



SNS COLLEGE OF TECHNOLOGY

Re-accredited by NAAC with A+ grade, Accredited by NBA(CSE, IT, ECE, EEE & Mechanical)
Approved by AICTE, New Delhi, Recognized by UGC, Affiliated to Anna University, Chennai



Create



Read



Update



Delete

CRUD

Department of Computer Applications

CRUD Operations with MongoDB

COURSE : 23CAT603- Database Management System

UNIT V : Column Oriented Database

CLASS : I Semester / I MCA

March 1, 2024

Column Oriented Database / DBMS / Dr.S.Sundararajan/ MCA/ SNSCT



Basic Operations



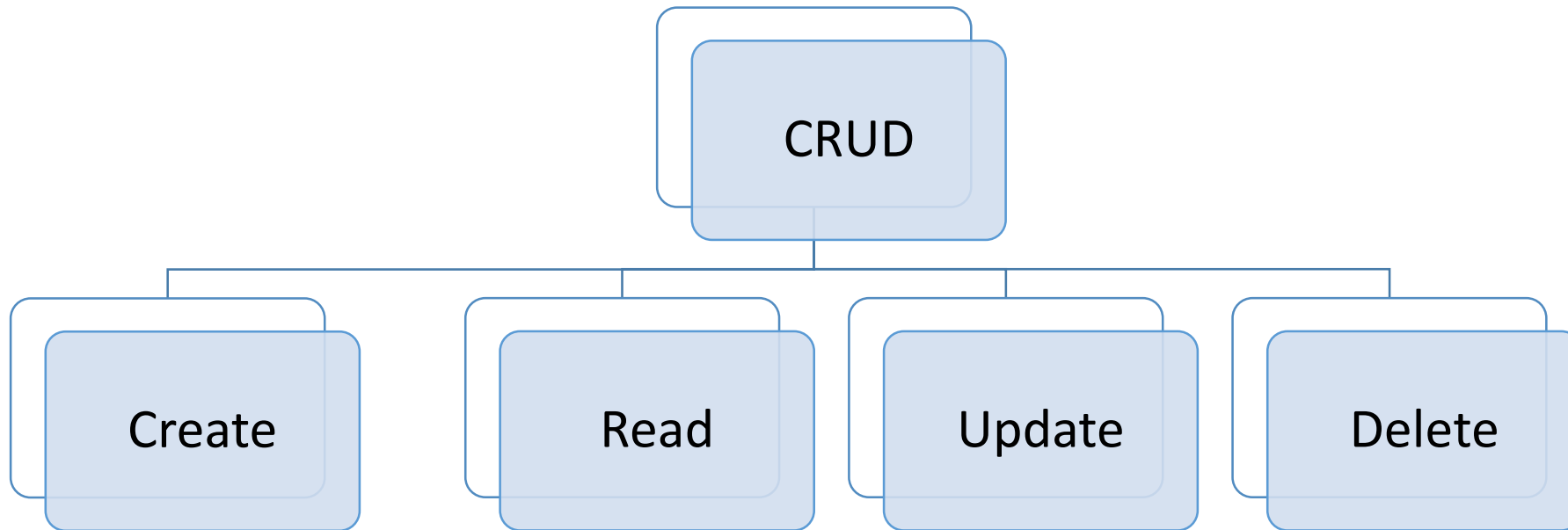
use	db	Show dbs	db.dropDatabase()
Switched to database	Check currently selected database	Display the list of databases	Drop the database



db.createCollection (name)	Show collections	db.collectionname.insert ({key,value})	Db.collection.drop ()
Create collection	List all collections	If no collection, automatically created when insert	Drop the collection



CRUD Operations





CRUD Operations - Basics

purpose	Commands	Remarks
Create a database	Use <database name>	Switch to db or create a new one
Drop a database	Db.dropdatabase	Delete the selected database
Show databases	Show dbs	Shows all the databases
Create a collection	Db.createCollection(name,options)	Create a collection with options indexing and memory
Drop a collection	Db.collectionname.drop()	Drop the collection
Show collections	Show collections	Shows all the collections



CRUD Operations - INSERT



- ❑ To insert data into collection, use

Db.collectionname.insert(doc)

- ❑ For example

Db.student.insert({name: "ram", course: "mca", age:15})

- ❑ If the document doesn't contain `_id` field, mongoDB automatically add this and assign objectID for document before inserting
- ❑ `_id` value (12 bytes field) is unique

4 bytes	3 bytes	2 bytes	3 bytes
Current timestamp	Machine id	Process id of mongodb server	Incremental value



CRUD Operations - INSERT



To insert single document

```
db.student.insert({name:"rem", age:21, addr: "22, x street"})
```

To insert multiple documents

```
db.student.insert(  
  {name: "ram", course: "mca", age:21},  
  {name: "gita", course: "mba", age:22}, )
```

To copy all documents from old collection to new. If new collection not exists, it will be created automatically

```
db.source.copyTo(target)
```

To insert one or many

```
db.student.insertOne({name:"rem", age:21, addr: "22, x street"})  
db.student.insertMany({name:"rem", age:21, addr: "22, x street"},  
  {name:"preem", age:22, addr: "15, K street"})
```



CRUD Operations - INSERT



- ❑ To insert multi columns

```
db.student.insert({name:"rem", address {city: "cbe", place:"mtp"}} )
```

- ❑ To insert multivalued attribute

```
db.student.insert(  
  {name:"rem",  
  hobbies["singing", "dancing"]  
  })
```

- ❑ To insert multivalued with multicolumn attribute

```
db.student.insert(  
  {name:"rem",  
  award[ {Rank:1, year: 2008, skill: "singing"},  
         {Rank:3, year: 2008, skill: "drawing"},  
         {Rank:3, year: 2011, skill: "drawing"} ]  
  })
```



CRUD Operations – READ/FIND



- ❑ To display data from collection

`db.collectionname.find()`

- ❑ To display data from collection in the formatted way

`db.collectionname.find().pretty()`

- ❑ To use query the document

`{<field>:<value>}`

- ❑ For example

`Db.student.find({name:"rag"})`



CRUD Operations – READ/FIND



\$eq	equal
\$gt	Greater than
\$lt	Less than
\$gte	Greater than or equal to
\$lte	Less than or equal to
\$ne	Not equal
\$in	Matches any one of value in an array
\$nin	Matches none of the values in an array

- For example

db.student.find({rno: {\$gt: 501}})

db.student.find({rno: {\$gt: 501, \$lt: 600}})

To find particular column

db.student.find({rno: 501})



CRUD Operations – READ/FIND



- ❑ To return no.of documents in the collection by

`db.student.find().count()`

To return no.of documents in the collection with condition by

`db.student.find({rno: 501}).count()`

To return first two documents in the collection by

`db.student.find({rno: 501}).limit(2)`

To return documents by skipping first 5 by

`db.student.find({rno: 501}).skip(5)`

To return first document by

`db.student.findOne()`



CRUD Operations – READ/FIND



- ❑ To return documents whose attribute value is either one

```
db.student.find({name:["riya", "jaya"]})
```

To return documents whose information not in

```
db.student.find( {rno:{$nin[510,515]}}
```

To return documents with distinct value

```
db.student.distinct("address")
```

To return documents by skipping first 5 by

```
db.student.find( {rno: 501}).skip(5)
```

To return first document by

```
db.student.findOne()
```



CRUD Operations – UPDATE



❑ Syntax for it

```
Db.collectionname.update (  
    <query/condition>  
    <update with $set or $unset>  
    {  
        upsert: <Boolean>  
        Multi:<Boolean>  
    }  
)
```

- ❑ **Upsert** : if set to true, creates new document if no matches found
- ❑ **Multi**: if set to true, update multiple documents meets criteria

❑ Example

```
Db.student.update({_id:101}, {$set{age:23}})
```

```
Db.student.update({_id:101}, {$unset{age:1}})
```



CRUD Operations – DELETE



- ❑ To remove all documents

`db.student.remove({})`

- ❑ To remove all documents which matches condition

`db.student.remove({type:"food"})`

- ❑ To remove a single document which matches condition

`db.student.remove({type:"food"},1)`

To return documents by skipping first 5 by

`db.student.find({rno: 501}).skip(5)`

To return first document by

`db.student.findOne()`



CRUD Operations – DELETE



- <https://www.tutorialspoint.com/NoSQL-Databases>
- <https://www.geeksforgeeks.org/nosql-data-architecture-patterns>
- <https://www.mongodb.com/nosql-explained>
- Shannon Bradshaw, Eoin Brazil, and Kristina Chodorow, “MongoDB: the Definitive Guide”, O’Reilly Media, 3rd Edition



March 1, 2024

