

se on elements array
luation theory

PRACTICAL NO : 4

Aims- practice exercise on element array base
and evaluation theory operator

Theory :- MongoDB Query operators:-
\$exist, \$type, \$%, \$regex

MONGODB provides various query operators
to filter documents based on
specific conditions.

1. \$exists operator :-

- ↳ checks whether a particular field exists in the document
- ↳ returns true if the field is present else false

e.g:- find documents where the age field exists

db.collection.find(

{ "age": { "\$exists": true } }

2. \$type operator :-

- ↳ matches documents where the field is of specified BSON data type.
- ↳ helps ensure fields are of expected type (string, no, array etc)

e.g:- find documents where salary is stored as no

db.collection.find(

{ "salary": { "\$type": "number" } }

"\$string" :- matches string fields

"\$number" :- matches integer values

"\$array" :- matches array fields

"\$bool" :- matches boolean values

3. \$mod (Modulo operator, as %)

↳ used to find numbers

divisible by given value

↳ returns documents where field % divisor
== remainder

e.g:- find documents where

employee_id is even (divisible by 2)

→ db.collection.find(

{ "employee_id": { "\$mod": [2, 0] } }

Here \$mod [2, 0] means $\text{employee_id} \% 2 == 0$

4. '\$regex' (regular expression matching) :-

↳ allows pattern matching
with string fields

↳ useful for text search and
case insensitive matching.

e.g:- find users whose name starts with "A"

→ db.collection.find({ "name": { "\$regex":
"^A", "\$options": "i" } })

* RESULT:- Hence successfully practised array and
evaluation base operators.

PRACTICAL NO : 5

Aims- exercise on MongoDB shell command
and user management command.

Theory- Mongodb shell commands

1. Database operations

↳ MongoDB organizes data into database

- o Show all databases :-

⇒ show dbs

- o Switch to a database :-

⇒ use myDatabase

- o Check the current database

⇒ db

- o Delete a database

⇒ db.dropDatabase()

2. Collection Operations :-

- o Create a new collection

db.createCollection("Students")

- o Show all collections in current database

show collections

- o Delete a collection

db.Students.drop()

3. Document operations (CRUD)

- Insert a single document

⇒ db.students.insertOne({name: "Adil", age: 20})

- Retrieve all documents

⇒ db.students.find().pretty()

- Find documents with a condition

⇒ db.students.find({age: {\$gt: 21}})

- Update document

db.students.updateOne({name: "Adil"},
{\$set: {age: 21}})

- Delete document

⇒ db.students.deleteOne({name: "Alice"})

* MongoDB User Management Commands.

- Create user -

⇒ use companyDB
db.createUser({

user: "adminUser",

pwd: "password123",

roles: [

{role: "readWrite", db: "companyDB"}]

- List all users :-

⇒ db.getUsers()

- Authenticate a user

⇒ db.auth("adminUser", "password123")

2. Managing user roles :-

- grant additional role to existing user
→ db.grantRolesToUser("adminuser",
[{"role": "dbAdmin", "db": "companyDB"}])
- revoke a role from a user
→ db.revokeRoleFromUser("adminuser",
[{"roles": "rdbAdmin", "db": "companyDB"}])
- delete a user from database
→ db.dropUser("adminuser")

* RESULT :-

Hence we have successfully practised
MongoDB and User Management
Commands.

RAISONI GROUP
— a vision beyond —