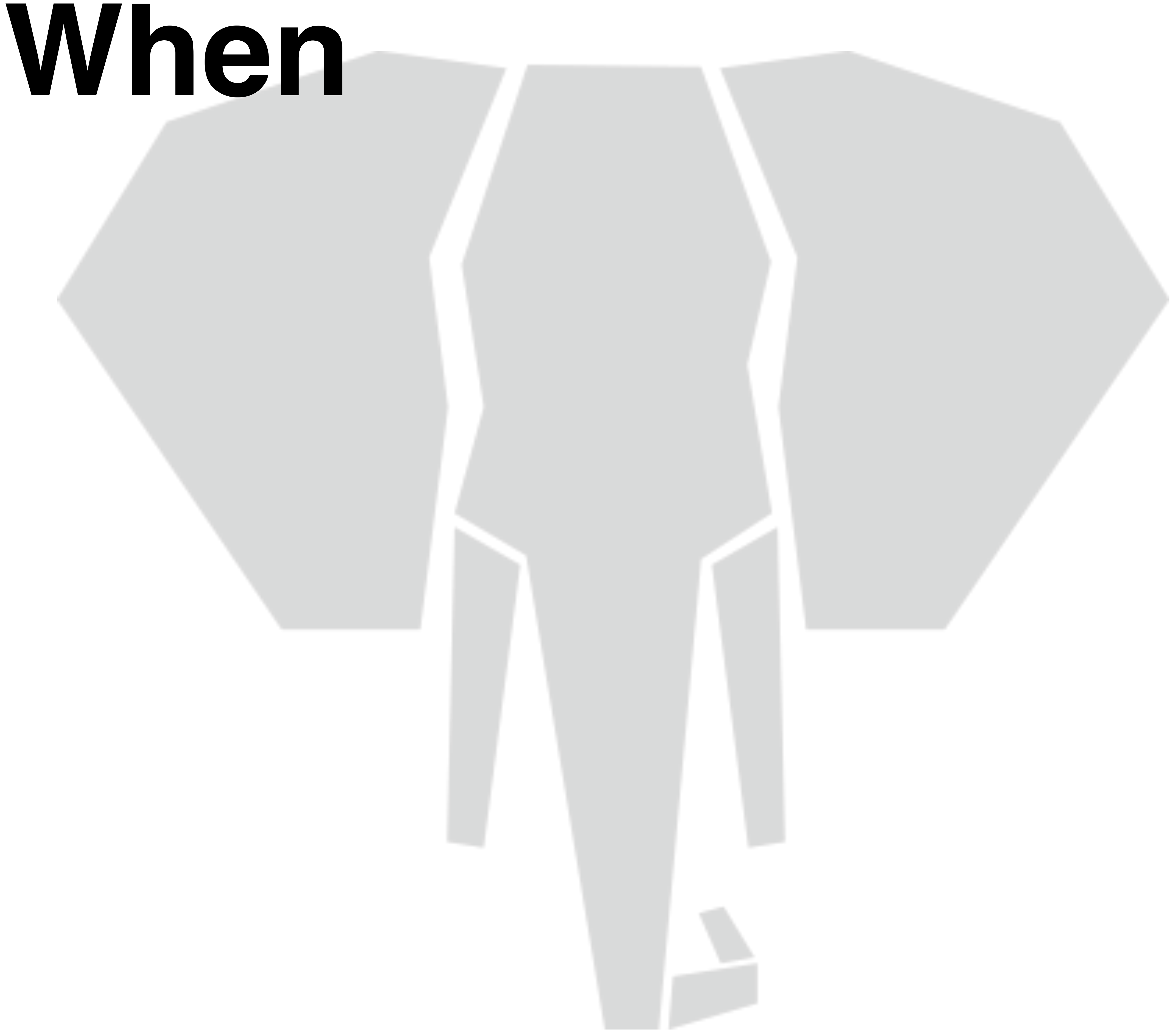
When to NoSQL and When to Know SQL

Simon Elliston Ball
Head of Big Data

@sireb
#noSQLknowSQL

http://nosqlknowsql.io



redgate
ingeniously simple

what is NoSQL?

SQL

NoSQL

Not only SQL

No, SQL

Many many things

before SQL

files

multi-value

ur… hash maps?

after SQL

everything is relational

ORMs fill in the other data structures

scale up rules

data first design

and now NoSQL

datastores that suit applications

polyglot persistence: the right tools

scale out rules

APIs not EDWs

why should you care?

data growth

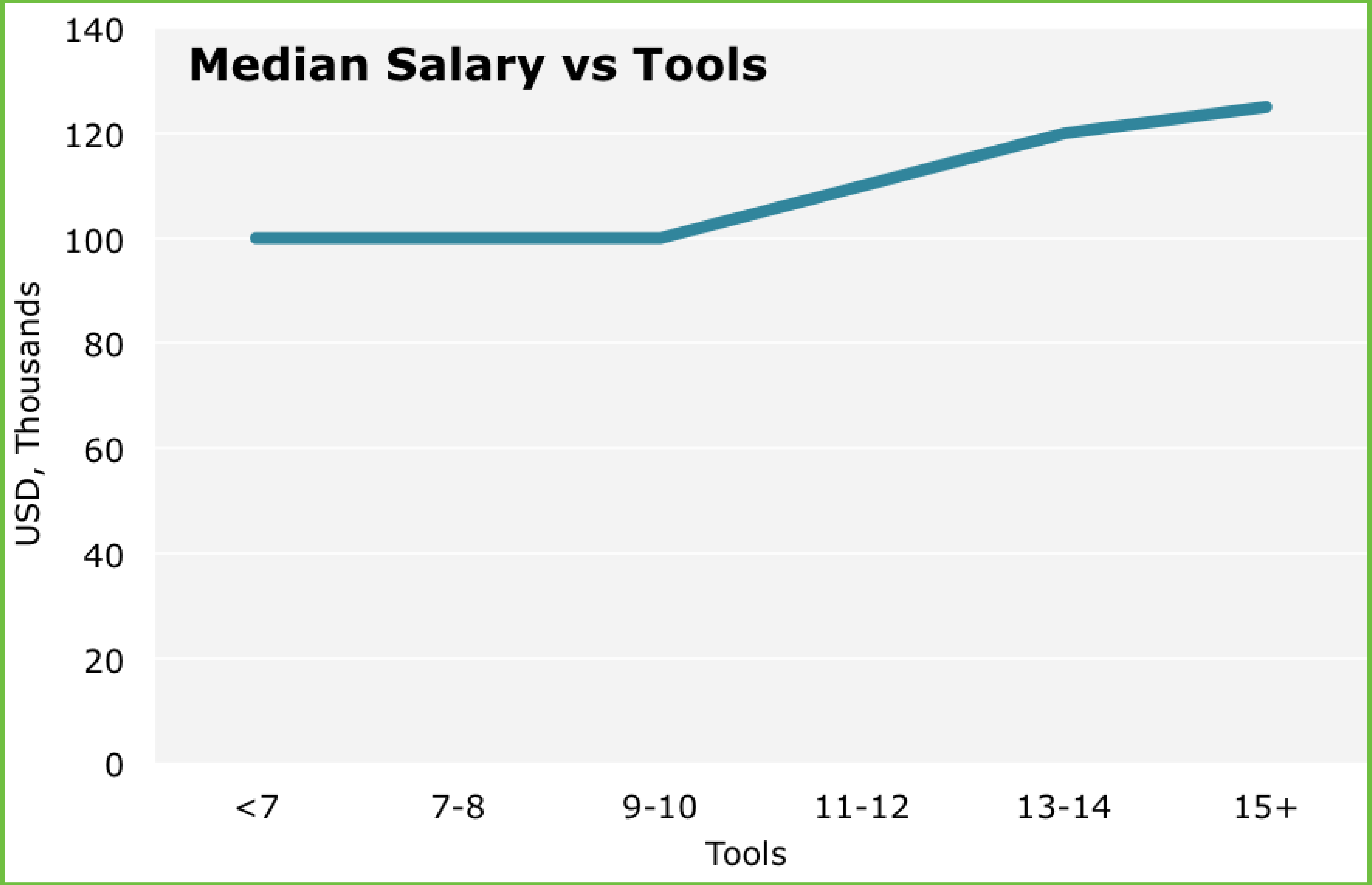
rapid development

fewer migration headaches… maybe

machine learning

social

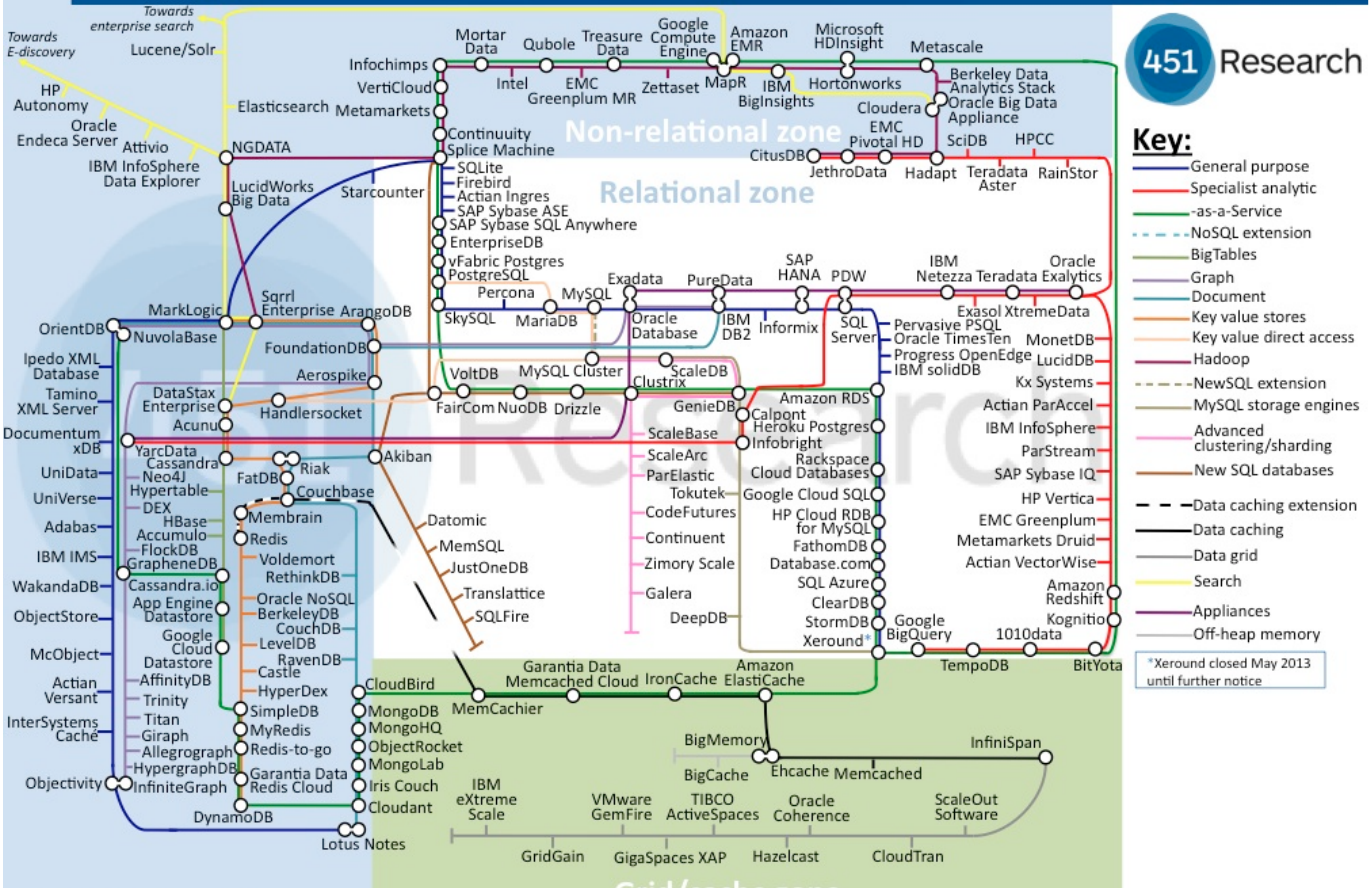
big bucks.



O'Reilly 2013 Data Science Salary Survey

So many NoSQLs…

Database Landscape Map – June 2013



document databases

document databases

rapid development

JSON docs

complex, variable models

known access pattern

document databases

learn a very new query language

denormalize

document form

joins? **JUST DON’T**

http://www.sarahmei.com/blog/2013/11/11/why-you-should-never-use-mongodb/

document vs SQL

what can SQL do?

query all the angles

sure, you can use blobs…

… but you can’t get into them

documents in SQL

SQL xml fields

mapping xquery paths is painful

native JSON

- but still structured

query everything: search

class of database database

full-text indexing

elasticsearch.

Apache Solr

you know google, right…

range query

span query

keyword query

you know the score

scores

```
"query": {
  "function_score": {
    "query": {
      "match": { "title": "NoSQL"}
    },
    "functions": [
      "boost": 1,
      "gauss": {
        "timestamp": {
          "scale":  "4w"
        }
      },
      "script_score" : {
        "script" : "_score * doc['important_document'].value ? 2 : 1"
      }
    ],
    "score_mode": "sum"
  }
}
```

SQL knows the score too

scores

```
declare @origin float = 0;
declare @delay_weeks float = 4;

SELECT TOP 10 * FROM (
  SELECT title,
    score *
    CASE
      WHEN p.important = 1 THEN 2.0
      WHEN p.important = 0 THEN 1.0
    END
    * exp(-power(timestamp-@origin,2)/(2*@delay*7*24*3600))
    + 1
    AS score
  FROM posts p
  WHERE title LIKE '%NoSQL%'
) as found
ORDER BY score
```

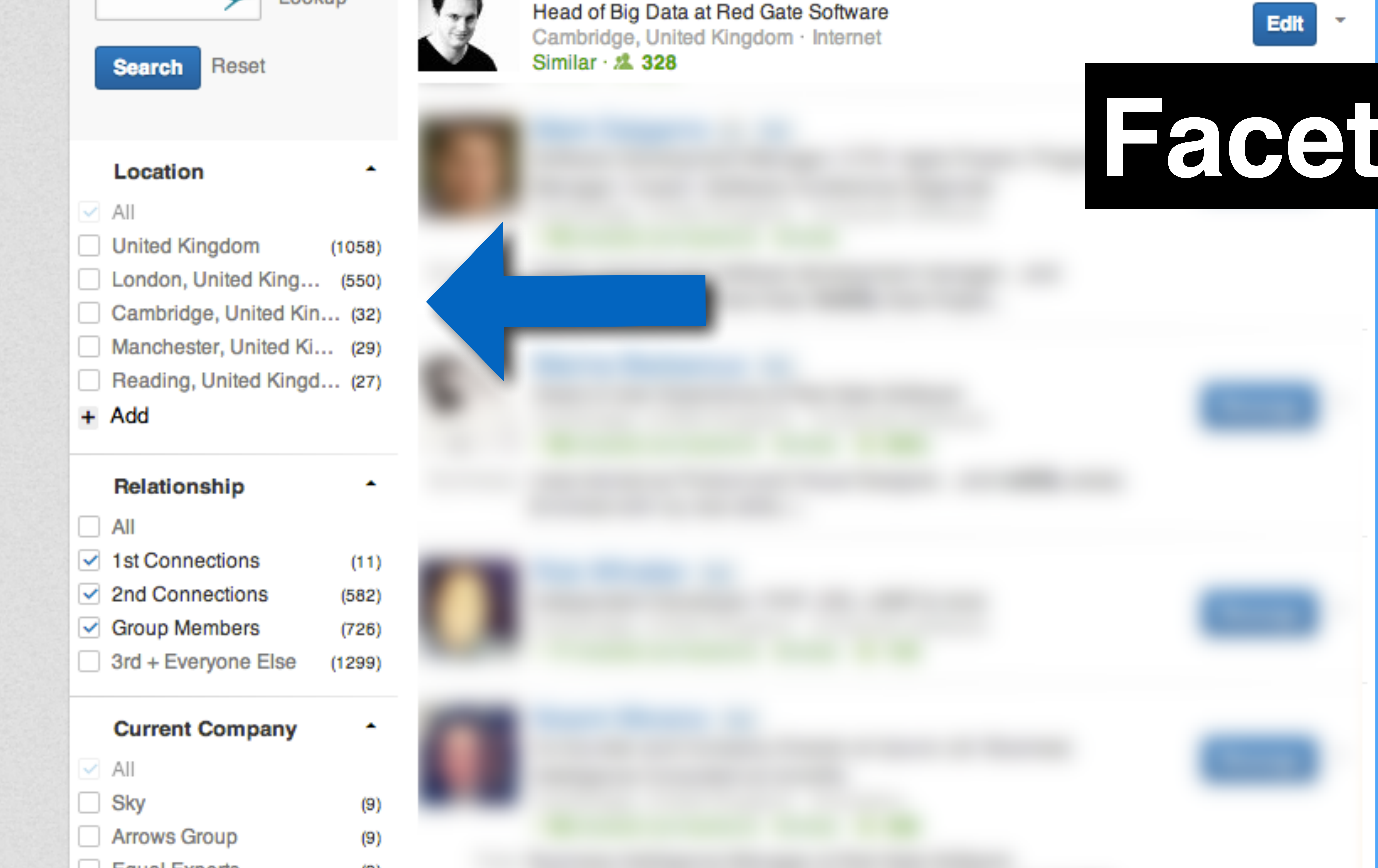
you know google, right…

more like this: instant tf-idf

```
{
    "more_like_this" : {
        "fields" : ["name.first", "name.last"],
        "like_text" : "text like this one",
        "min_term_freq" : 1,
        "max_query_terms" : 12
    }
}
```

Facets

Facets



Lookup

Search Reset

Location

☑ All

□ United Kingdom (1058)

□ London, United King... (550)

□ Cambridge, United Kin... (32)

□ Manchester, United Ki... (29)

□ Reading, United Kingd... (27)

+ Add

Relationship

□ All

☑ 1st Connections (11)

☑ 2nd Connections (582)

☑ Group Members (726)

□ 3rd + Everyone Else (1299)

Current Company

☑ All

□ Sky (9)

□ Arrows Group (9)

□ Equal Experts (9)

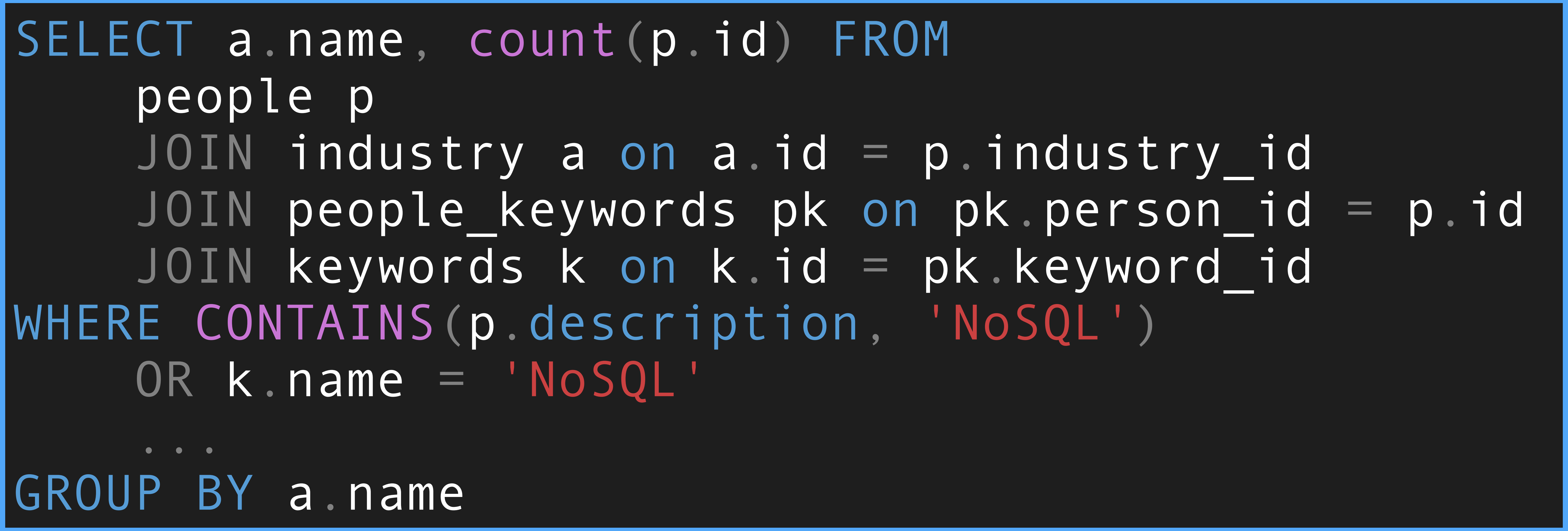
Head of Big Data at Red Gate Software

Cambridge, United Kingdom · Internet

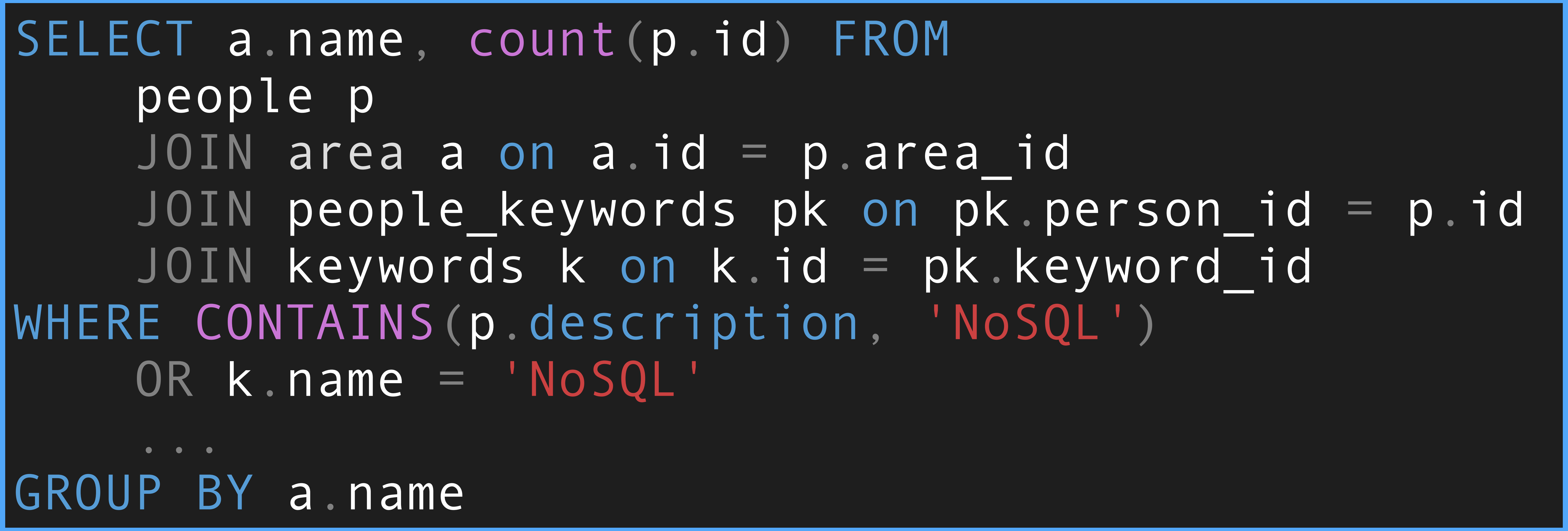
Similar · 328

Edit

SQL:



```
SELECT a.name, count(p.id) FROM
    people p
    JOIN industry a on a.id = p.industry_id
    JOIN people_keywords pk on pk.person_id = p.id
    JOIN keywords k on k.id = pk.keyword_id
WHERE CONTAINS(p.description, 'NoSQL')
    OR k.name = 'NoSQL'
    ...
GROUP BY a.name
```



```
SELECT a.name, count(p.id) FROM
    people p
    JOIN area a on a.id = p.area_id
    JOIN people_keywords pk on pk.person_id = p.id
    JOIN keywords k on k.id = pk.keyword_id
WHERE CONTAINS(p.description, 'NoSQL')
    OR k.name = 'NoSQL'
    ...
GROUP BY a.name
```

Facets

x lots

Facets

elasticsearch.

Elastic search:

```
{
  "query": {
    “query_string": {
       “default_field”: “content”,
       “query”: “keywords”
    }
  },
  "facets": {
    “myTerms": {
      "terms": {
        "field" : "lang",
        "all_terms" : true
      }
    }
  }
}
```

logs

untyped free-text documents

timestamped

semi-structured

discovery

aggregation and statistics

key: value

close to your programming model

distributed map | list | set

keys can be objects

redis

riak

SQL extensions

PostgreSQL

hash types

hstore

SQL and polymorphism

inheritance

ORMs hide the horror

turning round the rows

columnar databases

physical layout matters

turning round the rows

key	value	type
1	A	Home
2	B	Work
3	C	Work
4	D	Work

Row storage

00001	1	A	Home	00002	2	B	Work	00003	3	C	Work	…

Column storage

A	B	C	D	Home	Work	Work	Work	…

teaching an old SQL new tricks

MySQL **InfoBright**

SQL Server **Columnar Indexes**

```
CREATE NONCLUSTERED COLUMNSTORE INDEX idx_col
ON Orders (OrderDate, DueDate, ShipDate)
```

Great for your data warehouse, but no use for OLTP

column for hadoop and other animals

ORC files

Parquet http://parquet.io

column families

wide column databases

Cassandra

millions of columns

eventually consistent

CQL

set | list | map **types**

http://cassandra.apache.org/

http://www.datastax.com/

cell level security

SQL: so many views, so much confusion

accumulo https://accumulo.apache.org/

Time series

time

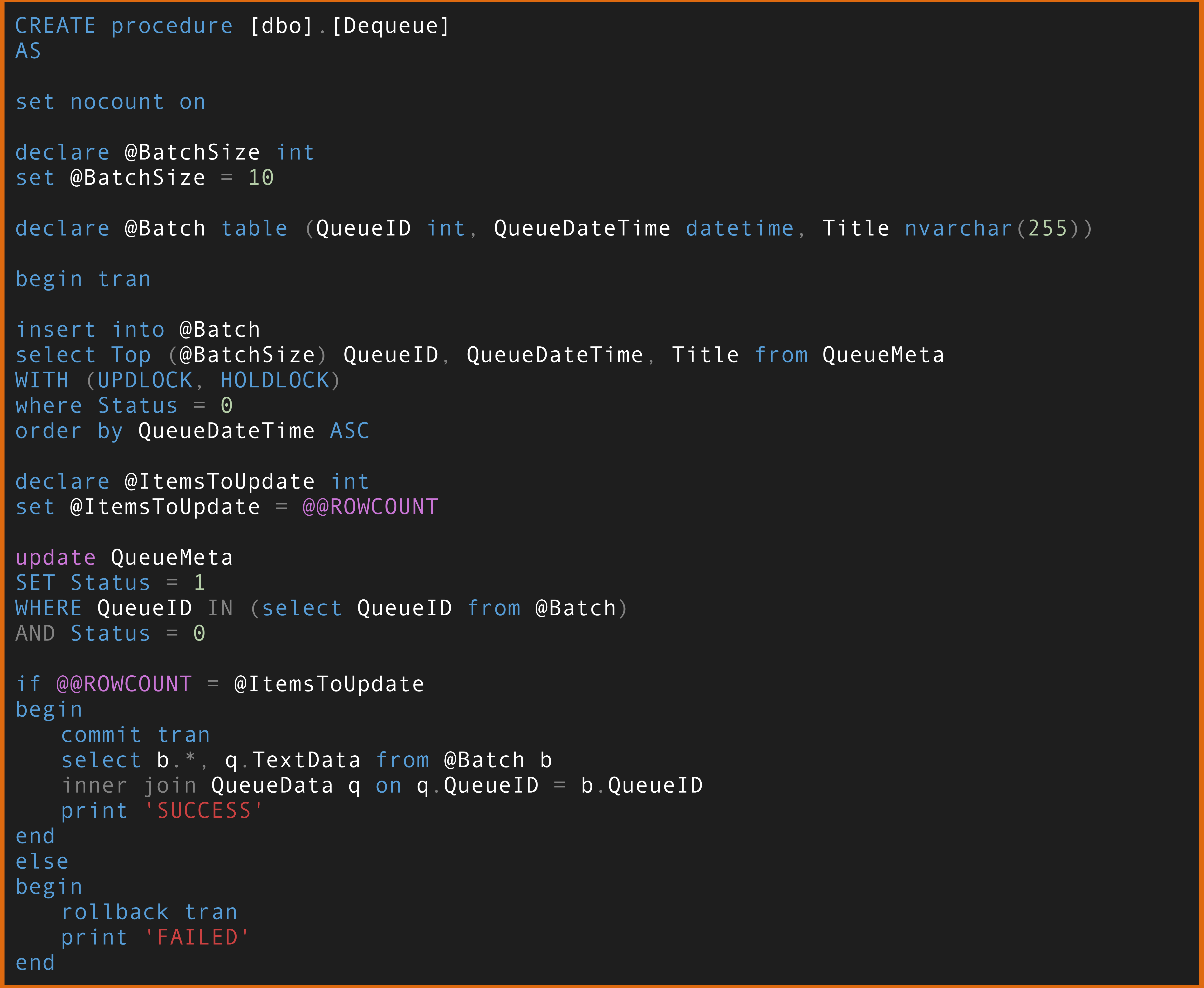
retrieving time series and graphs

window functions

```
SELECT business_date, ticker,
  close,
  close /
    LAG(close,1) OVER  PARTITION BY ticker ORDER BY business_date ASC)
    - 1 AS ret
FROM sp500
```

Queues

queues in SQL



```sql
CREATE procedure [dbo].[Dequeue]
AS

set nocount on

declare @BatchSize int
set @BatchSize = 10

declare @Batch table (QueueID int, QueueDateTime datetime, Title nvarchar(255))

begin tran

insert into @Batch
select Top (@BatchSize) QueueID, QueueDateTime, Title from QueueMeta
WITH (UPDLOCK, HOLDLOCK)
where Status = 0
order by QueueDateTime ASC

declare @ItemsToUpdate int
set @ItemsToUpdate = @@ROWCOUNT

update QueueMeta
SET Status = 1
WHERE QueueID IN (select QueueID from @Batch)
AND Status = 0

if @@ROWCOUNT = @ItemsToUpdate
begin
   commit tran
   select b.*, q.TextData from @Batch b
   inner join QueueData q on q.QueueID = b.QueueID
   print 'SUCCESS'
end
else
begin
   rollback tran
   print 'FAILED'
end
```

queues in SQL

index fragmentation is a problem

but built in logs of a sort

message queues

specialised apis

capabilities like fan-out

routing

acknowledgement

Windows Azure

ØMQ

amazon web services™

RabbitMQ™
Messaging that just works

relationships count

Graph databases

Neo4j

GraphX

APACHE GIRAPH

relationships count

trees and hierarchies

overloaded relationships

fancy algorithms

hierarchies with SQL

adjacency lists

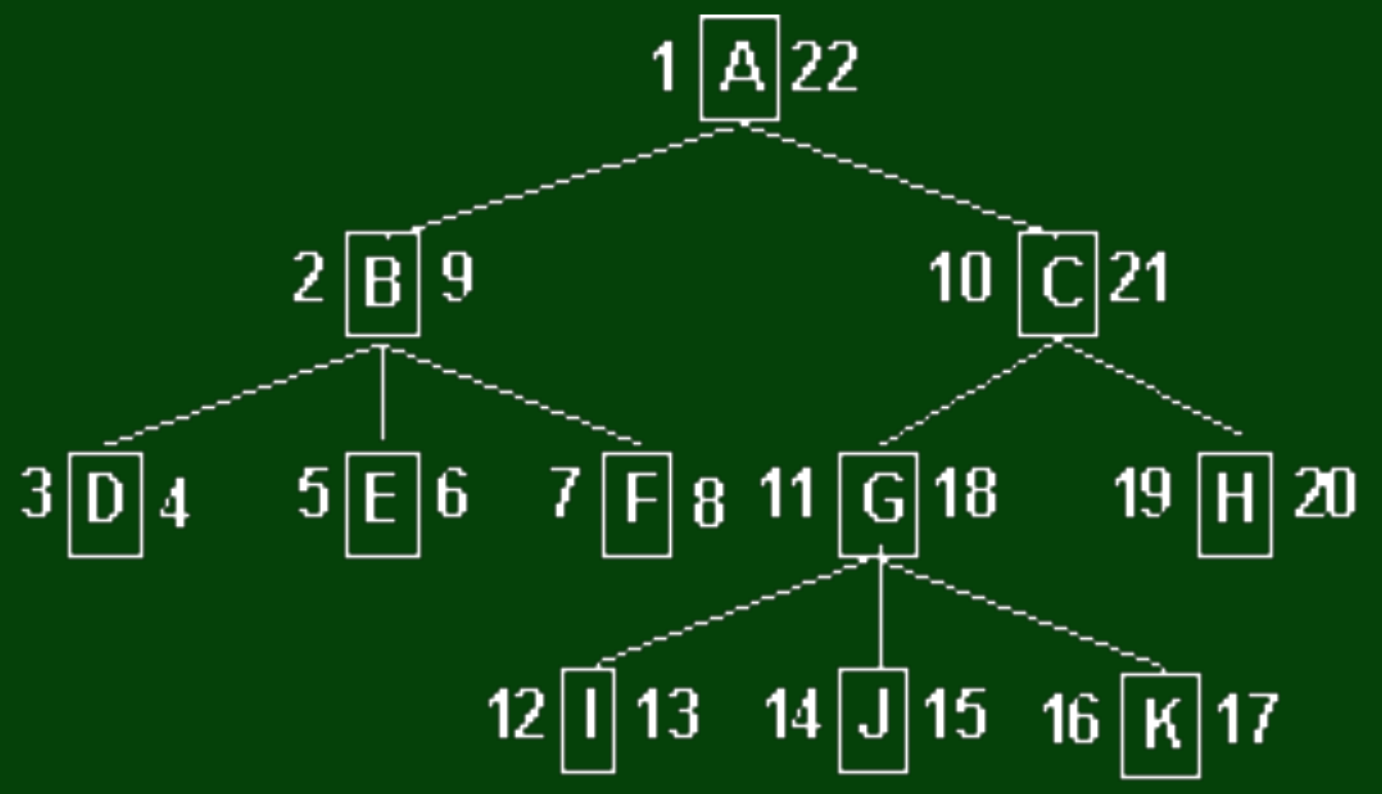
```
CONSTRAIN fk_parent_id_id
FOREIGN KEY parent_id REFERENCES some_table.id
```

materialised path

```
path = 1.2.23.55.786.33425
```

nested sets (MPTT)

Node	Left	Right	Depth
A	1	22	1
B	2	9	2
C	10	21	2
D	3	4	3
E	5	6	3
F	7	8	3
G	11	18	3
H	19	20	3
I	12	13	4
J	14	15	4
K	16	17	4



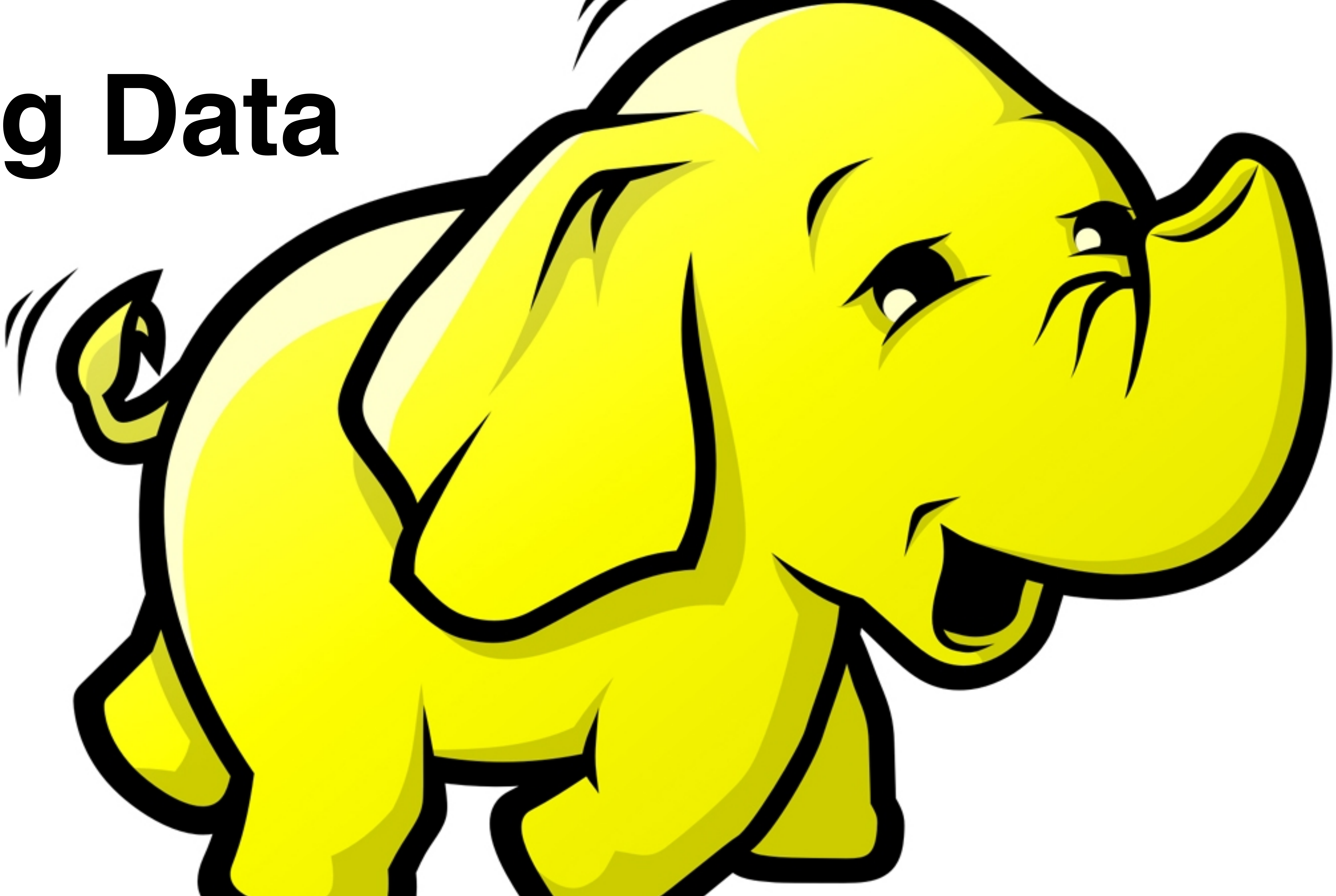
Velocity

https://www.flickr.com/photos/jiteshjagadish

when locks attack…

Don’t get ACID on the cuts

Big Data



SQL on Hadoop

More than SQL

Shark

Drill

Cascading

Map Reduce

System issues,
Speed issues,
Soft issues

the ACID, BASE litmus

Atomic

Consistent

Isolated

Durable

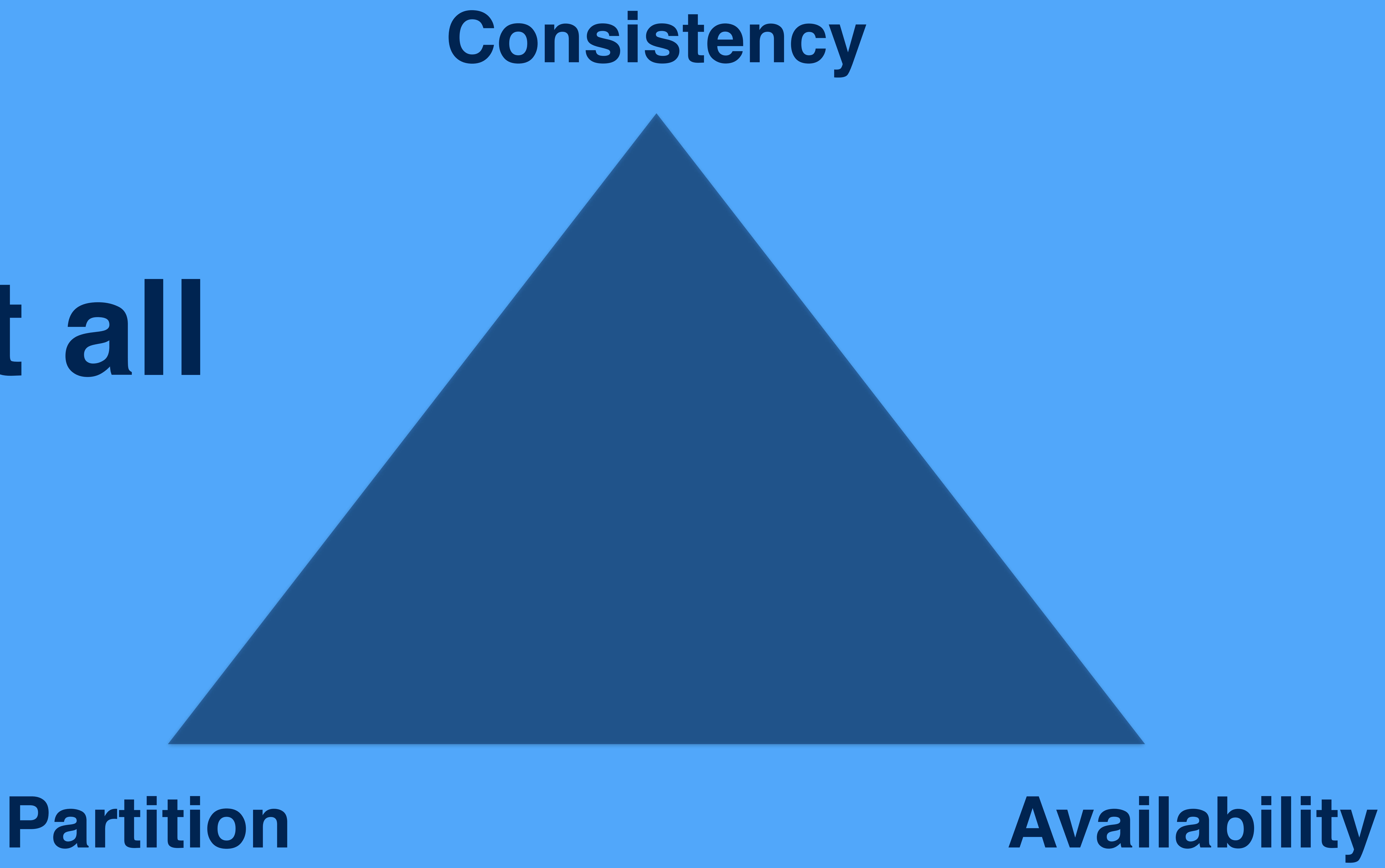
Basically **A**vailable

Soft-state

Eventually consistent

what matters to you?

CAP it all



write fast, ask questions later

SQL writes cost a lot

mainly write workload: NoSQL

low latency write workload: NoSQL

is it web scale?

most NoSQL scales well

but clusters still need management

are you facebook? one machine is easier than n

can ops handle it? app developers make bad admins

who is going to use it?

analysts: they want SQL

developers: they want applications

data scientists: they want access

choose the right tool

Photo: http://www.homespothq.com/

Thank you!

Simon Elliston Ball
simon@simonellistonball.com

@sireb
#noSQLknowSQL

http://nosqlknowsql.io

redgate
ingeniously simple

Questions

Simon Elliston Ball
simon@simonellistonball.com

@sireb

http://nosqlknowsql.io

redgate
ingeniously simple