

1) Los conceptos de RDBMS que existen en MongoDB es la Base de Datos, los demás conceptos tienen otra alternativa :

Tabla/Relación -> Colección

Fila/Tupla -> Documento

Columna -> Campo

-- Puede ser win s y copiar lo de teoria

2) Primero hay que aclarar la organización conceptual de MongoDB, su estructura se presenta con

0 o mas base de datos las cuales tienen 0 o más colecciones que tambien pueden tener 0 o mas documentos y estos tienen 0 o mas campos.

MongoDB trabaja de forma atomica sobre cada uno de los documentos, ofreciendoles a ellos el modelo transaccional.

El alcance de una transacción en MongoDB es básicamente disponible,estado suave y eventualmente consistente().

3) MongoDB permite soportar indices por cualquier campo o subcampo del documento, los tipos de indices son :

-

Single

field

-

Compound

index

-

Multikey

index

-

Geospatial

index

-

Text

index

-

Hashed

index

4) No, se reemplaza la clave foranea por una referencia .

5)

```
> db.nurses.insertOne(
  {name:"Morella Crespo", experience:9})
< { acknowledged: true,
  insertedId: ObjectId("62a7cc7cd8e6cf9bd574657f") }
> db.nurses.find()
< { _id: ObjectId("62a7cc7cd8e6cf9bd574657f"),
  name: 'Morella Crespo',
  experience: 9 }
> db.nurses.find().pretty()
< { _id: ObjectId("62a7cc7cd8e6cf9bd574657f"),
  name: 'Morella Crespo',
  experience: 9 }
```

6)

```
> db.nurses.insertMany(
  [
    { "name" : "Gale Molina", "experience" : 8, "vaccines" : ["AZ", "Moderna"] },
    { "name" : "Honoría Fernández", "experience" : 5, "vaccines" : ["Pfizer", "Moderna", "Sputnik V"] },
    { "name" : "Gonzalo Gallardo", "experience" : 3 },
    { "name" : "Altea Parra", "experience" : 6, "vaccines" : ["Pfizer"] }
  ])
< { acknowledged: true,
  insertedIds:
    { '0': ObjectId("62a7cda0d8e6cf9bd5746580"),
      '1': ObjectId("62a7cda0d8e6cf9bd5746581"),
      '2': ObjectId("62a7cda0d8e6cf9bd5746582"),
      '3': ObjectId("62a7cda0d8e6cf9bd5746583") } }
> db.nurses.find()
< { _id: ObjectId("62a7cc7cd8e6cf9bd574657f"),
  name: 'Morella Crespo',
  experience: 9 }
{ _id: ObjectId("62a7cda0d8e6cf9bd5746580"),
  name: 'Gale Molina',
  experience: 8,
  vaccines: [ 'AZ', 'Moderna' ] }
{ _id: ObjectId("62a7cda0d8e6cf9bd5746581"),
  name: 'Honoría Fernández',
  experience: 5,
  vaccines: [ 'Pfizer', 'Moderna', 'Sputnik V' ] }
{ _id: ObjectId("62a7cda0d8e6cf9bd5746582"),
  name: 'Gonzalo Gallardo',
  experience: 3 }
{ _id: ObjectId("62a7cda0d8e6cf9bd5746583"),
  name: 'Altea Parra',
  experience: 6,
  vaccines: [ 'Pfizer' ] }
```

- de 5 años de experiencia o menos

```
> db.nurses.find({ "experience" : { $gt : 5 } })
< { _id: ObjectId("62a7cc7cd8e6cf9bd574657f"),
  name: 'Morella Crespo',
  experience: 9 }
{ _id: ObjectId("62a7cdacd8e6cf9bd5746580"),
  name: 'Gale Molina',
  experience: 8,
  vaccines: [ 'AZ', 'Moderna' ] }
{ _id: ObjectId("62a7cdacd8e6cf9bd5746583"),
  name: 'Altea Parra',
  experience: 6,
  vaccines: [ 'Pfizer' ] }
```

- que hayan aplicado la vacuna “Pfizer”

```
> db.nurses.find({ vaccines : "Pfizer" })
< { _id: ObjectId("62a7cdacd8e6cf9bd5746581"),
  name: 'Honorio Fernández',
  experience: 5,
  vaccines: [ 'Pfizer', 'Moderna', 'Sputnik V' ] }
{ _id: ObjectId("62a7cdacd8e6cf9bd5746583"),
  name: 'Altea Parra',
  experience: 6,
  vaccines: [ 'Pfizer' ] }
```

- que no hayan aplicado vacunas (es decir, que el atributo vaccines esté ausente)

```
> db.nurses.find({ vaccines : {$exists: false} })
< { _id: ObjectId("62a7cc7cd8e6cf9bd574657f"),
  name: 'Morella Crespo',
  experience: 9 }
{ _id: ObjectId("62a7cdacd8e6cf9bd5746582"),
  name: 'Gonzalo Gallardo',
  experience: 3 }
```

- de apellido 'Fernández'

```
> db.nurses.find({ name : /Fernández/ })
< { _id: ObjectId("62a7cdacd8e6cf9bd5746581"),
  name: 'Honorio Fernández',
  experience: 5,
  vaccines: [ 'Pfizer', 'Moderna', 'Sputnik V' ] }
```

- con 6 o más años de experiencia y que hayan aplicado la vacuna 'Moderna'

```
> db.nurses.find( { "experience" : { $gte : 6 } , vaccines : "Moderna" })
< { _id: ObjectId("62a7cdacd8e6cf9bd5746580"),
  name: 'Gale Molina',
  experience: 8,
  vaccines: [ 'AZ', 'Moderna' ] }
```

```
> db.nurses.find( { experience : { $gte : 6 } , vaccines : "Moderna" }, { "name" : 1 , "_id" : 0 })
< { name: 'Gale Molina' }
```

7)

```
> db.nurses.updateOne( { name : "Gale Molina"}, { $set : { experience : 9}})
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
> db.nurses.find( {"name" : "Gale Molina" } )
< { _id: ObjectId("62a7cdacd8e6cf9bd5746580"),
  name: 'Gale Molina',
  experience: 9,
  vaccines: [ 'AZ', 'Moderna' ] }
```

8)

```

> db.nurses.updateOne(
  { name : "Gonzalo Gallardo" },
  { $set : { vaccines: [] } }
)
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
> db.nurses.find( { name : "Gonzalo Gallardo" } )
< { _id: ObjectId("62a7cdacd8e6cf9bd5746582"),
  name: 'Gonzalo Gallardo',
  experience: 3,
  vaccines: [] }

```

9)

```

> db.nurses.update({ name : "Altea Parra" }, { $push :{ vaccines : "AZ" } } )
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
> db.nurses.find({ name : "Altea Parra"})
< { _id: ObjectId("62a7cdacd8e6cf9bd5746583"),
  name: 'Altea Parra',
  experience: 6,
  vaccines: [ 'Pfizer', 'AZ' ] }

```

10)

```

> db.nurses.find({ vaccines : "Pfizer" })
< { _id: ObjectId("62a7cdacd8e6cf9bd5746581"),
  name: 'Honoría Fernández',
  experience: 5,
  vaccines: [ 'Pfizer', 'Moderna', 'Sputnik V' ] }
{ _id: ObjectId("62a7cdacd8e6cf9bd5746583"),
  name: 'Altea Parra',
  experience: 6,
  vaccines: [ 'Pfizer', 'AZ' ] }
> db.nurses.updateMany({ vaccines: "Pfizer" }, { $mul : { experience : 2 } })
< { acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0 }
> db.nurses.find({ vaccines : "Pfizer" })
< { _id: ObjectId("62a7cdacd8e6cf9bd5746581"),
  name: 'Honoría Fernández',
  experience: 10,
  vaccines: [ 'Pfizer', 'Moderna', 'Sputnik V' ] }
{ _id: ObjectId("62a7cdacd8e6cf9bd5746583"),
  name: 'Altea Parra',
  experience: 12,
  vaccines: [ 'Pfizer', 'AZ' ] }

```

11)

```

> db.doses.getIndexes()
< [ { v: 2, key: { _id: 1 }, name: '_id_' } ]

```

12)

```

> db.doses.createIndex({"nurse": 1})
< 'nurse_1'

```

-con índice en nurse

```

> db.doses.find( { nurse : /11/ } ). explain("executionStats")
< { explainVersion: '1',
  queryPlanner:
    { namespace: 'test.doses',
      indexFilterSet: false,
      parsedQuery: { nurse: BSONRegExp("11", "") },
      maxIndexedOrSolutionsReached: false,
      maxIndexedAndSolutionsReached: false,
      maxScansToExplodeReached: false,
      winningPlan:
        { stage: 'FETCH',
          inputStage:
            { stage: 'IXSCAN',
              filter: { nurse: BSONRegExp("11", "") },
              keyPattern: { nurse: 1 },
              indexName: 'nurse_1',
              isMultiKey: false,
              multiKeyPaths: { nurse: [] },
              isUnique: false,
              isSparse: false,
              isPartial: false,
              indexVersion: 2,
              direction: 'forward',
              indexBounds: { nurse: [ '["", {}]', '[/11/, /11/]' ] } } },
          rejectedPlans: [] },
      executionStats:
        { executionSuccess: true,
          nReturned: 14025,
          executionTimeMillis: 218,
          totalKeysExamined: 228133,
          totalDocsExamined: 14025,
          executionStages:
            { stage: 'FETCH',
              nReturned: 14025,
              executionTimeMillisEstimate: 3,
              works: 228134,

```

-sin indice en nurse

```

> db.doses.dropIndex("nurse_1")
< { nIndexesWas: 2, ok: 1 }
> db.doses.getIndexes()
< [ { v: 2, key: { _id: 1 }, name: '_id_' } ]

> db.doses.find( { nurse : /11/ } ). explain("executionStats")
< { explainVersion: '1',
  queryPlanner:
    { namespace: 'test.doses',
      indexFilterSet: false,
      parsedQuery: { nurse: BSONRegExp("11", "") },
      maxIndexedOrSolutionsReached: false,
      maxIndexedAndSolutionsReached: false,
      maxScansToExplodeReached: false,
      winningPlan:
        { stage: 'FETCH',
          inputStage:
            { stage: 'IXSCAN',
              filter: { nurse: BSONRegExp("11", "") },
              keyPattern: { nurse: 1 },
              indexName: 'nurse_1',
              isMultiKey: false,
              multiKeyPaths: { nurse: [] },
              isUnique: false,
              isSparse: false,
              isPartial: false,
              indexVersion: 2,
              direction: 'forward',
              indexBounds: { nurse: [ '["", {}]', '[/11/, /11/]' ] } } },
          rejectedPlans: [] },
      executionStats:
        { executionSuccess: true,
          nReturned: 14025,
          executionTimeMillis: 234,
          totalKeysExamined: 228133,
          totalDocsExamined: 14025,
          executionStages:
            { stage: 'FETCH',
              nReturned: 14025,
              executionTimeMillisEstimate: 17,
              works: 228134,

```


