Backend developer (inter) – Assessment

By : Rakshith Venkatachalapathy

1. Database structure for storing the data.
2. For relational database I will be using SQL with MySQL database.

The give data should be normalized and then stored in the relational database.

To do the same we will be creating two tables

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| userId  (Primary key) | timestamp | message |  | Tagid  (primary key) | userId  (Foreign key) | tags |
| 1 | 174adee | Foo |
| 174adee | 1402272598 | This is a sample data |
| 2 | 174adee | world |
| 3 | 174adee | Bar |
| 4 | 174adee | Demo |
| 5 | 174adee | hello |

Table 1 Table 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id  (primary key) | Userid  (foreign key) | | Fname | | lname |
| 1 | 174adee | John | | Doe | |

Table 1 contains: userId – primary key

Timestamp

Message

Table 2 contains: userId – Foreign key (*To link Table 1 and 2*)

Tags

Tagid – primary key of Table 2

The foreign key is used to link the two tables.

For every tag there will be a corresponding row in Table with the ‘userId’ to link the same.

1. For non-relation database, I will be using MongoDB.

In MongoDB, data is stored as documents. These documents are stored in MongoDB in JSON format.

JSON documents support embedded fields, so related data and lists of data can be stored with the document instead of an external table.

{

"message": "This is sample data",

"tags": [ “foo”, “bar” , “demo”],

"timestamp": “1402272598”,

"userId": "174adee "

}

2.Queries

1. OR clause for all the tags in relational Database

SELECT messagecol

FROM message m

join tags t using (userid)

where tags in ('foo','bar','demo')

group by messagecol;

OR clause for all the tags in non-relational Database:

To select the messagecol column in mongodB we use the below query

The below query corresponds to performing ‘OR’ operation over all the tags

* *db.inventory.find( { tags: { $in: [ ‘foo’, ‘bar’,’demo’ ] } } ,{messagecol : 1})*

2.AND clause for all the tags in relational database:

SELECT messagecol

FROM message m

join tags1 t using (userid)

where t.tags ='foo' and t.tags='bar' and t.tags ='demo' and m.userid = t.userid

group by messagecol;

AND clause for all the tags in non-relational database:

To select the messagecol column in mongodB we use the below query

The below query corresponds to performing ‘AND’ operation over all the tags

* *db.inventory.find( { tags: { $all: [ ‘foo’, ‘bar’,’demo’ ] } } ,{messagecol : 1})*

3.To fetch information on a user

SELECT messages.userId, users.fname, users.lname

FROM messages

RIGHT JOIN users

ON messages.userId = messages.userId

ORDER BY messages.userId;

1. To help mitigate traffic in a network or cloud computing, load balancers can be used.

* They improve the overall performance of applications by decreasing the burden on servers associated with managing and maintaining application and network sessions, as well as by performing application-specific tasks.
* To achieve high availability, sustainable as you grow you need at least two backend servers for high availability, and your load balancer will ensure that if one backend isn’t functioning, the traffic will be directed to the other backend.
* This technique ensures that even if one backend fails the traffic will be directed to other backend with the help of load balancer.
* Load balancers are also capable of detecting unavailable servers and redirecting traffic to those still operational.

1. To handle multiple write requests coming in, the techniques used to ensure uptime and data availability to clients are

* Running concurrent writes/ parallel writes to increase the number of writes happening to the database.
* Implementing batch writing techniques to increase the number of writes to the database.
* Speeding up recovery times and backing up the data regularly also helps.