PAC 1

M2.952 - Arquitectures de bases de dades no tradicionals

Alumne: Carlos A. García Pérez

Exercici 1

Pregunta 1

Les bases de dades transaccionals segueixen el model ACID (Atomicity, Consistency, Isolation, Durability). Això no és òptim si el que volem es garantir la disponibilitat del servei en un sistema distribuït. Per a garantir la transaccionalitat total, es segueix el model 2 Phase Commit. Aquest model assegura que tota transacció distribuïda és consistent. En contraposició, és bloquejant. Mentre es fa un commit, la base de dades no pot ser accedida. En molt de models de negocis es preferible una consisténcia de transacció relaxada a perdre disponibilitat.

El model BASE no garanteix la consistència global a un clúster, però sí la disponibilitat: un client por llegir diferents valors depenent de a quin node de base de dades accedeixi. Un procés (en el moment d'escriptura, lectura,… depèn del model) s'encarregarà de retornar la consistència en base a qualque criteri pre-establert. Es prioritza la disponibilitat davant de la consistència total, que és l'objectiu de les bases de dades NoSQL.

Pregunta 2

Gilbert i Lynch varen formalitzar i demostrar el teorema de Brewer; aquest diu que una base de dades en cluster no pot garantir la disponibilitat, consistència i tolerància a particions; només pot garantir dues de les tres. De fet, varen demostrar que les úniques combinacions possibles eren: AP (disponibilitat i tolerància a particions), CP (consistència i tolerància a) i AC (disponibilitat i consistència).

Pregunta 3

S'aconsegueix una consistència forta amb el compliment de W+R>N (4+5>9). Normalment, els models relacionals tenen W=N (fins que no s'escriu a tots els nodes no es considera finalitzada l'escriptura), que no es compleix en aquest cas.

Que W sigui 4 vol dir que l'escriptura es donarà per bona quan s'hagi realitzat a 4 dels 5 nodes. Això vol dir que, en el moment de l'escriptura, un node pot estar inconsistent. Es guanya, en canvi, temps de resposta. No cal que els 5 validin l'operació.

Que R sigui 5 vol dir que la lectura ha de ser consistent a tots els nodes. Això vol dir que hem de garantir que la dada sigui la correcta als 5 nodes. En aquest model se garanteix la consistència en la lectura, no en l'escriptura.

Pregunta 4

- Key-value: Hazelcast i Voldemort poden tenir persistència només en memòria.
- Graph-oriented databases. Els frameworks de processament de grafs carreguen tota la informació a memòria RAM. Una vegada el procés s'ha completat, les modificacions fetes sobre el graf de memòria no es persisteixen.

Exercici 2

Afirmació 1

Cert. L'agregació de dades implica, preferiblement, que s'accedeixi a les dades agregades simultàniament (pàgina 29 del llibre). El model d'agregació no té gaire sentit si no sabem quines consultes es realitzaran i, per tant, quines dades es consultaran a la vegada (pàgines 14 i 31).

Afirmació 2

Fals. Si dues aplicacions accedeixen a una base de dades sense esquema, ambdues han de conèixer i respectar el model implícit (pàgina 29). Si una aplicació canvia l'estructura de dades, les altres es veuran afectades si accedeixen a les mateixes dades i no realitzen el mateix canvi. El model ideal és que a una base de dades sense esquema explícit només hi accedeixi una aplicació.

Afirmació 3

Fals. L'existència d'un esquema (present a les bases de dades tradicionals) implica que el tipus de dades és conegut (numèric, caràcter, data, etc.). Qualsevol intent de guardar altra informació pot provocar un error o tenir un efecte sobre les dades (pàgina 10). Com a nota apart, i fora del llibre, dir que les bases de dades tradicionals també permeten guardar la informació en JSON o XML, malgrat no és el seu funcionament "estàndard".

Afirmació 4

Cert. La persistència políglota (pàgina 11) està en contra de la solució "one solution fits all"; és a dir: aplicar la mateixa recepta a tots els casos. Cada cas ha de ser tractar de forma diferent, aplicant les solucions i productes més adients.

Exercici 3

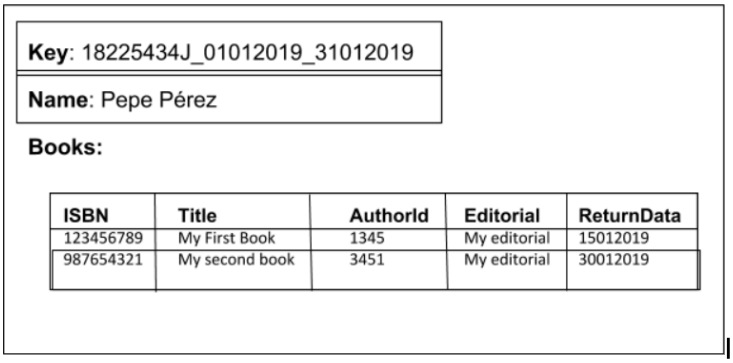
Cas 1

Proposo com a estructura la següent:

- **Una clau (key) formada pel DNI d'usuari** (PK de l'usuari i part de la informació que es vol obtenir) **i les dates d'inici i fi de la cerca**. D'aquesta forma s'afavoreixen les cerques per usuari i període.
- **Nom**: És l'altra característica de l'usuari que es vol; la situem a la mateixa alçada.
- **Llista de llibres**. Informació agregada a l'usuari. Totes les característiques que es demanen.
- **Exemple de JSON i representació gràfica**:


```
{
 "key":"18225434J_01012019_31012019",
 "name": "Pepe Pérez",
 "books":[
   {

     "ISBN":123456789,
     "title": "My first book",
     "authorId": 1345,
     "editorial": "My editorial",
     "returnData": 15012019
   },
   {
     "ISBN":987654321,
     "title": "My second book",
     "authorId": 3451,
     "editorial": "My editorial",
     "returnData": 30012019
   }
  ]
}
```

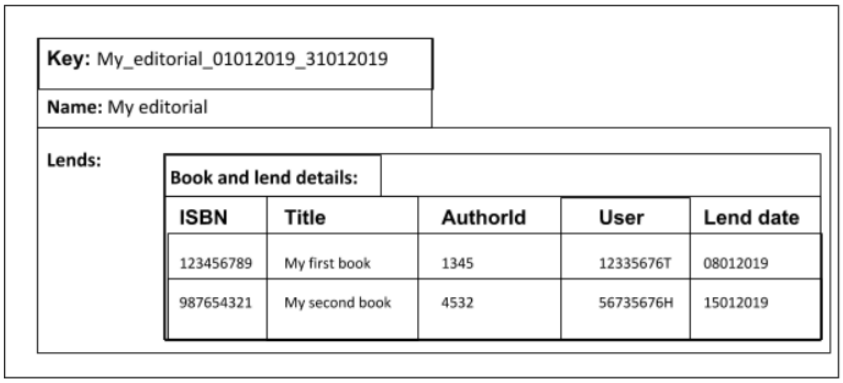



Cas 2

Proposo com a estructura la següent:

- **Una clau formada pel nom de l'editorial i les dates del període**. Si tenim problemes amb els espais en blanc, els substituiria per un símbol i afegiria el nom de l'editorial com a camp.
- **Una llista dels llibres prestats**. Agrupo els atributs dins de book.
- **El id de l'usuari** (DNI en el nostre cas) **i la data de préstec**.
- **Exemple de JSON i representació gràfica**:


```
{
 "key":"My_editorial_01012019_31012019",
 "name": "My editorial",
 "lends":[
    {
     "book":{
      "ISBN":123456789,
      "title": "My first book",
      "authorId": 1345
     },
     "user": "12335676T",
     "lendDate": "08012019"
    },
    {
     "book":{
      "ISBN":987654321,
      "title": "My second book",
      "authorId": 4532
     },
     "user": "56735676H",
     "lendDate": "15012019"
    }
  ]
}
```

Key: My_editorial_01012019_31012019

Name: My editorial

Lends:

Book and lend details:

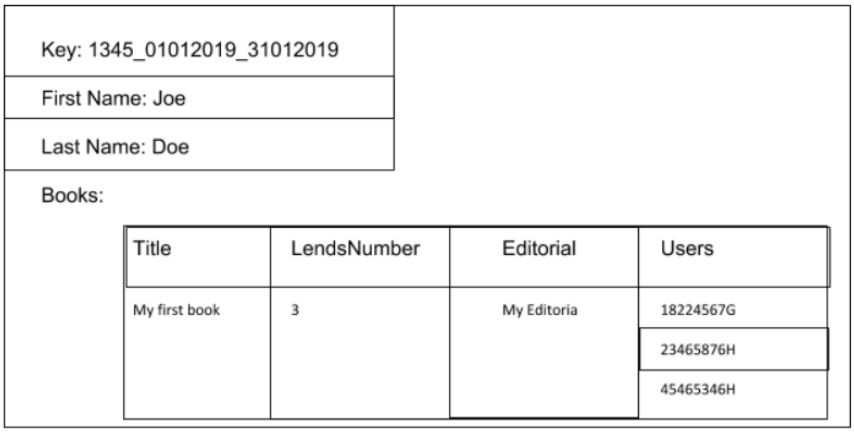
ISBN	Title	AuthorId	User	Lend date
123456789	My first book	1345	12335676T	08012019
987654321	My second book	4532	56735676H	15012019

Cas 3:

Proposo com a estructura la següent:

- **Una clau formada pel codi de l'autor i les dates del període**.
- **Les altres dades de l'autor** a la mateixa alçada
- **Una llista dels llibres prestats**. La llista és books.
- **Dins de l'anterior llista, s'inclouen els DNIs dels usuaris**.
- **Exemple de JSON**:

```
{
 "key":"1345_01012019_31012019",
 "firstName": "Joe",
 "lastName": "Doe",
 "books":[
    {
     "title":"My first book",
     "lendsNumber":3,
     "editorial":"My Editorial",
     "users":[
      {"DNI":"18224567G"},
      {"DNI":"23465876H"},
      {"DNI":"45465346H"}
     ]
    },
    {
     "title":"My second book",
     "lendsNumber":2,
     "editorial":"My Editorial",
     "users":[
      {"DNI":"18224567G"},
      {"DNI":"45465346H"}
     ]
    }
  ]
}
```



Exercici 4

Descriu el problema de persistència de dades que s'ha resolt

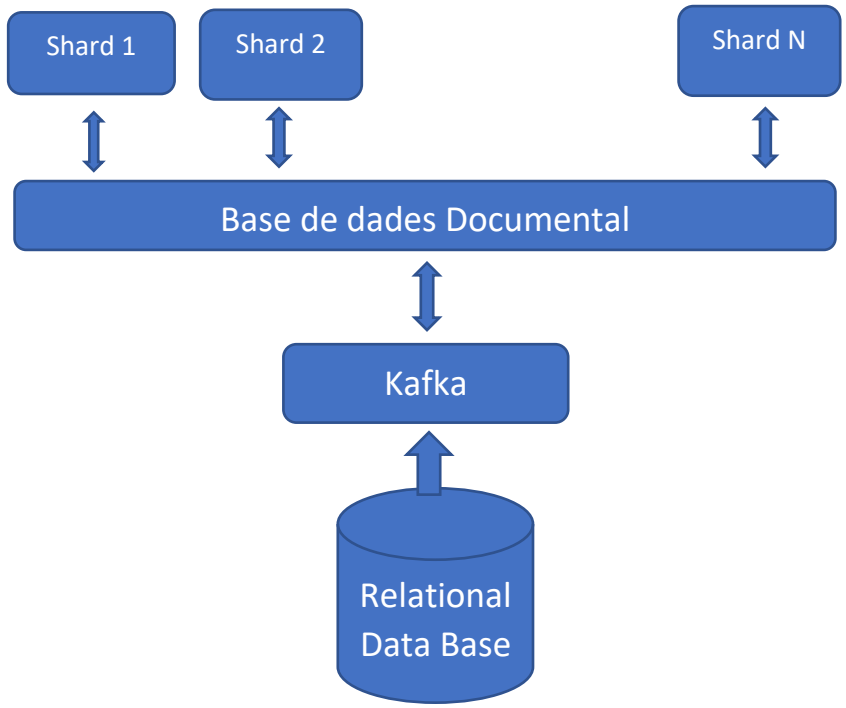
El nostre negoci és el de majorista d'habitacions d'hotel. Les agències de viatges (tant on-line com off-line) i els TTOO no negocien directament amb els hotelers; això ho fem nosaltres. Incorporem aquest producte (disponibilitat d'habitacions hoteleres) a una plataforma a la qual poden accedir els nostres clients.

Els nostres clients munten les seves pàgina web en funció de la disponibilitat de producte que nosaltres (i altres empreses) li proporcionem. La informació ha de ser correcta i actualitzada; no podem oferir disponibilitat d'un hotel que té un tancament de ventes, per exemple. Una consulta molt habitual és: "donem tots els hotels d'un destí". Podem tenir milers de milions de consultes diàries de disponibilitat des de clients de tot el món.

Tenim una base de dades relacional que garanteix: una habitació no es ven dues vegades, qualsevol canvi de contracte s'aplica immediatament, etc. Evidentment, si llencem totes les consultes sobre la base de dades, aquesta caurà.

La solució és una base de dades documental que repliqui les dades de la base de dades relacional. Ha d'estar distribuïda (data centers a Amèrica, Àsia i Europa), seguint una estratègia de sharding. Això és així perquè el producte es sol contractar per proximitat: la majoria del producte Americà es contracta des d'Amèrica, per exemple. Podem pensar en diferents documents agregats: per destí, per hotel, per temàtica, ... Els documents inclouen tots els detalls del producte.

La contractació d'un producte només es pot fer a la base de dades relacional. La base de dades NoSQL només és de consulta. Les dades s'actualitzen mitjançant un servei de missatgeria (Kafka).



Justifica les raons per les quals no és recomanable l'ús d'una base de dades relacional

Les tres principals raons són:

- És necessària una estratègia de clúster per a poder distribuir les dades. Les bases de dades relacionals no estan pensades per a aquesta topologia.
- Sempre es consulta la mateixa informació conjuntament. L'estratègia d'agregació és perfecta per a això. Les joins a la base de dades relacional no tindrien el mateix rendiment.
- El rati consultes vs modificacions és molt elevat. No és necessari accedir a la base de dades garantit la consistència per a totes les consultes. És necessari que totes les consultes donin una resposta i que ho faci molt aviat.

Justifica les raons per les quals és recomanable la base de dades que s'ha utilitzat com a solució

S'adapta perfectament als requisits del problema:

- L'accés per clau és molt ràpid; tant com a una base de dades de clau-valor. La majoria de les cerques (destí, hotel, contracte) són conegudes i això permet construir un document que s'adapti al que vol el client. Cada consulta es pot repetir milers de vegades.
- Permet de forma natural el sharding. Es pot distribuir el producte així com interessi.
- No és necessària la consistència total. Normalment, els productes no es contracten immediatament després de la consulta. Si això passa, la capa de persistència de la base de dades relacional garanteix que no es produeixen errors.
- La topologia de shards pot canviar depenent de l'època de l'any i la demanda. Aquestes bases de dades s'adapten de forma natural a aquest requisit.

Indicar les referències utilitzades per desenvolupar l'exercici

He de dir que l'exemple no és real, però està inspirat en el nostre model de negoci. Així que la primera referència és el meu lloc de feina.

A més:

Book
NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence
Addison-Wesley Professional ©2012
ISBN:0321826620 9780321826626
Chapters: 1-3

Persisting big-data: The NoSQL landscape
Author: Alejandro Corbellini, Cristian Mateos, Alejandro Zunino, Daniela Godoy, Silvia Schiaffino
ISISTAN (CONICET-UNCPBA) Research Institute1, UNICEN University, Campus Universitario, Tandil B7001BBO, Argentina