NoSQL Databases.

ORACLE, MySQL } RDBMS (MySQL uses SQL)

Ex:- MongoDB

jfg-(introduction to nosql)

NoSQL Stands for → non-SQL (or) not only SQL (or) non-relational SQL Databases

DBMS

NoSQL → it is type of Database Management System that is designed to handle and store large volumes of unstructured and semi-structured data.

NoSQL Databases are generally classified into 4 main Categories

1) Document Databases:-

These Databases store data as semi-structured documents such as JSON or XML

2) Key-value stores:-

These Databases store data as key-value pairs.

3) Column-family stores:-

These Databases store data as column-families which are sets of columns that are treated as a single entity.

4) Graph databases:-

These databases store data as nodes and edges.

MongoDB // Javatpoint.com - mongodb tutorial

→ MongoDB is an open source cross-platform document oriented no-SQL database.

→ It provides a language called as MQL stands for MongoDB Query Language.

Installation of MongoDB // gfg - how to install MongoDB

Step 1:- Go to MongoDB download Center
Select '4.4.28 version.

not follow step 9th and 10th (exactly do)

Step 11:- cmd prompt → C:/Program files/MongoDB/Server/

in C:data/db → create db folder in data folder
↓
create new folder in C drive

4.4/mongoD
↓
execute mongoD cmd

- - - - /4.4/mongoD
↓
run this command.

in another cmd prompt

- - /4.4/mongo
↓
run this command

SQL terms / concepts	MongoDB terms / concept
1) database	1) database
2) Table	2) collection
3) row	3) Document or JSON document
4) Column	4) Field
5) primary-key	5) Primary-key
6)	

Create Database in MongoDB

The command for creating the database is

use databasename.

↘ if dB is not created then it creates. if it is already created then it connects

dB → Shows

Show dbs → list out all the databases created.

↓ atleast it should have one document to list out

db.collectionname.insert({"name":"xyz"})

↓

it creates a collection and inserts the document

For droping the database the command is

db.dropDatabase(); //drops the current database

Creation of Collection:-

db.createCollection("name");

To see the collections that are created:-

Show collections;

for droping the collection:-

db.collection.drop;

→ using array inserting multiple documents

(Array insertion explore

CRUD operations:-

↓

insertMany → for inserting many documents (tuples at the same time)

C → Create :- inserting the document

R → Read

U → Update :-

D → delete

for updating db.collectionname.update({'course': 'Java'}, {$set:{'course':'android'}})

↑ Key-value

Syntax for deleting the documents.

db.collection.remove({})

↓

removes all the documents in the collection

if we want to delete particular document:

db.collection.remove({'course': 'Java'})

if we want to delete only first occurrence then

db.collection.remove({'course': 'Java'}, 1)

Query the Document:- $gt 10

db.collection.find() → list out all the documents in the collection.

db.collection.find().pretty() → list out the documents in formatted way

upsert operation

↓

it is a boolean

upsert:- The document we need to update and then it is not present then it inserts when update = 1.

Program 1:

1) Create "Crew" collection and perform the following CRUD operations:

a) Create a document

b) Create two (or) more documents at the same time.

c) Update a document with crew number 10 to 20

d) Delete all the crews with strength 10

e) Retrieve the crews with strength greater than equal to 20.

```
use studio;
db.createCollection("Crew");
show collections; // display collections
```

a) db.crew.insertOne({
    crewno:10,
    strength:15,
    sector:"Thriller"});

b) db.crew.insertmany([
{crew_no:20, strength:20, sector:"thriller"},
{crew_no:25, strength:28, sector:"action"},
{crew_no:26, strength:23, sector:"action"}]);

c) db.crew.updateOne(
    {crew_no:10},
    {$set:{crew_no:20}});

d) db.crew.deleteMany({strength:10});

e) db.crew.find({strength:{$gte:20}});

Program - 2

Create "Studio" location and perform the follow CRUD operations.

a) create a document

b) create two (or) more documents at the same time

c) Update a document with studios with locat ~~xyz~~ name 'std1' to 'std2'

d) Delete all the studios with location 'xyz'

e) Retrieve the studio with location equal to 'xyz'

a) db.studio.insertOne({
name: "std1",
branch: "branch1",
location: "Mandya"});

b) db.studio.insertMany([
{name: "std1", branch: "branch1", location: "Sira"
{name: "std3", branch: "branch2", location: "tmk"
{name: "std4", branch: "branch3", location: "Hasan"
]);

c) db.studio.updateOne(
{name: "std1"},
{$set:{name: "std2"}});

d) db.studio.deleteMany({location: "xyz"});

e) db.studio.find({location: "xyz"});

Program - 3

3) Create "CartoonSerial" collection and perform following CRUD operations:

a) Create a document

b) Create a two or more documents at the same time.

c) Update a document with cartoon serial title "std1" to "std2"

d) Delete all the cartoon serials with title "xyz"

e) Retrieve the cartoon serial with sensor number less than 100

a) db.cartoon_serial.insertOne({
title: "Std1", S_no: 15, yop: "2020"});

b) db.cartoon_serial.insertMany([
{title: "std2", S_no: 10, yop: 2011},
{title: "std1", S_no: 20, yop: 2013},
{title: "std3", Sno: 30, yop: 2023}]);

c) db.cartoon_serial.updateOne(
{title: "std1"}
{$set: {title: "std2"}});

d) db.cartoon_serial.deleteMany
({title: "xyz"});

e) db.cartoon_serial.find({S_no: {$lt: 100}});

Program-4

Create "Car" collection and perform the following CRUD operations:

a) create a document
b) create two/more documents at the same time
c) Update a document with car reg_no 10 to 20
d) Delete all the cars with model 'xyz'
e) Retrieve the cars with color green.

a) db.car.insertOne({reg_no:10, model:'xyz',
color: "yellow"});

b) db.car.insertMany([
{reg-no:12, model: "thar", color: "black"},
{reg_no: 15, model: "Swift", color: "white"},
{reg_no: 20, model: "thar", color: "green"}]);

c) db.car.updateOne(
{reg-no:10}, {$set: {reg_no:20}});

d) db.car.deleteMany({model: "xyz"});

e) db.car.find({color: "green"});

Program - 5

Create "dog" collection and perform the following CRUD operations:

a) Create a document.

b) Create two (or) more documents at the same time

c) Update a document with dog name 'xyz' to 'abc'

d) Delete all the dogs with gender male.

e) Retrieve the dogs with gender female.

a) db.dog.insertOne({name: "xyz", gender: "male"});

b) db.dog.insertMany([
{name: "Charlie", gender: "male"},
{name: "Bunty", gender: "male"},
{name: "Chinnu", gender: "female"}
]);

c) db.dog.updateOne(
{name: "xyz"}, {$set: {name: "abc"}});

d) db.dog.deleteMany({gender: "male"});

e) db.dog.find({gender: "female"});