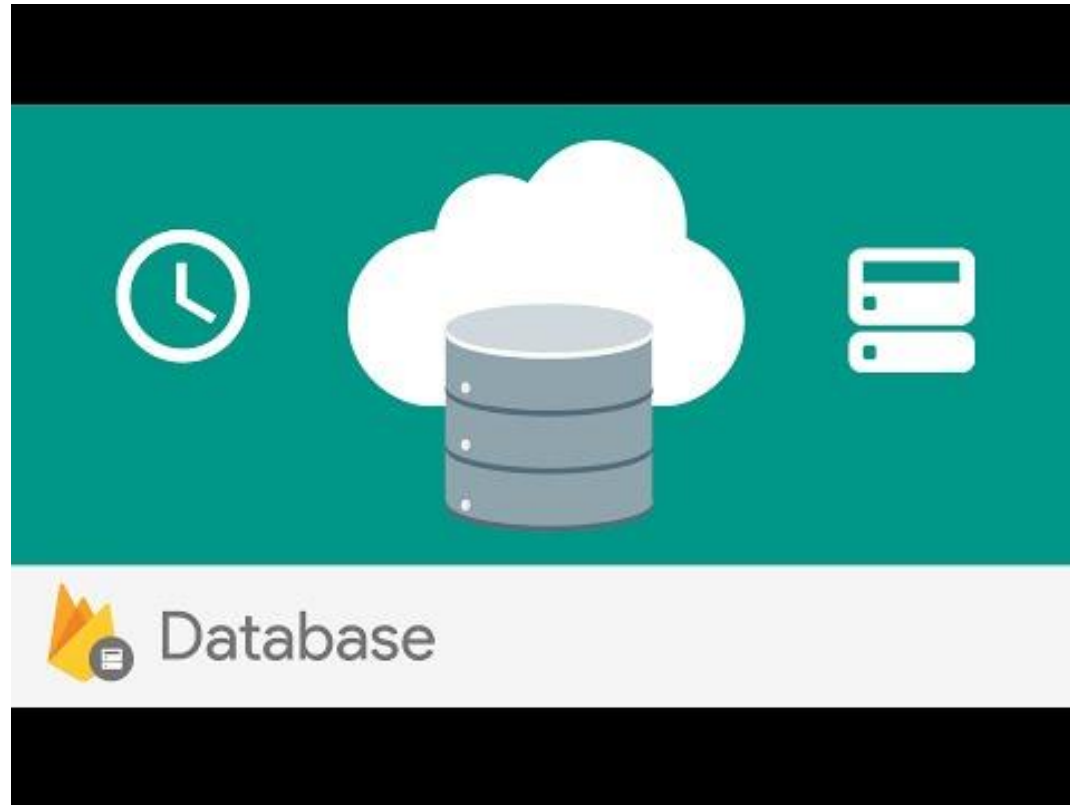




firebase





Not Only SQL

All in the NoSQL Family

NoSQL databases are geared toward managing large sets of varied and frequently updated data, often in distributed systems or the cloud. They avoid the rigid schemas associated with relational databases. But the architectures themselves vary and are separated into four primary classifications, although types are blending over time.



Document databases

Store data elements in document-like structures that encode information in formats such as JSON.



Common uses include content management and monitoring Web and mobile applications.



EXAMPLES:

Couchbase Server, CouchDB, MarkLogic, MongoDB



Graph databases

Emphasize connections between data elements, storing related "nodes" in graphs to accelerate querying.



Common uses include recommendation engines and geospatial applications.



EXAMPLES:

Allegrograph, IBM Graph, Neo4j



Key-value databases

Use a simple data model that pairs a unique key and its associated value in storing data elements.



Common uses include storing clickstream data and application logs.



EXAMPLES:

Aerospike, DynamoDB, Redis, Riak



Wide column stores

Also called table-style databases—store data across tables that can have very large numbers of columns.



Common uses include Internet search and other large-scale Web applications.



EXAMPLES:

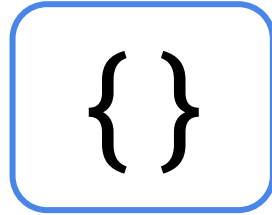
Accumulo, Cassandra, HBase, Hypertable, SimpleDB

FIREBASE implementa una base de datos basada en documentos bajo el formato de JSON.

Básicamente Firebase es una enorme bola de



Base de datos inicial de Firebase



Práctica

The logo for 'The Simpsons' is displayed in a bright yellow, hand-drawn, bubble-style font. The word 'the' is smaller and positioned above the word 'SIMPSONS'. The entire logo is set against a light gray, rounded rectangular background.



Profesora

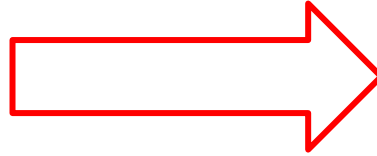
Edna krabappel

44 años

Le dicen Edna

Altura: 1.64

JSON



```
{  
  "nombre": "Edna Krabapple",  
  "edad": "44",  
  "apodo": "Edna",  
  "altura": "1,64"  
}
```



Profesora

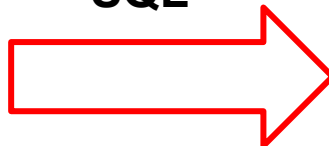
Edna Krabappel
44 años
Altura: 1.64



Profesor

Dewey Largo
46 años
Altura: 1.80

SQL

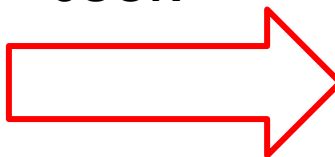


PROFESORES

ID	NOMBRE	EDAD	ALTURA
1	Edna Krabappel	44	1.64
2	Dewey Largo	46	1.80

```
{  
  "profesores":  
    [{  
      "nombre": "Edna  
Krababple",  
      "edad": "44",  
      "altura": "1,64"  
    },  
    {  
      "nombre": "Dewey Largo",  
      "edad": "46",  
      "altura": "1,80"  
    }  
  ]  
}
```

JSON





Profesora

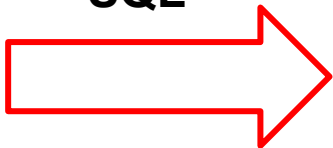
Edna krabappel

44 años

Altura: 1.64

Apodo: Edna

SQL



PROFESORES

ID	NOMBRE	EDAD	ALTURA	Apodo
1	Edna Krabappel	44	1.64	Edna
2	Dewey Largo	46	1.80	null

{

"profesores":

[{

"nombre": "Edna
Krabapple",

"edad": "44",

"altura": "1,64",

"apodo": "Edna"

},

{

"nombre": "Dewey Largo",

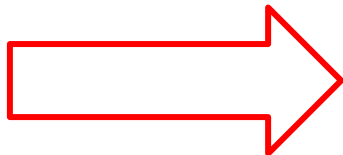
"edad": "46",

"altura": "1,80"

}]

}

JSON



¿Qué conclusión podemos sacar?



A las bases de datos en forma de documentos se le pueden agregar atributos a una sola entidad sin modificar los demás objetos.

No hay una tabla predefinida (una estructura).

A esto se lo denomina schemaless.



Profesora

Edna krabappel

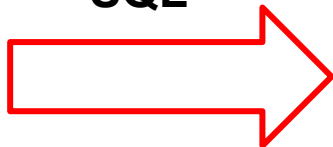
44 años

Altura: 1.64

Apodos:

Edna,edi, ed

SQL



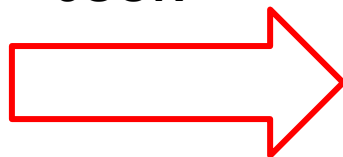
PROFESORES

ID	NOMBRE	EDAD	ALTURA	Apodo
1	Edna Krabappel	44	1.64	Edna,edi,ed
2	Dewey Largo	46	1.80	Null

"profesores":

```
[{
  "nombre": "Dewey Largo",
  "edad": "46",
  "altura": "1,80"
},
{
  "nombre": "Edna
Krabapple",
  "edad": "44",
  "altura": "1,64",
  "apodo": ["Edna", "edi", "ed"]
}]
}
```

JSON



Modelar el siguiente contexto en una base de datos basada en Documentos

Profesora

Edna krabappel
44 años
Altura: 1.64
Apodo: Edna



Profesor

Dewey Largo
46 años
Altura: 1.80



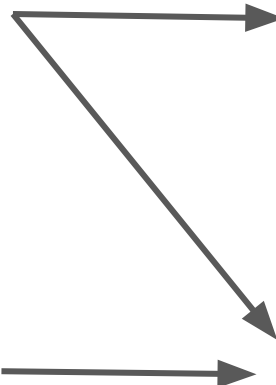
Curso

Android
Duracion 210hs



Curso

iOS
Duracion 200hs



Solución 1

Agregar los cursos como atributo al profesor

```
{
  "profesores":
  [{
    "nombre": "Edna Krabapple",
    ...
    "cursos": [
      {"nombre": "android",
        "duracion": "210"},
      {"nombre": "iOS",
        "duracion": "200"}
    ]
  },
  {
    "nombre": "Dewey Largo",
    ...
    "cursos": []
  }
]
```


Otras soluciones

Solución 2

Agregar los cursos y poner a los docentes como atributos.

Solución 3

Poner ids a ambas entidades y agregar un objeto que represente las relaciones

Solución 4

Juntar la 1 con la 2.

De todas las soluciones.. ¿Cuál conviene?



Colecciones en Firebase

Posición	0	1	2	3	4	5
Valor	Juan	Miguel	Maria	Marta	Mariano	Cesar

El
cuarto!!...está
despedido!!



The fourth
one...YOU
ARE
FIRED!!!



Colecciones en Firebase

Posición	0	1	2	3	4
Valor	Juan	Miguel	Maria	Marta	Cesar

El
cuarto!!...está
despedido!!



The fourth
one...YOU
ARE
FIRED!!!



Colecciones en Firebase

Posición	0	1	2	3
Valor	Juan	Miguel	Maria	Marta

El
cuarto!!...está
despedido!!



The fourth
one...YOU
ARE
FIRED!!!



¿Qué pasó? ¿Cómo lo resuelvo?

