Belajar MongoDB

1. Pengenalan MongoDB
   1. MongoDB merupakan free dan opensource database management system
   2. MongoDB merupakan database management system berbasis document
   3. Dikembangkan oleh perusahaan bernama 10gen tahun 2007
   4. Dirilis ke public tahun 2009
   5. Saat ini perusahaan 10gen sudah berganti nama menjadi MongoDB Inc
   6. MongoDB hampir mendukung semua bahasa pemrograman sebagai client nya
   7. MongoDB tidak menggunakan SQL, namun menggunakan Javascript sebagai bahasa utama untuk manipulasi document
   8. https://github.com/mongodb/mongo
2. Apa Itu Document Oriented Database
   1. Document oriented database merupakan sistem database yang digunakan untuk memanipulasi data dalam bentuk document (semi structured data)
   2. Biasanya document disimpan dalam bentuk JSON atau XML
   3. Document oriented database biasanya bertolak belakang dengan relational database
   4. Relational database biasanya menyimpan data dalam bentuk table, dan menyimpan relasi nya di table lain
   5. Document oriented database biasanya menyimpan data dalam bentuk JSON atau XML, dan menyimpan relasi nya sebagai embedded object didalam document yang sama
3. Istilah Relational Database(MySql) vs Document Database (MongoDB)

|  |  |
| --- | --- |
| Relational Database (Mysql) | Document Database (MongoDB) |
| Database | Database |
| Table | Collection |
| Column | Field |
| Row, Record | Document (JSON, XML, dan Lain-Lain) |
| Join Table | Embedded Document , Reference |
| SQL | Javascript (MongoDB) |

1. Menginstall MongoDB
   1. Linux : Documentasi resmi mongodb
   2. Mac : Documentasi resmi mongodb
   3. Windows : Documentasi resmi mongodb
2. Menginstall MongoDB Menggunakan Docker
   1. MongoDB Docker Image : Document resmi docker
3. Mongo Shell
   1. MongoDB menyediakan aplikasi mongo client berupa command line interface untuk terkoneksi ke MongoDB server dengan nama mongo shell
   2. Mongo shell sangat bermanfaat saat kita tidak harus konek ke mongo server tanpa GUI
   3. Mongo shell menggunakan bahasa pemrograman javascript
4. Menggunakan Mongo Shell
   1. mongo --host localhost --port 27017
5. MongoDB GUI Client
   1. Jika kita terbiasa menggunakan GUI, ada beberapa aplikasi yang bisa kita gunakan sebagai mongo client, seperti:
      1. MongoDB Compass
      2. Jetbrain DataGrip
      3. MongoDB for Visual Studio Code
      4. Robo 3T
6. Database
   1. Database adalah tempat menyimpan collection
   2. Semua collection harus disimpan di database
   3. Biasa nya database digunakan untuk memisahkan data secara logical per aplikasi, artinya biasanya satu aplikasi akan memiliki satu database
   4. Jarang sekali kita akan menggunakan satu database untuk beberapa aplikasi
7. Membuat Database
   1. Kita tidak perlu secara eksplisit membuat database
   2. MongoDB akan secara otomatis membuatkan database sesuai dengan nama database yang kita pilih
   3. Untuk memilih nama database, kita bisa menggunakan perintah “use” diikuti nama database
8. Memilih Database
   1. > use belajar
   2. > use tutorial
   3. > show database
9. Database Methods

|  |  |
| --- | --- |
| Database Methods | Keterangan |
| db.dropDatabase() | Menghapus Database |
| db.getName() | Mengambil Nama Database |
| db.hostInfo() | Mengambil informasi host tempat mongodb |
| db.version() | Mengambil versi database |
| db.stats() | Mengambil statistik penggunaan database |

Referensi lengkap doc resmi

1. Collection
   1. Collection adalah tempat menyimpan document
   2. Maximum per document yang bisa disimpan adalah 16mb
   3. Maximum level nested document yang bisa disimpan adalah 100 level
2. Database Methode untuk Collection

|  |  |
| --- | --- |
| Database Methods Untuk Collection | Keterangan |
| db.getCollectionNames() | Mengambil Semua Nama Collection |
| db.createCollection(name) | Membuat Collection Baru |
| db.getCollection(name) | Mendapatkan Object Collection |
| db.<name> | Sama dengan db.getCollection(<name>) |
| db.getCollectionInfos() | Mendapat informasi semua collection |

Referensi lengkap doc resmi

1. Collection Methods

|  |  |
| --- | --- |
| Database Mothods Untuk Collection | Keterangan |
| db.<collection>.find() | Mengambil semua document |
| db.<collection>.count() | Mengambil jumlah document |
| db.<collection>.drop() | Menghapus collection |
| db.<collection>.totalSize() | Mengambil total ukuran collection |
| db.<collection>.stats() | Mengambil informasi statistik collection |

Referensi lengkap doc resmi

1. Kenapa Perlu Mengerti Data Modeling
   1. Pindah dari relational database ke document database bukanlah hal yang sesederhana hanya dengan memindahkan semua table ke collection
   2. Penggunaan document database tidak akan mendatangkan manfaat besar jika kita tidak mengerti cara memodelkan data untuk kebutuhan aplikasi kita
   3. Saat memodelkan data menggunakan relational database, biasanya kita mengacu ke database normalization
   4. Saat memodelkan data menggunakan document database, kita harus mengacu ke penggunaan aplikasi dalam melakukan query, update dan memproses data
2. Schema Yang Fleksibel
   1. Tidak seperti di relation database, di MongoDB kita bisa memasukan data ke collection secara langsung tanpa mendefinisikan schema collection-nya
   2. Schema untuk collection di MongoDB sangat fleksibel, tiap document bisa berbeda. Tidak seperti tabel di relational database yang harus sama tiap record
   3. Namun pada praktek-nya, sangat di rekomendasikan menggunakan jenis data yang sama untuk tiap collection, walaupun bisa berbeda - beda di collection yang sama
3. Primary Key
   1. Saat membuat document di MongoDB, kita wajib menambahkan primary key
   2. Tidak seperti relational database yang bebas membuat column untuk primary key, di MongoDB primary key wajib menggunakan field\_id
   3. Selain itu primary key tidak bisa lebih dari 1 field, hanya bisa field\_id, jadi jika kita ingin membuat composite primary key, maka kita hanya bisa melakukan dengan menggunakan 1 field\_id
4. Struktur Document - Embedded

{

\_id: <ObjectId1>,

username: “123xyz”,

contact {

phone: “123-456-789”,

email: “xyz@example.com

},

Access {

level: 5,

group: “dev”

}

}

Embedded Sub-Document

Embedded Sub-Document

1. Struktur Document - Reference

Contact Document

{

\_id: <ObjectId2>,

userId: <ObjectId1>,

phone: “123-456-789-0”,

email: “xyz@example.com”

}

User Document

{

\_id: <ObjectId1>,

username: “123xyz”

}

Access Document

{

\_id: <ObjectId3>,

userId: <ObjectId1>,

level: 5,

group: “dev”

}

1. Embedded vs Reference

|  |  |
| --- | --- |
| Gunakan Embeddded Jika | Gunakan Reference Jika |
| Antar document saling ketergantungan | Antar document bisa berdiri sendiri dan tidak terlalu ketergantungan satu sama lain |
| Tidak bisa langsung melakukan perubahan ke embedded document | Bisa melakukan manipulasi data langsung terhadap reference document |
| Embedded document selalu dibutuhkan ketika mengambil data document | Reference document tidak selalu dibutuhkan saat mengambil document |

1. BSON
   1. BSON singkatan dari Binary JSON, yaitu binary-encoded serialization document seperti JSON
   2. Sama seperti JSON, di BSON juga kita bisa menggunakan embedded object, array dan lain-lain
   3. <http://bsonspec.org/>
   4. <https://docs.mongodb.com/manual/reference/bson-types/>
2. Tipe Data di BSON

|  |  |
| --- | --- |
| Tipe Data | Alias |
| Double | Double |
| String | string |
| Object | object |
| Array | array |
| Binary Data | binData |
| ObjectId | objectid |
| Boolean | bool |
| Date | date |
| Null | null |
| Regular Expression | regex |
| Javascript | javascript |
| Javascript With Scope | javascriptWithScope |
| 32 Bit Integer | int |
| Timestamp | timestamp |
| ui64 Bit Integer | long |
| Decimal 128 | decimal |
| Min Key | minKey |
| Max Key | maxKey |

1. ObjectId
   1. ObjectId adalah random data yang unik, cepat di generate dan terurut
   2. Nilai ObjectId memiliki ukuran panjang 12 byte, konsisten terdiri dari informasi 4 byte timestamp, 5 byte random value, dan 3 byte incrementing counter
   3. ObjectId digunakan sebagai default\_id(primary key) di document jika kita tidak secara eksplisit menyebutkan \_id document nya
2. Date dan ISODate
   1. BSON Date adalah 64 bit integer yang mempresentasikan angka milisecond sejak Unix epoch(1 januari 1970)
   2. NIlai ini bisa mempresentasikan waktu dengan jarak 290 juta tahun sebelum dan setelah unix epoch
   3. ISODate merupakan representasi waktu yang digunakan oleh MongoDB
   4. Date ini kompatibel dengan Date di Javascript
   5. <https://developer.mozila.org/en-US/docs/Web/Javascript/Reference/GlobalObjects/Date>
3. Insert Document
   1. Untuk menyimpan data ke MongoDB, kita perlu membuat document dalam bentuk JSON
   2. Field\_id tidak wajib dimasukan, jika kita tidak memasukan field\_id, maka secara otomatis MongoDB akan membuat \_id baru secara random dengan tipe data ObjectId
   3. Atau kita juga bisa secara eksplisit membuat ObjectId baru dengan menggunakan perintah “new ObjectId()”
4. Insert Document Function

|  |  |
| --- | --- |
| Function | Keteranagan |
| db.<collection>.insertOne(document) | Menambah document ke collection |
| db.<collection>.insertMany(array<document>) | Menambah semua document di array ke collection |
| db.<collection>.insert(document / array) | Menambah satu document atau banyak document |

1. Query Document
   1. Sama seperti relation database, di MongoDB pun kita bisa melakukan query atau pencarian document yang sudah kita simpan di collection
2. Query Document Function

|  |  |
| --- | --- |
| Function | Keteranagan |
| db.<collection>.find(query) | Mencari document secara spesifik dengan query |
| db.<collection>.find() | Mencari seluruh document |

1. Sd
2. Csd
3. C
4. Sdc
5. Dsc
6. Sd
7. Cds
8. C
9. Sd
10. cs