

NoSQL: CouchDB

CouchDB

- dokumentorientierte Datenbank
- Ziel: Verknüpfung des einfachen Datenmodells einer dokumentenorientierten Datenbank mit der Skalierbarkeit und Leistung einer relationalen Datenbank
- Speicherung der Daten in Dokumenten, die aus JSON-Objekten bestehen
- Einsatzgebiete: Softwareprojekte, Webseiten, Facebook-Anwendungen

Installation und Start des Servers: <https://docs.couchdb.org/en/stable/install/index.html>

JavaScript Object Notation

- kompaktes Datenformat; benutzerfreundlich, rein textbasiert, leicht lesbar
- Einsatzgebiet: Speicherung/Übertragung strukturierter Daten
- plattform- und sprachunabhängig, Parser in allen verbreiteten Sprachen

JSON-Objekt

- von geschweiften Klammern umgeben
- enthält Schlüssel-Wert-Paare, die durch " : " getrennt sind
- Schlüssel in Anführungszeichen
- mögliche Werte: JSON-Objekte, Zahlen, Zeichenfolgen, Arrays, null, true, false

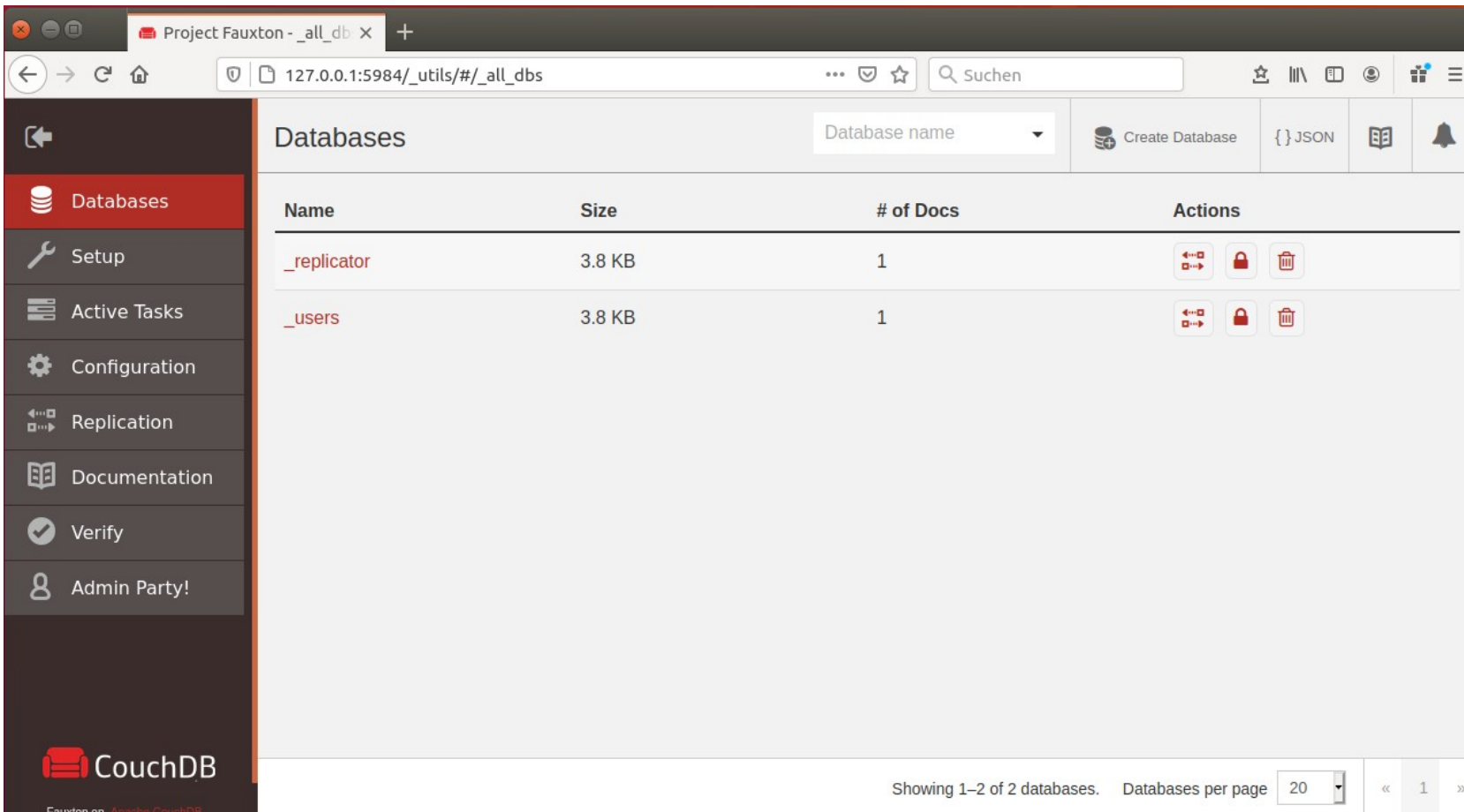
```
{
  "key1": "value1",
  "key2": "value2",
  ...
}
```

Beispiel für Werte verschiedener Datentypen:

```
{
  "myString" : "mystring",
  "myInt" : 5,
  "myArrayOfDoubles" : [ 3.14, 2.72 ],
  "myArrayOfStrings" : ["red" ,"green", "blue"],
  "objectInObject" : {}
}
```

Fauxton

- JavaScript-UI zur Administration der CouchDB ab Version 2.0
- erreichbar im Browser über http://127.0.0.1:5984/_utils/
- ausführliche visualisierte Anleitung für die Arbeit mit Fauxton:
<https://couchdb.apache.org/fauxton-visual-guide/#intro>



The screenshot shows the Fauxton web interface for CouchDB. The browser address bar displays `127.0.0.1:5984/_utils/#/_all_dbs`. The left sidebar contains navigation links: Databases (selected), Setup, Active Tasks, Configuration, Replication, Documentation, Verify, and Admin Party!. The main content area is titled 'Databases' and features a table with the following data:

Name	Size	# of Docs	Actions
<code>_replicator</code>	3.8 KB	1	[Replicate] [Lock] [Delete]
<code>_users</code>	3.8 KB	1	[Replicate] [Lock] [Delete]

At the bottom of the interface, it indicates 'Showing 1-2 of 2 databases.' and 'Databases per page' set to 20. The CouchDB logo is visible in the bottom left corner.

Databases

- Datenbank ist dokumentbasiert
- Datensätze sind in sich geschlossene Dokumente (Design Documents)

Beispiel: Anlegen einer Datenbank buecher im Fauxton

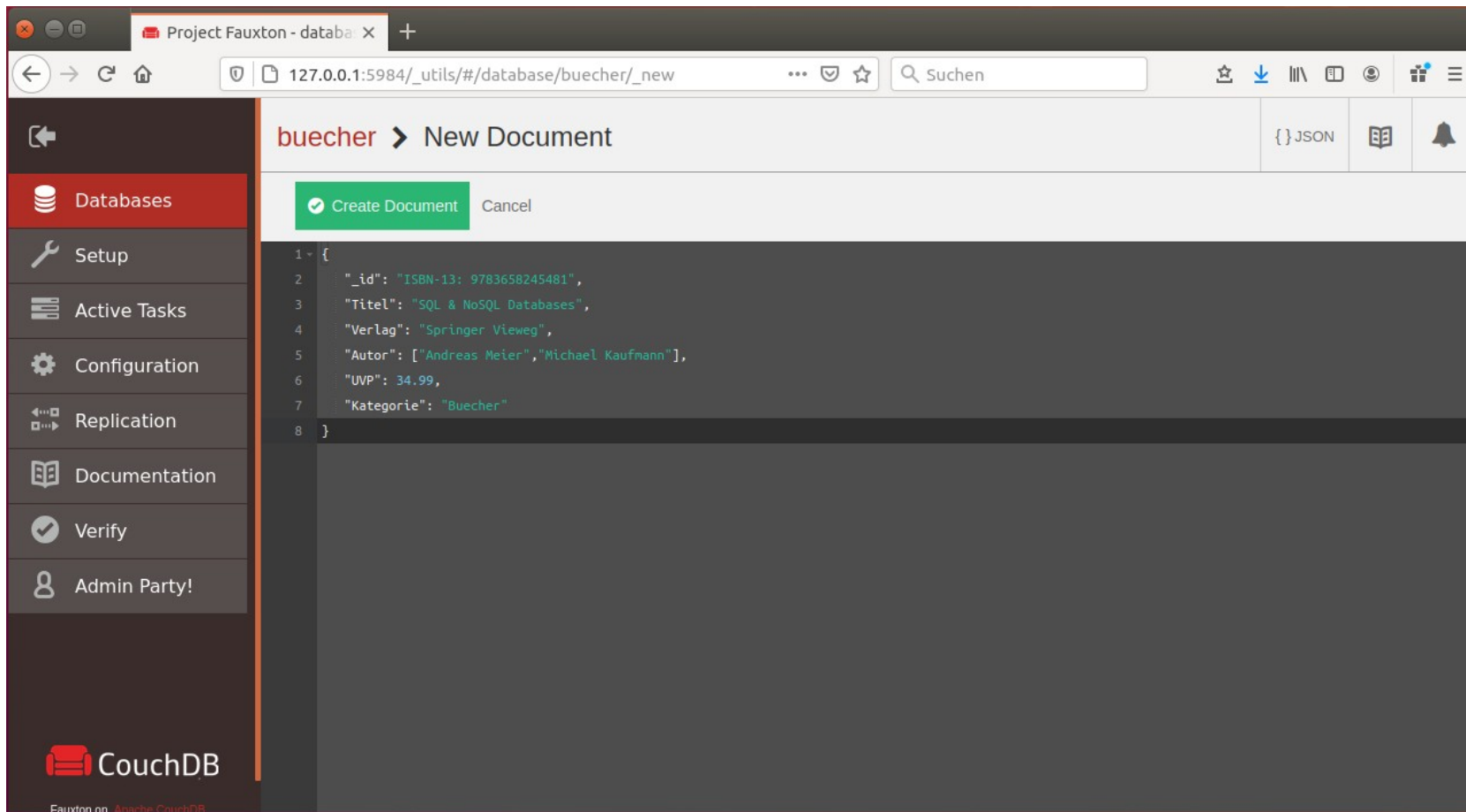
The screenshot shows the Fauxton web interface for a CouchDB instance. The browser tab is 'Project Fauxton - _all_db'. The address bar shows '127.0.0.1:5984/_utils/#/_all_dbs'. The left sidebar contains navigation links: Databases, Setup, Active Tasks, Configuration, Replication, Documentation, Verify, and Admin Party!. The main content area is titled 'Databases' and features a table with the following data:

Name	Size	# of Docs
_replicator	3.8 KB	1
_users	3.8 KB	1


A 'Create Database' modal is open, showing a text input field with the value 'buecher' and a red 'Create' button. The modal also includes icons for zooming, locking, and deleting. At the bottom of the interface, it says 'Showing 1-2 of 2 databases. Databases per page 20'.

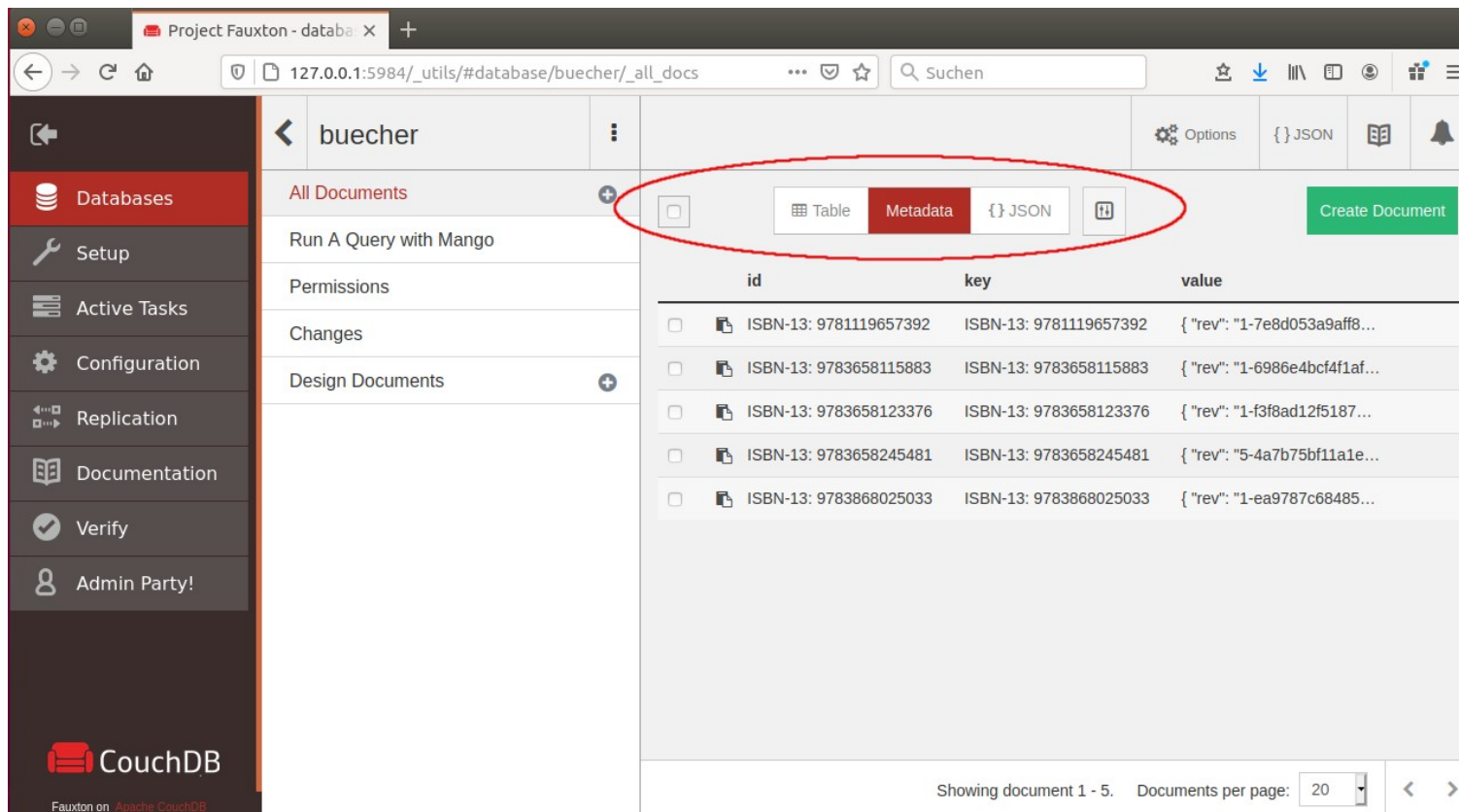
Documents I

- Daten werden in Form von JSON-Dokumenten mit einer `_id` in der Datenbank gespeichert
- beim Anlegen oder Bearbeiten von Dokumenten wird von CouchDB zusätzlich eine Versionsnummer (`_rev`-Attribut) automatisch zugewiesen
- die Versionsnummer ist ein Hashwert, der angibt, wie oft das Dokument bereits aktualisiert wurde








Documents II

- Anzeige der Dokumente in verschiedenen Formaten:
 - Metadata-Register: als (key,value)-Paar
 - Table-Register: komplette Datensätze
 - {}JSON-Register: im JSON-Format
- die meisten Werte sind aufgrund von Platzmangel abgeschnitten - beim MouseOver werden sie vollständig angezeigt
- über den Button  kann die Anzeige-Grösse verstellt werden

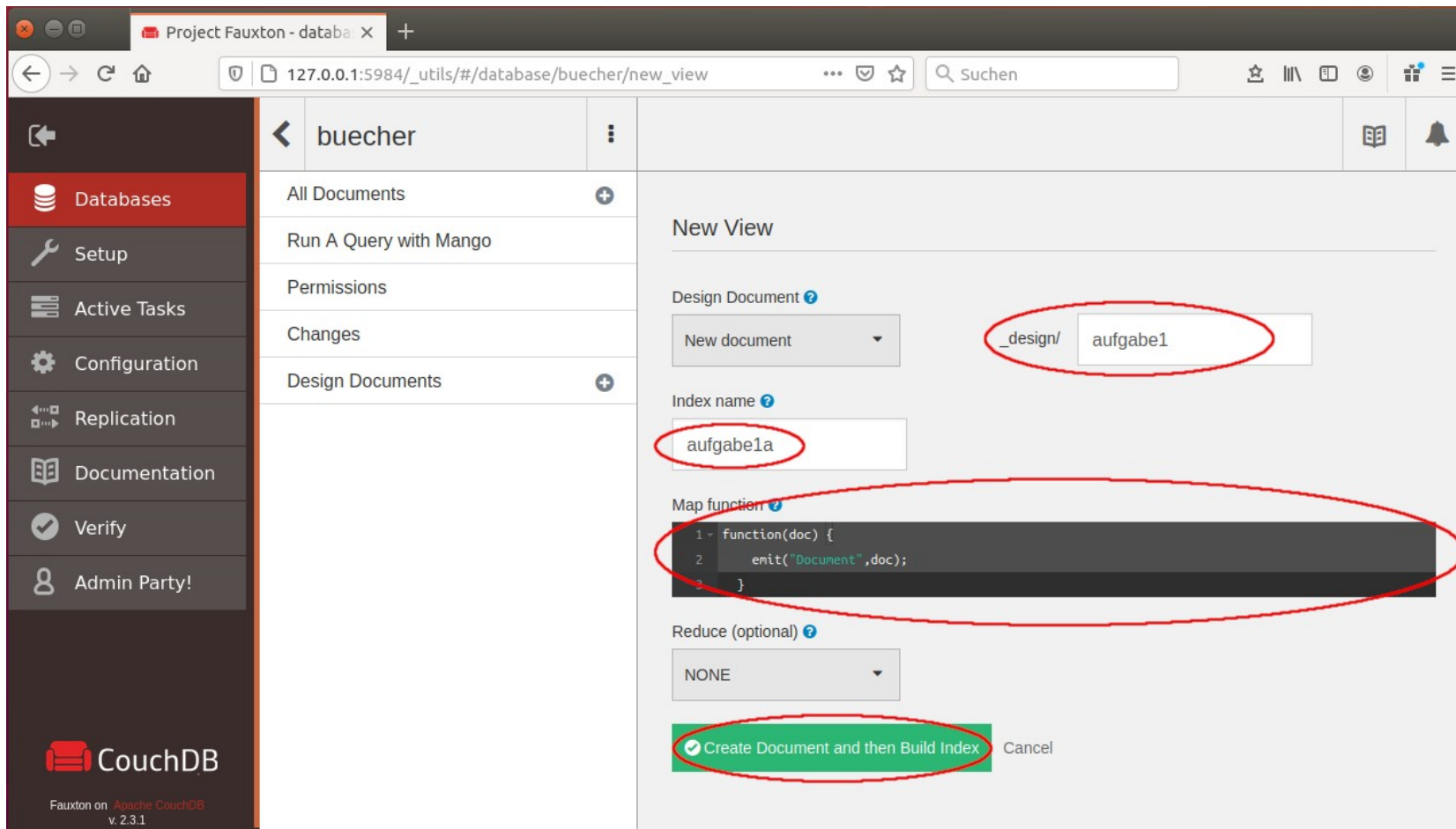


The screenshot shows the Project Fauxton web interface for a CouchDB database named 'buecher'. The left sidebar contains navigation links: Databases, Setup, Active Tasks, Configuration, Replication, Documentation, Verify, and Admin Party!. The main content area displays a list of documents under the 'buecher' database. At the top of the document list, there are buttons for 'Table', 'Metadata', and '{} JSON'. The 'Table' button is currently selected and highlighted with a red oval. Below these buttons, a table displays document data with columns 'id', 'key', and 'value'. The 'value' column contains truncated JSON objects, which are shown in full on mouseover. At the bottom right, there is a 'Create Document' button and a pagination control showing 'Showing document 1 - 5' and 'Documents per page: 20'.

id	key	value
<input type="checkbox"/>  ISBN-13: 9781119657392	ISBN-13: 9781119657392	{ "rev": "1-7e8d053a9aff8..." }
<input type="checkbox"/>  ISBN-13: 9783658115883	ISBN-13: 9783658115883	{ "rev": "1-6986e4bcf4f1af..." }
<input type="checkbox"/>  ISBN-13: 9783658123376	ISBN-13: 9783658123376	{ "rev": "1-f3f8ad12f5187..." }
<input type="checkbox"/>  ISBN-13: 9783658245481	ISBN-13: 9783658245481	{ "rev": "5-4a7b75bf11a1e..." }
<input type="checkbox"/>  ISBN-13: 9783868025033	ISBN-13: 9783868025033	{ "rev": "1-ea9787c68485..." }

Anfragen I

- Anfragen werden als Views in **DesignDocuments** gespeichert
- Erzeugung unter DesignDocuments → New View



Anfragen II

- Anzeige der **Lösung** zur Anfrage im Register Metadata
- Anfragen können editiert werden

The screenshot shows the Project Fauxton interface for a CouchDB database named 'buecher'. The left sidebar contains navigation links: Databases, Setup, Active Tasks, Configuration, Replication, Documentation, Verify, and Admin Party!. The main content area displays the 'buecher' database with a list of documents. The 'aufgabe1' view is selected, and the 'aufgabe1a' view is highlighted. The 'Metadata' tab is selected, showing a table with columns 'id', 'key', and 'value'. A tooltip shows the JSON metadata for a document.

id	key	value
ISBN-13: 9781119657392	Document	{ "_id": "ISBN-13: 9781119657392", "rev": "1-7e8d053a9aff818af86eba43e902ffd6", "Titel": "Hands-On NoSQL", "Verlag": "JOHN WILEY & SONS INC.", "Autor": ["Arsames Qajarl", "Dan Sullivan"], "UVP": 37.5, "Kategorie": "Buecher" }
ISBN-13: 9783658115883	Document	{ "_id": "ISBN-13: 9783658115883", "rev": "1-7e8d053a9aff818af86eba43e902ffd6", "Titel": "Hands-On NoSQL", "Verlag": "JOHN WILEY & SONS INC.", "Autor": ["Arsames Qajarl", "Dan Sullivan"], "UVP": 37.5, "Kategorie": "Buecher" }
ISBN-13: 9783658123376	Document	{ "_id": "ISBN-13: 9783658123376", "rev": "1-7e8d053a9aff818af86eba43e902ffd6", "Titel": "Hands-On NoSQL", "Verlag": "JOHN WILEY & SONS INC.", "Autor": ["Arsames Qajarl", "Dan Sullivan"], "UVP": 37.5, "Kategorie": "Buecher" }
ISBN-13: 9783658245481	Document	{ "_id": "ISBN-13: 9783658245481", "rev": "1-7e8d053a9aff818af86eba43e902ffd6", "Titel": "Hands-On NoSQL", "Verlag": "JOHN WILEY & SONS INC.", "Autor": ["Arsames Qajarl", "Dan Sullivan"], "UVP": 37.5, "Kategorie": "Buecher" }
ISBN-13: 9783868025033	Document	{ "_id": "ISBN-13: 9783868025033", "rev": "1-7e8d053a9aff818af86eba43e902ffd6", "Titel": "Hands-On NoSQL", "Verlag": "JOHN WILEY & SONS INC.", "Autor": ["Arsames Qajarl", "Dan Sullivan"], "UVP": 37.5, "Kategorie": "Buecher" }

Map-Funktion

- Anfrage-Funktion, wird in JavaScript geschrieben
- einziges Funktionsargument: `doc` - enthält ein einzelnes Dokument der Datenbank

```
function(doc) {  
  emit("Document",doc);  
}
```

- die `emit()`-Funktion erwartet zwei Argumente: `key` (Schlüssel) und `value` (Wert)
- der Aufruf `emit(key,value)` erzeugt einen Eintrag in der View-Ergebnismenge
- Ergebnis der Anfrage sind wieder Schlüssel-Wert-Paare
- innerhalb einer Map-Funktion kann `emit()` auch mehrmals aufgerufen werden

Beispiel 1: Ausgabe von Titel und Autor aller Bücher aus dem Verlag "Springer Vieweg"

```
function (doc) {  
  if(doc.Verlag=="Springer Vieweg")  
    emit(doc.Titel,doc.Autor);  
}
```

Beispiel 2: Ausgabe von Titel und Preis aller Bücher von "Andreas Meier"

```
function(doc) {  
  for (var x in doc.Autor){  
    if (doc.Autor[x] == "Andreas Meier")  
      emit(doc.Titel, doc.UVP);  
  }  
}
```

Reduce

- Reduce-Funktionen werden immer auf die Values einer vorangegangenen Map-Funktion angewendet
- dienen der Zusammenfassung von Ergebnissen
- built-in-Funktionen `_sum`, `_count`, `_stats` etc.
- benutzerdefinierte Reduce-Funktionen möglich

Beispiel: Anzahl der Bücher aus dem Verlag "Springer Vieweg"

Map: Anfrage mit value=1

```
function (doc) {  
    if(doc.Verlag=="Springer Vieweg")  
        emit(doc.Verlag,1);  
}
```

Reduce: Berechnung der Anzahl mit `_sum`

```
function(key,values,reduce) {  
    return sum(values)  
}  
  
//oder nur:  
  
_sum
```

Reduce im Fauxton I

- Erzeugung der Map-Funktion
- Auswahl der Reduce-Methode

The screenshot shows the Fauxton web interface for a CouchDB database named 'buecher'. The left sidebar contains navigation links: Databases, Setup, Active Tasks, Configuration, Replication, Documentation, Verify, and Admin Party!. The main content area is titled 'Edit View' and displays the configuration for a design document named '_design/aufgabe1'. The 'Index name' is set to 'aufgabe1d'. The 'Map function' is a JavaScript function that emits the value 1 for documents where 'doc.Verlag' is 'Springer Vieweg'. The 'Reduce (optional)' is set to '_sum'. The 'Map function' and 'Reduce (optional)' fields are circled in red. At the bottom, there is a green button labeled 'Save Document and then Build Index' and a 'Cancel' button.

```
function (doc) {  
  if(doc.Verlag=="Springer Vieweg")  
    emit(doc.Verlag,1);  
}
```

Reduce (optional): _sum

Save Document and then Build Index Cancel

Reduce im Fauxton II

- Ergebnisliste wird (ohne Reduce) angezeigt

The screenshot shows the Fauxton II web interface for a CouchDB database named 'buecher'. The left sidebar contains navigation links: Databases, Setup, Active Tasks, Configuration, Replication, Documentation, Verify, and Admin Party!. The main content area displays a table of documents under the 'aufgabe1' design document. The table has three columns: 'id', 'key', and 'value'. The 'value' column contains the number 1 for all three documents, which are circled in red. The 'Options' button in the top right is also circled in red.

id	key	value
ISBN-13: 9783658115883	Springer Vieweg	1
ISBN-13: 9783658123376	Springer Vieweg	1
ISBN-13: 9783658245481	Springer Vieweg	1

Showing document 1 - 3. Documents per page: 20

Reduce im Fauxton III

- Aktivierung von Reduce in den Options

The screenshot shows the Fauxton III web interface for a CouchDB database named 'buecher'. The left sidebar contains navigation links: Databases, Setup, Active Tasks, Configuration, Replication, Documentation, Verify, and Admin Party!. The main content area displays a list of documents under the 'aufgabe1' view, with columns for 'id' and 'ISBN-13'. A 'Query Options' dialog box is open, showing settings for the query. The 'Reduce' option is selected and circled in red. The 'Update' field is set to 'true'. The 'Run Query' button is also circled in red. The bottom status bar indicates 'Showing document 1 - 3' and 'Documents per page: 20'.

Project Fauxton - databa X 3.1. Design Documents - X +

127.0.0.1:5984/_utils/#database/buecher/_design/aufgabe1/_view ... Suchen

buecher

Options {} JSON

Query Options ?

☐ Include Docs ☒ Reduce Group Level Exact

☐ Stable Update true

Keys

By Key(s) Between Keys

Additional Parameters

Limit None

☐ Descending Skip # of rows

Run Query Cancel

CouchDB

Fauxton on Apache CouchDB v. 2.3.1

Showing document 1 - 3. Documents per page: 20

Reduce im Fauxton IV

- Ergebnis wird angezeigt

The screenshot shows the Fauxton IV web interface for a CouchDB database named 'buecher'. The left sidebar contains navigation links: Databases, Setup, Active Tasks, Configuration, Replication, Documentation, Verify, and Admin Party!. The main content area displays a list of documents under the 'buecher' database. The 'aufgabe1' document is selected, and its 'Views' section is expanded, showing four views: 'aufgabe1a', 'aufgabe1b', 'aufgabe1c', and 'aufgabe1d'. The 'aufgabe1d' view is currently selected, and its content is displayed in the main area. The content shows a single document with the key 'Springer Vieweg' and the value '3'. The interface includes a search bar at the top, a 'Create Document' button, and a footer indicating 'Showing document 1 - 1. Documents per page: 20'.

key	value
Springer Vieweg	3