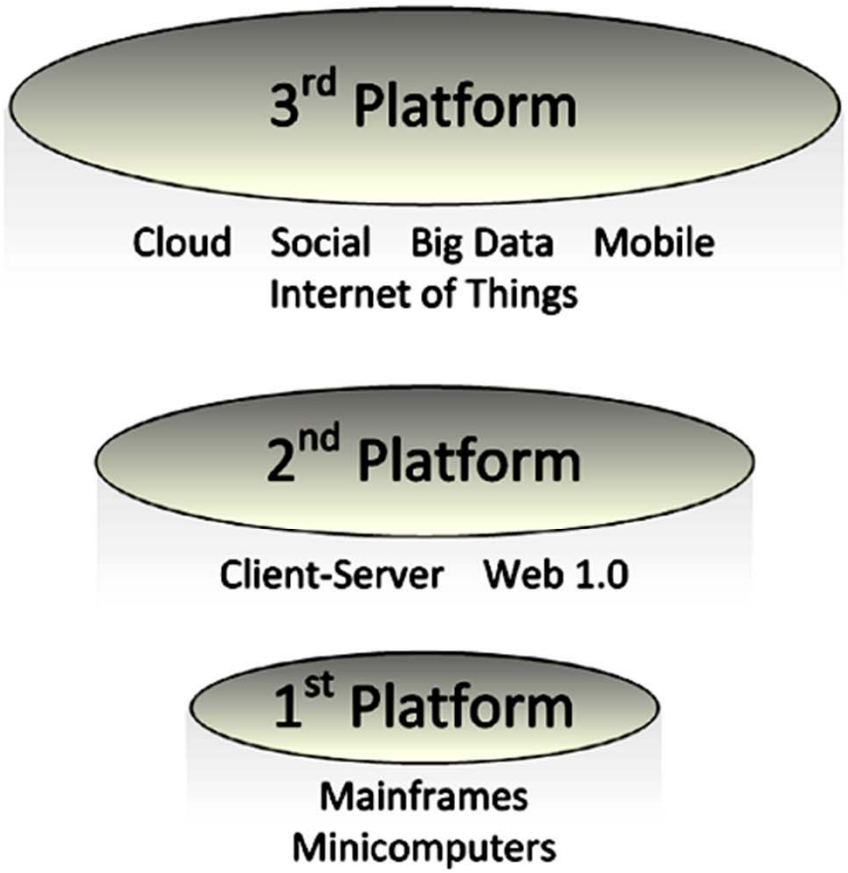
Nosql Database

عادل قاضی خانی
هیئت علمی گروه کامپیوتر دانشگاه امام رضا (ع)

تعریف

- **نوعی پایگاه داده** که از مفاهیم **پایگاه داده های رابطه ای** استفاده نمی کند

نسلهای پایگاه داده ها



Relational Database
NoSQL
NewSQL
Big Data platforms

Relational Database

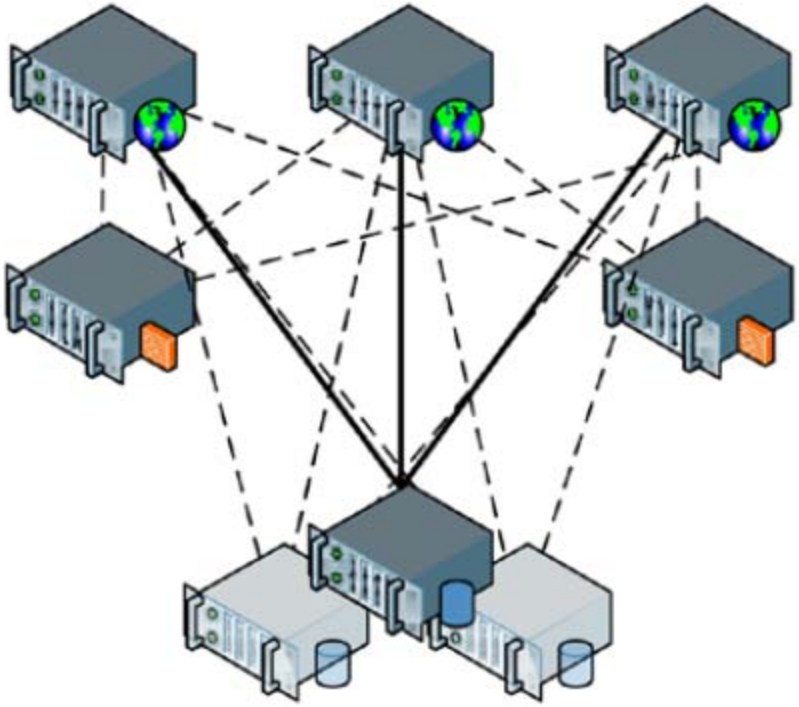
Hierarchical Database
Network Database
ISAM files

تاریخچه

- در دهه ۱۹۹۰ وب با **صفحات ثابت** مرتبط شروع شد (HTML)
- در ادامه **سایتهای پویا** مبتنی بر پایگاه داده غالب شد (CGI,ASP.NET,PHP,J2EE,...)
- **با بزرگتر شدن** برنامه‌ها وب، مقیاس پذیری مهم شد

مقیاس پذیری

Single database & single web server



Scaling up by adding 2 web servers, 2 Memcached servers, and 2 read-only database slaves

Read only traffic — Read/Write traffic

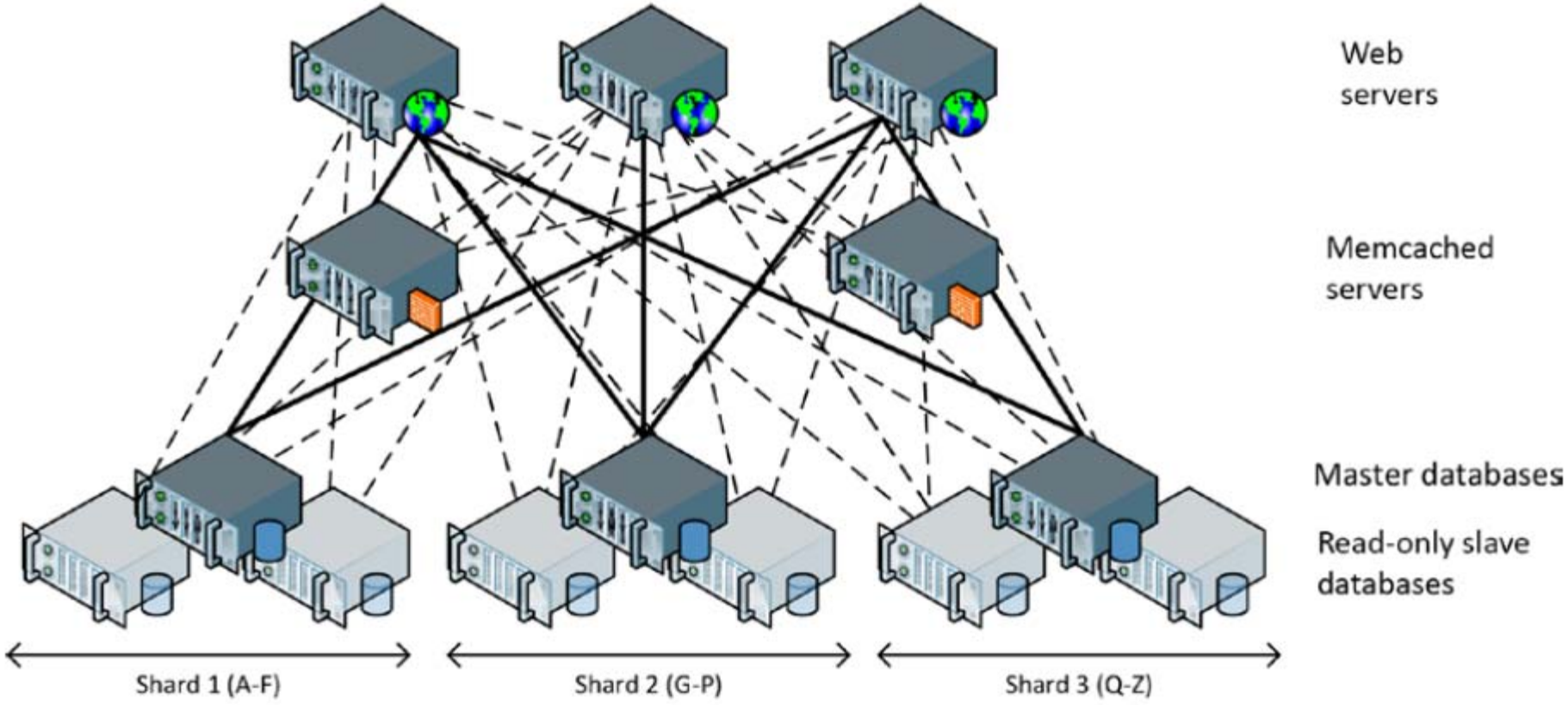
مقیاس پذیری

- **Memchached server**
  - تکنولوژی که خواندن داده‌ها از پایگاه داده را کاهش می دهد.
- **Replication**
  - ایجاد کپی از داده ها برای مقیاس‌پذیر کردن خواندن داده ها
  - این دو تکنولوژی **دسترسی خواندن را بهبود می دهد نه نوشتن**

مقیاس پذیری

- **Sharding**
- تکنولوژی برای توزیع فیزیکی پایگاه داده بر روی چند سرویس دهنده
- به هر بخش پایگاه داده shard گفته می شود
- بر اساس کلید مقدار
- نرم‌افزاری که به پایگاه داده متصل می‌شود باید بداند داده ها در کدام سرویس دهنده قرار دارند

مقیاس پذیری



Scaling with 3 web servers, 2 Memcached servers, and 3 database shards with 2 read-only database slaves per shard

--read-only traffic --read/write traffic

دلایل گرایش به Nosql

- **مقیاس‌پذیری**: حجم بالای داده‌ها و تراکنش‌ها ما را نیازمند به مقیاس‌پذیری می‌کند
- **هزینه**: پایگاه داده‌های رابطه‌ای اکثرا متن باز نیستند
- **انعطاف پذیری**: ...
- **سرعت**:...
- **دسترس‌پذیری**: ...

دلایل گرایش به Nosql

- **انعطاف پذیری:**
- در پایگاه داده رابطه‌ای ابتدا باید داده‌ها مدل شوند و سپس برنامه‌نویس مطابق با آن برنامه خود را بنویسد. در برخی کاربردها برنامه‌نویس می‌خواهد بتواند در شرایط خاص تغییراتی در داده‌ها ایجاد کند که در پایگاه داده رابطه‌ای کار هزینه‌بری است.

دلایل گرایش به Nosql

- **سرعت:**
- در پایگاه داده رابطه‌ای جامعیت موضوع مهمی است که داده‌های ما همیشه درست باشند. این مسئله مستلزم اجرای قوانینی توسط پایگاه داده است و سبب کند شدن پایگاه داده می‌شود.
- در برخی کاربردها این محدودیت‌ها برای ما مهم نیست بنابراین می‌توانیم سرعت بالاتری داشته باشیم

دلایل گرایش به Nosql

- **سرعت:**
- در پایگاه داده رابطه‌ای بدلیل مدلسازی خاص داده‌ها و مفهوم رابطه نرمال نیازمند عملیات زمانبری مثل پیوند هستیم.
- اگر داده‌های مرتبط به شکلی کنار هم قرار گیرند عملیات پیوند در پایگاه داده کاهش و سرعت افزایش می‌یابد

دلایل گرایش به Nosql

- **دسترس پذیری**: یکی از مهم‌ترین مسائل در مورد نرم افزارها بویژه در محیط وب در دسترس بودن همیشگی است.
- این مسئله با نصب پایگاه داده بر روی یک کلاستر امکان‌پذیر است.
- پایگاه داده رابطه‌ای برای نصب در کلاستر امکانات مختلفی دارد (تهیه نسخه‌ها کپی از داده‌ها و ...) اما این مسئله توام با پیچیدگی هایی است.

تولد Nosql

- در سال ۲۰۰۶ شرکت گوگل در مقاله ای پایگاه داده توزیع شده **Bigtable** را معرفی کرد
- Google said :“Bigtable is a distributed storage system for managing structured data that is designed to scale to a very large size: petabytes of data across thousands of commodity servers.”

تولد Nosql

- در سال ۲۰۰۷ آمازون **Dynamo** را بعنوان مدل ذخیره‌سازی خود معرفی کرد.
- Amazon said: “Dynamo is used to manage the state of services that have very high reliability requirements and need tight control over the tradeoffs between availability, consistency, cost-effectiveness and performance.”

انواع پایگاه داده Nosql

- **چهار نوع اصلی:**
- **مبتنی بر کلید مقدار**
- **مبتنی بر سند**
- **مبتنی بر ستون**
- **مبتنی بر گراف**

پایگاه داده Nosql، کلید مقدار

- ساده‌ترین مدل Nosql

- در این پایگاه داده فرمت داده‌ها کلید مقدار است

- **کلید مقدار؟**

پایگاه داده Nosql، کلید مقدار

- **کلید مقدار،** ساختار داده‌ای است که توسعه یافته آرایه است
- در آرایه یک لیست مرتب از مقادیر داریم. اندیس عدد صحیح است و مقادیر از یک نوع.

1	True
2	True
3	False
4	True
5	False
6	False
7	False
8	True
9	False
10	True

```
exampleArray[1] = 'Goodbye world.'
exampleArray[2] = 'This is a test.'
exampleArray[5] = 'Key-value database'
exampleArray[9] = 'Elements can be set in any order.'
```

پایگاه داده Nosql، کلید مقدار

- **کلید مقدار،** در واقع یک آرایه انجمنی است.
- در آرایه انجمنی اندیس فقط عدد صحیح نیست و مقادیر می‌تواند متفاوت باشد.

```
exampleAssociativeArray['Pi'] = 3.1415
exampleAssociativeArray['CapitalFrance'] = 'Paris'
exampleAssociativeArray['ToDoList'] = { 'Alice' : 'run
  reports; meeting with Bob', 'Bob' : 'order inventory;
  meeting with Alice' }
exampleAssociativeArray[17234] = 34468
```

پایگاه داده Nosql، کلید مقدار

- **کلید:** داده‌ای است برای ارجاع به داده دیگر که همان مقدار است.
- **مثلاً** آدرس یک ساختمان. خیابان کوهسنگی ۱۰، پلاک ۲۰ شبیه یک کلید است برای یک ساختمان.

پایگاه داده Nosql، کلید مقدار

- **کلید،** می‌تواند انواع مختلف باشد که وابسته به پایگاه داده است.
- **مثلاً** در پایگاه داده Redis نسخه 2.8.13 انواع کلیدها عبارتند از:
- رشته
- لیست
- مجموعه
- مجموعه مرتب
- هش
- آرایه بیتی

پایگاه داده Nosql، کلید مقدار

- **نحوه ساخت کلید؟**

Entity Name + ‘:’ + Entity Identifier +’:’ + Entity Attribute

- مثال: نام مشتری

Cust:12387:firstName

پایگاه داده Nosql، کلید مقدار

- **مقدار،** داده ای است که منتسب به یک کلید است و می‌تواند انواع متنوعی از داده‌ها از قبیل عدد، رشته، JSON، تصویر و ... باشد

پایگاه داده Nosql، کلید مقدار

- **مثال از مقدار،**
- آدرس مشتری

```
‘1232 NE River Ave, St. Louis, MO’

(‘1232 NE River Ave’, ‘St. Louis’, ‘MO’)

{ ‘Street:’ : ‘1232 NE River Ave’, ‘City’ : ‘St. Louis’,:
  ‘State’ : ‘MO’ }
```

پایگاه داده Nosql، کلید مقدار

- **چگونه یک مقدار از کلید تعیین می شود؟**
- استفاده از تابع درهم سازی

Key	Hash Value
customer:1982737: firstName	e135e850b892348a4e516cfcb385eba3bfb6d209
customer:1982737: lastName	f584667c5938571996379f256b8c82d2f5e0f62f
customer:1982737: shippingAddress	d891f26dcdb3136ea76092b1a70bc324c424ae1e
customer:1982737: shippingCity	33522192da50ea66bfc05b74d1315778b6369ec5
customer:1982737: shippingState	239ba0b4c437368ef2b16ecf58c62b5e6409722f
customer:1982737: shippingZip	814f3b2281e49941e1e7a03b223da28a8e0762ff

پایگاه داده Nosql، کلید مقدار

- **فضای نام (namespace):** مجموعه ای از کلید مقدارها که در آنها کلید تکراری وجود ندارد.

Database

Bucket 1	
‘Foo1’	‘Bar’
‘Foo2’	‘Bar2’
‘Foo3’	‘Bar7’

Bucket 2	
‘Foo1’	‘Baz’
‘Foo4’	‘Baz3’
‘Foo6’	‘Baz2’

Bucket 3	
‘Foo1’	‘Bar7’
‘Foo4’	‘Baz3’
‘Foo7’	‘Baz9’

پایگاه داده Nosql، کلید مقدار

- چگونه می توانیم یک پایگاه داده رابطه ای را به کلید مقدار تبدیل کنیم؟
- مثلا جدولی برای مشتریان داریم ۷ صفت. یکی از آنها شماره مشتری است. ۶ صفت دیگر در شکل زیر مشخص است.

```
customer:1982737:firstName
customer:1982737:lastName
customer:1982737:shippingAddress
customer:1982737:shippingCity
customer:1982737:shippingState
customer:1982737:shippingZip
```

پایگاه داده Nosql، کلید مقدار

- **مزایا:**

❖سادگی

❖سرعت

❖مقیاس پذیری

پایگاه داده Nosql، کلید مقدار

- **مزایا:**

❖**سادگی**

- ذخیره سازی حداقل داده های مورد نیاز.
- عدم نیاز به طراحی و تغییر مدل داده.
- نوع داده صفات می تواند متغیر باشد

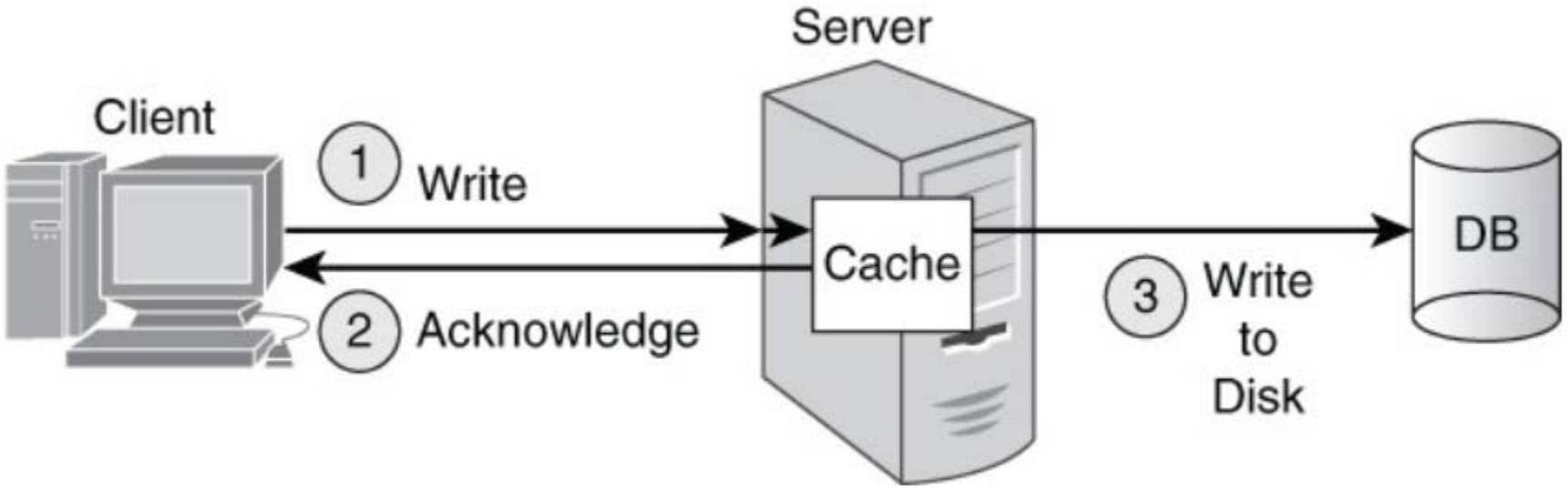
```
shoppingCart[cart:1298:customerID] = 1982737
shoppingCart[cart:3985:customerID] = 'Johnson, Louise'
```

پایگاه داده Nosql، کلید مقدار

- **مزایا:**

❖**سرعت**

❑استفاده از حافظه کش



پایگاه داده Nosql، کلید مقدار

- **عیب:**

❖**زبانی برای جستجو داده ها نداریم (SQL)**

❖تمام عملیات جستجو توسط برنامه نویس باید طراحی شود

پایگاه داده Nosql، کلید مقدار

- **چند نمونه:**

- Riak
- Redis

پایگاه داده Nosql، مبتنی بر سند

- در این نوع پایگاه داده، داده ها در اسناد ذخیره می‌شوند.

- **سند؟**

پایگاه داده Nosql، مبتنی بر سند

- **سند:**
- **مجموعه ای** از کلید مقدارهای مرتب

- **مثال برای مجموعه بودن کلید مقدارها**
- { ‘foo’ : ‘a’, ‘bar’ : ‘b’, ‘baz’ : ‘c’} **(مجموعه هست)**
- {‘foo’ : ‘a’, ‘bar’ : ‘b’, ‘baz’ : ‘c’, ‘foo’ : ‘a’} **(مجموعه نیست)**

پایگاه داده Nosql، مبتنی بر سند

- **مثال سند:**

- { ‘employeeName’ : ‘Janice Collins’,
  ‘department’ : ‘Software engineering’
  ‘startDate’ : ‘10-Feb-2010’,
  ‘pastProjectCodes’ : [189847, 187731, 176533, 154812]
  }

پایگاه داده Nosql، مبتنی بر سند

- **مثال سند:**

- { ‘employeeName’ : ‘Janice Collins’,
  ‘department’ : ‘Software engineering’,
  ‘startDate’ : ‘10-Feb-2010’,
  ‘pastProjects’ : {
  {‘projectCode’ : 189847,
  ‘projectName’ : ‘Product Recommendation System’,
  ‘projectManager’ : ‘Jennifer Delwiney’ },
  {‘projectCode’ : 187731,
  ‘projectName’ : ‘Finance Data Mart version 3’,
  ‘projectManager’ : ‘James Ross’},
  {‘projectCode’: 176533,
  ‘projectName’ : ‘Customer Authentication’,
  ‘projectManager’ : ‘Nick Clacksworth’},
  {‘projectCode’: 154812,
  ‘projectName’ : ‘Monthly Sales Report’,
  ‘projectManager’: ‘Bonnie Rendell’}
  }

پایگاه داده Nosql، مبتنی بر سند

- **دو مزیت نسبت به پایگاه داده کلید مقدار:**

❖**صفات مرتبط در کنار هم قرار دارند**(مشابه صفات جدول در پایگاه داده رابطه ای)

❖ { ‘employeeName’ : ‘Janice Collins’,
‘department’ : ‘Software engineering’
‘startDate’ : ‘10-Feb-2010’,
‘pastProjectCodes’ : [189847, 187731, 176533, 154812]
}

پایگاه داده Nosql، مبتنی بر سند

- **دو مزیت نسبت به پایگاه داده کلید مقدار:**

❖**اسناد مرتبط به هم در یک مجموعه (Collection) کنار هم قرار می گیرند** (مشابه سطرهای جدول در پایگاه داده رابطه ای، اما قویتر)

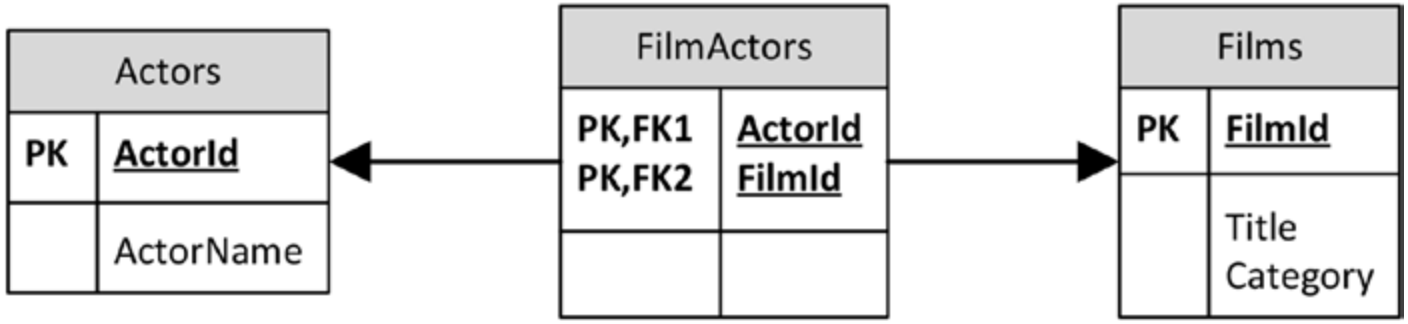
```
{
   {
    “customer_id”:187693,
    “name”: “Kiera Brown”
    “address” : {
                “street” : “1232 Sandy Blvd.”,
                “city” :  “Vancouver”,
                “state” :  “WA”,
                “zip” :  “99121”


                },
      “first_order” : “01/15/2013”,
      “last_order” : ” 06/27/2014”
   }
```

```
 {
{
 “customer_id”:187694,
 “name”: “Bob Brown”,
 “address” : {
             “street” : “1232 Sandy Blvd.”,
             “city” :  “Vancouver”,
             “state” :  “WA”,
             “zip” :  “99121”
             },
“first_order” : “02/25/2013”,
“last_order” : ” 05/12/2014”
}
 }
```

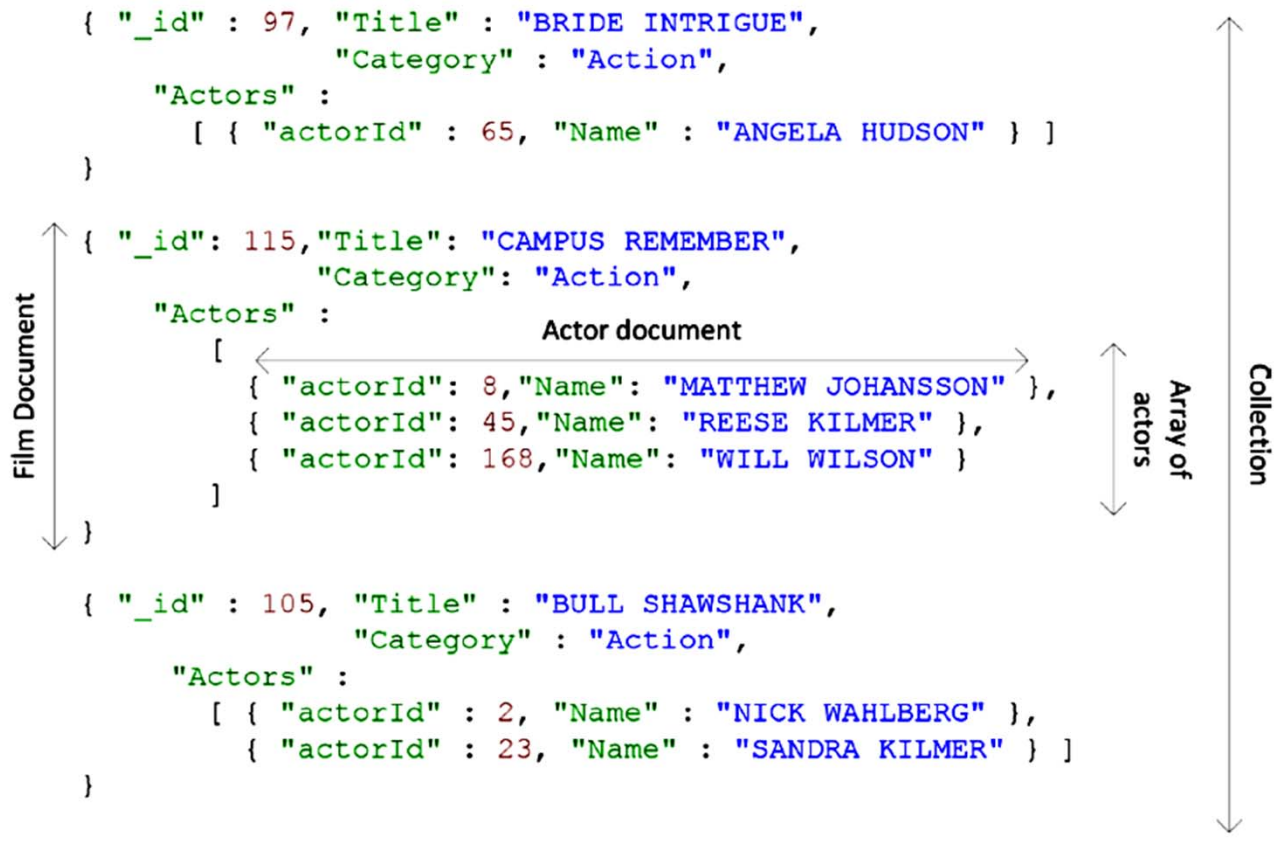
پایگاه داده Nosql، مبتنی بر سند

- **مثال در پایگاه داده رابطه ای و مبتنی بر سند:**
- **رابطه ای**



پایگاه داده Nosql، مبتنی بر سند

- **مثال در پایگاه داده رابطه ای و مبتنی بر سند:**
- **مبتنی بر سند**



پایگاه داده Nosql، مبتنی بر سند

- **مزایا:**

❖انعطاف‌پذیری (عدم نیاز به طراحی مدل داده)

❖مقیاس‌پذیری

❖عملیات ساده تر نسبت به پایگاه داده کلید مقدار (بدلیل داشتن امکاناتی شبیه به پایگاه داده رابطه ای)

پایگاه داده Nosql، مبتنی بر سند

- **چند نمونه:**

- eXist (XML)
- MarkLogic (XML)
- CouchDB (JSON)
- MongoDB (JSON)

پایگاه داده Nosql، مبتنی بر ستون

پایگاه داده Nosql، مبتنی بر گراف

پایگاه داده مبتنی بر سند

- در این نوع پایگاه داده داده ها بصورت اسناد ساختیافته معمولاً XML یا JSON ذخیره می شوند
- مختار هستند که محدودیتهای پایگاه داده رابطه ای را اعمال کنند یا نه

پایگاه داده مبتنی بر سند

- **پایگاه داده XML**
- بر اساس استاندارد سند XML
- قبل از JSON
- استانداردی برای انتقال داده ها

XML

- **ابزارها و استانداردها**
- XML شامل مجموعه ای از ابزار و استانداردها برای تولید، اعتبارستجی، جستجو و تبدیل اسناد XML هستند شامل:
- Xpath
- Xquery
- XML schema
- XSLT
- DOM

- XPath: ابزاری برای دسترسی به بخشهای مختلف یک سند XML

XML

- **ابزارها و استانداردها**
- **XPath:** ابزاری برای دسترسی به بخشهای مختلف یک سند XML
- **XQuery:** یک زبان پرس و جو برای سند XML
- **XML schema:** برای اعتبارسنجی سند XML
- **XSLT:** برای تبدیل سند XML به یک فرمت دیگر
- **DOM** : یک API برای تعامل با XML

پایگاه داده XML

- **مثالی از XQuery**

query

```
1  xquery version "3.0";
2  collection("/db/apps/demo/data")//address[contains(city, "Berlin")]
3
```

/db/query

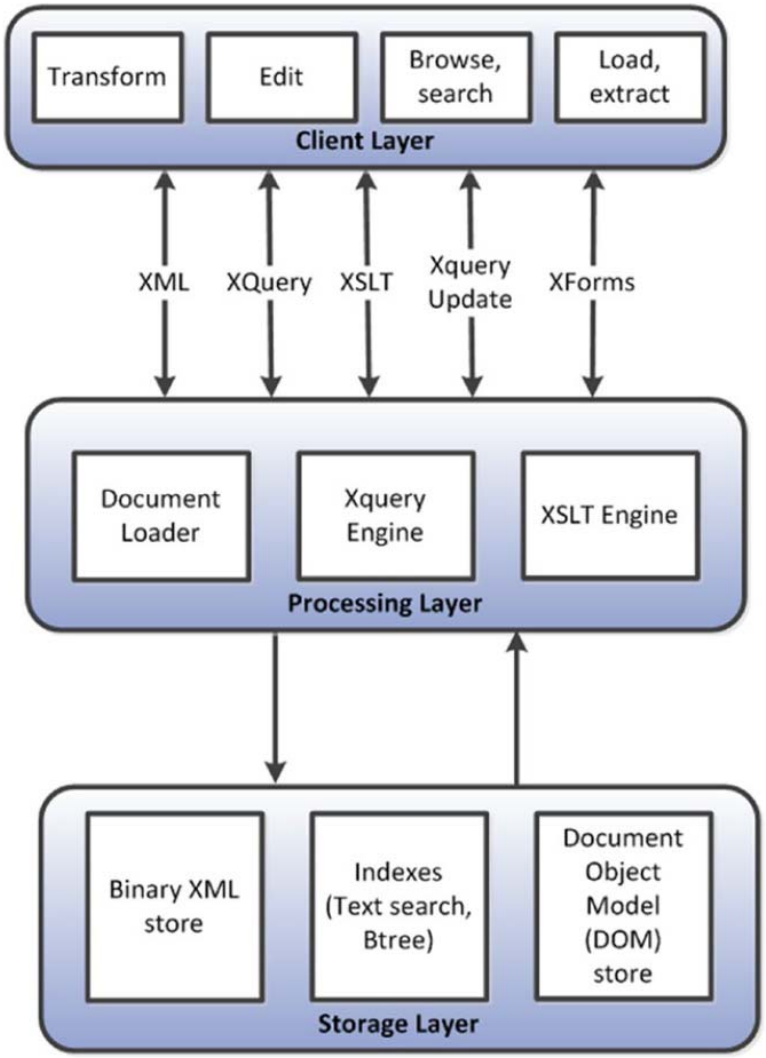
XML Output ▾ Live Preview

```
1  <address id="0ff8612a-b998-4677-84a3-73e9ef84ba5f">
       <name>Biene Maja</name>
       <street>Wiesenweg 33</street>
       <city>Berlin</city>
   </address>
```

پایگاه داده XML

- **معماری پایگاه داده**

- **دو نمونه:**
- **eXist**
- **MarkLogic**



پایگاه داده XML

- این پایگاه داده بیشتر برای مدیریت محتوی ، سازماندهی و نگهداری مجموعه ای از فایلهای متنی در فرمت XML

پایگاه داده مبتنی بر سند

- **پایگاه داده JSON**
- در XML مصرف فضای زیاد و پردازش سنگین
- *JavaScript Object Notation(JSON)*

JSON

- در ابتدا توسط Douglas Crockford برای ایجاد برنامه های وب قویتر ارائه شد و جایگزین XML در برنامه نویسی AJAX

مقایسه ظاهر JSON وXML

films.json

```
{
    "Category": "Documentary",
    "Description": "A Epic Drama of
    "Length": "86",
    "Rating": "PG",
    "Rental Duration": "6",
    "Replacement Cost": "20.99",
    "Special Features": "Deleted Sce
    "Title": "ACADEMY DINOSAUR",
    "_id": 1,
    "Actors":
    [ {
            "First name": "PENELOPE"
            "Last name": "GUINESS",
            "actorId": 1
        },{
            "First name": "CHRISTIAN
            "Last name": "GABLE",
            "actorId": 10
        },{
            "First name": "LUCILLE",
            "Last name": "TRACY",
            "actorId": 20
        },{
            "First name": "SANDRA",
            "Last name": "PECK",
            "actorId": 30
        },{
            "First name": "JOHNNY",
```

films.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
    <Category>Documentary</Category>
    <Description>A Epic Drama of a Femi
    <Length>86</Length>
    <Rating>PG</Rating>
    <Rental Duration>6</Rental Duration
    <Replacement Cost>20.99</Replacemen
    <Special Features>Deleted Scenes,Be
    <Title>ACADEMY DINOSAUR</Title>
    <_id>1</_id>
    <Actors>
        <First name>PENELOPE</First nam
        <Last name>GUINESS</Last name>
        <actorId>1</actorId>
    </Actors>
    <Actors>
        <First name>CHRISTIAN</First na
        <Last name>GABLE</Last name>
        <actorId>10</actorId>
    </Actors>
    <Actors>
        <First name>LUCILLE</First name
        <Last name>TRACY</Last name>
        <actorId>20</actorId>
    </Actors>
    <Actors>
        <First name>SANDRA</First name>
        <Last name>PECK</Last name>
```

Design | Source

پایگاه داده JSON

- در این نوع پایگاه داده مهمترین مسئله ذخیره داده ها در فرمت JSON است
- دو عنصر مهم در JSON
- ***document***
- ***collection***

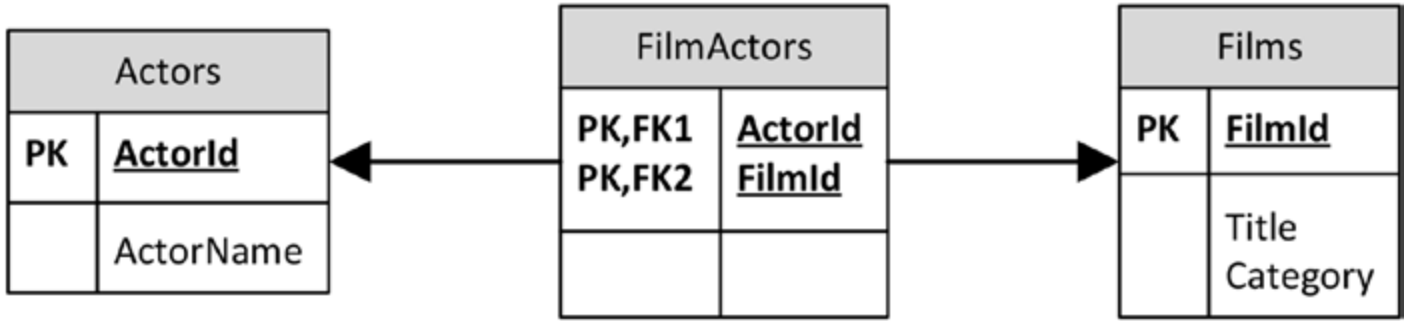
پایگاه داده JSON

- ***document :***
- واحد پایه ای ذخیره سازی، شامل یک یا چند کلید – مقدار ، همچنین می تواند شامل اسناد و آرایه های تو در تو (سطر جدول در RDBMS)
- ***collection:***
- مجموعه ای از اسناد که یک هدف مشترک دارند (جدول در RDBMS)

پایگاه داده JSON

- **مثال: پایگاه داده فیلم**

- **در RDBMS**



پایگاه داده JSON

- **مثال: پایگاه داده فیلم**
- **در JSONDB**

```
{ "_id" : 97, "Title" : "BRIDE INTRIGUE",
              "Category" : "Action",
    "Actors" :
      [ { "actorId" : 65, "Name" : "ANGELA HUDSON" } ]
}

{ "_id": 115,"Title": "CAMPUS REMEMBER",
            "Category": "Action",
    "Actors" :
       [
         { "actorId": 8,"Name": "MATTHEW JOHANSSON" },
         { "actorId": 45,"Name": "REESE KILMER" },
         { "actorId": 168,"Name": "WILL WILSON" }
       ]
}

{ "_id" : 105, "Title" : "BULL SHAWSHANK",
               "Category" : "Action",
     "Actors" :
       [ { "actorId" : 2, "Name" : "NICK WAHLBERG" },
         { "actorId" : 23, "Name" : "SANDRA KILMER" } ]
}
```

Actor document

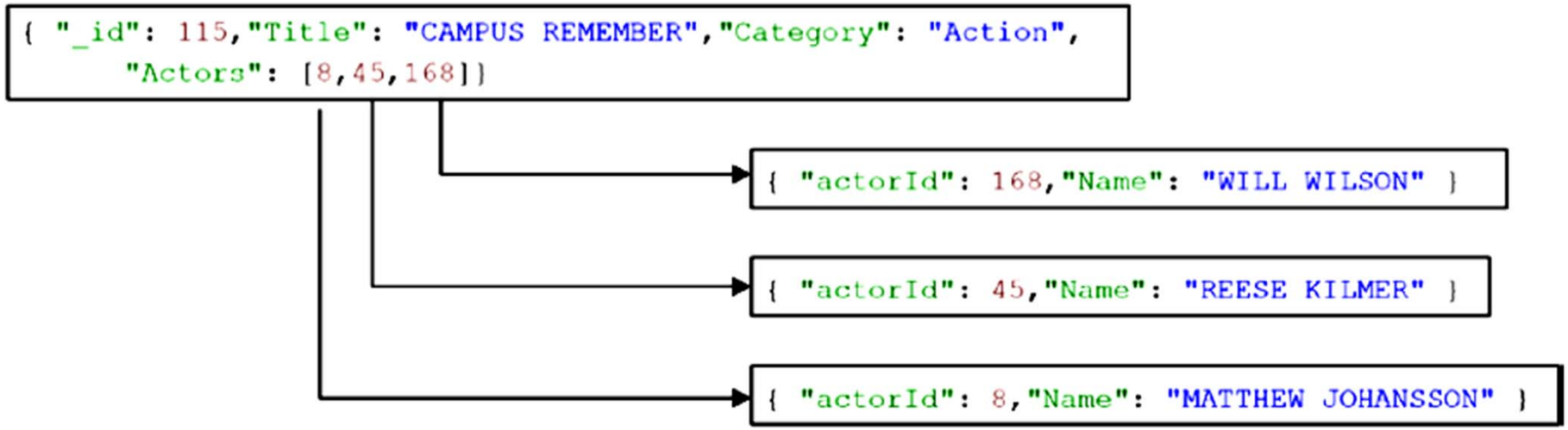
Film Document

Array of actors

Collection

پایگاه داده JSON

- **مثال: پایگاه داده فیلم**
- **در JSONDB ، می توان اسناد را مرتبط کرد**



پایگاه داده JSON

- **چند نمونه**

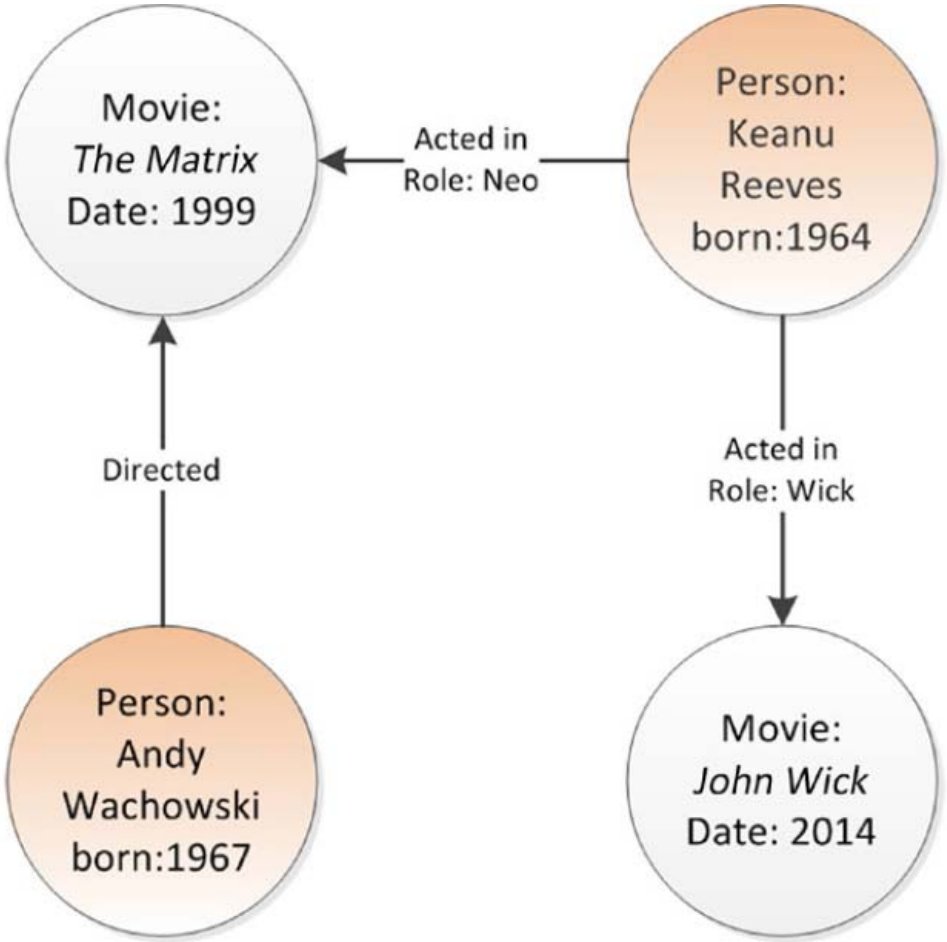
- CouchDB
- MemBase and CouchBase
- MongoDB

پایگاه داده مبتنی بر گراف

- مبتنی بر تئوری گراف
- گراف شامل: رئوس، یالها که می توانند صفاتی داشته باشند

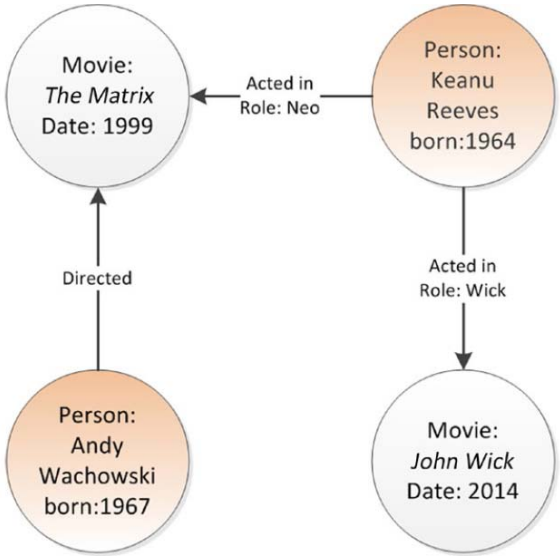
پایگاه داده مبتنی بر گراف

- **مثال**

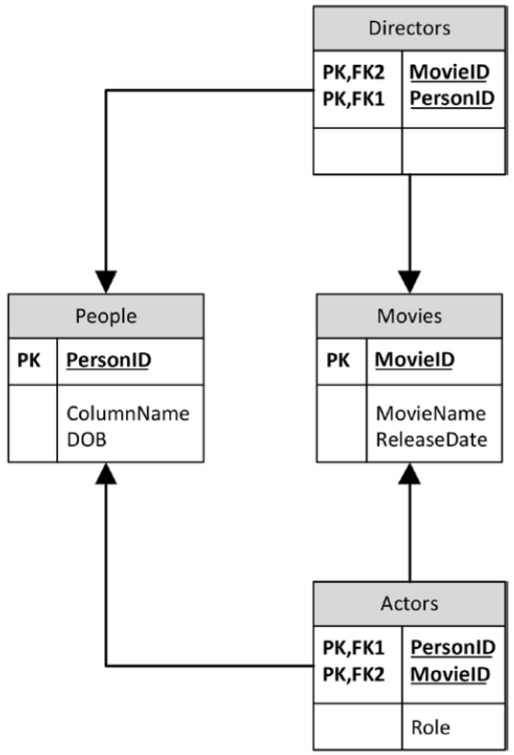


پایگاه داده مبتنی بر گراف

- **مثال در RDBMS؟**



=



?

- **عملیات گراف در SQL مشکل است**

پایگاه داده مبتنی بر گراف

- انواع

- **RDF**
- **Neo4j**

پایگاه داده RDF

- در RDF اطلاعات با سه گانه موجودیت ، صفت و مقدار بیان می شوند

- **مثال**

```
TheMatrix: is :Movie
Kenau: is :Person
Kenau: starred in: TheMatrix
```

پایگاه داده RDF

- **انواع**

- AllegroGraph
- Ontotext GraphDB
- StarDog
- Oracle Spatial

پایگاه داده RDF

- زبان پرس و جو : ***SPARQL***

- **مثال:**

Virtuoso SPARQL Query Editor

About | Namespace Prefixes | Inference rules | iSPARQL

Default Data Set Name (Graph IRI)

Query Text

```
SELECT ?object
  FROM <http://dbpedia.org>
 WHERE { <http://dbpedia.org/resource/Edgar_F._Codd>
         <http://purl.org/dc/terms/subject> ?object }
```

(Security restrictions of this

Results Format:

Execution timeout:

Options:

(The result can only be sent

Run Query | Reset

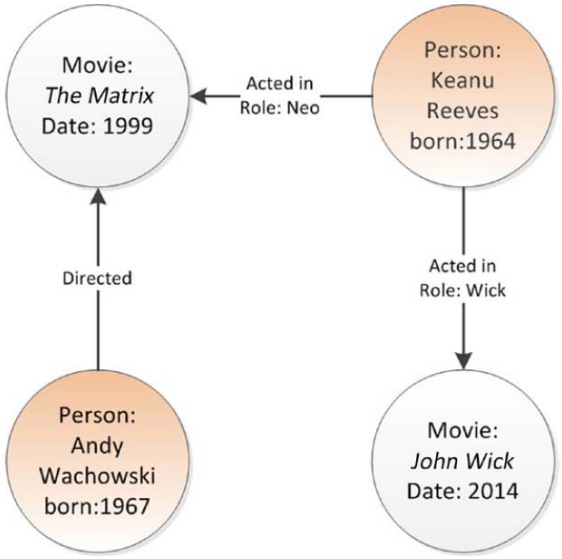
object
http://dbpedia.org/resource/Category:1923_births
http://dbpedia.org/resource/Category:2003_deaths
http://dbpedia.org/resource/Category:Alumni_of_Exeter_College,_Oxford
http://dbpedia.org/resource/Category:British_computer_scientists
http://dbpedia.org/resource/Category:Cellular_automatists
http://dbpedia.org/resource/Category:Deaths_from_myocardial_infarction
http://dbpedia.org/resource/Category:Fellows_of_the_Association_for_Computing_M
http://dbpedia.org/resource/Category:IBM_Fellows
http://dbpedia.org/resource/Category:IBM_employees
http://dbpedia.org/resource/Category:Turing_Award_laureates
http://dbpedia.org/resource/Category:University_of_Michigan_alumni
http://dbpedia.org/resource/Category:Database_researchers
http://dbpedia.org/resource/Category:People_educated_at_Poole_Grammar_School
http://dbpedia.org/resource/Category:People_from_San_Jose,_California

پایگاه داده Neo4j

- بر مبنای گراف خاصیت
- زبان پرس و جو : ***Cypher***
- نتایج دستورات Cypher گراف است

پایگاه داده Neo4j

- **مثال**

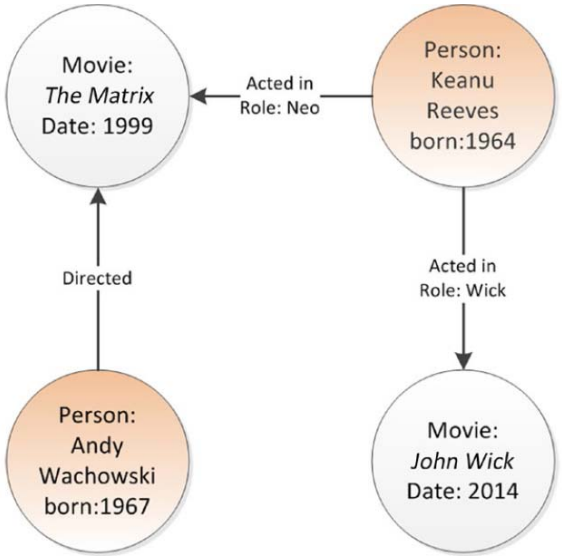


Cypher Query

```
CREATE (TheMatrix:Movie {title:'The Matrix', released:1999,
        tagline:'Welcome to the Real World'})
CREATE (JohnWick:Movie {title:'John Wick', released:2014,
        tagline:'Silliest Keanu movie ever'})
CREATE (Keanu:Person {name:'Keanu Reeves', born:1964})
CREATE (AndyW:Person {name:'Andy Wachowski', born:1967})
CREATE
  (Keanu)-[:ACTED_IN {roles:['Neo']}]->(TheMatrix),
  (Keanu)-[:ACTED_IN {roles:['John Wick']}]->(JohnWick),
  (AndyW)-[:DIRECTED]->(TheMatrix)
```

پایگاه داده Neo4j

- **مثال**

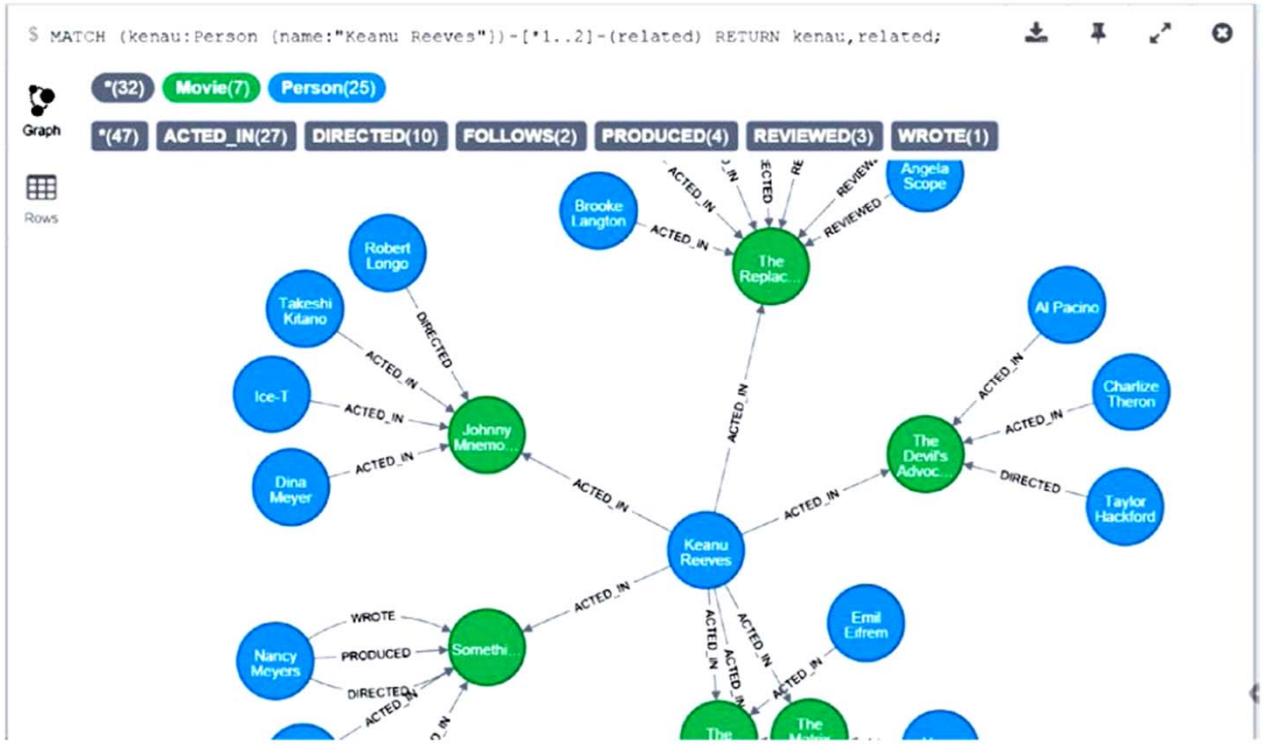


Cypher Query

```
neo4j-sh (?)$ MATCH (kenau:Person {name:"Keanu Reeves"})
>             RETURN kenau;
+--------------------------------------+
| kenau                                |
+--------------------------------------+
| Node[1]{name:"Keanu Reeves",born:1964} |
+--------------------------------------+
```

پایگاه داده Neo4j

- خروجی گراف Cypher

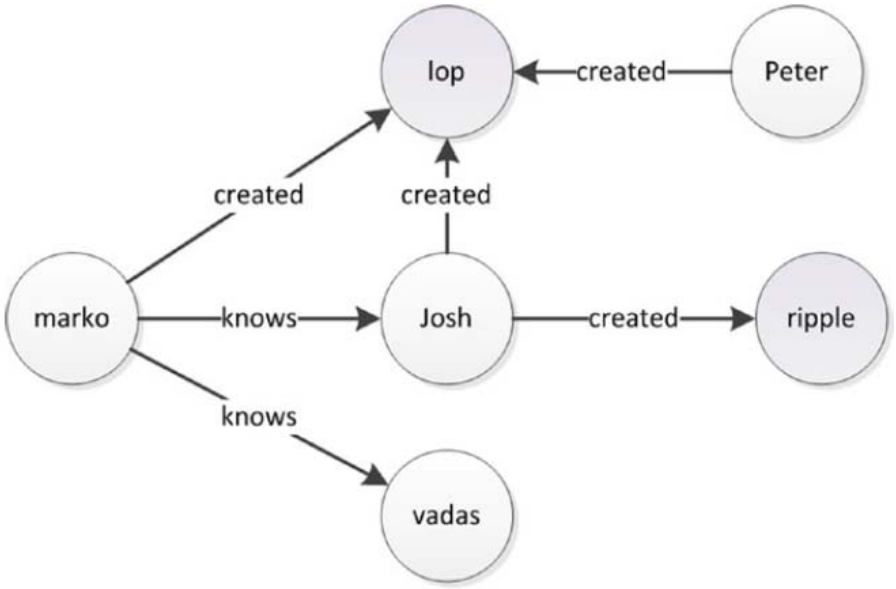


پایگاه داده Neo4j

- یک زبان پرس و جوی دیگر : ***Gremlin***
- نسبت به Cypher رویه ای تر است

پایگاه داده Neo4j

- **مثال**



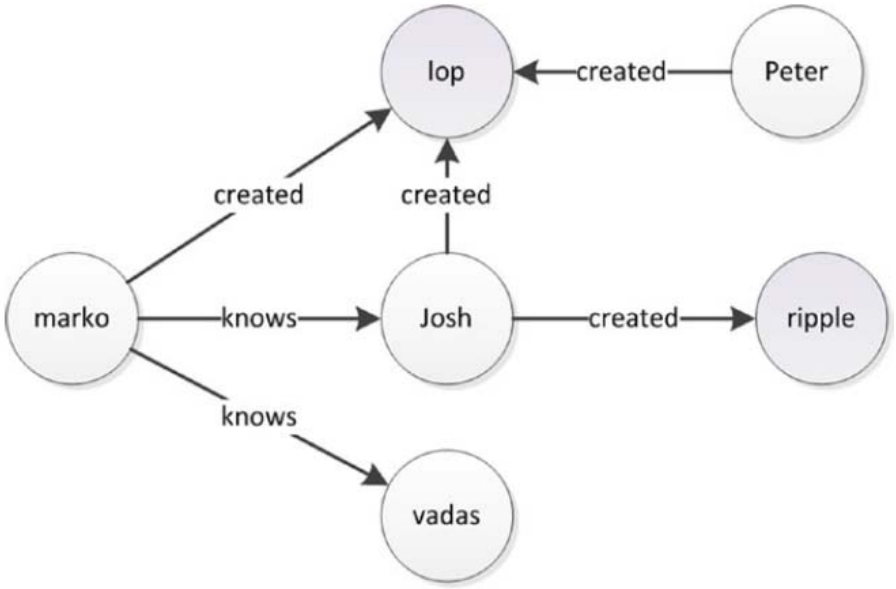
Gremlin Query

```
gremlin> myGraph.V.map
==>{name=marko, age=29}
==>{name=vadas, age=27}
==>{name=lop, lang=java}
==>{name=josh, age=32}
==>{name=ripple, lang=java}
==>{name=peter, age=35}

gremlin> myGraph.E
==>e[11][4-created->3]
==>e[12][6-created->3]
==>e[7][1-knows->2]
==>e[8][1-knows->4]
==>e[9][1-created->3]
==>e[10][4-created->5]
```

پایگاه داده Neo4j

- **مثال**



Gremlin Query

```
gremlin> marko=myGraph.v(1)
==>v[1]
gremlin> marko.out('knows').name
==>vadas
==>josh
```

موتورهای محاسباتی گراف

- برای پردازش گرافهای حجیم از این موتورها استفاده می شود:
  - Apache Giraph **in Hadoop**
  - GraphX **in Spark**
  - Gelly **in Flink**
  - Titan in Big Data storage engines like **Hbase, Cassandra,..**