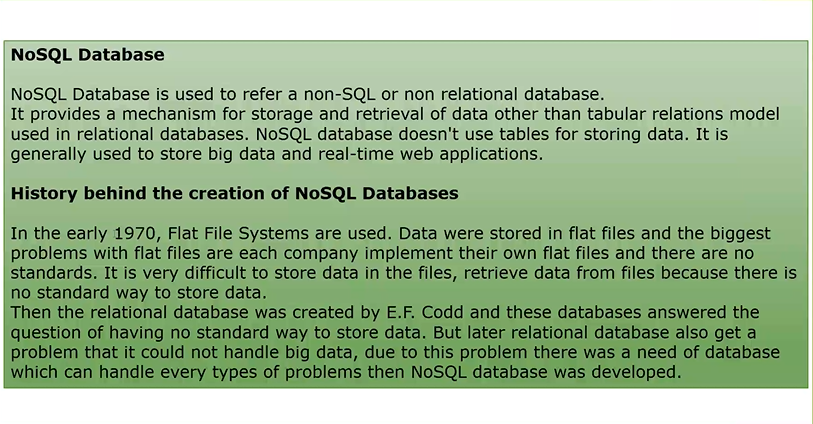
**What is MongoDB?**

MongoDB is a NoSQL, document-oriented database developed by MongoDB Inc. It stores data in JSON-like documents (BSON) rather than traditional rows and columns (as in SQL databases).

**Key Features:**

* Schema-less: No fixed structure; documents in a collection can vary.
* High Performance: Optimized for read/write speed and horizontal scaling.
* Horizontal Scalability: Supports sharding to distribute data across servers.
* Rich Query Language: Allows filtering, sorting, aggregation, indexing, and joins

****

Example:

{

"\_id": ObjectId("..."),

"name": "Harish",

"email": "harish@example.com",

"skills": ["Python", "SQL", "MongoDB"]

}

The SQl will keep the data in various tables where the mongo DB keep at one

If the SQL has the tables

Table 1=Customer

Table 2=Orders

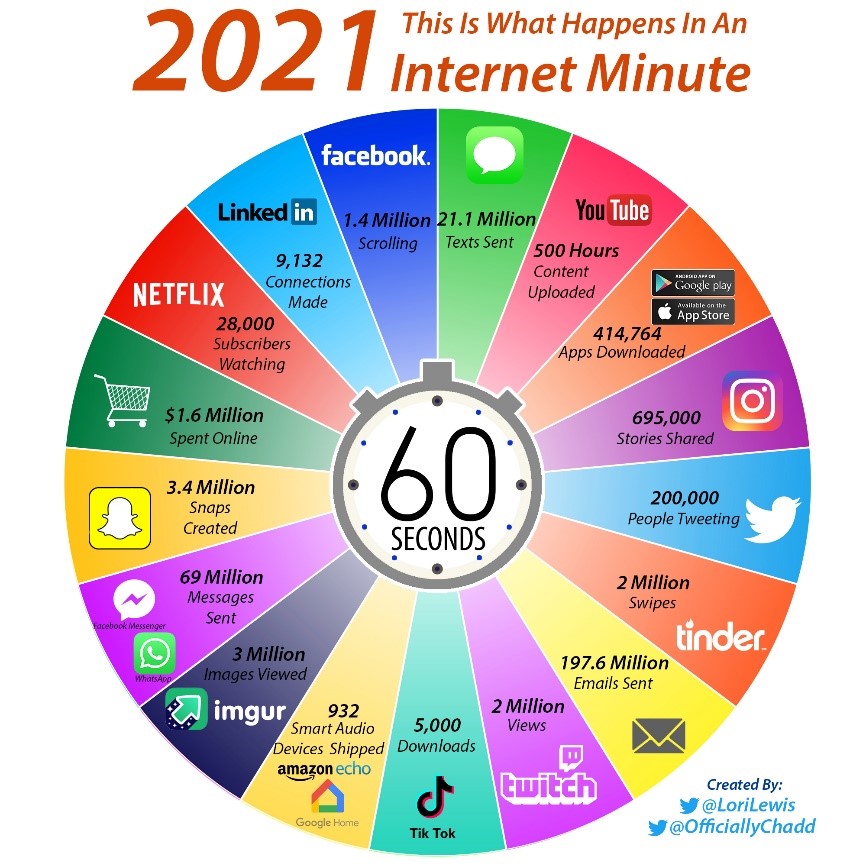
Table 3=Products

The mongo db will be

Customer{

Orders{

Products{}

}

**Work Out**

1.create database and collection

2.Add data – insert document – enter the data in the key value pair

Inserting data

{

"name": "Harish",

"department": "Engineering",

"salary": 50000,

"age": 22

}

**Working on shell**

3. open mongo shell

Use company DB #(database name)

# insert many records

db.employees.insertMany([  
  {  
    name: "Neha Reddy",  
    department: "Marketing",  
    salary: 45000,  
    age: 28  
  },  
  {  
    name: "Faizan Ali",  
    department: "Engineering",  
    salary: 58000,  
    age: 32  
  },  
  {  
    name: "Divya Mehta",  
    department: "HR",  
    salary: 40000,  
    age: 29  
  },  
  {  
    name: "Ravi Verma",  
    department: "Sales",  
    salary: 35000,  
    age: 26  
  }  
]);

# # the employees used in the below queries are the collection name in the database where the data are stored

# To see all the data

db.employees.find()

# To find the specific one data

db.employees.findone({name:”Amit Sharma”})

# To display the top or first record

db.employees.findone() -- which displays the top or the first data

# To show the specific rows in the collection

db.employees.find({},{name:1,department:1}) – if we give 0 instead of 1 it will not display we can use 1 to 10

# To fetch the data based on condition

db.employees.find({salary:{$gt:50000}}) -- greater than 50000

db.employees.find({salary:{$lt:50000}}) -- lower than 50000

db.employees.find({salary:{$eq:50000}}) -- equal to 50000

db.employees.find({age:{$gte:28,$lte:32}}) – age b/w 28 and 32

db.employees.find({department:{$in:[“HR”,”Sales”]}}) – employees in the hr and sales department

db.employees.find({department:{$ne:”Marketing”}}) – not belongs to marketing

db.employees.find({name:{$regex:”^A”}}) – name starts with “A”

# ordering and limiting

db.employees.find().sort({salary:-1}) – sort according to the salary desc

db.employees.find().sort({salary:-1}).limit(3) – limit the records to 3

# To update

db.employees.updateOne({name:”Ravi verma”},{$set:{salary:35000}}) – update the one record

if two records on the same name the top will update

# To update multiple record we can use the updateMany()

db.employees.updateMany({department:”Engineering”},{$set:{salary:5000}})

# delete

db.employees.deleteOne({name:”Divya Mehta”}) – delete one record

db.employyes.deleteMany({department:”Sales”}) – delete many record