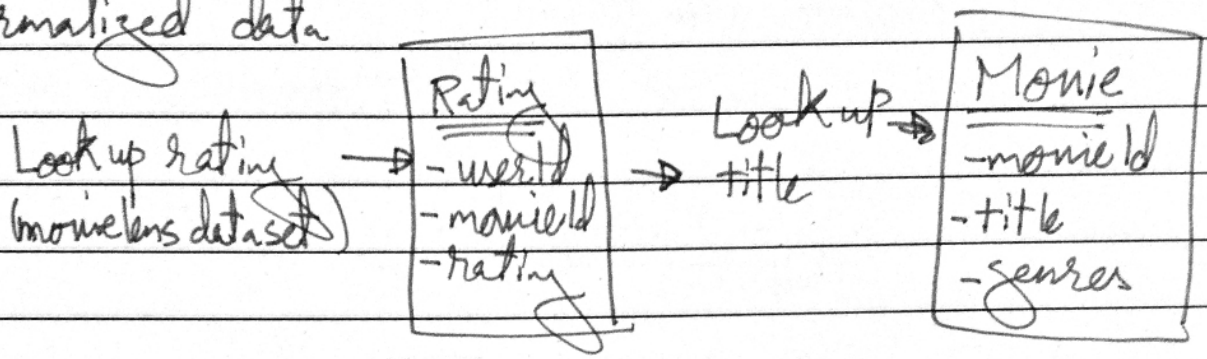
Shafjeel

Date _____ 20 ____

Data Modeling

Strategies for relational data

- Normalized data



- Min storage, makes it easy to change
- require 2 queries, storage cheap

doubles traffic on cluster - page response time

Denormalized data

look up rating → Rating
userId
rating
title

duplicated title, 1 query
changing title difficult

Date _____ 20 ____

Parent / Child relationship

Movie franchises etc

Flattened Datatype

scaling w/ large # of subfields

Flattened subfield: avoid every subfield as individual fields use one subfield

elastic search's performance suffers if too many inner fields - mapping explosion

use when unknown or large # of fields

Cluster collection of elasticsearch nodes
cluster state passed b/w nodes so clusters
run smoothly. master node sends node state
to other nodes, they ack

Shahjeel

Date _____ 20 ____

after each cluster state change nodes need to be synced

frequently adding new fields to index causes:
- cluster state to grow
- triggers cluster state updates across all nodes which can cause delays

w/o update cluster state nodes can't index, search etc can cause
- memory issues
- poor performance
- cluster crash (mapping explosion)

map entire object w/ inner field into single field → flattened

Date _____ 20 ____

limitation of flattened

Keys treated as keywords
no analyzers and tokenizer

Supported queries for flattened data types

- term, terms & term_set
- prefix
- range (non numerical range ops)
- match & multi_match (supports exact keywords)
- query_string & simple_query_string
- exist

result highlighting feature not enabled for flattened data type

Date _____ 20 ____

Mapping Exceptions

Mapping

↳ Defining how JSON doc will be stored

Res:
= actual metadata resulting from the definition process

Explicit Mapping

predefined fields & types

Dynamic Mapping

fields & types defined by ES

Date _____ 20 ____

What could go wrong?

Explicit

Mapping Exceptions when mismatch

Dynamic

mapping explosion

Dead Queue Pattern

store failed documents that throw mapping exception in seperate queue

handle on app level or use logstash DL queue

allows to still process failed docs

Date _____ 20 ____

limits of mapping

default # 1000 # of fields in a mapping

changing to greater than 1000 may cause performance degradation & high memory pressure