

Document Database Basics Couchbase

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25,
  "address":
  {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021"
  },
  "phoneNumber":
  [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "fax",
      "number": "646 555-4567"
    }
  ]
}
```

JSON Documents

- Map more closely to external API
- CRUD Operations, lightweight and implicit schema

```
myDocument = {  
  "fields" : ["with basic types", 3.14159, true],  
  "like" : "your favorite language.",  
  "status": {  
    "apis" : true,  
    "databases" : "document" }  
}
```

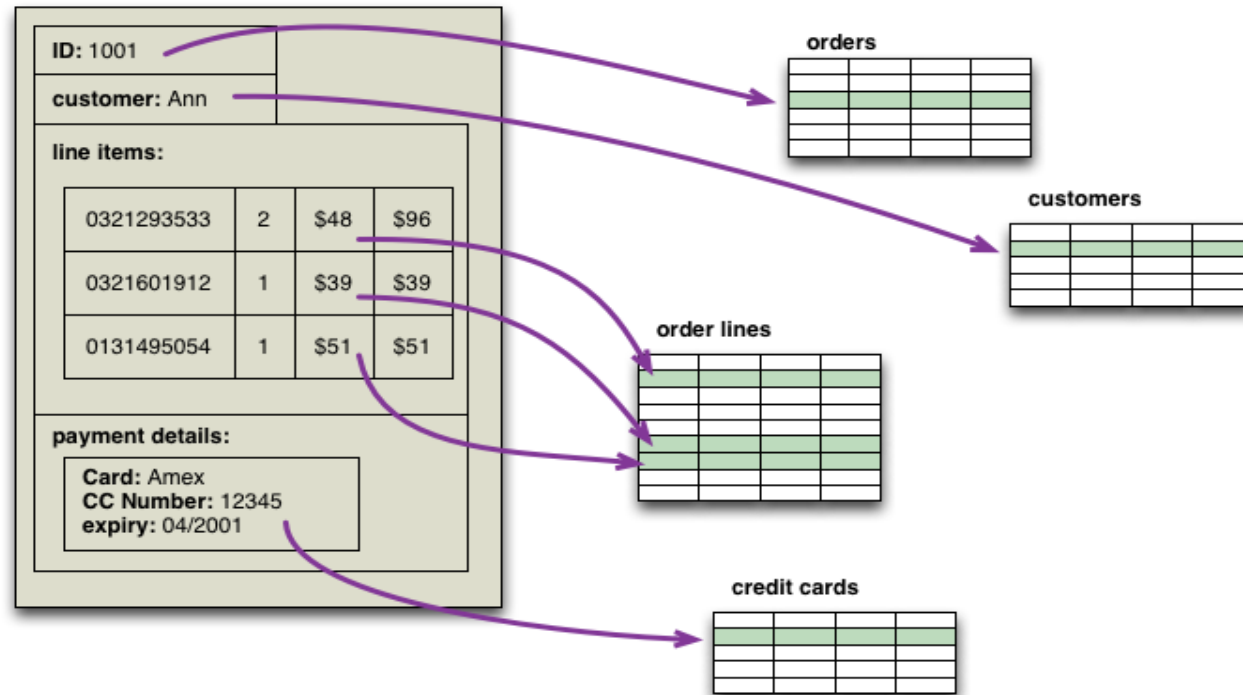
- Stored under a unique identifier key

```
client.set("mydocumentid", myDocument);  
mySavedDocument = client.get("mydocumentid");
```

Benefits of JSON

- Can Represent Complex Objects and Data Structures
- Very simple notation, lightweight, compact, readable
- The most common API return type for Integrations
 - Facebook, Twitter, you name it, return JSON
 - Native to Javascript (can be useful)
 - Can be inserted straight into Couchbase (faster development)
- Serialization and Deserialization are very fast

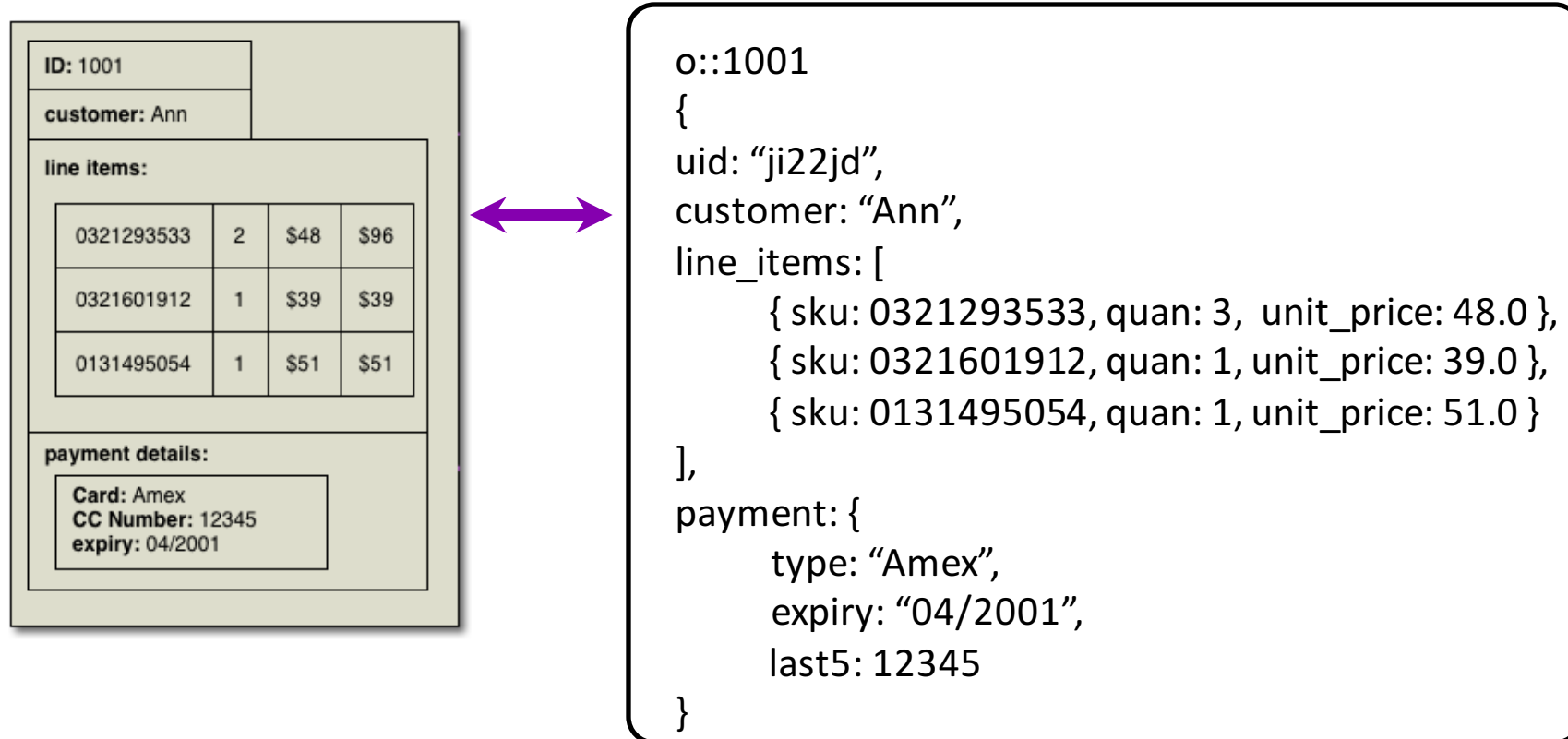
The relational approach to model data



Relational databases were not designed with clusters in mind, which is why people have cast around for an alternative. Storing aggregates as fundamental units makes a lot of sense for running on a cluster.

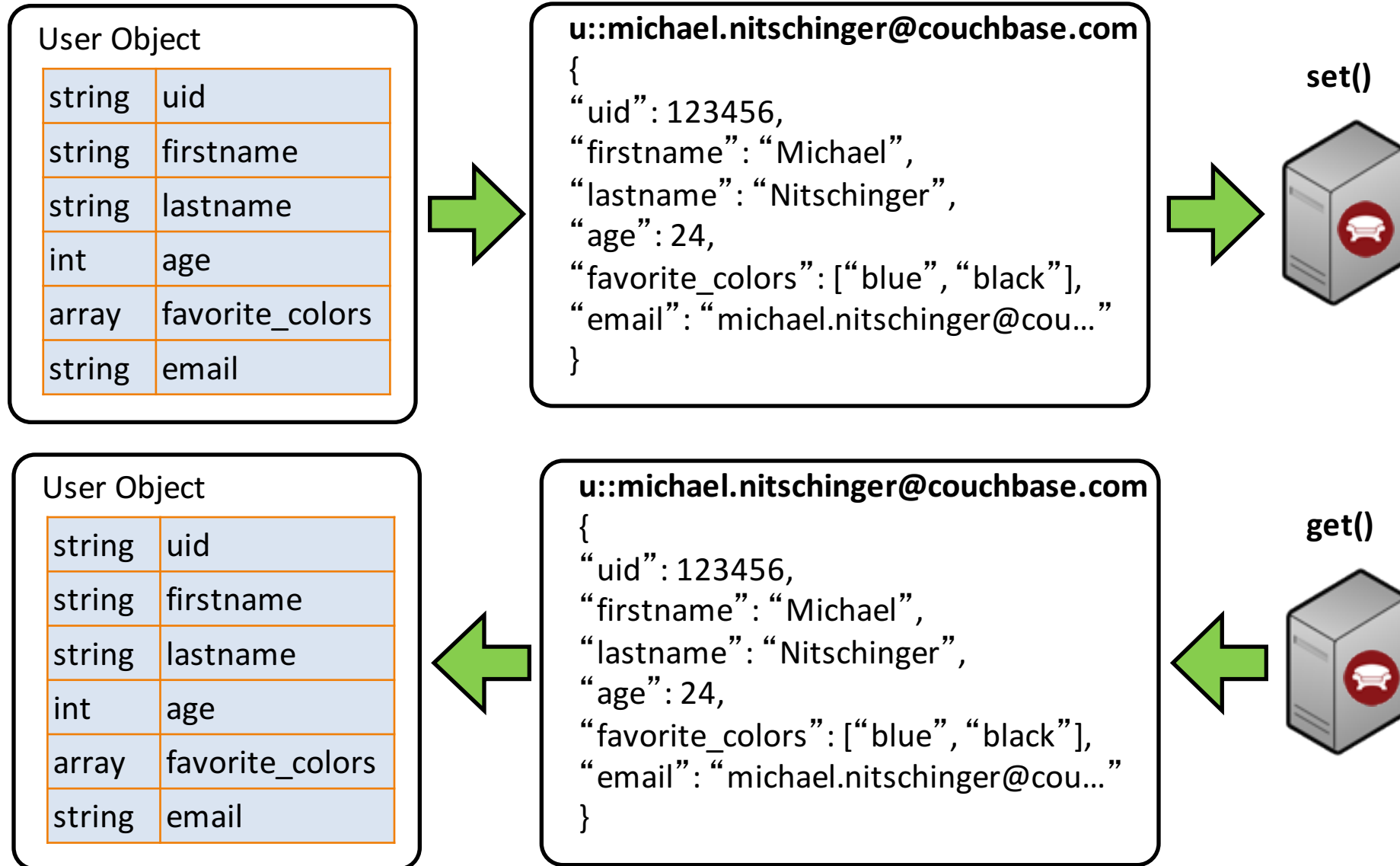
<http://martinfowler.com/bliki/AggregateOrientedDatabase.html>

Compared to a Document Database



- Easy to distribute data
- Makes sense to application programmers

Object to JSON back to Object



Using a Document-Oriented Approach

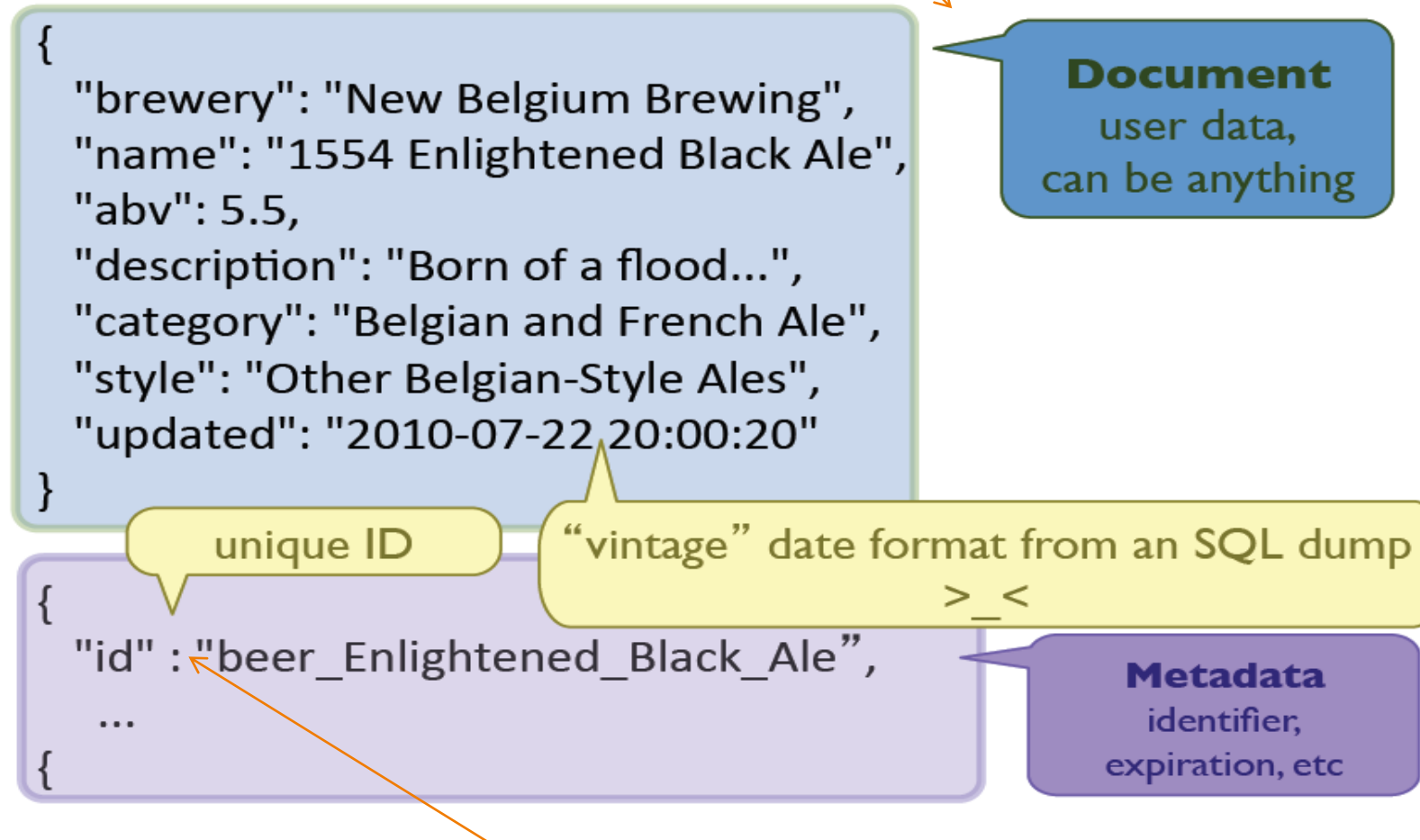
- use documents in a specific format that stores data representing an object graph.
- documents are stored in JSON format.
- we do not represent the fields with null values.

```
{  
  "id": "2d18e287-d5fa-42ce-bca5-b26eae0d7d4",  
  "type": "rant",  
  "userName": "JerryS",  
  "rantText": "Why do they call it Ovaltine? The mug is round. The jar is round. They should  
call it Roundtine."  
}
```

Not bound to a schema.

Meta + Document Body

Most Recent In Ram And Persisted To Disk



All Keys Unique and Kept in RAM

Keys and Metadata

- **Keys** are unique identifiers for a document, and values are either JSON documents or byte stream etc
- By default, all documents contain three types of metadata which are provided by the Couchbase Server
 - **Cas Value** : - unique identifier associated with a document, and verified by the Couchbase Server before a document is deleted or changed.
 - **Time to Live (ttl)** — This is an expiration for a document typically specified in seconds
 - **Flags** — These are SDK- specific flags which are used to provides a variety of options during storage, retrieval, update, and removal of documents

Keys and Metadata

- **Id:** Couchbase Server saves the document's key as part of the metadata.
- **Type:** Internally, Couchbase saves data in two formats:
 - **JSON:** Used whenever a valid JSON value is being saved in the database. JSON is also used to store counters that are created using the increment/decrement functionality in the SDKs.
 - **Base64:** All other data is being saved as Base64 encoded strings.
- **Rev:** This value is an internal revision ID used by Couchbase Server.

No More alter table

- ✓ More productive developers!
- ✓ Emergent schema—
- ✓ Schema driven by code

Creating Your First Document

activity help Administrator

10.143.192.101 > Documents CLASSIC EDITOR ADD DOCUMENT

Dashboard

Servers

Buckets

XDCR

Security

Settings

Logs

Documents

Query

Search

Analytics

Eventing

Views

Indexes

Bucket

show top keys

Limit 10

Offset 0

Document ID optional...

show range

































N1QL WHERE optional...

Retrieve Docs

10 Results for `select meta().id from `travel-sample` data order by meta().id limit 10 offset 0`

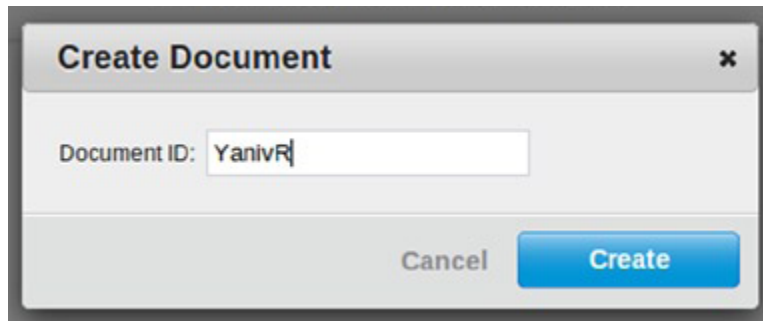
enable field editing

< prev batch | next batch >

	id	
   	airline_10	<code>{"callsign":"MILE-AIR","country":"United States","iata":"Q5","icao":"MLA","id":10,"type":"airline"}</code>
   	airline_10123	<code>{"callsign":"TXW","country":"United States","iata":"TQ","icao":"TXW","id":10123,"type":"airline"}</code>
   	airline_10226	<code>{"callsign":"atifly","country":"United States","iata":"A1","icao":"A1F","id":10226,"type":"airline"}</code>
   	airline_10642	<code>{"callsign":null,"country":"United Kingdom","iata":null,"icao":"JRB","id":10642,"type":"airline"}</code>
   	airline_10748	<code>{"callsign":"LOCAIR","country":"United States","iata":"ZQ","icao":"LOC","id":10748,"type":"airline"}</code>
   	airline_10765	<code>{"callsign":"SASQUATCH","country":"United States","iata":"K5","icao":"SQH","id":10765,"type":"airline"}</code>
   	airline_109	<code>{"callsign":"ACE AIR","country":"United States","iata":"KO","icao":"AER","id":109,"type":"airline"}</code>
   	airline_112	<code>{"callsign":"FLYSTAR","country":"United Kingdom","iata":"5W","icao":"AEU","id":112,"name":"Astraeus","type":"airline"}</code>

Creating Your First Document - TBD

- Document ID → Enter document detail



A dialog box titled "Create Document" with a close button (x) in the top right corner. It contains a text input field labeled "Document ID:" with the text "YanivR" entered. At the bottom, there are two buttons: "Cancel" and "Create".

Enter the document details



The interface shows a breadcrumb navigation bar with "ranter" and a dropdown arrow, followed by "> Documents". Below this is a document editor window titled "YanivR" with "Delete", "Save As...", and "Save" buttons. The editor contains a JSON document with the following content:

```
1 {  
2   "type": "user",  
3   "username": "YanivR",  
4   "email": "yaniv@ranter.com",  
5   "password": "12345",  
6   "shortDesc": "A software grouch",  
7   "imageUrl": "https://secure.gravatar.com/userimage/34395405/7856c192a706851d06f391be9050e008.jpg?size=200",  
8 }
```

Labs: Document Design (Optional)