

## Experiment No :- 10

**Title :-** NoSQL Database - CouchDB.

**Aim :-** Install CouchDB on windows, create and delete CouchDB database . Run CouchDB query.

**Theory :-**

- **NoSQL Database - CouchDB**

CouchDB is an open-source, NoSQL database developed by the Apache Software Foundation. It's designed to be highly fault-tolerant, distributed, and easily scalable, which makes it well-suited for web and mobile applications.

- **Key Features of CouchDB**

### 1. Document-Oriented Database

**NoSQL:** CouchDB stores data as JSON documents rather than in traditional rows and columns like relational databases. Each document is a self-contained data structure with a unique ID.

**Schema-Free:** Documents can have different structures (schemas), allowing flexible and dynamic data models. This is perfect for applications where the structure of data evolves over time.

### 2. RESTful HTTP API

CouchDB uses a **RESTful HTTP/JSON API** to interact with the database. You can perform CRUD (Create, Read, Update, Delete) operations through HTTP requests. This makes it easy to integrate with any web technology that supports HTTP.

Example API requests:

- **Create a document:** POST /db
- **Read a document:** GET /db/docid
- **Update a document:** PUT /db/docid
- **Delete a document:** DELETE /db/docid

### 3. Multi-Version Concurrency Control (MVCC)

CouchDB uses **MVCC** to manage concurrent updates. When a document is updated, CouchDB doesn't overwrite the existing data but creates a new version, keeping a history of previous versions. This ensures data integrity and allows for conflict resolution in distributed systems.

Each document includes a `_rev` field (revision number) that gets updated with every change, ensuring optimistic concurrency control.

#### 4. Replication and Synchronization

CouchDB is designed to be highly distributed and provides **master-master replication** (multi-node synchronization) where changes on one node can be replicated to others.

**Incremental replication** means that only changes are replicated rather than the entire database.

This makes CouchDB ideal for applications like offline-first mobile apps, where data can be synchronized across devices and servers.

#### 5. Mango Queries (Declarative Query Language)

CouchDB includes a declarative query language called Mango, which is similar to MongoDB's query syntax. It allows developers to run JSON-based queries without needing to write MapReduce views.

Mango queries are run using the `_find` endpoint, making querying easier for non-programmers.

#### 6. MapReduce Views for Querying

CouchDB uses **MapReduce functions** for building views to query data. Views are written in JavaScript and allow you to emit key-value pairs that CouchDB can index and query.

**Map function:** Defines the key-value pairs to be emitted based on the document.

**Reduce function:** Aggregates the emitted values.

Views can be stored in **design documents**, and CouchDB automatically updates them as documents change.

#### 7. Mango Queries (Declarative Query Language)

CouchDB includes a declarative query language called **Mango**, which is similar to MongoDB's query syntax. It allows developers to run JSON-based queries without needing to write MapReduce views.

Mango queries are run using the `_find` endpoint, making querying easier for non-programmers.

- **History of CouchDB:**

CouchDB was first released in 2005 and later became an [Apache Software Foundation](http://www.apache.org) project in 2008.

Unlike a [relational database](#), a CouchDB database does not store data and relationships in tables. Instead, each database is a collection of independent documents.

Official Online Resources <http://couchdb.apache.org> and [www.couchbase.com](http://www.couchbase.com).

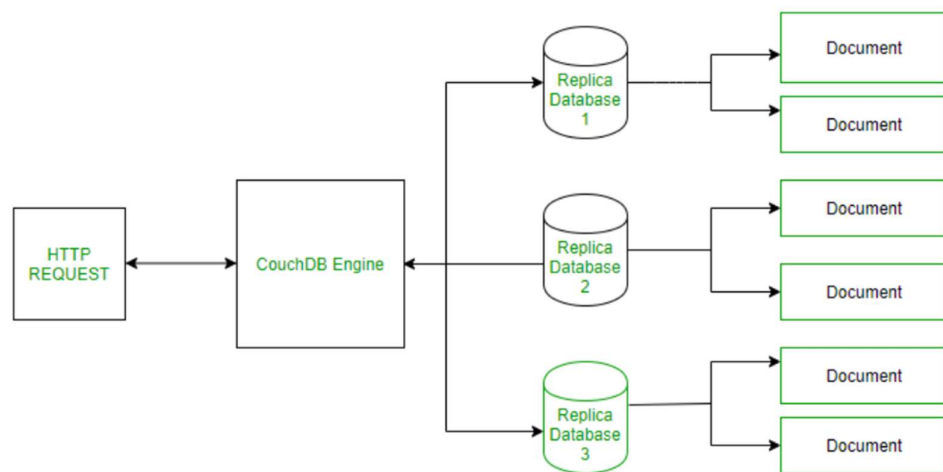
Most of the authors are part of Couchbase. Inc.

Access Methods Upholds REST above every other mechanism. Use standard web tools and clients to access the database, the same way as you access web resources

Open-Source License Apache License version 2.

Who Uses It-Apple, BBC, Canonical, Cern, and more at [http://wiki.apache.org/couchdb/CouchDB\\_in\\_the\\_wild](http://wiki.apache.org/couchdb/CouchDB_in_the_wild)

- **Architecture :**



**Architecture Of CouchDB**

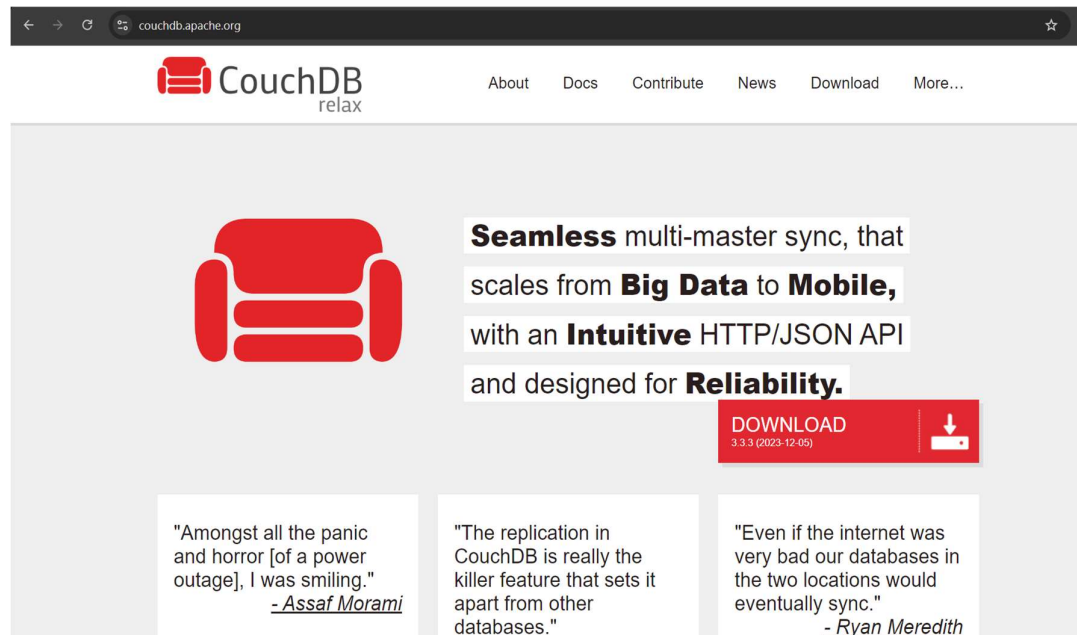
The architecture of CouchDB is described below:

1. **CouchDB Engine:** It is based on B-tree and in it, data is accessed by keys or key ranges which map directly to the underlying B-tree operations. It is the core of the system which manages to store internal data, documents, and views.
2. **HTTP Request:** It is used to create indices and extract data from documents. It is written in JavaScript that allows creating Adhoc views that are made of MapReduce jobs.
3. **Document:** It stores a large amount of data.
4. **Replica Database:** It is used for replicating data to a local or remote database and synchronizing design documents.

## 1. Install CouchDB on Windows

### Step 1: Download CouchDB

Go to the official CouchDB website: CouchDB Downloads.



Select the Windows version and download the installer.

### Step 2: Run the Installer

Once downloaded, run the installer.

Follow the installation prompts:

Choose the installation directory (default is usually fine).

Allow CouchDB to be installed as a Windows Service so it starts automatically when Windows starts.

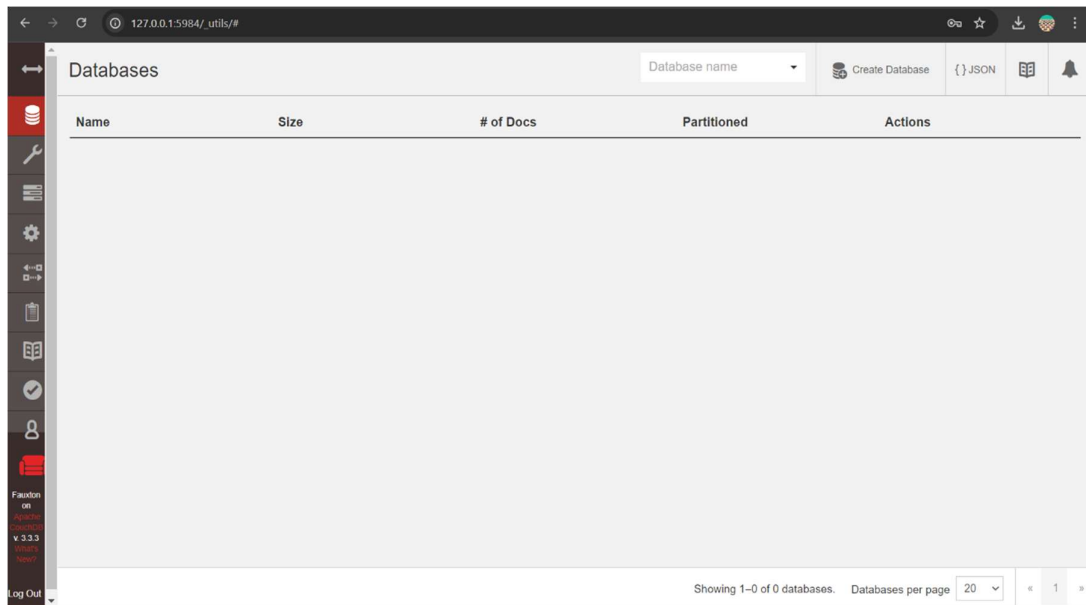
Choose to run CouchDB as a single node (if it's a personal or small project).

### Step 3: Verify Installation

After installation, CouchDB should start automatically.

Open a web browser and navigate to [http://127.0.0.1:5984/\\_utils/](http://127.0.0.1:5984/_utils/).

This should bring up the Fauxton web interface, CouchDB's graphical user interface.



## 2. Create a CouchDB Database

You can create a CouchDB database using either the Fauxton web interface or using an HTTP request via curl or Postman.

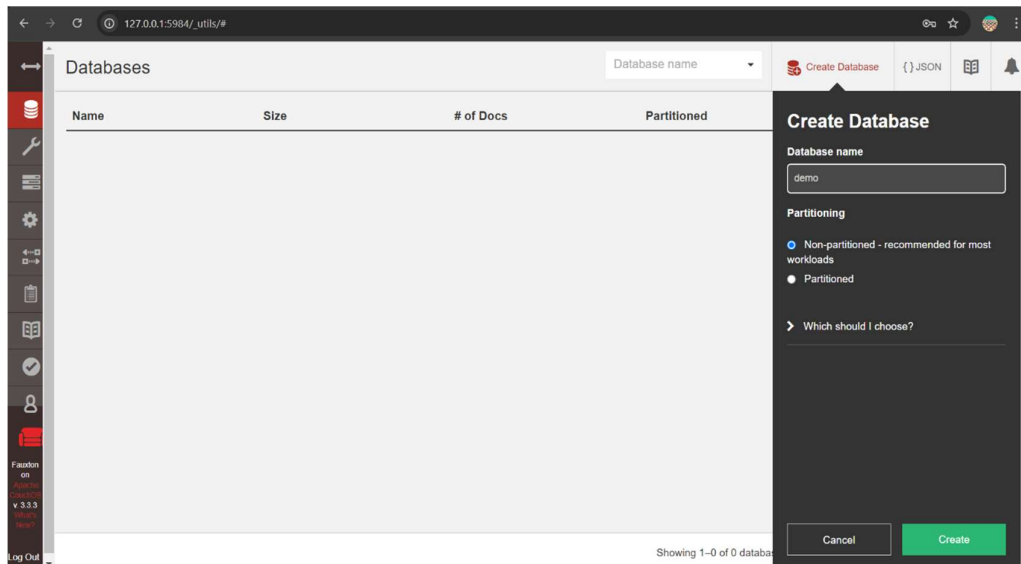
### Method 1: Using Fauxton Web Interface

Open Fauxton at [http://127.0.0.1:5984/\\_utils/](http://127.0.0.1:5984/_utils/).

In the left sidebar, click the Database icon.

Click the Create Database button.

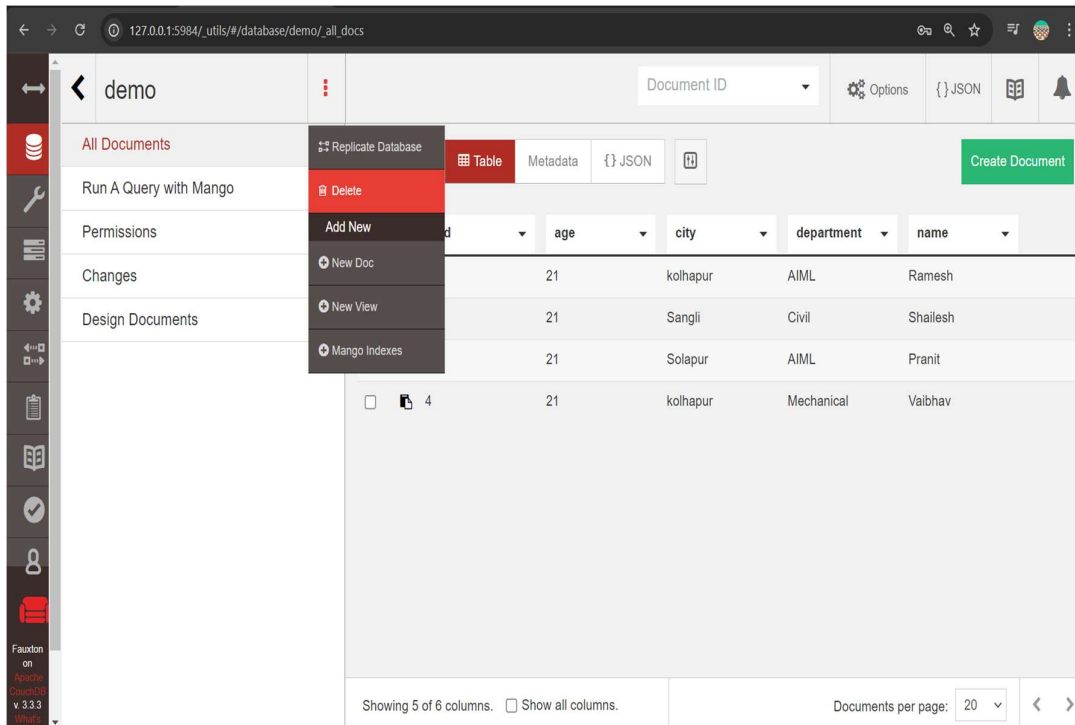
Enter a name for your database (e.g., mydatabase) and click Create.



### 3. Delete a CouchDB Database

#### Method 1: Using Fauxton Web Interface

1. Go to **Fauxton** at **[http://127.0.0.1:5984/\\_utils/](http://127.0.0.1:5984/_utils/)**.
2. In the left sidebar, click the **Database** icon.
3. Find the database you want to delete and click the **Delete** icon next to it.
4. Confirm the deletion.

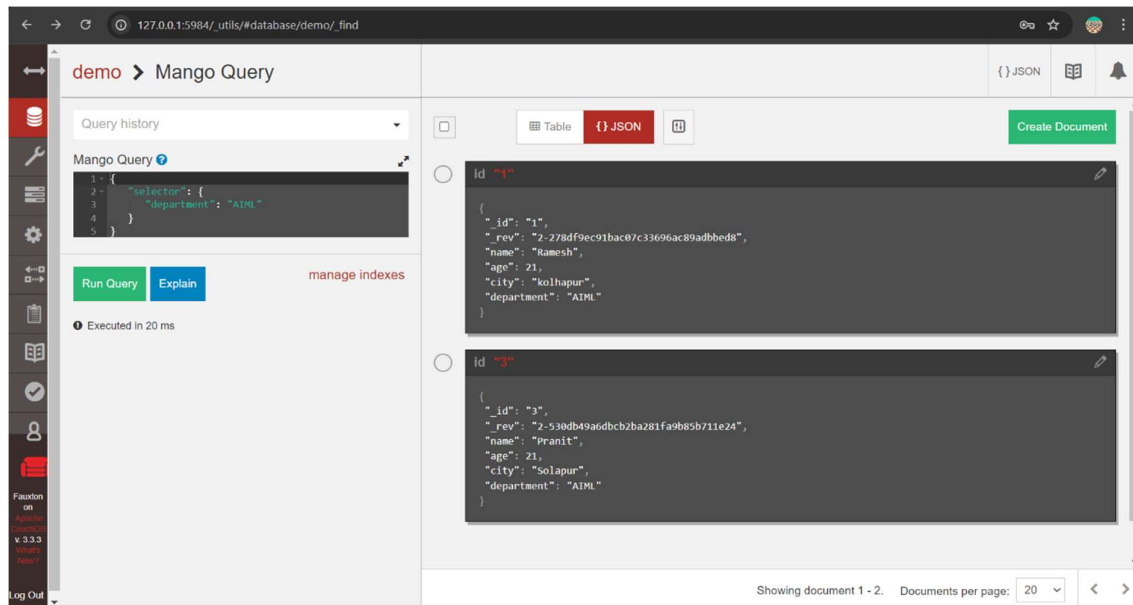


#### CouchDB Mango Queries:

Mango queries are structured JSON queries with a **selector** to filter the data based on the conditions you provide.

#### Example 1: Select All students from "AIML" Department

To retrieve all students who belong to the AIML department, the Mango query would look like this:



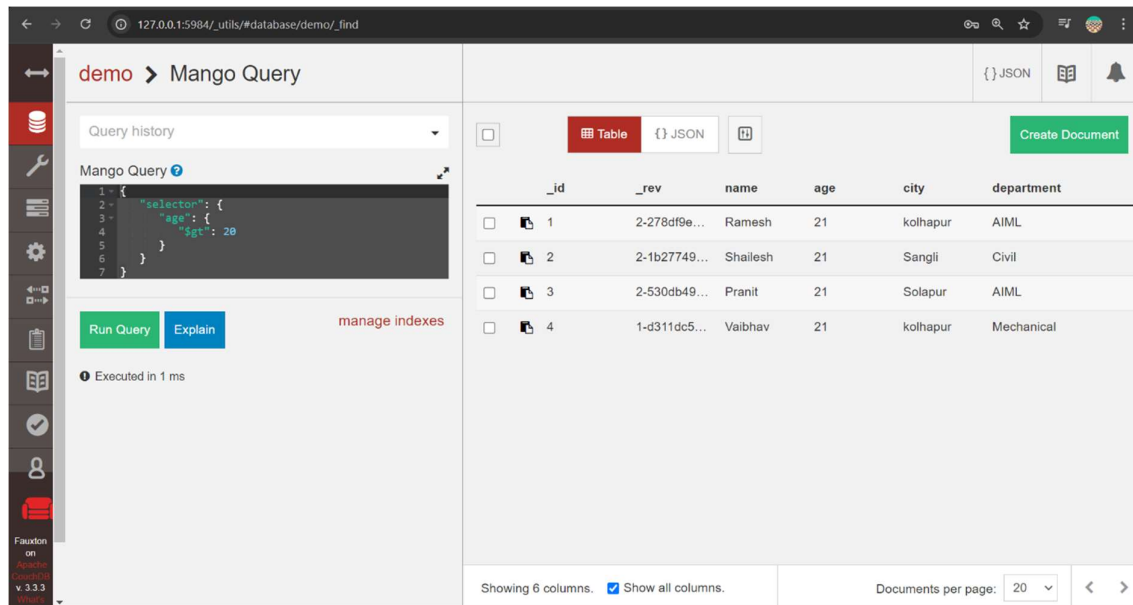
### Example 2: Find students Aged Greater Than 30

Let's now query for employees where age is greater than 30.

```
{
  "selector": {
    "age": { "$gt": 20 }
  }
}
```

#### Explanation:

- **"age": { "\$gt": 20 }:** This condition specifies that the field age must be greater than (\$gt) 20



- **Database Connectivity with Python:-**

### Steps to Connect CouchDB with Python:

#### 1. Install the CouchDB Python Library

First, you need to install the couchdb Python package using pip:

```
pip install couchdb
```

#### 2. Connecting to CouchDB

Here's a basic example to show how to connect to a CouchDB server and create a database or use an existing one:

```
import couchdb
```

```
# Step 1: Connect to CouchDB server
```

```
# Replace with your CouchDB server URL
```

```
couch = couchdb.Server("http://username:password@localhost:5984/")
```

```
# Step 2: Check if the database exists, else create it
```

```
db_name = 'my_database'
```

```
if db_name in couch:
```

```
    db = couch[db_name]
```

```
else:
```

```
    db = couch.create(db_name)
```

```
# Step 3: Insert a document
```

```
doc = {
```

```
    'type': 'person',
```



```

    'name': 'John Doe',
    'age': 30
}
doc_id, doc_rev = db.save(doc)
print(f'Document created with ID: {doc_id} and Revision: {doc_rev}')

```

#### **Explanation:**

1. **Connection to CouchDB:** Server("http://localhost:5984/") is used to connect to your CouchDB instance. If it's hosted remotely, replace the URL with the appropriate one.
2. **Database Access:** Check if the database exists with if db\_name in couch. If not, couch.create(db\_name) will create the database.
3. **Inserting Documents:** You can insert a document (data) into the database using db.save(doc).

### **3. Fetching Data from CouchDB**

Here's how you can retrieve documents from the database:

```
# Fetch all documents
```

```
for doc_id in db:
```

```
    doc = db[doc_id]
```

```
    print(doc)
```

This code retrieves all documents in the database and prints them.

### **4. Advanced Operations**

You can also perform various operations like updating, deleting documents, and querying using the \_id of the documents.

- **Updating a Document:**

```
doc = db[doc_id]
```

```
doc['age'] = 31 # Modify the document
```

```
db.save(doc) # Save the updated document
```

- **Deleting a Document:**

```
db.delete(doc) # Delete the document
```

- **Conclusion:**

Thus, we have studied about Installing CouchDB on windows, create and delete CouchDB database . Run CouchDB query