JUDCon

JBoss Users & Developers Conference

2012:Boston

Hibernate OGM

JPA for NoSQL

Emmanuel Bernard
Data Platform Architect
but actually doing things
JBoss By Red Hat

Before you leave

HIBERNATE OGM

- JPA for NoSQL
- Denormalization engine
- Use the JPA mapping semantic
- Does queries too (gradual ramp up)

Emmanuel Bernard

- JBoss: Hibernate, JCP
- Ceylon
- Podcasts
  - asylum.jboss.org
  - lescastcodeurs.com
- The rest is at http://emmanuelbernard.com
- @emmanuelbernard

(No)SQL tour

Relational databases

Relational databases

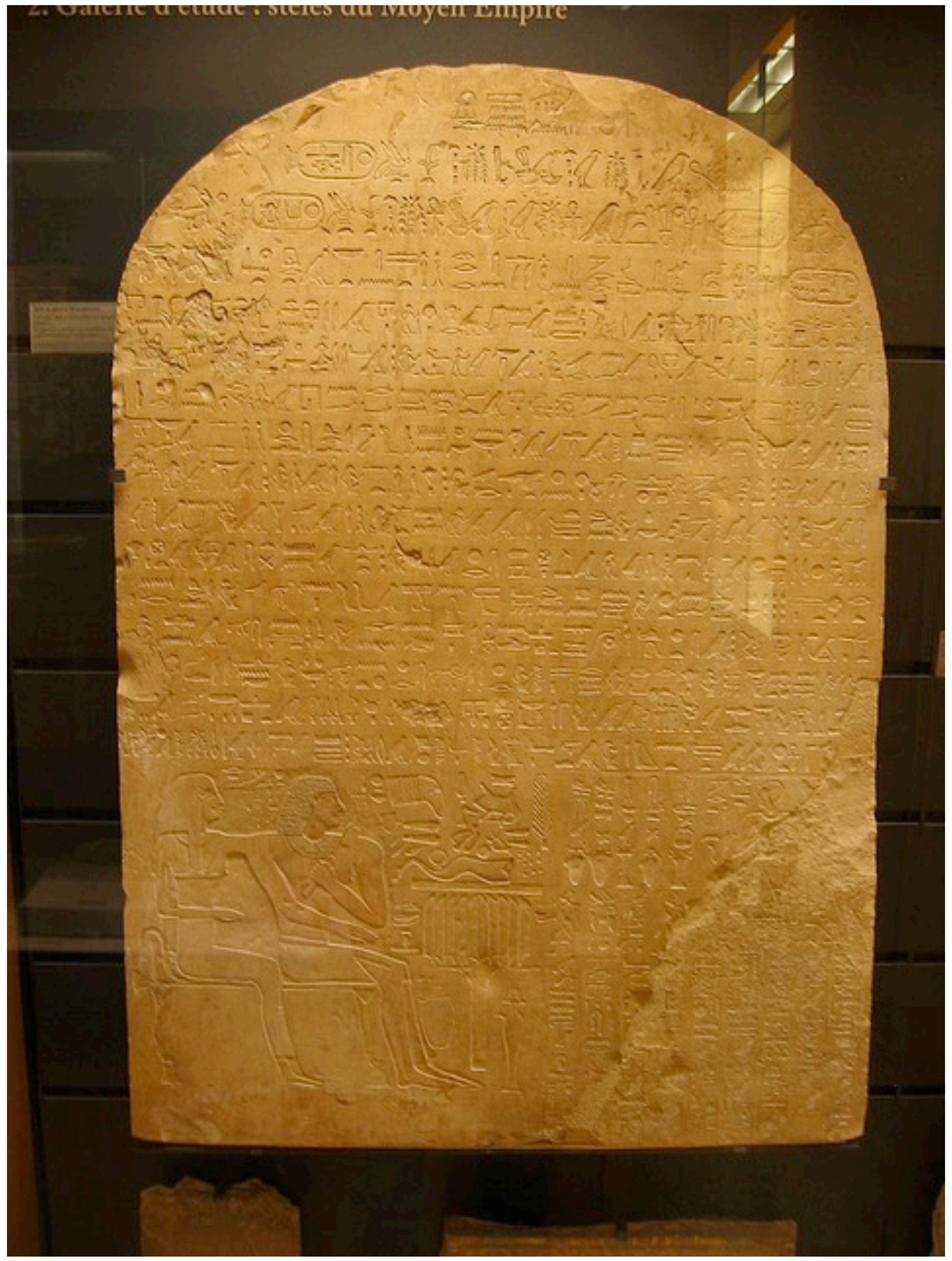
- Data structure abstraction
- Transaction, referential integrity
- Common query language
- (Simple) type
- Proven usefulness
  - tuning, backup, resilience

Relational databases

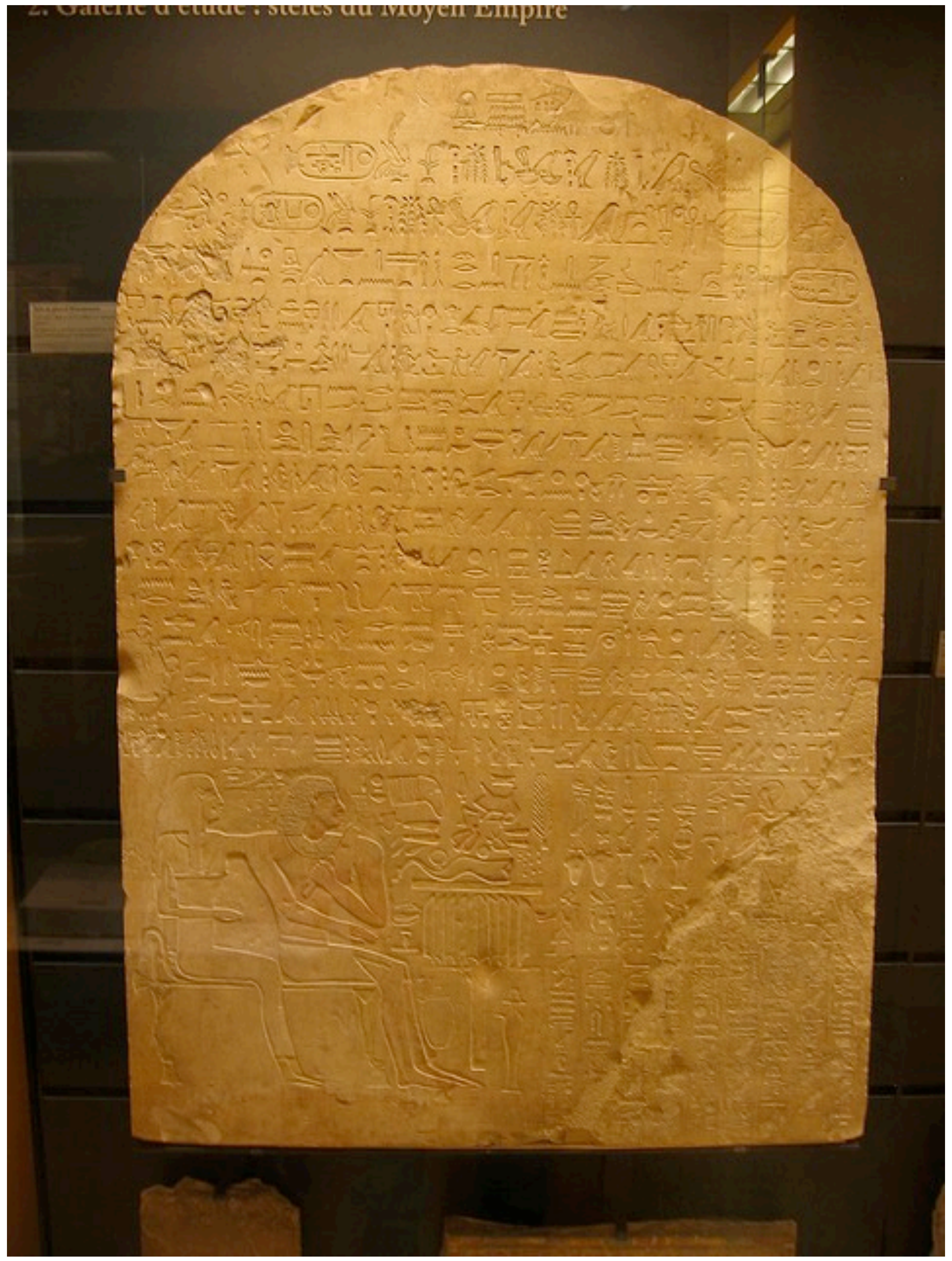
- (Some) limitations:
  - planning for scale is hard
  - data model changes are painful
- New needs
  - limitless data for later analysis
  - instant fame syndrome
  - less query demanding data

NoSQL is not new

NoSQL is not new



NoSQL is not new

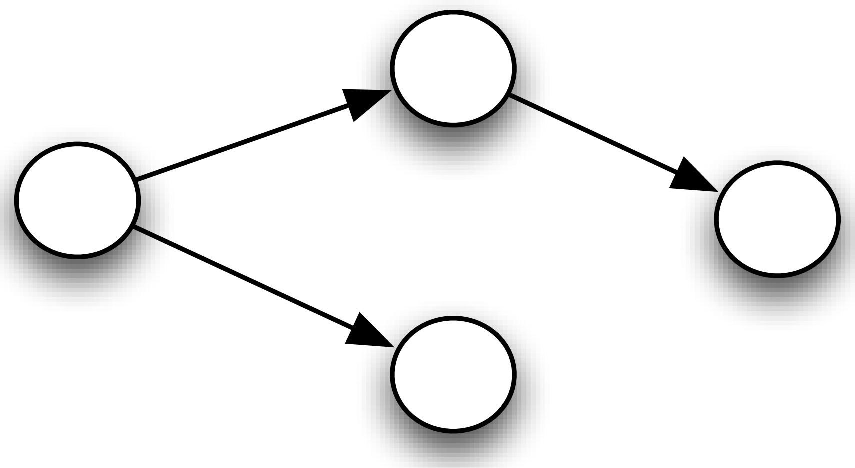


华佗
（公元141年 — 203年）
古代著名医生，字元化，沛国谯人。他精通内、外、妇、儿、针灸各科，能做各种手术，发明了麻醉药"麻沸散"和锻炼身体的"五禽戏"。
韩非
（公元前280年 — 前233年）
先秦法家思想的集大成者，著有《韩非子》。

NoSQL alternatives

- Web “giants” needs
- Very different Goals
  - data size / availability
  - low latency / higher throughput
- Optimize some data access patterns

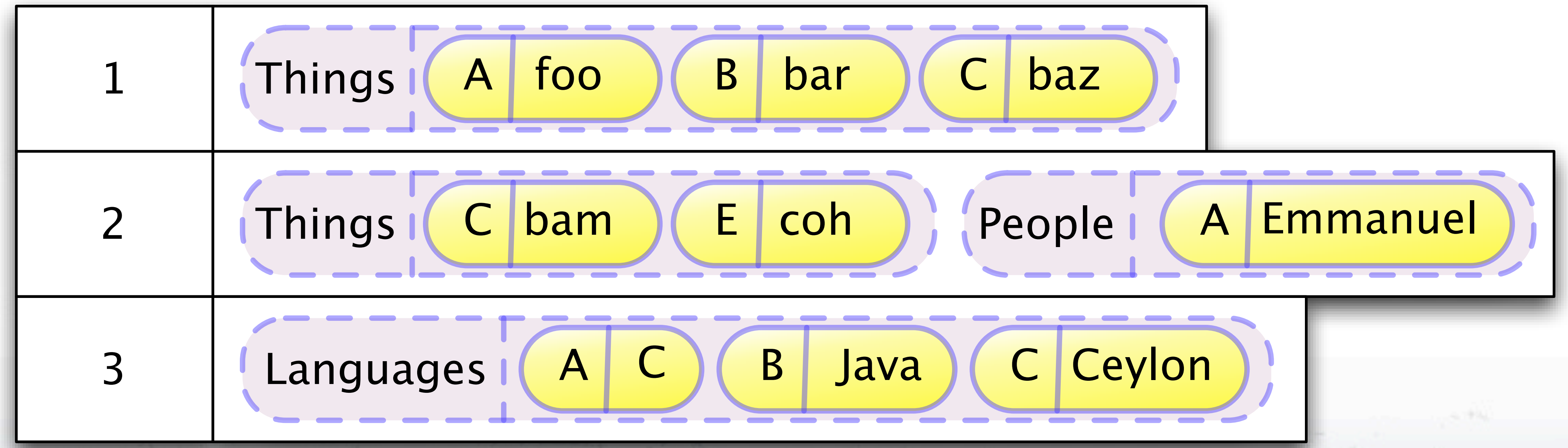
NoSQL families



- Graph oriented databases
- Key / value stores
- Document based stores
- BigTable-style

key	value
123	Address@23
126	“Booya”

```
{ "user" : {
"id": "124",
"name": "Emmanuel",
"addresses" : [
{ "city": "Paris", "country": "France" },
{ "city": "Atlanta", "country": "USA" }
]
}
```


1
Things A foo B bar C baz
2
Things C bam E coh People A Emmanuel
3
Languages A C B Java C Ceylon

Flexibility at a cost

Flexibility at a cost

- Programming model
  - no common API :(
  - query (Map Reduce, specific DSL, ...)
  - no schema => app driven schema
- Denormalization at the app layer

- Transaction / durability / consistency

JPA for NoSQL

HIBERNATE OGM

Demo

Goals

- Encourage new data usage patterns
  - volume, types etc
- Familiar environment
  - Full JPA support
  - easy to jump in (and out!)
- Declarative denormalization

What it does

- Today
  - JPA front end for Infinispan, EhCache and MongoDB
  - CRUD support for @Entities
  - Full-text queries
- Working on it
  - JP-QL queries (simple ones)
  - More NoSQL (Cassandra)
  - Explore denormalization

Not a silver bullet!

- But JPA matches quite nicely

Concepts

Schema or no schema?

- Schema-less
  - developer friendly
  - data structure migration?
  - need strict development guidelines
- Schema
  - strong documentation
  - share with other apps / tooling

Entities as serialized blobs?

- Store the whole graph?
- Consistency with duplicated objects

- Structure change and (de)serialization

OGM’s approach

- Keep what’s best from relational model
  - as much as possible
- Decorrelate object and data structure
  - object model evolution
- Data stored as (self-described) tuples
- Limited set of core types
- CRUD operations are key lookups

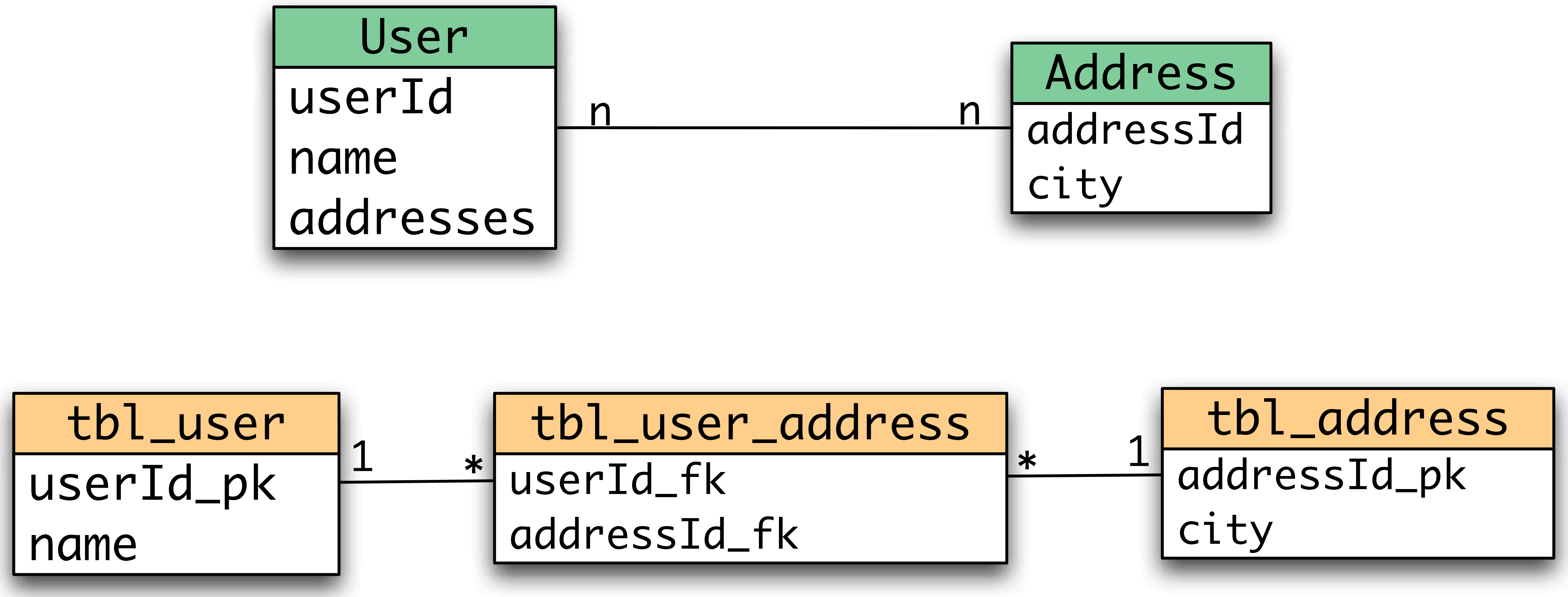
Hibernate OGM’s data structure

Storage - Entities

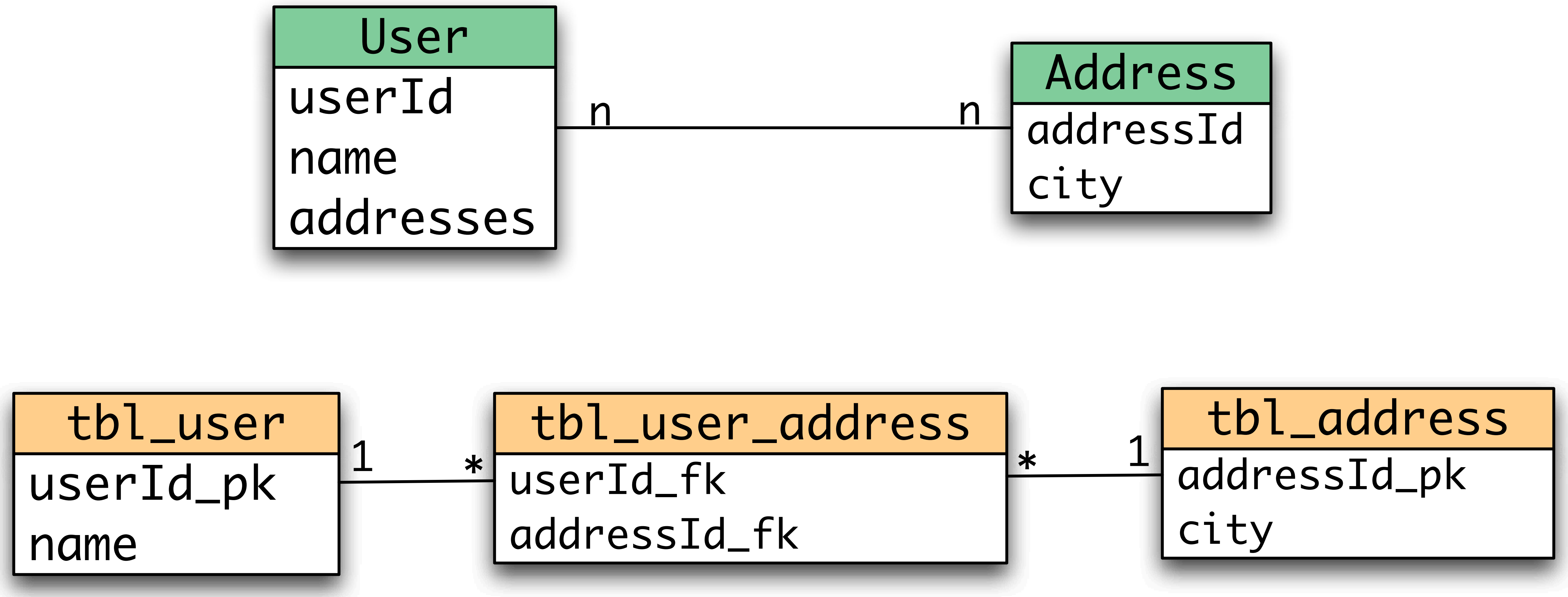
- Each entity in a unique key
  - table name
  - id column names and values
- Value is Map<String,Object>
  - String: column name
  - Object: simple type (serializable)

Storage - Associations

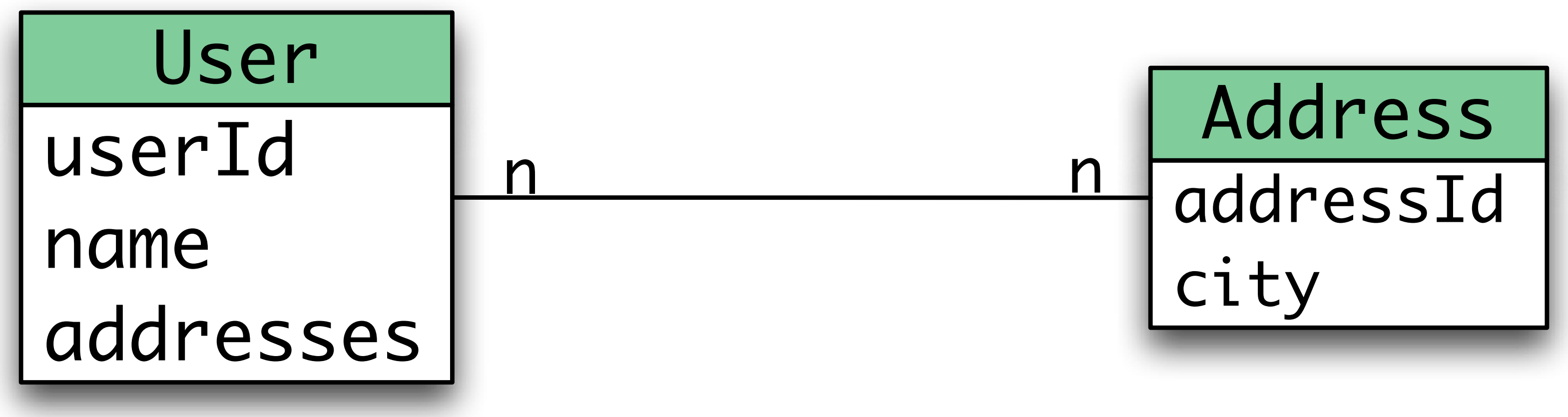
- Cannot store exactly like relational DBs
- Simulate navigation to associations
  - one key per navigation
- Value is the list of tuples
- Focus on speedy reads
  - writes involve several key lookups

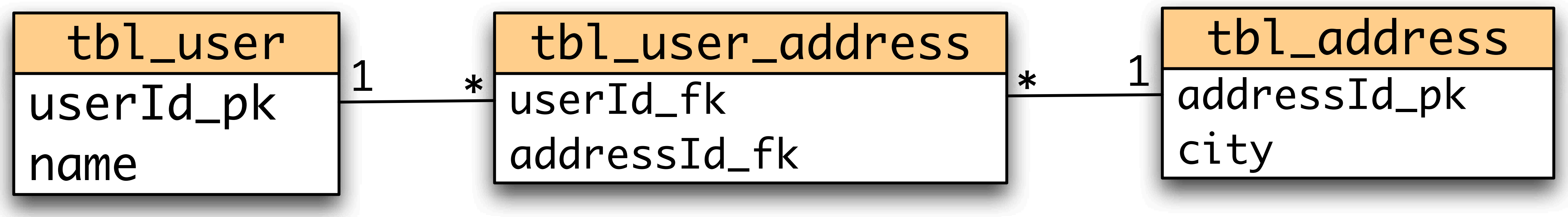


key	value
tbl_user,userId_pk,1	{userId_pk=1,name=”Emmanuel”}
tbl_user,userId_pk,2	{userId_pk=2,name=”Caroline”}
tbl_address,addressId_pk,3	{addressId_pk=3,city=”Paris”}
tbl_address,addressId_pk,5	{addressId_pk=5,city=”Atlanta”}



key	value
tbl_user,userId_pk,1	{userId_pk=1,name=”Emmanuel”}
tbl_user,userId_pk,2	{userId_pk=2,name=”Caroline”}
tbl_address,addressId_pk,3	{addressId_pk=3,city=”Paris”}
tbl_address,addressId_pk,5	{addressId_pk=5,city=”Atlanta”}
tbl_user_address,userId_fk,1	{ {userId_fk=1, addressId_fk=3}, {userId_fk=1, addressId_fk=5} }
tbl_user_address,userId_fk,2	{ {userId_fk=2, addressId_fk=3} }





key	value
tbl_user,userId_pk,1	{userId_pk=1,name=”Emmanuel”}
tbl_user,userId_pk,2	{userId_pk=2,name=”Caroline”}
tbl_address,addressId_pk,3	{addressId_pk=3,city=”Paris”}
tbl_address,addressId_pk,5	{addressId_pk=5,city=”Atlanta”}
tbl_user_address,userId_fk,1	{ {userId_fk=1, addressId_fk=3}, {userId_fk=1, addressId_fk=5} }
tbl_user_address,userId_fk,2	{ {userId_fk=2, addressId_fk=3} }
tbl_user_address,addressId_fk,5	{ {userId_fk=1, addressId_fk=5} }
tbl_user_address,addressId_fk,3	{ {userId_fk=1, addressId_fk=3}, {userId_fk=2, addressId_fk=3} }

Queries

- Hibernate Search indexes entities
- Store Lucene indexes in Infinispan
- JP-QL to Lucene query

- Works for simple-ish queries

```
select a from Animal a where a.size > 20
> animalQueryBuilder
  .range().onField(“size”).above(20).excludeLimit()
  .createQuery();

select u from Order o join o.user u
where o.price > 100 and u.city = “Paris”
> orderQB.bool()
  .must(
    orderQB.range().onField(“price”)
      .above(100).excludeLimit().createQuery() )
  .must(
    orderQB.keyword().onField(“user.city”)
      .matching(“Paris”).createQuery() )
  .createQuery();
```

Hibernate Search is awesome

- Full-text search made simple
  - fuzzy, ngram, phonetic
  - faceting, geolocation
- Nice and readable Query DSL
- Computed on app layer side
  - Clusterable

Future

- More NoSQL families
- More JP-QL support
- JP-QL to “primitives”
- API for operations in bulk
- More denormalization options
- Hybrid deployment options

Hibernate OGM

- JPA for NoSQL
- Denormalization engine
- Does queries too

- Status
  - CRUD support for Infinispan, EHCache, MongoDB
  - queries are the next frontier

More info

- Documentation
  - http://ogm.hibernate.org
    - including reference doc
  - Any good JPA book ;)
- Code
  - come and contribute or you’ll get 7 years of bad sex
  - https://github.com/hibernate/hibernate-ogm
- Q&A

References

- Pictures under creative commons
- http://www.flickr.com/photos/tomsaint/3415333390/
- http://www.flickr.com/photos/anniewong/26473161/
- http://www.flickr.com/photos/jdhancock/5002736203/
- http://www.flickr.com/photos/liutao/280498401
- http://www.flickr.com/photos/ehw/243631365/