Indexes

1. Tool to analyze how it executed the query.

* db.contacts.explain(‘executionStats’).find({‘dob.age’: {$gt: 60}})

1. Adding an index to a collection.

* db.contacts.createIndex({‘dob.age’: -1}), (-1 is descending order)

1. Getting rid of an index.

* db.contacts.dropIndex({'dob.age':1})

1. Compound index.

* db.contacts.createIndex({‘dob.age’: 1, gender: 1})

1. Can also use sort with indexes.

* db.contacts.find({‘dob.age’: 30}).sort({gender: 1})

1. Indexes can ensure data consistency and give you errors on inserts that violate that rule or when creating the index as well.

* db.contacts.createIndex({email: 1}, {unique: true})

1. Partial Index, speeds up inserts and updates partially.

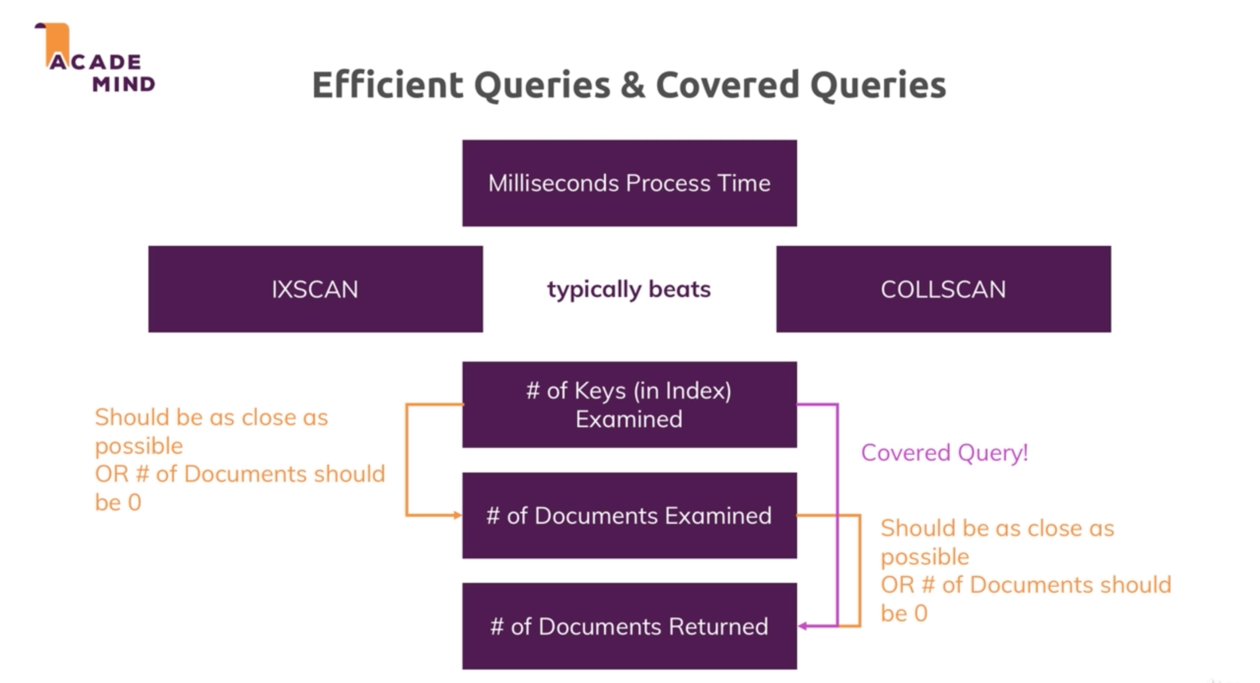
* db.contacts.createIndex({‘dob.age’: 1}, {partialFilterExpression: {gender: ‘male’}})

1. Apply partial index for consistency. (null values are treated as uniwue)

* Db.contacts.createIndex({email: 1}, {unique: true, partialFilterExpression: {email: {$exists: true}}})

1. Setting an index to autodelete after a certain amount of time. (Only works on date fields).

db.sessions.createIndex({createdAt: 1}, {expireAfterSeconds: 10})



1. When an index is created and you only want the filtered field which is in the index to be returned.

* db.customers.explain(‘executionStats’).find({name: ‘Max’}, {\_id: 0, name: 1})

1. See the details of every index you have created.

* db.customers.explain(‘allPlansExecution’).find({name: ‘Max’, age: 30})

1. Multi key indexes are for arrays within a document.

* db.contacts.createIndex({hobbies: 1})

1. Creating a text index. (Only one allowed per c ollection).

* db.contacts.createIndex({description: ‘text’})
* db.products.find({$text: {$search: {'awesome'}}})
* db.products.find({$text: {$search: '"red book"'}})
* db.products.find({$text: {$search: ‘awesome t-shirt’}, {$score: {$meta: ‘textScore’}}}).sort({score: {$meta: {‘textScore’}}})
* drop text indexes by name

1. Can merge text indexes into one.

* Db.products.createIndex({title: ‘text’, description: ‘text’})

1. Exclude words from searching filter

* db.products.find({$text: {$search: ‘awesome -t-shirt’}})

1. Adding weights for text filtering.

* db.products.createIndex({title: ‘text’, description: ‘text’}, {default\_language: ‘english’, weights: {title: 1, description: 10}})

1. Building indexes in the background so they can be  
    accessible at all times.

* db.ratings.createIndex({age: 1}, {background: true})