In this lecture, we will discuss…

- `unique` command
- `sparse` command
- TTL (time to live)

`unique` index

- `unique` property for an index causes MongoDB to reject duplicate values for the indexed field
- `db[:users].indexes.create_one( { "user_id": 1 }, { unique: true } )`

Note: By default, unique is false on MongoDB indexes

`unique` index

- The unique constraint applies to separate documents in the collection
- Prevents separate documents from having the same value for the indexed key

`sparse` index

- Contains entries for documents that have the indexed field, even if the index field contains a null value
- The index skips over any document that is missing the indexed field
- The index is “sparse” because it does not include all documents of a collection

`sparse` index

- `db[:users].indexes.create_one( { "user_id": 1 }, { sparse: true } )`

`TTL` index

- Special single-field indexes that MongoDB can use to automatically remove documents from a collection after a certain amount of time
- Data expiration is useful for certain types of information like machine generated event data, logs, and session information that only need to persist in a database for a finite amount of time

`TTL` index

- `db[:zips].indexes.create_one({ :state => 1 }, {expireAfterSeconds: 3600})`
- The documents are removed from the collection after 3600 seconds

Summary

- Index properties
  - Unique
  - Sparse
  - TTL

What's Next?

- Mongoid