6.148

Backend II: Introduction to Databases

A (very gentle) introduction to databases, before Firebase lecture

da·ta·base

ˈdatəˌbās,ˈdā-/

noun

a structured set of data held in a computer, especially one that is accessible in various ways.

Why databases?

Why databases?

```
var fakedb = {
  1: 'http://6.470.scripts.mit.edu/2015/css/img/logo.svg'
};

var counter = 2;

/* GET /photos/123 */
router.get('/photos/:id', function(req, res) {
  var photoId = req.param('id');
  res.render('photo', { url: fakedb[photoId] });
});
```

Why databases?

```
var fakedb = {
  1: 'http://6.470.scripts.mit.edu/2015/css/img/logo.svg'
};

var counter = 2;
```

```
/* GET /photos/123 */
router.get('/photos/:id', function(req, res) {
  var photoId = req.param('id');
  res.render('photo', { url: fakedb[photoId] });
});
```

What are some problems?

Server crash makes us lose our data.

Server crash makes us lose our data.

What if we write to files instead?

I want to pay Victor $50.

I want to pay Victor $50.

```
charles.balance -= 50;
victor.balance += 50;
```

```
victor.balance += 50;
charles.balance -= 50;
```

I want to pay Victor $50.

```
charles.balance -= 50;
victor.balance += 50;
```

```
victor.balance += 50;
charles.balance -= 50;
```

What if we crash in between?

What if simultaneously, Victor wants to pay Jeffrey?

Databases solve lots of hard problems for us.

reliability

server crashes
replicated data

performance

dedicated database servers
database optimizations
concurrency handling

consistency

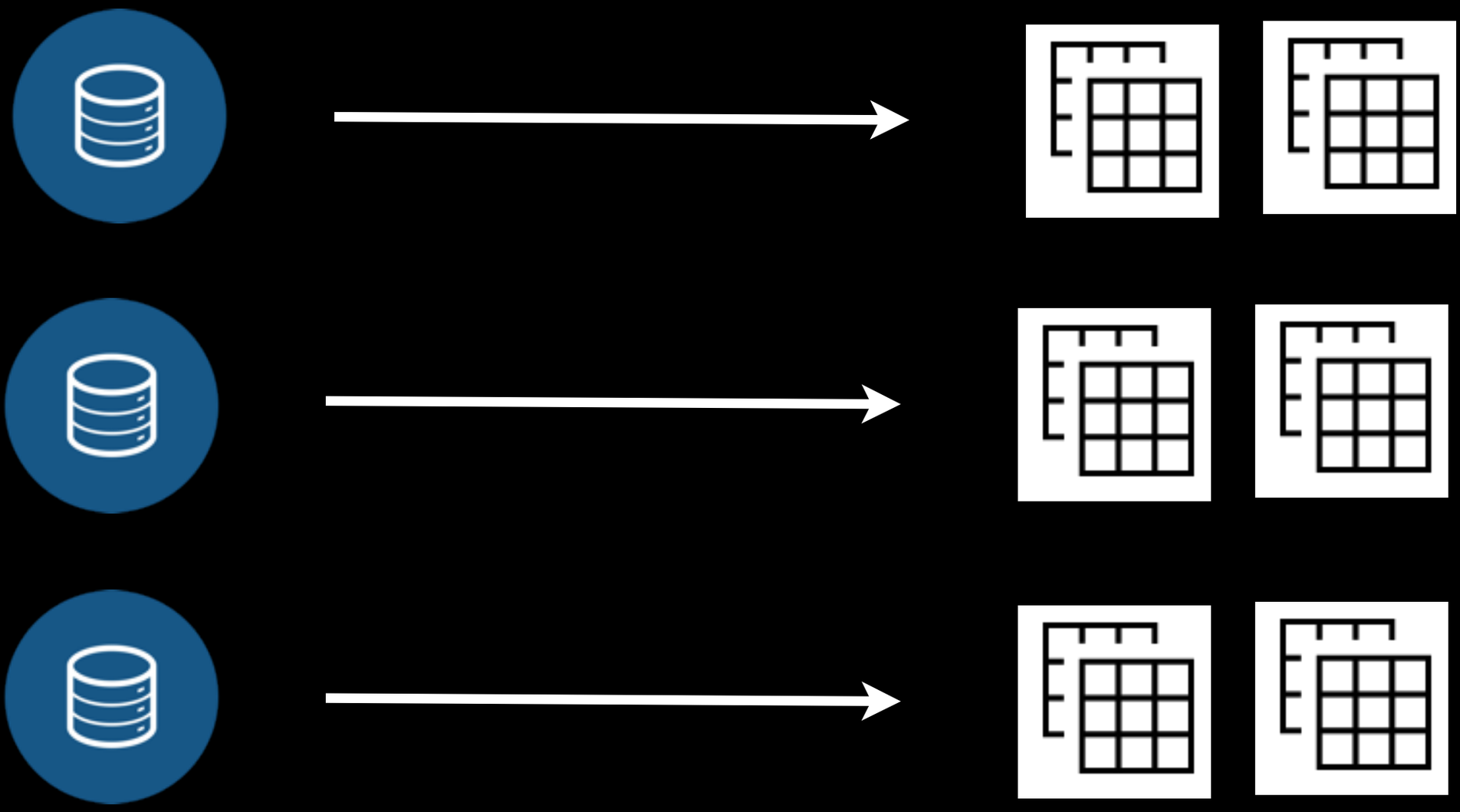
proper concurrent behavior

multi-step transactions

SQL or NoSQL?

- Two different approaches to databases.
- SQL (Structured Query Language): enforces a structure on your data
    - Default for Ruby on Rails
- NoSQL: no enforced structure; performance and scalability in some applications
    - Much more common in Javascript frameworks

SQL



Databases

Tables

SQL

id	username
1	steph
2	xander
3	alan

SQL

"primary key"

id	username
1	steph
2	xander
3	alan

id	content
1	"hello!"
2	"i'm hungry"
3	"wat"

The Schema

- Users:
  id: integer, primary key
  username: varchar(40)

- Posts:
  id: integer, primary key
  content: text

“Relational”

id	username
1	steph
2	xander
3	alan

“foreign key”

id	content	user_id
1	“hello!”	2
2	“i’m hungry”	2
3	“wat”	1

Some Queries

```
CREATE TABLE users (
    id int NOT NULL PRIMARY KEY ,
    username varchar(40),
    UNIQUE (username)
);

INSERT INTO users (username) VALUES ('charles');

SELECT username FROM users WHERE id=2;
```

More Queries

id	username
1	steph
2	xander
3	alan

id	content	user_id
1	“hello!”	2
2	“i’m hungry”	2
3	“wat”	1

What if we want to get the username of a post?

```
SELECT users.username
FROM users
JOIN posts
WHERE users.id = posts.user_id AND posts.id = 1
```

You (probably) won't have to actually write SQL.

Rails takes care of all of this for you!

NoSQL

- A group of databases that don't use SQL; one of the most well-known: MongoDB
- *"Collections"* of *"documents"*
- Almost always used with JS frameworks
    - Many similarities between JS objects and documents

Javascript, JSON, Documents

- Javascript object:
  { contents: “hello world”, time: “1/9/2015 11:00 AM” }
- JSON: “Javascript object notation” - turn the Javascript object into a string
  - Often used in client-server communication

Using MongoDB

```
db.users.insert({
    name: “charles”,
    age: 22,
});

db.users.insert({name: “victor”, course: “6-2”});

db.posts.find(
   { author: “charles” }
);
```

a document, also a JS object!

Relations in MongoDB?

Come to Backend III lecture next Tuesday!

MongoDB frameworks for Node.js

Firebase

"a powerful API to store and sync data in realtime"

Welcome Firebase!