

TRƯỜNG ĐẠI HỌC SƯ PHẠM TP.HCM

Khoa Công nghệ thông tin

Môn: Cơ sở dữ liệu nâng cao

Đề tài:

TÌM HIỂU COUCHDB



Giáo viên hướng dẫn: Thầy Lương Trần Hy Hiến

Sinh viên thực hiện: Nhóm 23

Châu Thái Bảo	-	42.01.104.012
Hà Chí Vĩ	-	42.01.104.290
Võ Quỳnh Mai Trang	-	42.01.104.176
Bùi Phương Dung	-	42.01.104.020
Võ Thị Hoàng Linh	-	42.01.104.237

TP.HCM Tháng 11 năm 2018

LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn thầy Lương Trần Hy Hiến đã tận tình giảng dạy, trang bị, cung cấp cho chúng em những kiến thức nền tảng, chuyên môn cần thiết giúp chúng em hiểu sâu hơn về bộ môn Cơ sở dữ liệu nâng cao.

Mặc dù đã cố gắng hoàn thành tốt công việc được phân công trong phạm vi và khả năng cho phép nhưng chắc chắn không tránh khỏi những thiếu sót. Rất mong nhận được sự góp ý của thầy để đề tài được hoàn thiện hơn, đồng thời bổ sung vốn kinh nghiệm cho chúng em trên con đường sắp tới.

Xin chân thành cảm ơn!

Sinh viên thực hiện

Nhóm 23

∞ MỤC LỤC ∞

Nội dung

	Trang
CHƯƠNG I: KHÁI QUÁT CƠ SỞ DỮ LIỆU NOSQL	3
1.1 Khái niệm	3
1.2 Đặc điểm	3
1.3 Phân loại NoSQL database	3
CHƯƠNG II: COUCHDB	5
2.1. Khái niệm	5
2.2. Tại sao chúng ta lại cần CouchDB?	5
2.3. Mô hình dữ liệu	5
2.4. Tính năng của CouchDB	5
2.5. Ưu điểm và nhược điểm của CouchDB	6
2.6. Hướng dẫn cài đặt và xử lý dữ liệu trên CouchDB	7
2.7. Cách truy vấn và xử lý dữ liệu của CouchDB bằng Javascript	16
2.8. Demo	16
CHƯƠNG III: CÀI ĐẶT DEMO	
TÀI LIỆU THAM KHẢO	23

CHƯƠNG I: KHÁI QUÁT CƠ SỞ DỮ LIỆU NOSQL

Trong vài năm qua chúng ta đã thấy sự gia tăng của một loại cơ sở dữ liệu mới, đó là cơ sở dữ liệu NoSQL - mà đang thách thức sự thống trị của cơ sở dữ liệu quan hệ. Cơ sở dữ liệu quan hệ đã thống trị ngành công nghiệp phần mềm trong một thời gian dài khi đã cung cấp cơ chế để lưu trữ dữ liệu liên tục, đồng thời kiểm soát, giao dịch, giao diện được chuẩn hóa và được tích hợp vào các hệ thống dữ liệu ứng dụng, báo cáo. Tuy nhiên, ưu thế đó đã không còn tồn tại cho cơ sở dữ liệu quan hệ nữa.

1.1 Khái niệm

NoSQL là một khái niệm chỉ về một lớp các hệ cơ sở dữ liệu không sử dụng mô hình quan hệ (RDBMS). RDBMS vốn tồn tại khá nhiều nhược điểm như có hiệu năng không tốt nếu kết nối dữ liệu nhiều bảng lại hay khi dữ liệu trong một bảng là rất lớn.

NoSQL ra đời năm 1998 bởi Carlo Strozzi khi ông lập mới một hệ cơ sở dữ liệu quan hệ mã nguồn mở nhanh và nhẹ không liên quan đến SQL. Vào năm 2009, Eric Evans, nhân viên của Rackspace giới thiệu lại thuật ngữ NoSQL khi Johan Oskarsson của Last.fm muốn tổ chức một hội thảo về cơ sở dữ liệu nguồn mở phân tán. Thuật ngữ NoSQL đánh dấu bước phát triển của thế hệ CSDL mới: phân tán (distributed) + không ràng buộc (non-relational).

1.2 Đặc điểm

- NoSQL lưu trữ dữ liệu của mình theo dạng cặp giá trị “key – value”. Sử dụng số lượng lớn các node để lưu trữ thông tin
- Chấp nhận dữ liệu bị trùng lặp do một số node sẽ lưu cùng thông tin giống nhau
- Phi quan hệ – không có ràng buộc nào cho việc nhất quán dữ liệu
- Có hiệu suất cao (high performance) và tính sẵn sàng cao (high availability)

1.3 Phân loại NoSQL database

- *Key – value data stores*: Dữ liệu lưu dưới dạng cặp key – value. Giá trị được truy xuất thông qua key.
 - Ví dụ : Redis, Dynamite, Project Voldemort.
 - Thường cho: Content caching Applications
 - Ưu điểm: Tìm kiếm rất nhanh
 - Nhược điểm: Lưu dữ liệu không theo khuôn dạng (schema) nhất định
- *Column-based – Tabular*: Cơ sở dữ liệu tổ chức dưới dạng các bảng. Gần giống với mô hình RDBMS. Tuy nhiên, Chúng lưu dữ liệu bởi các cột chứ

không phải bằng các dòng. Nó khá thích hợp với để hiển thị bằng các phần mềm quản lý kho dữ liệu

- Ví dụ: Apache Hbase, Apache Cassandra, Hypertable
- Thường cho: các hệ phân tán file
- Ưu điểm: Tìm kiếm nhanh, Phân tán dữ liệu tốt
- Nhược điểm: Hỗ trợ được với rất ít phần mềm
- *Document-based*: Dữ liệu (bán cấu trúc hay semi-structured) được lưu trữ và tổ chức dưới dạng một tập hợp các document. Các document này linh hoạt, mỗi document có một tập nhiều trường.
 - Ví dụ: Apache CouchDB và MongoDB
 - Thường cho: Web applications
 - Ưu điểm: Dùng khi dữ liệu nguồn không được mô tả đầy đủ
 - Nhược điểm: Hiệu năng truy vấn, Không có cú pháp chuẩn cho câu truy vấn dữ liệu
- *Graph-based data-stores*: Những CSDL này áp dụng lý thuyết đồ thị trong khoa học máy tính để lưu trữ và truy xuất dữ liệu. Chúng tập trung vào tính rời rạc giữa các phần dữ liệu. Các phần tử đơn vị dữ liệu được biểu thị như một nút và liên kết với các thành phần khác bằng các cạnh.
 - Ví dụ: Neo4j, InfiniteGraph, DEX
 - Thường cho: Social networking, Hệ trợ giúp
 - Ưu điểm: Ứng dụng các thuật toán trên đồ thị như Đường đi ngắn nhất, liên thông, ...
 - Nhược điểm: Phải duyệt nội bộ đồ thị, để trả lời lại các truy vấn. Không dễ để phân tán

CHƯƠNG II: COUCHDB

2.1. Khái niệm

- CouchDB là 1 cơ sở dữ liệu dạng NoSQL mã nguồn mở được phát triển bởi nền tảng phần mềm Apache (được viết bằng ngôn ngữ lập trình Erlang) database lưu trữ dữ liệu dạng document/JSON.
- CouchDB được thiết kế nhằm tới tính dễ sử dụng và phục vụ cho môi trường web.

2.2. Tại sao chúng ta lại cần CouchDB?

- CouchDB có API dạng RESTful giúp cho việc giao tiếp với cơ sở dữ liệu được đơn giản.
- Các RESTful API rất trực quan và dễ thao tác.
- Dữ liệu được lưu dưới cấu trúc document rất mềm dẻo, chúng ta không cần phải lo lắng về cấu trúc dữ liệu.
- Map/reduce giúp việc lọc, tìm, tổng hợp dữ liệu dễ hơn bao giờ hết.
- Nhân bản / đồng bộ là sức mạnh đặc biệt của CouchDB mà hiếm database nào có.

2.3. Mô hình dữ liệu

- Database là cấu trúc dữ liệu lớn nhất của CouchDB.
- Mỗi database là 1 danh sách các document độc lập.
- Document bao gồm dữ liệu người dùng thao tác lẫn thông tin về phiên bản của dữ liệu để tiện việc merge dữ liệu.
- CouchDB sử dụng cơ chế phiên bản hoá dữ liệu để tránh tình trạng khoá dữ liệu khi đang ghi.

2.4. Tính năng của CouchDB

- Lưu dưới dạng document
- CouchDB là một NoSQL database dạng document. Document là một đơn vị dữ liệu (giống như 1 object của Javascript), mỗi field có một tên riêng không trùng nhau, chứa các loại dữ liệu như chữ, số, Boolean, danh sách... Không có bất kỳ giới hạn nào về dung lượng text hay số field trong 1 document.
- CouchDB cung cấp 1 RESTful API cho việc đọc và ghi (thêm, sửa, xóa) document.

- Sau đây là 1 ví dụ về 1 document

```
{  
  "title": "Macbook",  
  "price": 1500,  
  "SKU": "abcd1234"  
}
```

- Các thuộc tính ACID

Khi dữ liệu được ghi xuống ổ cứng thì nó sẽ không bị ghi đè. Bất kì thay đổi nào (thêm, sửa, xóa) đều theo chuẩn Atomic, có nghĩa là dữ liệu sẽ được lưu lại toàn diện hoặc không được lưu lại. Database không bao giờ thêm hay sửa một phần dữ liệu.

Hầu hết các cập nhật đều được serialized để đảm bảo tất cả người dùng có thể đọc document mà không bị chờ đợi hoặc gián đoạn.

- Khả năng nén (Compaction)

Nén là 1 hành động giúp giải phóng dung lượng ổ cứng được sử dụng bằng cách xóa đi các dữ liệu không còn được sử dụng. Khi tiến hành nén dữ liệu ở 1 file thì 1 file mới với định dạng **.compaction** sẽ được tạo ra và dữ liệu sẽ được sao chép vào file mới này. Khi quá trình copy hoàn thành thì file cũ sẽ được xóa bỏ. Database vẫn online trong quá trình nén và các thao tác thay đổi / đọc dữ liệu vẫn diễn ra bình thường.

- Views

Dữ liệu trong CouchDB được lưu trữ trong các document. Bạn có thể tưởng tượng như 1 database là 1 table và 1 document (chứa fields và attachments) là 1 row. Khi chúng ta muốn trình bày dữ liệu bằng nhiều góc nhìn khác nhau thì chúng ta cần 1 phương pháp để filter, tổ chức để hiển thị kết quả cuối cùng. Để giải quyết vấn đề này, CouchDB sử dụng mô hình View. View là 1 phương pháp tổng hợp dữ liệu trong các document ở 1 database. Các View được build động và không ảnh hưởng đến dữ liệu đã ghi của các document nên chúng ta có thể có bao nhiêu View tùy ý tùy vào nhu cầu trình bày dữ liệu.

2.5. Ưu điểm và nhược điểm của CouchDB

❑ Ưu điểm

- Các điểm thu hút chính của CouchDB là nó sẽ tiếp tục vận hành thậm chí kết nối mạng chập chờn. Do tính chất này, CouchDB thích hợp lưu trữ dữ liệu cục bộ.
- Là sự lựa chọn tốt khi người dùng tạo các báo cáo động từ các yếu tố thay đổi rất thường xuyên. Một ví dụ tốt khi dùng document là phân tích theo thời gian thực trong mạng xã hội hoặc hệ thống quản lý nội dung.
- CouchDB có API dạng RESTful giúp cho việc giao tiếp với cơ sở dữ liệu được đơn giản (Các RESTful API rất trực quan và dễ thao tác).
- Dữ liệu được lưu dưới cấu trúc document rất mềm dẻo, linh hoạt chúng ta không cần phải lo lắng về cấu trúc dữ liệu, không bị bó buộc về số lượng field, kiểu dữ liệu,...
- Map/reduce giúp việc lọc, tìm, tổng hợp dữ liệu dễ hơn bao giờ hết.
- Nhân bản / đồng bộ là sức mạnh đặc biệt của CouchDB mà hiếm database nào có.

❑ Nhược điểm:

- CouchDB không có tính ràng buộc như RDBMS nên dễ làm sai dữ liệu, cần tính cẩn thận cao..
- Cần phải tạo các view cho mỗi và mọi truy vấn, tức là các truy vấn đặc biệt như các truy vấn động WHERE và SORT trong một SQL) không có sẵn.
- Sử dụng nhiều bộ nhớ dữ liệu lưu dưới dạng key-value nên key sẽ bị lặp lại, thừa dữ liệu.

2.6. Hướng dẫn cài đặt và xử lý dữ liệu trên CouchDB

➤ Cài đặt

- Download CouchDB: <http://couchdb.apache.org/>



[About](#) [Docs](#) [Contribute](#) [News](#) [Download](#) [More...](#)



Seamless multi-master sync, that
scales from **Big Data** to **Mobile**,
with an **Intuitive** HTTP/JSON API
and designed for **Reliability**.

DOWNLOAD
2.2.0 (2018-08-08)



Download CouchDB 2.2.0

↓ Source

Version 2.2.0 | [Release Notes](#) | [PGP Signature](#) | [SHA256](#)

↓ Windows (x64)

Erlang/OTP 19.3 | Version 2.2.0 | [Release Notes](#) | [PGP Signature](#)

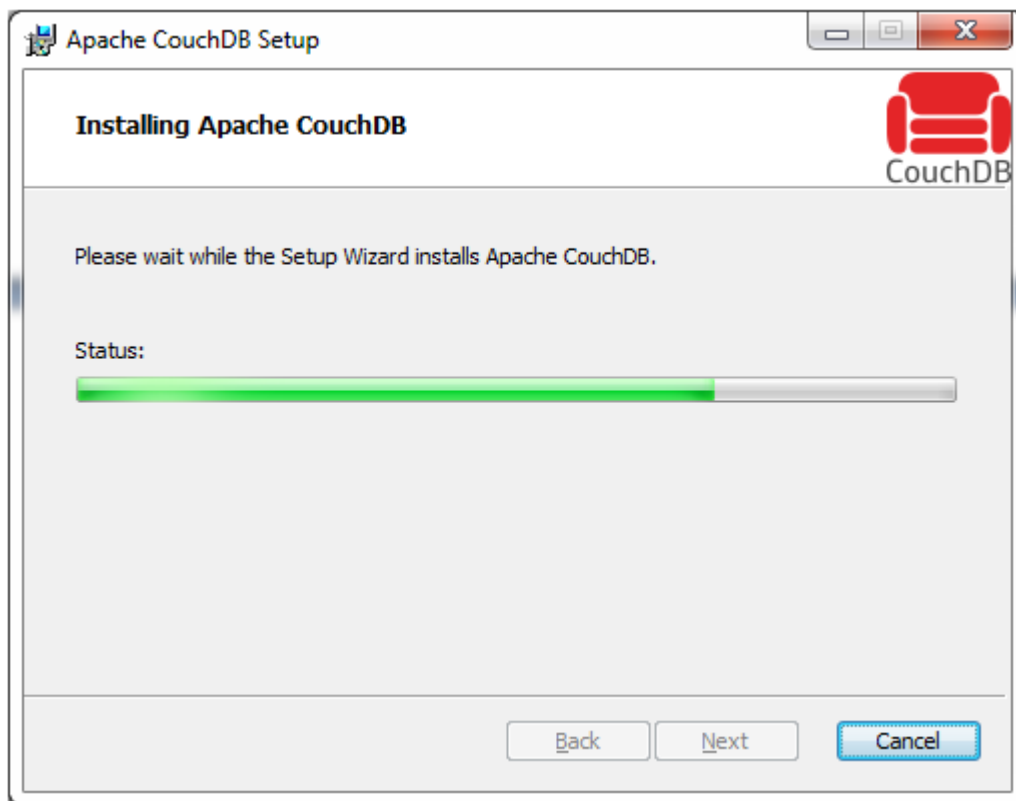
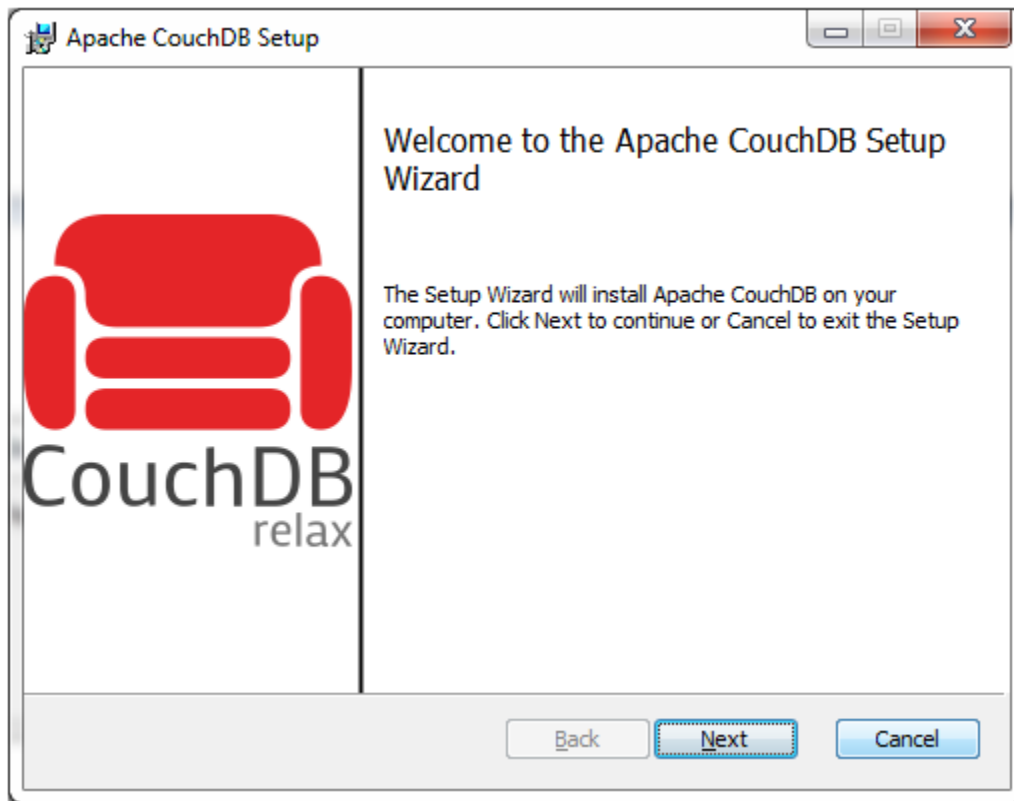
↓ macOS (10.10+)

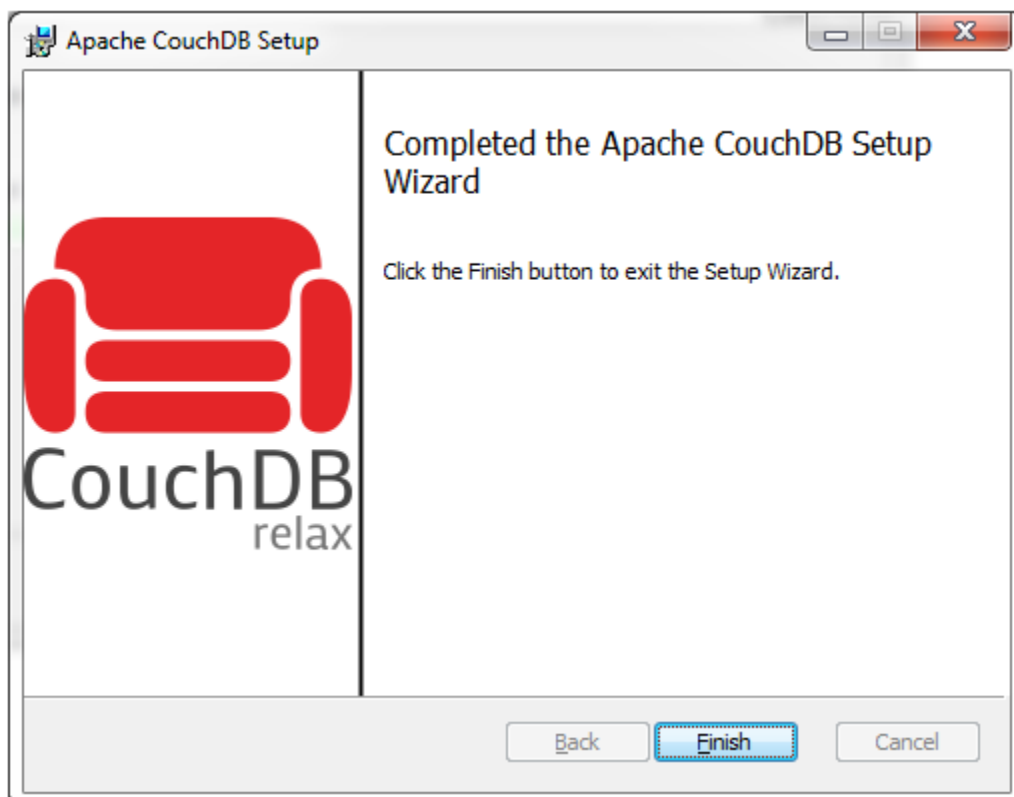
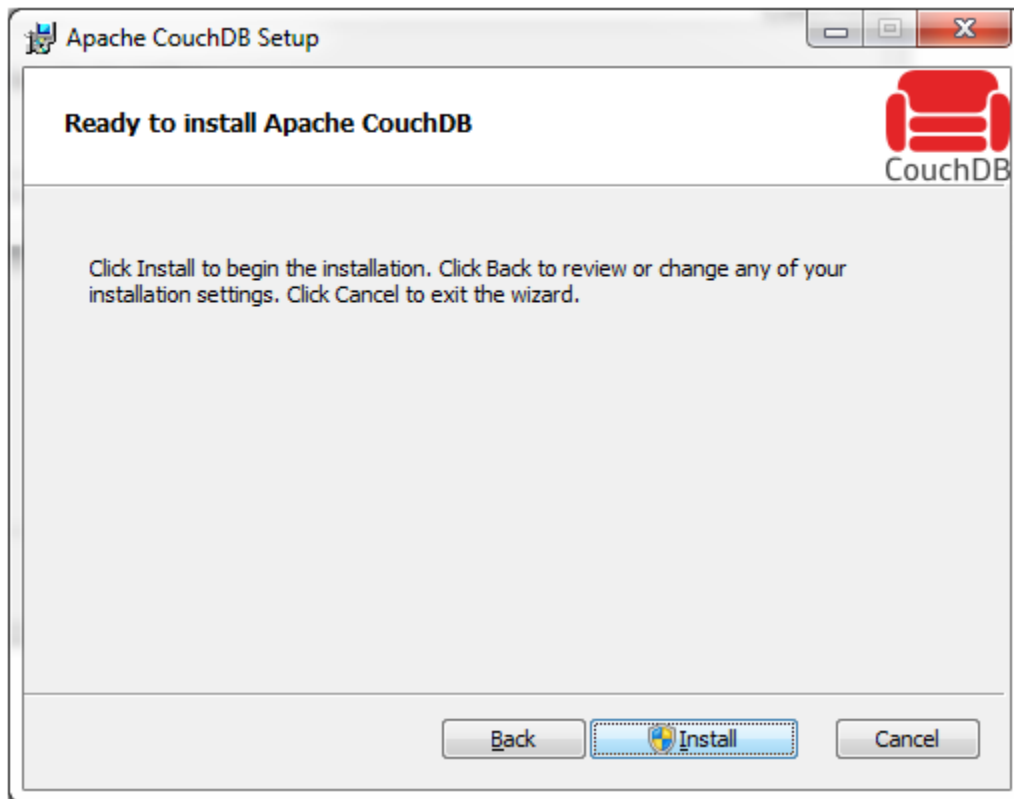
Erlang/OTP 19.3 | Version 2.2.0 | [Release Notes](#) | [PGP Signature](#) | [SHA256](#) | [SHA512](#)

↓ Debian/Ubuntu/RHEL/CentOS packages

Erlang/OTP 19.3 | Version 2.2.0 | [Release Notes](#)

[Instructions for verifying downloads.](#)





- Sau khi hoàn tất. Truy cập theo đường link: <http://localhost:5984/>.

- Trang web sẽ hiển thị nội dung sau

```
{"couchdb":"Welcome","version":"2.2.0","git_sha":"2a16ec4","features":
["pluggable-storage-engines","scheduler"],"vendor":{"name":"The Apache Software
Foundation"}}
```

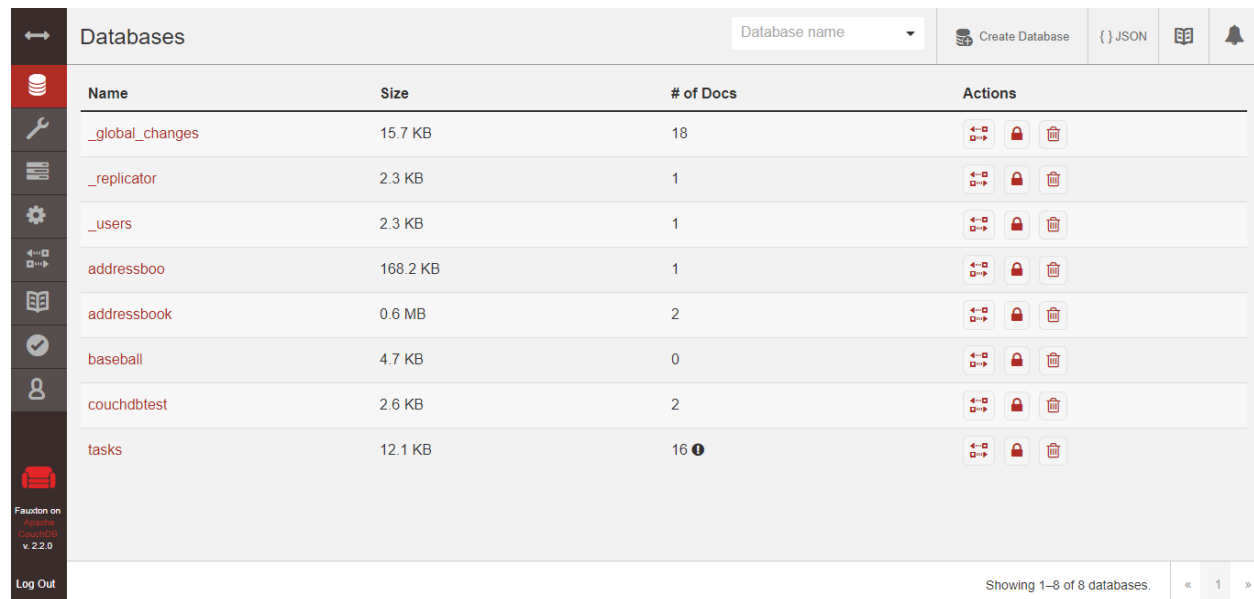
- Để tương tác giao diện CouchDB truy cập link: http://localhost:5984/_utils/

- Có 2 cách để tương tác với CouchDB: Fauxton và cURL

➤ Tương tác CouchDB bằng Fauxton

- Fauxton là một giao diện quản trị tích hợp trên web của CouchDB. Fauxton cung cấp một giao diện đồ họa đơn giản để tương tác với CouchDB, cung cấp quyền truy cập đầy đủ vào tất cả các tính năng của CouchDB.

- Truy cập http://localhost:5984/_utils/ sẽ hiển thị trang chủ của Fauxton



The screenshot shows the CouchDB Fauxton interface. On the left is a sidebar with navigation icons and a 'Log Out' button. The main area is titled 'Databases' and contains a table listing databases. The table has columns for Name, Size, # of Docs, and Actions. The databases listed are: _global_changes (15.7 KB, 18 docs), _replicator (2.3 KB, 1 doc), _users (2.3 KB, 1 doc), addressboo (168.2 KB, 1 doc), addressbook (0.6 MB, 2 docs), baseball (4.7 KB, 0 docs), couchdbtest (2.6 KB, 2 docs), and tasks (12.1 KB, 16 docs). Each row has three action icons: a double arrow, a lock, and a trash can. At the bottom right, it says 'Showing 1-8 of 8 databases.' and has a page number '1'.

Name	Size	# of Docs	Actions
_global_changes	15.7 KB	18	
_replicator	2.3 KB	1	
_users	2.3 KB	1	
addressboo	168.2 KB	1	
addressbook	0.6 MB	2	
baseball	4.7 KB	0	
couchdbtest	2.6 KB	2	
tasks	12.1 KB	16	

- CouchDB Fauxton tạo điều kiện cho bạn thực hiện các hoạt động sau:

- Databases:

✓ Create databases

The screenshot shows the CouchDB Databases interface. A table lists databases with columns for Name, Size, and # of Docs. A 'Create Database' modal is open, showing a text input for 'Name of database' and a 'Create' button. The modal is highlighted with a yellow circle.

Name	Size	# of Docs
_global_changes	15.7 KB	18
_replicator	2.3 KB	1
_users	2.3 KB	1
addressboo	168.2 KB	1
addressbook	0.6 MB	2
baseball	4.7 KB	0
couchdbtest	2.6 KB	2
tasks	12.1 KB	16 ⓘ

Showing 1–8 of 8 databases. « 1 »

✓ Delete databases

The screenshot shows the CouchDB Databases interface. A table lists databases with columns for Name, Size, # of Docs, and Actions. The 'Delete' icon (trash can) for the 'couchdbtest' database is highlighted with a yellow circle. A tooltip 'Delete couchdbtest' is visible over the icon.

Name	Size	# of Docs	Actions
_global_changes	This database failed to load.		
_replicator	2.3 KB	1	
_users	2.3 KB	1	
addressboo	168.2 KB	1	
addressbook	0.6 MB	2	
baseball	4.9 KB	0	
couchdbtest	2.6 KB	2	
tasks	12.1 KB	16 ⓘ	

Showing 1–8 of 8 databases. « 1 »

○ Documents:

✓ Create documents

localhost:5984/_utils/#/database/couchdbtest/new

Showing document 1 - 2. Documents per page: 30

✓ Update documents

```
1 {
2   "_id": "c5bf3d23c23958d6072323dfbd00431c",
3   "_rev": "4-bf7eee8ba671243f1f5eeb33a1784490",
4   "_attachments": {
5     "thru.php": {
6       "content_type": "application/octet-stream",
7       "revpos": 3,
8       "digest": "md5-tKV+8RI2ySn5+mFSKqub+g==",
9       "length": 1331,
10      "stub": true
11    }
12  }
13 }
```

Log Out

✓ Edit documents

The screenshot shows the CouchDB web interface. On the left is a sidebar with navigation links: All Documents, Run A Query with Mango, Permissions, Changes, Design Documents, and a bottom section for Faustus on Apache CouchDB v.2.2.0 with a Log Out button. The main area has a top bar with 'couchdbtest', a 'Document ID' dropdown, 'Options', 'JSON' view selector, and a 'Create Document' button. Below the top bar are tabs for 'Table', 'Metadata', and 'JSON'. The 'JSON' tab is active, displaying a document with the following JSON structure:

```
{
  "id": "c5bf3d23c23958d6072323dfbd00431c",
  "key": "c5bf3d23c23958d6072323dfbd00431c",
  "value": {
    "rev": "4-bf7eee8ba671243f1f5eeb33a1784490"
  },
  "doc": {
    "_id": "c5bf3d23c23958d6072323dfbd00431c",
    "_rev": "4-bf7eee8ba671243f1f5eeb33a1784490",
    "_attachments": {
      "thu.php": {
        "content_type": "application/octet-stream",
        "revpos": 3,
        "digest": "md5-tKV+8RIZySn5+mfSKqub+g==",
        "length": 1331,
        "stub": true
      }
    }
  }
}
```

An 'Edit document' button (pencil icon) is circled in yellow in the top right corner of the document view. At the bottom, it says 'Showing document 1 - 2. Documents per page: 30'.

✓ Delete documents

The screenshot shows the CouchDB web interface in the 'Table' view. The sidebar is the same as in the previous screenshot. The main area has a top bar with 'couchdbtest', a 'Document ID' dropdown, 'Options', 'JSON' view selector, and a 'Create Document' button. Below the top bar are tabs for 'Table', 'Metadata', and 'JSON'. The 'Table' tab is active, displaying a list of documents. A red trash can icon (delete button) is circled in yellow in the top left corner of the document list. The table has columns for '_id' and '_attachments'. Two documents are listed:

	_id	_attachments
<input checked="" type="checkbox"/>	c5bf3d23c23958d6072323dfbd00431c	{ "thu.php": { "content_type": "application/octet-stream", "rev..."
<input checked="" type="checkbox"/>	c5bf3d23c23958d6072323dfbd0054fd	

At the bottom, it says 'Showing 2 columns. Show all columns. Showing document 1 - 2. Documents per page: 30'.

➤ Tương tác CouchDB bằng cURL

- Công cụ này được sử dụng để truyền dữ liệu đến máy chủ bằng cách sử dụng một trong các giao thức được hỗ trợ như (HTTP, HTTPS, FTP, FTPS, TFTP, DICT, TELNET, LDAP hoặc FILE). Lệnh này được thiết kế để hoạt động mà không có sự tương tác của người dùng. cURL cung cấp một loạt các thủ thuật hữu ích như hỗ trợ proxy, xác thực người dùng, tải lên ftp, bài đăng HTTP, kết nối SSL (https :), cookie, tiếp tục truyền tệp và hơn thế nữa.

- cURL có sẵn cho các hệ điều hành như UNIX, Linux, Mac OS X và Windows. Nó là một command-line utility để truy cập giao thức HTTP ngay từ command-line utility.

○ Databases:

- ✓ Create databases

```
curl -X PUT http://127.0.0.1:5984/database_name
```

- ✓ Delete databases

```
curl -H 'Content-Type: application/json' \  
-X DELETE http://127.0.0.1:5984/database_name
```

○ Documents:

- ✓ Create documents

```
curl -X PUT http://127.0.0.1:5984/database_name/"id" -d '{ document }'
```

```
curl -H 'Content-Type: application/json' \  
-X PUT http://127.0.0.1:5984/my_database/"001" -  
d'{"Name": "Ajeet", "age": "23", "Designation": "Designer"}'
```

- ✓ Update documents

```
curl -X PUT http://127.0.0.1:5984/database_name/document_id/ -d '{ "field" : "value", "_rev" : "revision id" }'
```


- ✓ Delete documents

```
curl -H 'Content-Type: application/json' \  
-X DELETE http://127.0.0.1:5984/database_name/database_id?_rev
```

2.7. Cách truy vấn và xử lý CouchDB bằng Javascript

- Server Information

JavaScript example:

```
$.couch.info({  
  success: function(data) {  
    console.log(data);  
  }  
});
```

Console output:

```
{  
  "couchdb": "Welcome",  
  "version": "1.1.0"  
}
```

- ALL DBs

JavaScript example:

```
$.couch.allDbs({  
  success: function(data)  
    console.log(data);  
  }  
});
```

Console output:

```
[  
  "_replicator",  
  "_users"  
]
```

- User Signup

```
var userDoc = {
  _id: "org.couchdb.user:bob",
  name: "bob"
};
$.couch.signup(userDoc, "supersecurepassword",
  success: function(data) {
    console.log(data);
  },
  error: function(status) {
    console.log(status);
  }
});
```

Console output:

```
{
  "ok":true,
  "id":"org.couchdb.user:bob",
  "rev":"1-230dc0625bd3c4aac735846cc152c296"
}
```

- Login

```
$.couch.login({
  name: "bob",
  password: "supersecurepassword",
  success: function(data) {
    console.log(data);
  },
  error: function(status) {
    console.log(status);
  }
});
```

Console output:

```
{
  "ok":true,
  "name":"bob",
  "roles":[]
}
```

- Logout

```
$.couch.logout({
  success: function(data) {
    console.log(data);
  }
});
```

Console output:

```
{
  "ok":true
}
```

- Server Configuration

```
$.couch.config({
  success: function(data) {
    console.log(data);
  }
}, "uuids", "algorithm");
```

Console output:

```
"sequential"
```

- Create Database

```
$.couch.db("mydb").create({
  success: function(data) {
    console.log(data);
  },
  error: function(status) {
    console.log(status);
  }
});
```

Console output:

```
{
  "ok":true
}
```

- Save new document

```
var doc = {};  
$.couch.db("mydb").saveDoc(doc, {  
  success: function(data) {  
    console.log(data);  
  },  
  error: function(status) {  
    console.log(status);  
  }  
});
```

Console output:

```
{  
  "ok":true,  
  "id":"d12ee5ea1df6baa2b06451f44a019ab9",  
  "rev":"1-967a00dff5e02add41819138abb3284d"  
}
```

- Open Document

```
$.couch.db("mydb").openDoc("d12ee5ea1df6baa2b06451f44a019ab9", {  
  success: function(data) {  
    console.log(data);  
  },  
  error: function(status) {  
    console.log(status);  
  }  
});
```

- Save updated document

```
var doc = {
  _id: "d12ee5ea1df6baa2b06451f44a019ab9",
  _rev: "1-967a00dff5e02add41819138abb3284d",
  foo: "bar"
};
$.couch.db("mydb").saveDoc(doc, {
  success: function(data) {
    console.log(data);
  },
  error: function(status) {
    console.log(status);
  }
});
```

- Remove document

```
var doc = {
  _id: "d12ee5ea1df6baa2b06451f44a019ab9",
  _rev: "2-13839535feb250d3d8290998b8af17c3"
};
$.couch.db("mydb").removeDoc(doc, {
  success: function(data) {
    console.log(data);
  },
  error: function(status) {
    console.log(status);
  }
});
```

Console output:

```
{
  "id":d12ee5ea1df6baa2b06451f44a019ab9
  "ok":true,
  "rev":"3-1f04f977685e1108b4664f70b09c6f65"
}
```

- Drop database

```
$.couch.db("mydb").drop({
  success: function(data) {
    console.log(data);
  },
  error: function(status) {
    console.log(status);
  }
});
```

CHƯƠNG III: CÀI ĐẶT DEMO

TÀI LIỆU THAM KHẢO

- [1]. <http://docs.couchdb.org>
- [2]. <https://www.tutorialspoint.com/couchdb>
- [3]. <https://www.javatpoint.com>
- [4]. <https://www.w3schools.com>
- [5]. <https://www.ibm.com/developerworks/library/os-php-couchdb>