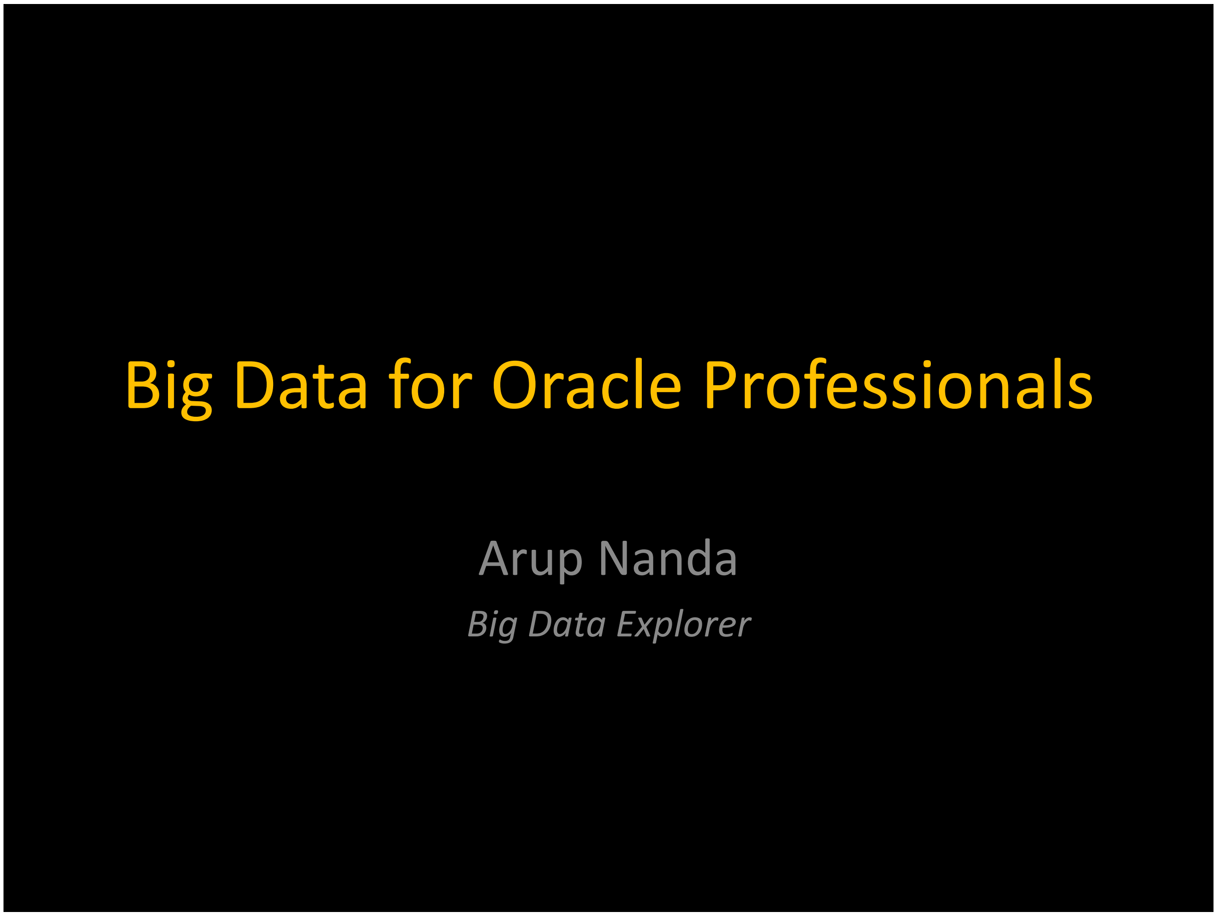
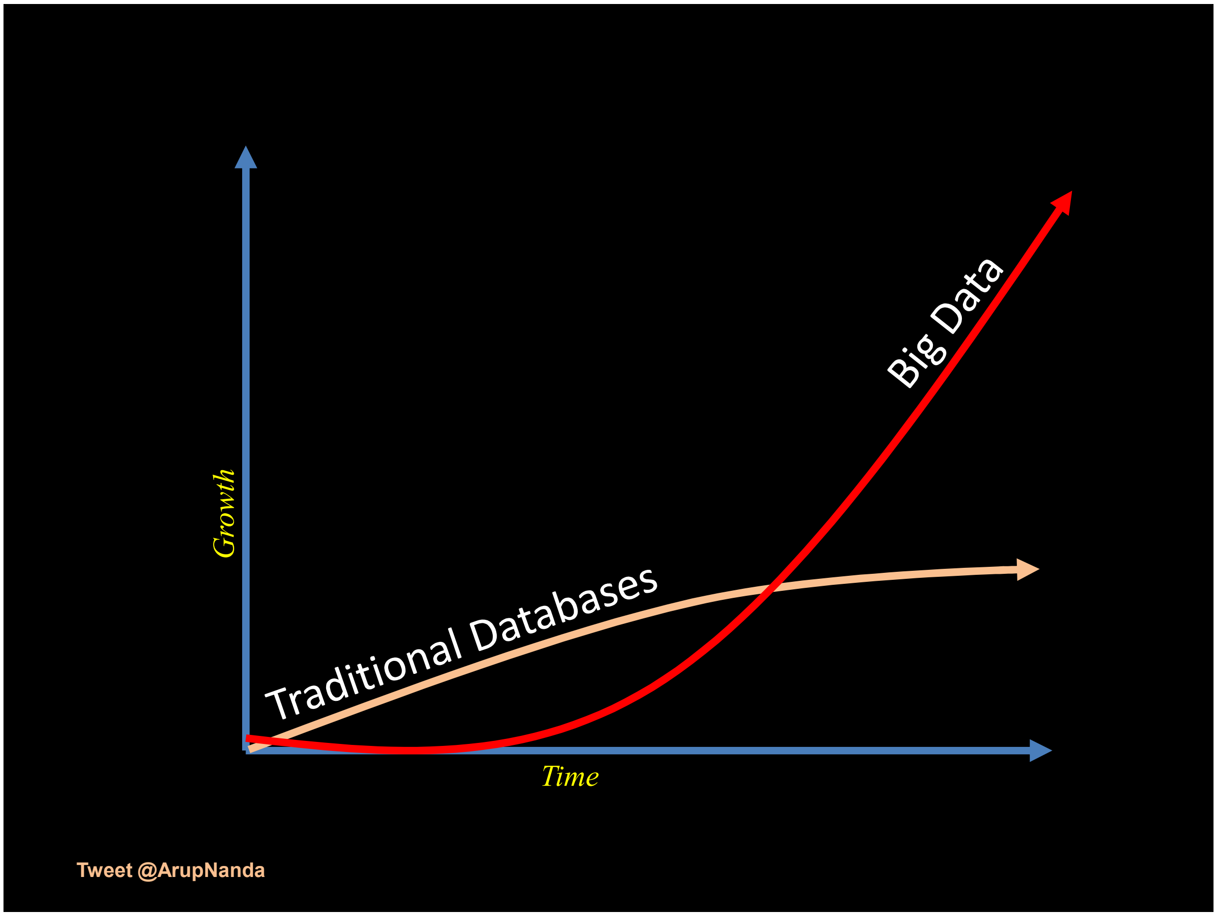


Big Data for Oracle Professionals

Arup Nanda

Big Data Explorer



Tweet @ArupNanda

NoSQL

YARN

Hadoop

Map/Reduce

Spark

Flume.

Tweet @ArupNanda

YAHOO!

```
fcrawler.looksmart.com - - [26/Apr/2000:00:00:12 -0400] "GET /contacts.html
HTTP/1.0" 200 4595 "-" "FAST-WebCrawler/2.1-pre2 (ashen@looksmart.net)"
fcrawler.looksmart.com - - [26/Apr/2000:00:17:19 -0400] "GET /news/news.html
HTTP/1.0" 200 16716 "-" "FAST-WebCrawler/2.1-pre2 (ashen@looksmart.net)"

ppp931.on.bellglobal.com - - [26/Apr/2000:00:16:12 -0400] "GET
/download/windows/asctab31.zip HTTP/1.0" 200 1540096
"http://www.htmlgoodies.com/downloads/freeware/webdevelopment/15.html"
"Mozilla/4.7 [en]C-SYMPA  (Win95; U)"

123.123.123.123 - - [26/Apr/2000:00:23:48 -0400] "GET /pics/wpaper.gif HTTP/1.0"
200 6248 "http://www.jafsoft.com/asctortf/" "Mozilla/4.05 (Macintosh; I; PPC)"
123.123.123.123 - - [26/Apr/2000:00:23:47 -0400] "GET /asctortf/ HTTP/1.0" 200
8130 "http://search.netscape.com/Computers/Data_Formats/Document/Text/RTF"
"Mozilla/4.05 (Macintosh; I; PPC)"
123.123.123.123 - -
```

Tweet @ArupNanda

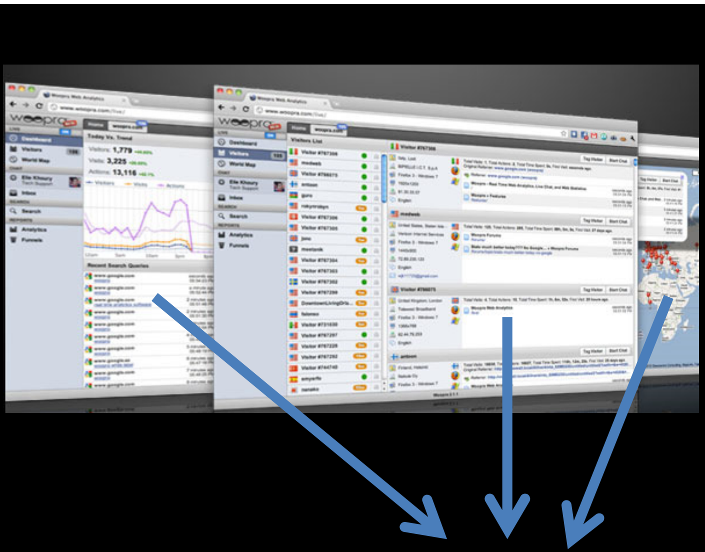
```
fcrawler.looksmart.com - - [26/Apr/2000:00:00:12 -0400] "GET
/contacts.html HTTP/1.0" 200 4595 "-" "FAST-WebCrawler/2.1-pre2
(ashen@looksmart.net)"
fcrawler.looksmart.com - - [26/Apr/2000:00:17:19 -0400] "GET
/news/news.html HTTP/1.0" 200 16716 "-" "FAST-WebCrawler/2.1-pre2
(ashen@looksmart.net)"

ppp931.on.bellglobal.com - - [26/Apr/2000:00:16:12 -0400] "GET
/download/windows/asctab31.zip HTTP/1.0" 200 1540096
"http://www.htmlgoodies.com/downloads/freeware/webdevelopment/15.
html" "Mozilla/4.7 [en]C-SYMPA  (Win95; U)"

123.123.123.123 - - [26/Apr/2000:00:23:48 -0400] "GET
/pics/wpaper.gif HTTP/1.0" 200 6248
"http://www.jafsoft.com/asctortf/" "Mozilla/4.05 (Macintosh; I;
PPC)"
123.123.123.123 - - [26/Apr/2000:00:23:47 -0400] "GET /asctortf/
HTTP/1.0" 200 8130
"http://search.netscape.com/Computers/Data_Formats/Document/Text/
RTF" "Mozilla/4.05 (Macintosh; I; PPC)"
123.123.123.123 - -
```

petabytes
unpredictable format
transient.

Tweet @ArupNanda



Metadata Repository

Google

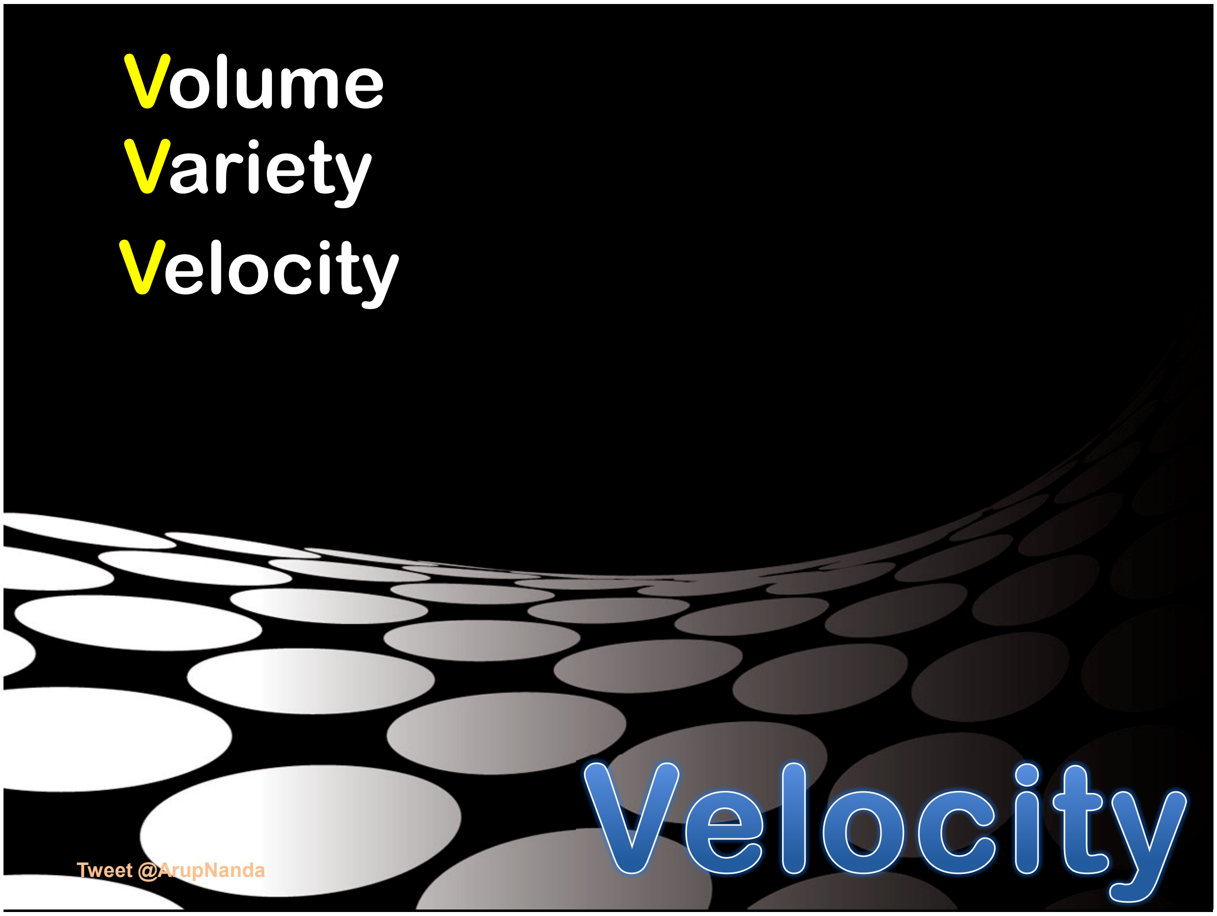
Tweet @ArupNanda

Volume

Tweet @ArupNanda

Variety

Tweet @ArupNanda

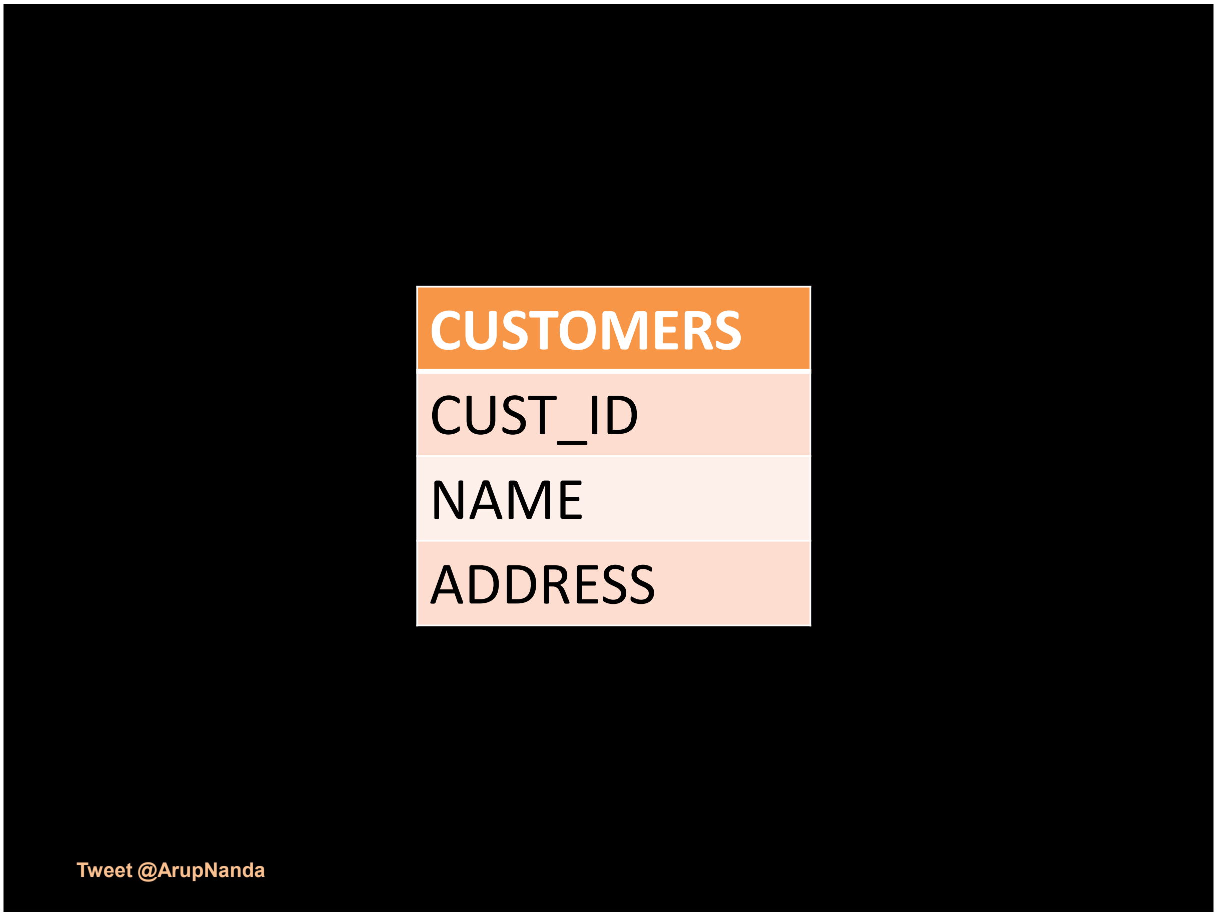


Volume

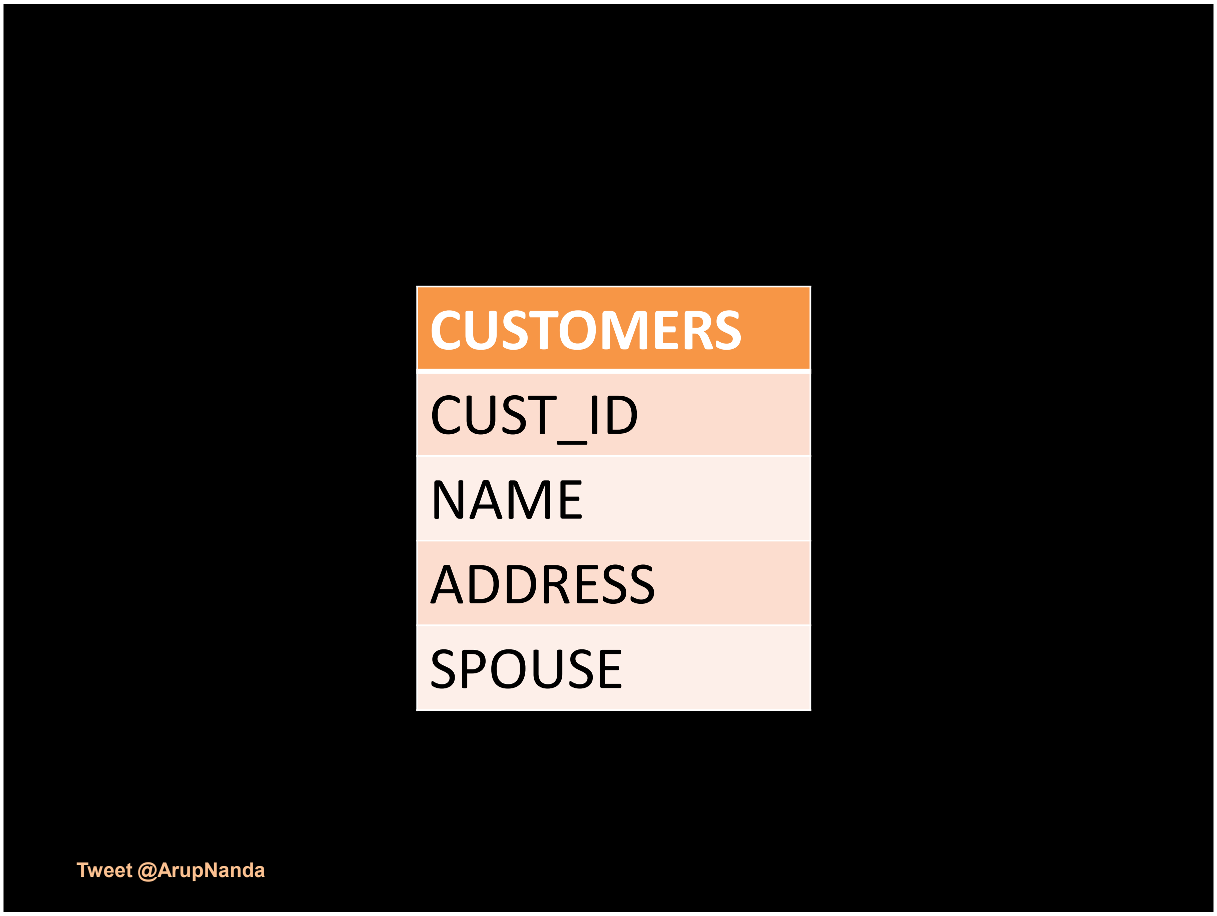
Variety

Velocity

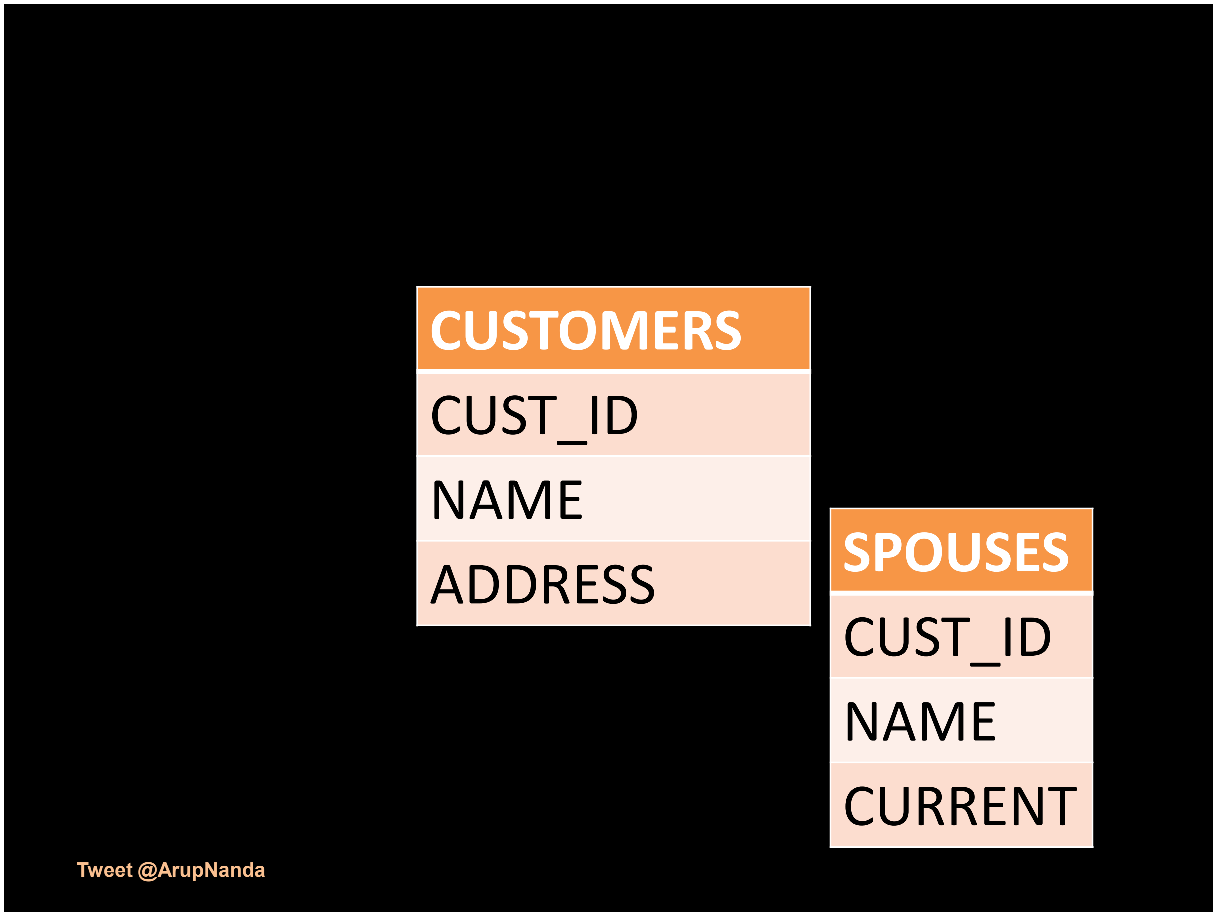
Velocity



CUSTOMERS
CUST_ID
NAME
ADDRESS



CUSTOMERS
CUST_ID
NAME
ADDRESS
SPOUSE



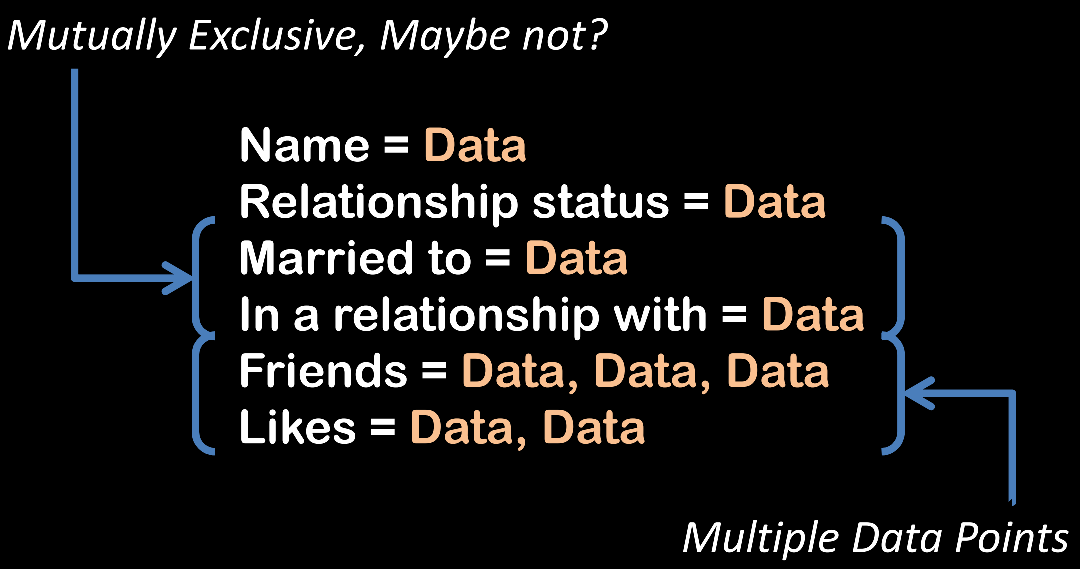
CUSTOMERS
CUST_ID
NAME
ADDRESS

SPOUSES
CUST_ID
NAME
CURRENT

EMPLOYERS
CUST_ID
NAME
CURRENT

CUSTOMERS
CUST_ID
NAME
ADDRESS

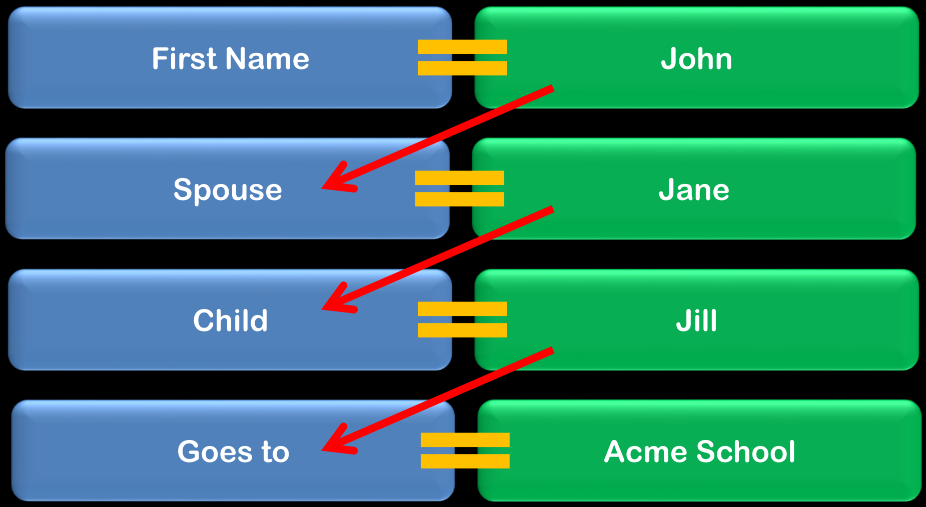
SPOUSES
CUST_ID
NAME
CURRENT



Mutually Exclusive, Maybe not?

Name = Data
Relationship status = Data
Married to = Data
In a relationship with = Data
Friends = Data, Data, Data
Likes = Data, Data

Multiple Data Points

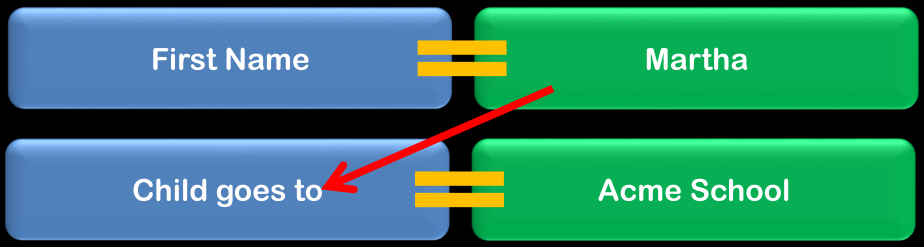


First Name = John

Spouse = Jane

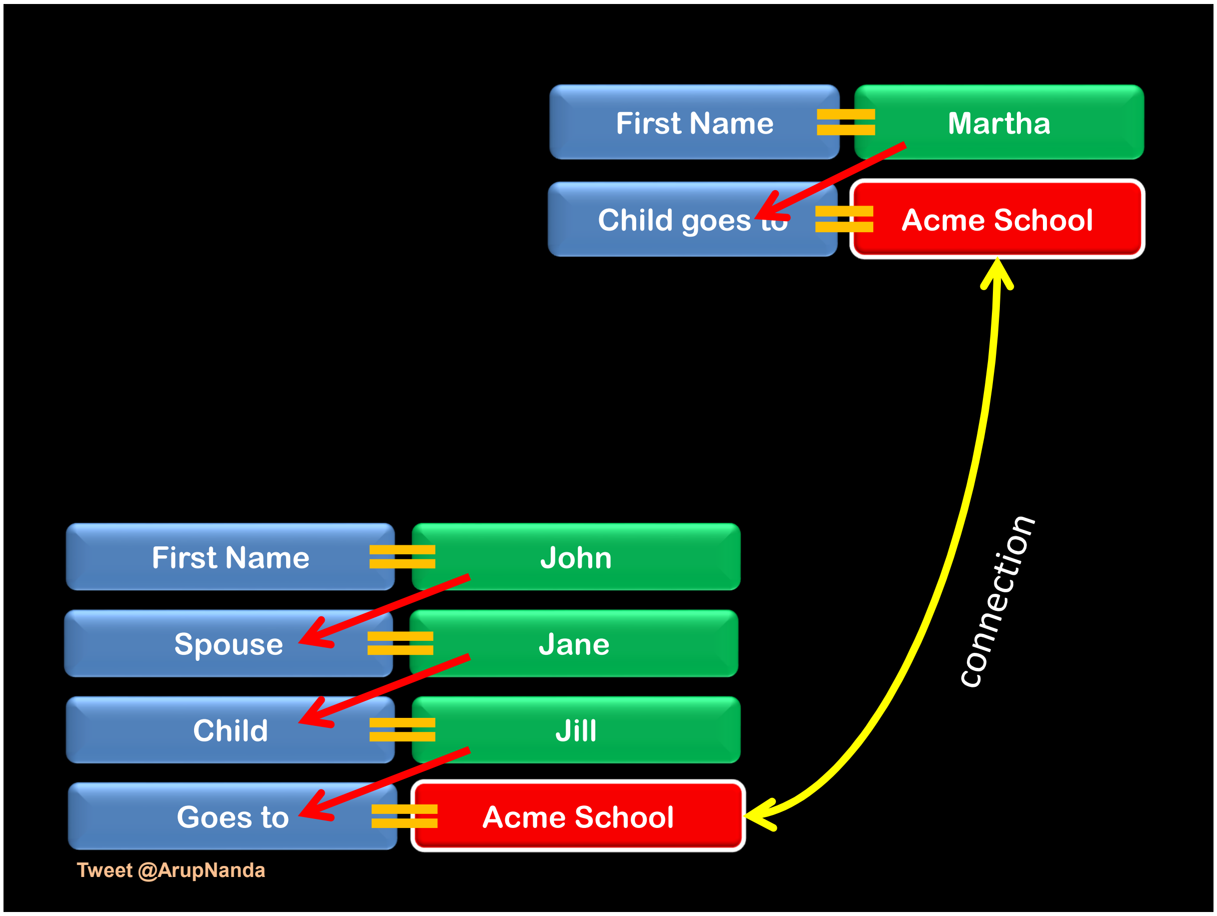
Child = Jill

Goes to = Acme School

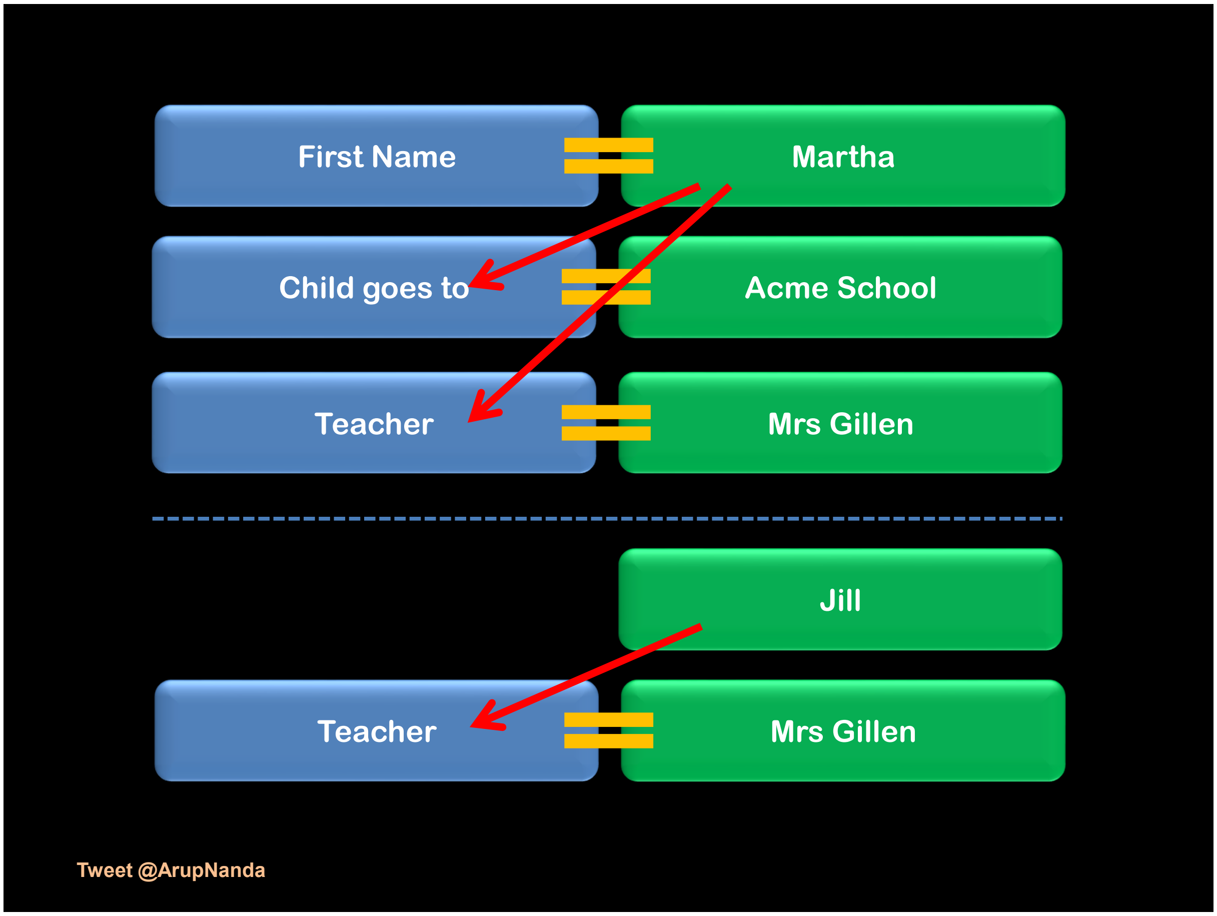


First Name = Martha

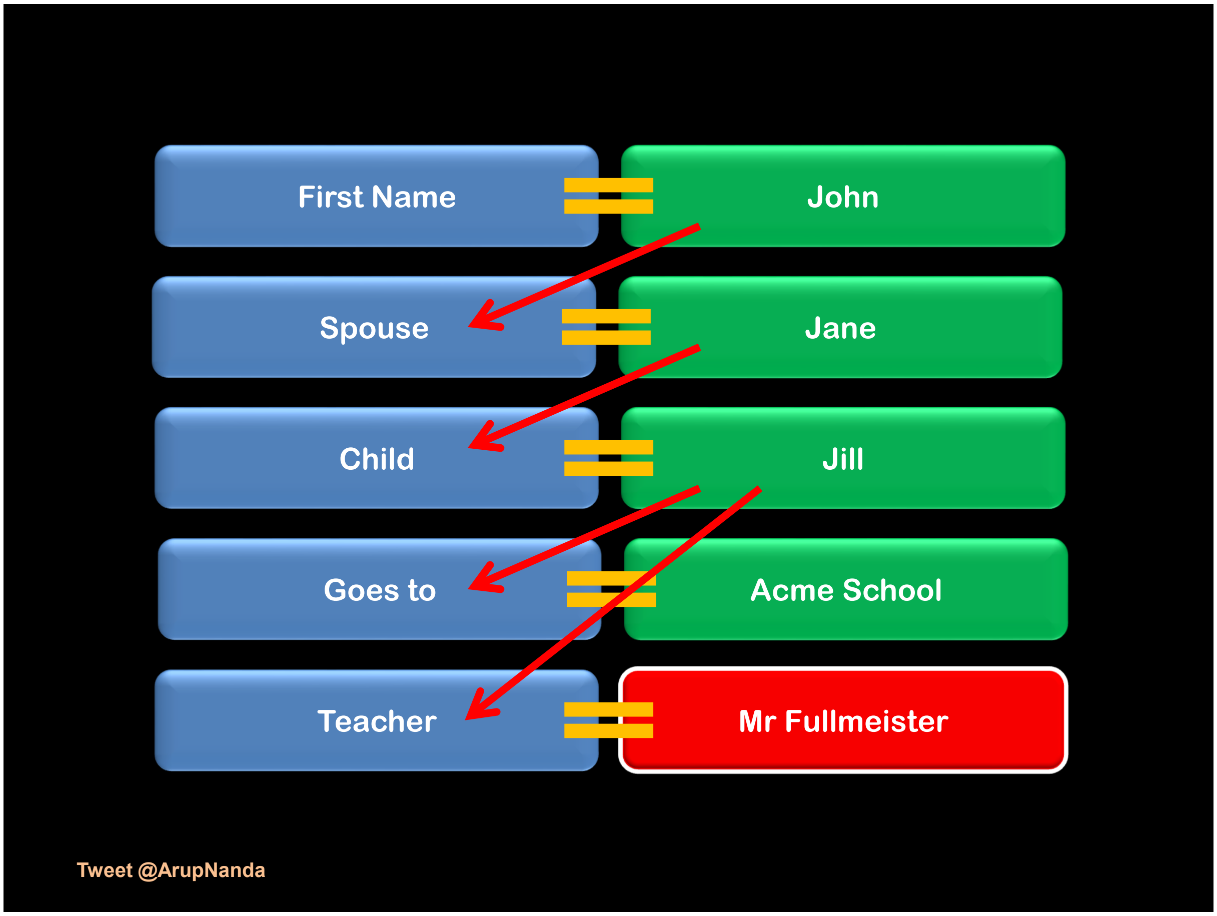
Child goes to = Acme School



Tweet @ArupNanda

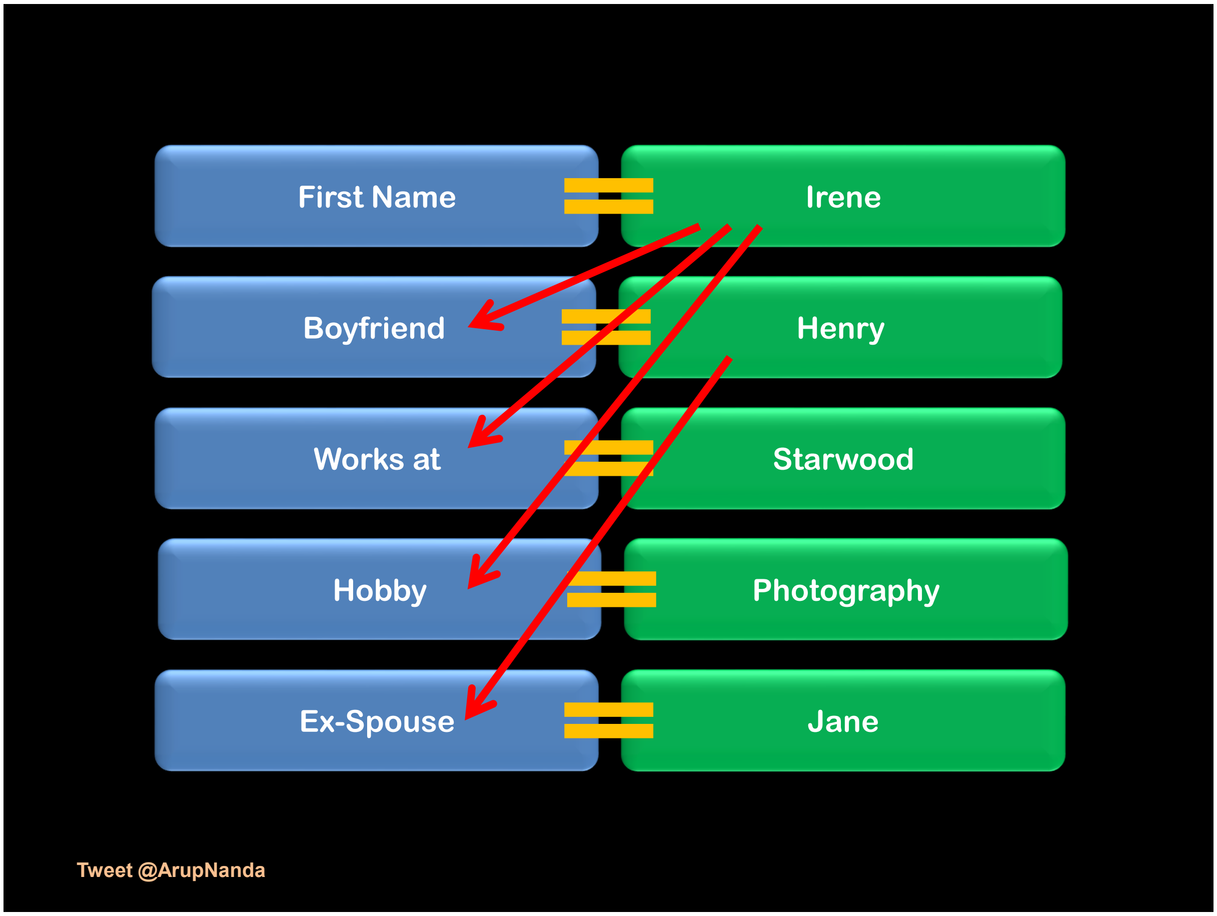


Tweet @ArupNanda



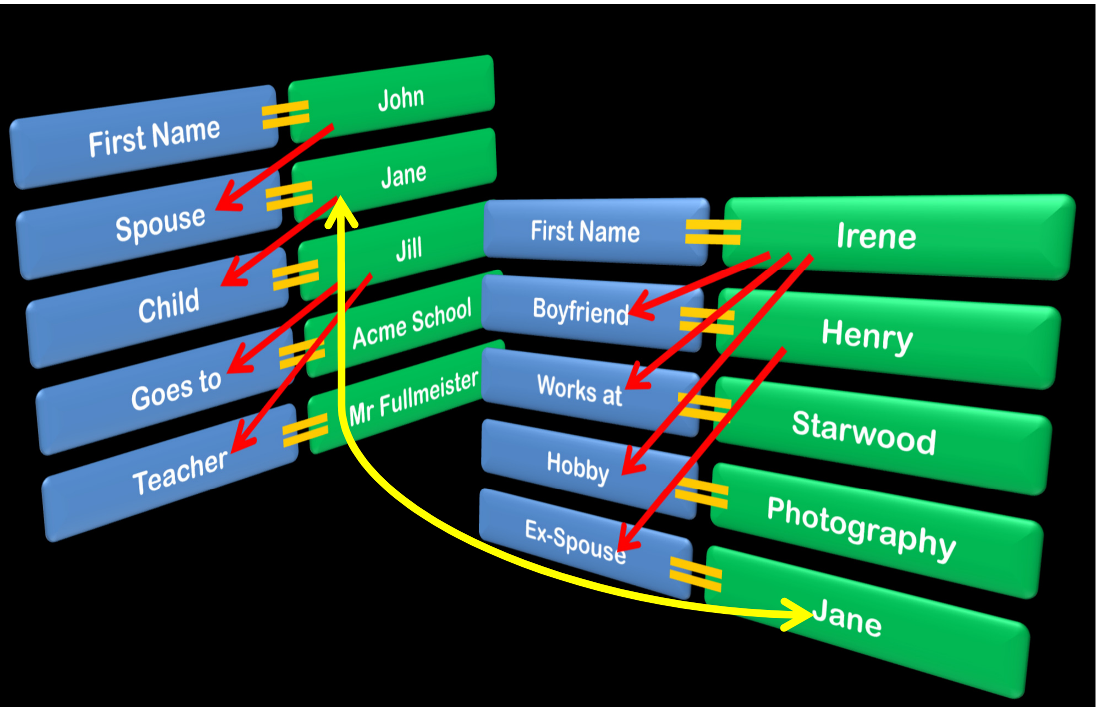
First Name	John
Spouse	Jane
Child	Jill
Goes to	Acme School
Teacher	Mr Fullmeister

Tweet @ArupNanda



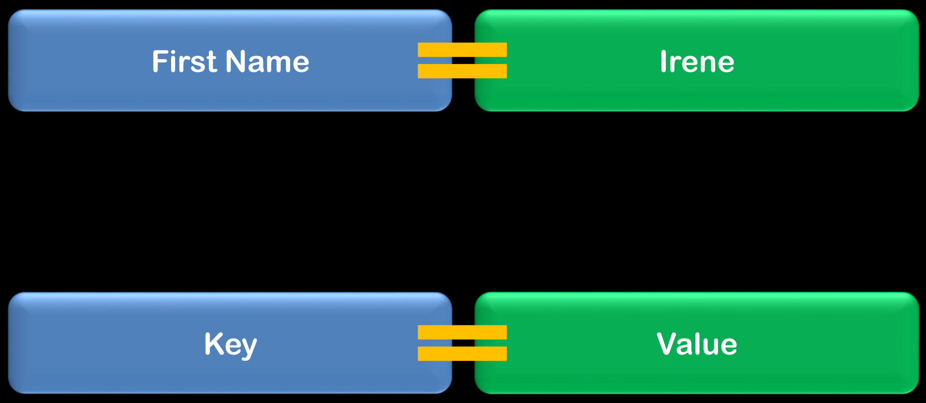
First Name	Irene
Boyfriend	Henry
Works at	Starwood
Hobby	Photography
Ex-Spouse	Jane

Tweet @ArupNanda



First Name = John
Spouse = Jane
Child = Jill
Goes to = Acme School
Teacher = Mr Fullmeister

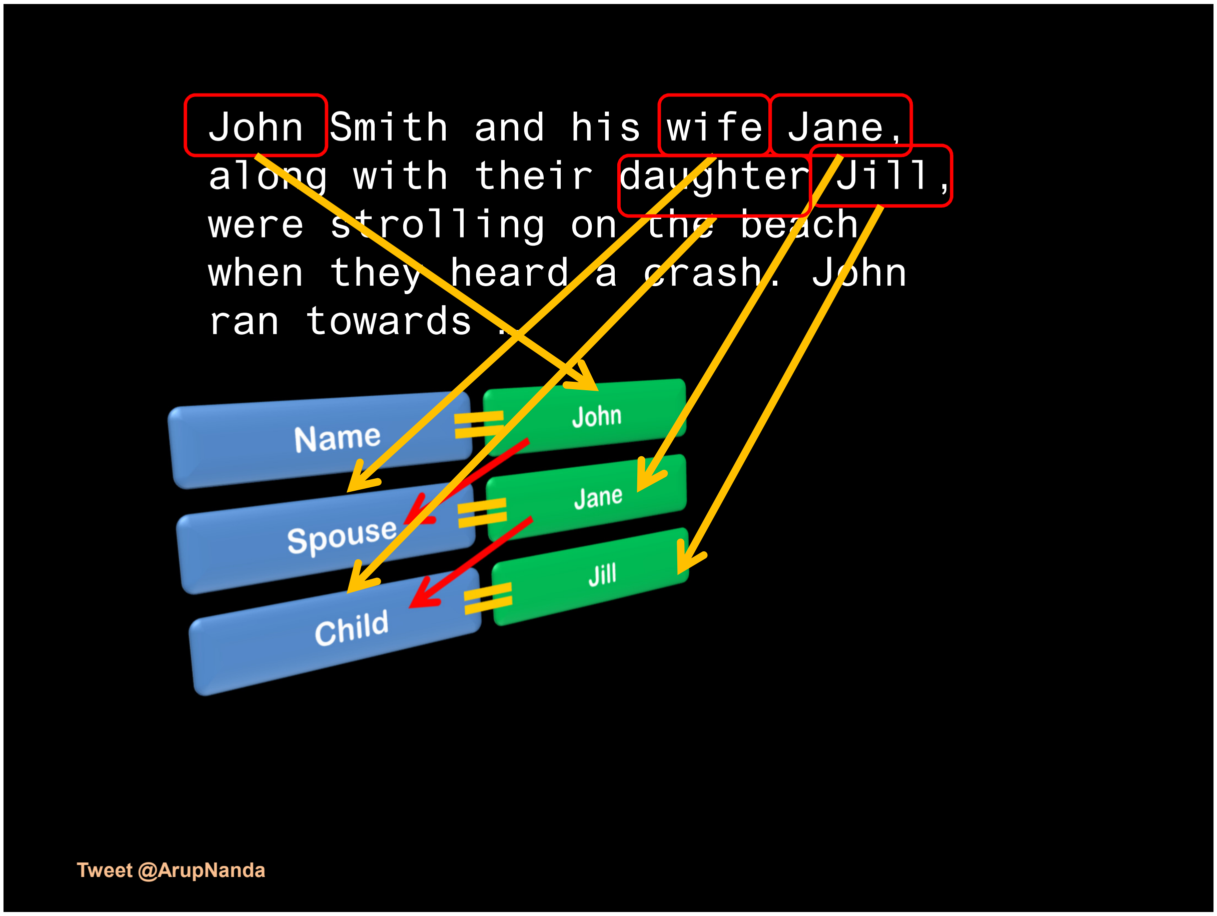
First Name = Irene
Boyfriend = Henry
Works at = Starwood
Hobby = Photography
Ex-Spouse = Jane



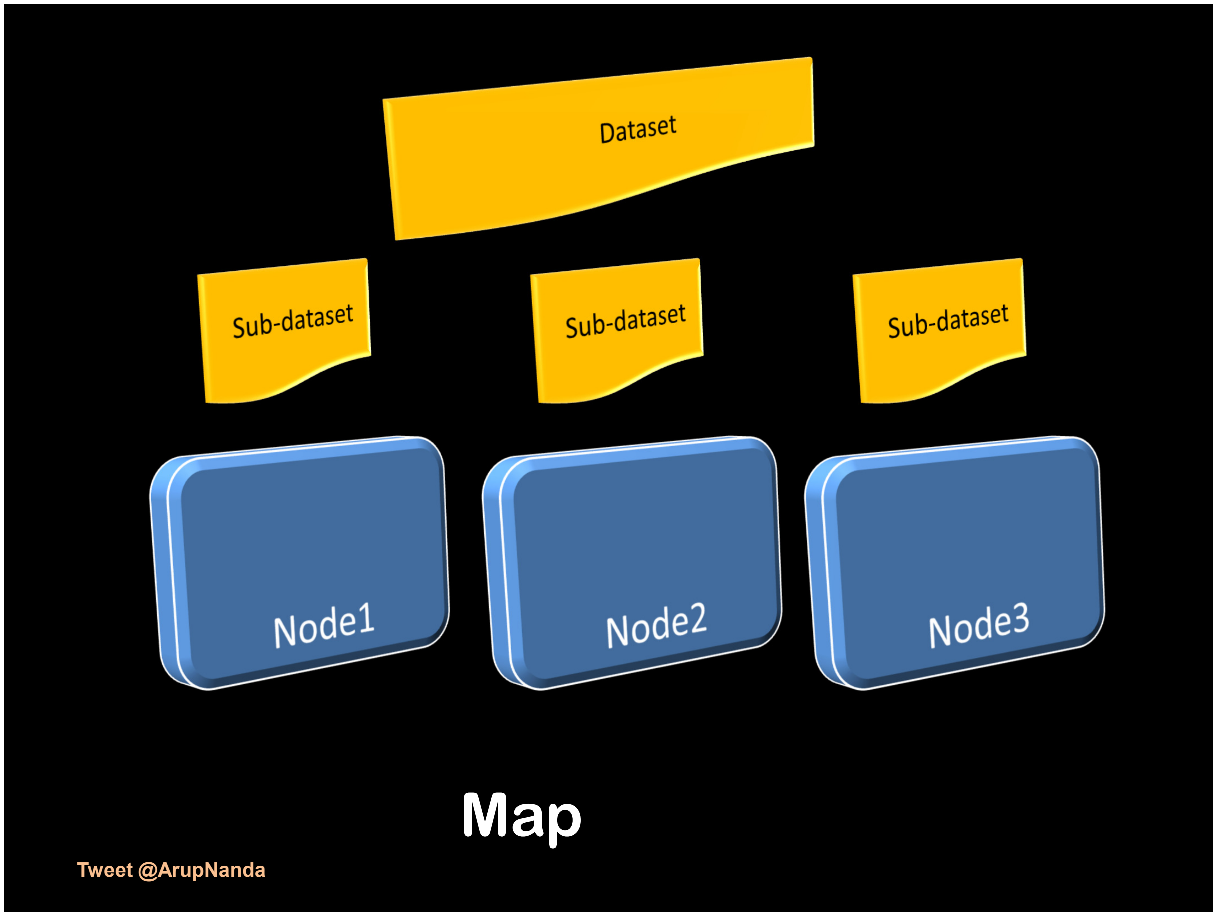
First Name = Irene

Key = Value

Key-Value Pair



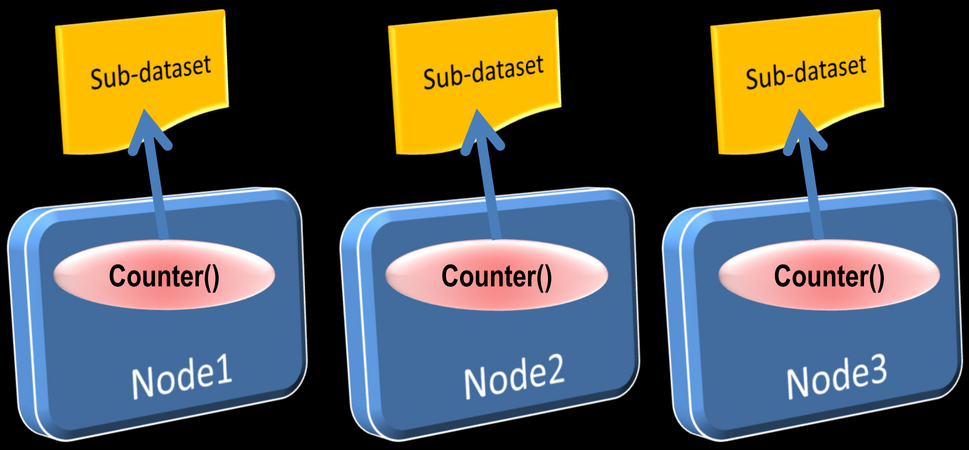
John Smith and his wife Jane, along with their daughter Jill, were strolling on the beach when they heard a crash. John ran towards .

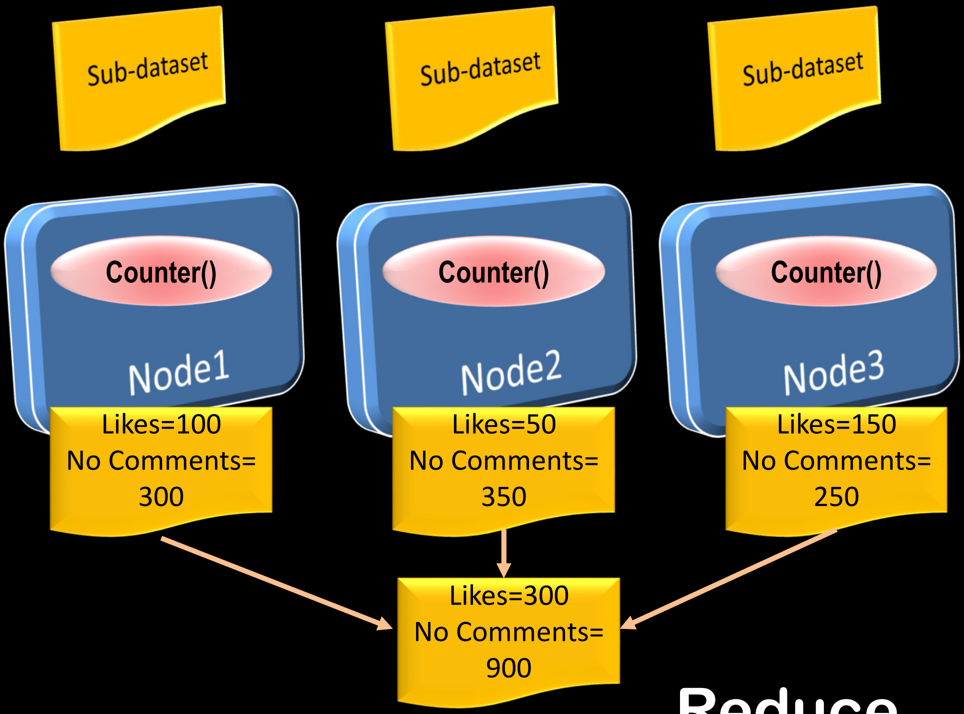


Map

Counter()

```
begin
  get post
  while (there_are_remaining_posts) loop
     extract status of "like" for the specific post
     if status = "like" then
         like_count := like_count + 1
     else
         no_comment := no_comment + 1
     end if
  end loop
end
```





Sub-dataset

Sub-dataset

Sub-dataset

Counter()

Node1

Counter()

Node2

Counter()

Node3

Likes=100
No Comments=
300

Likes=50
No Comments=
350

Likes=150
No Comments=
250

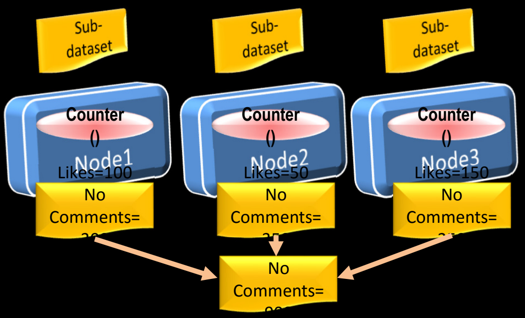
Likes=300
No Comments=
900

Reduce

Dividing the work among different nodes

Map / Reduce

Collating the results to get final answer



- Divide the workload
- Submit and track the jobs
- If a job fails, restart it on another node
- …

Hadoop

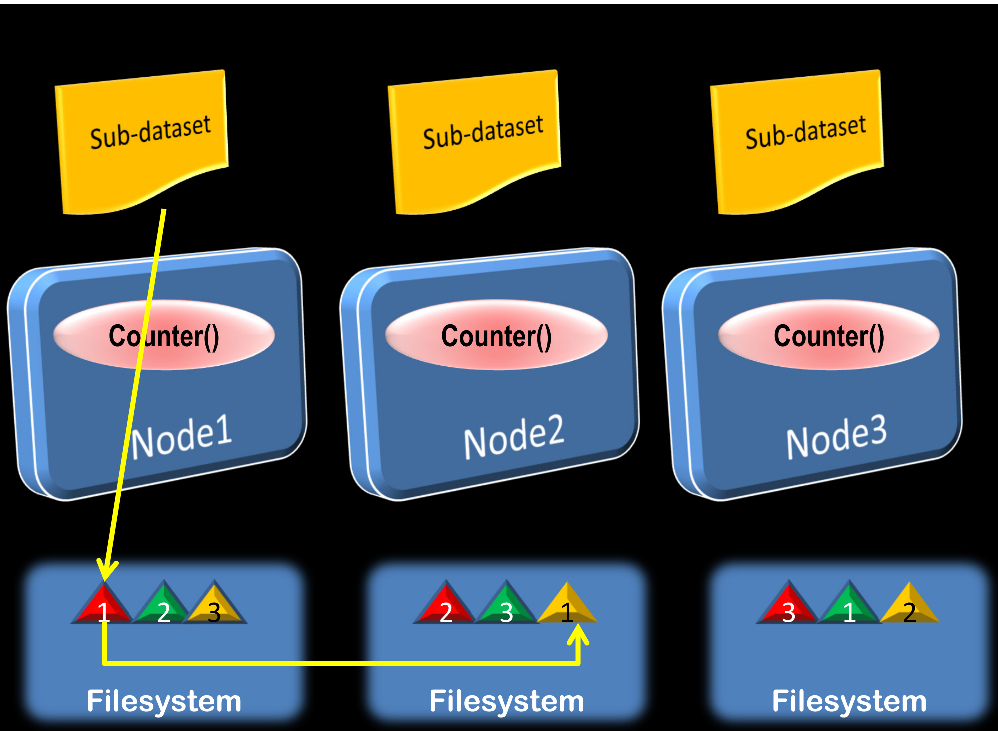
Resource Management

YARN

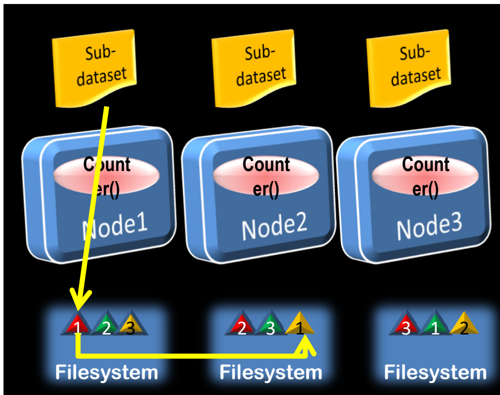
Yet Another Resource Negotiator

Map Reduce v2.

Applications

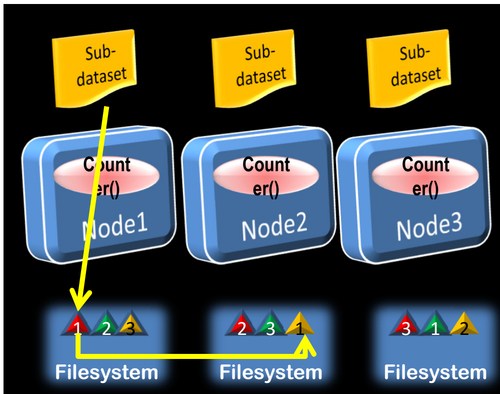


Hadoop Distributed Filesystem (HDFS)



Comparison with RAC

- *Not shared storage*
- *Data is discrete*
- *Version control not required*
- *Concurrency not required*
- *Transactional integrity across nodes not required.*



Advantages of Hadoop

- Processors need not be super-fast
- Immensely scalable
- Storage is redundant by design
- No RAID level required.

Tweet @ArupNanda

Scalable?

ACID Properties

Reliability at a cost

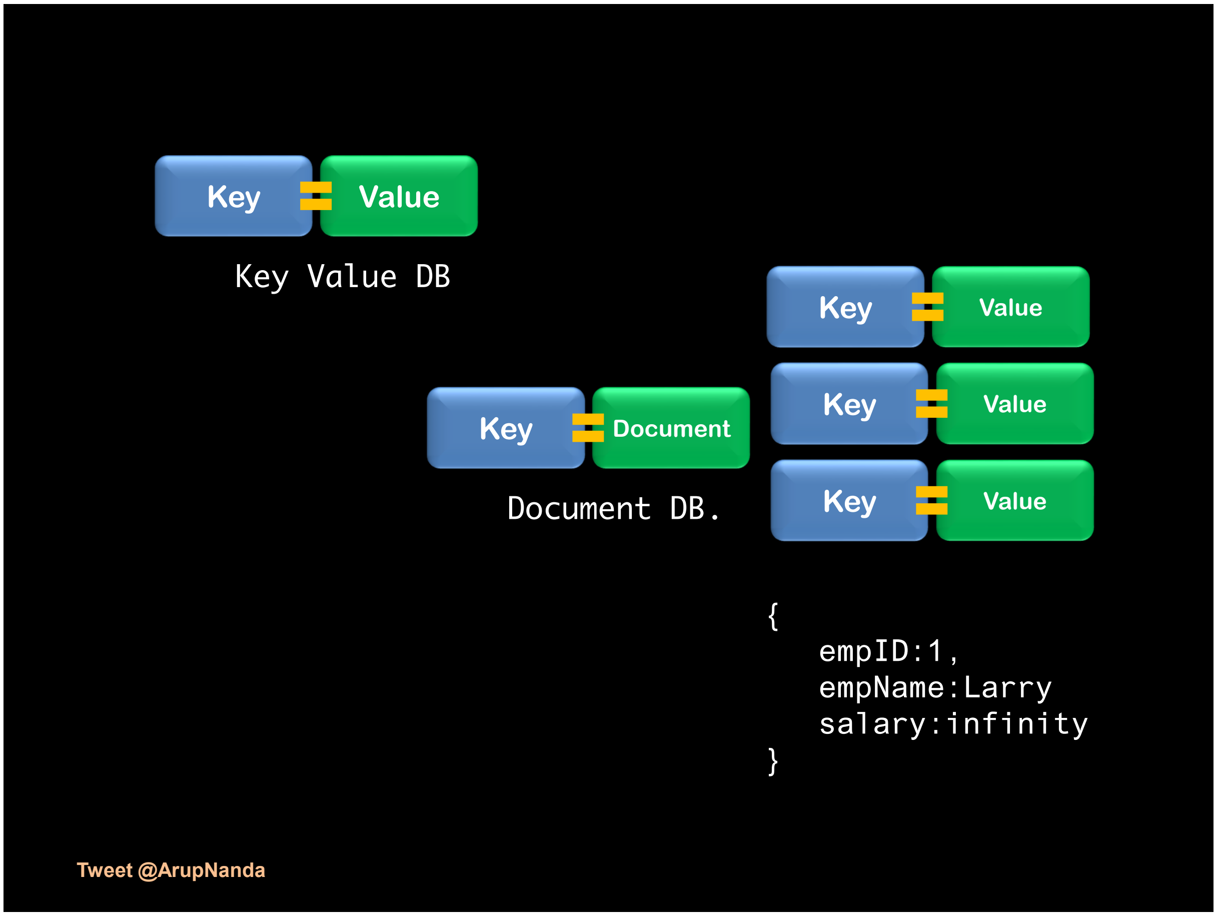
Large overhead in data processing

Tweet @ArupNanda

Website logs
Combine with structured data
SOAP Messages
Twitter, Facebook …

Data Access: through programs

NoSQL Databases



Key Value DB

Document DB.

```
{
    empID:1,
    empName:Larry
    salary:infinity
}
```

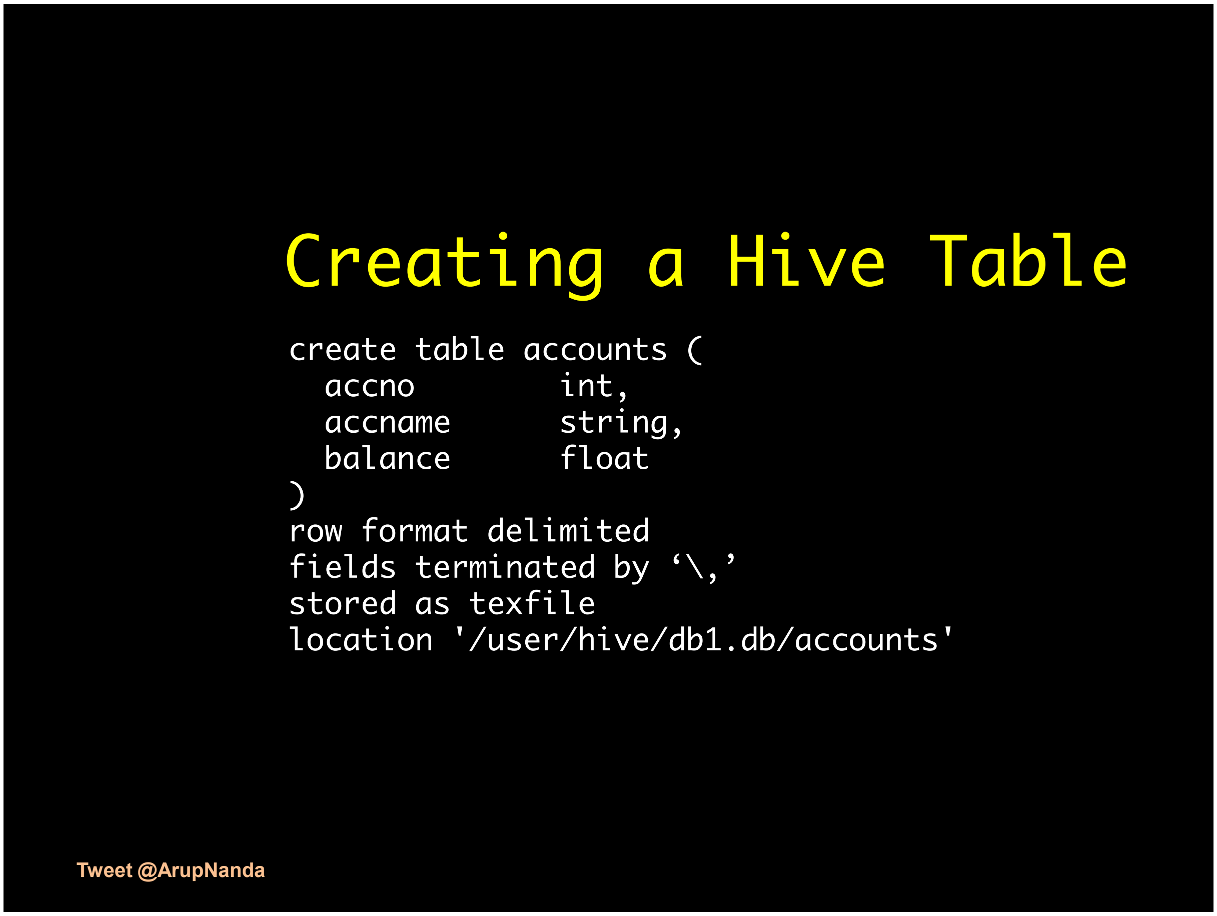
Tweet @ArupNanda

SQL-interface required

Hive

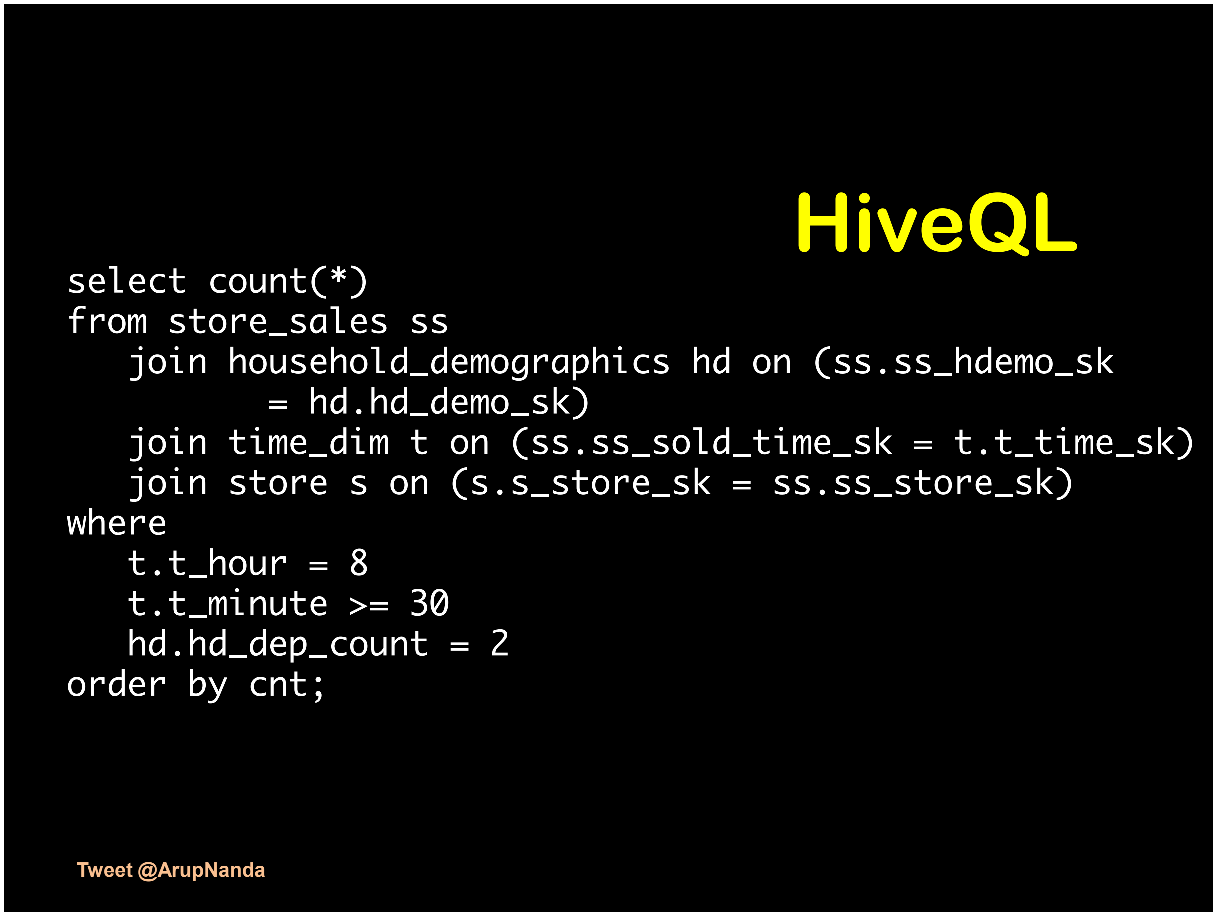
HiveQL

Tweet @ArupNanda



Creating a Hive Table

```
create table accounts (
  accno        int,
  accname      string,
  balance      float
)
row format delimited
fields terminated by ‘\,’
stored as texfile
location '/user/hive/db1.db/accounts'
```



HiveQL

```
select count(*)
from store_sales ss
   join household_demographics hd on (ss.ss_hdemo_sk
          = hd.hd_demo_sk)
   join time_dim t on (ss.ss_sold_time_sk = t.t_time_sk)
   join store s on (s.s_store_sk = ss.ss_store_sk)
where
   t.t_hour = 8
   t.t_minute >= 30
   hd.hd_dep_count = 2
order by cnt;
```

Map/Reduce

Divide the work and collate the results } *Needs development in Java, Python, Ruby, etc.*

A framework to work on the dataset in parallel **Pig**

Pig Latin *Scripting language for Pig*

Tweet @ArupNanda

SQL

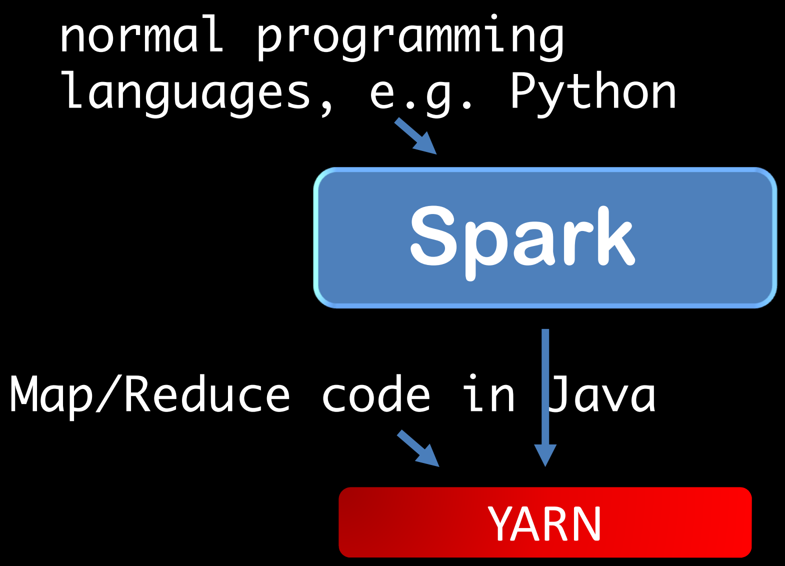
```
select category, avg(pagerank)
from urls
where pagerank > 0.2
group by category
having count(*) > 1000000
```

Pig Latin

```
good_urls = FILTER urls BY pagerank > 0.2;
groups = GROUP good_urls BY category;
big_groups = FILTER groups BY
COUNT(good_urls)>1000000;
output = FOREACH big_groups GENERATE category,
AVG(good_urls.pagerank);
```

Tweet @ArupNanda

HBase	A database built on Hadoop
HiveQL	An SQL-like (but not the same) query language
Pig	Procedural Logic without M/R Code.

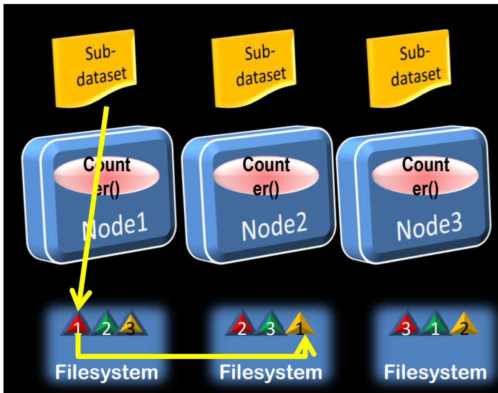


normal programming languages, e.g. Python

Spark

Map/Reduce code in Java

YARN



Hadoop processing in files

Memory is cheaper

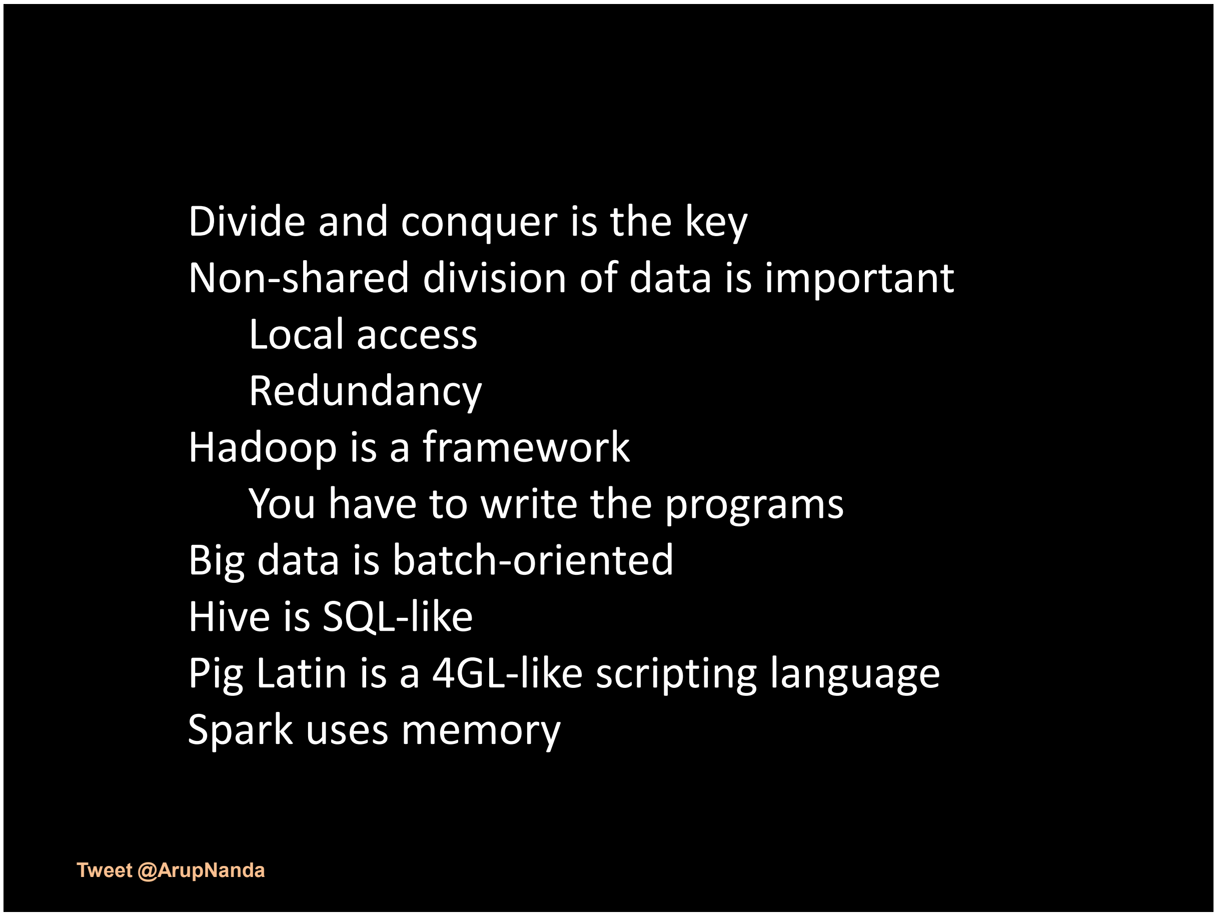
Interactive processing needs faster access.

SparkShell SparkSQL MLib SparkR PySpark

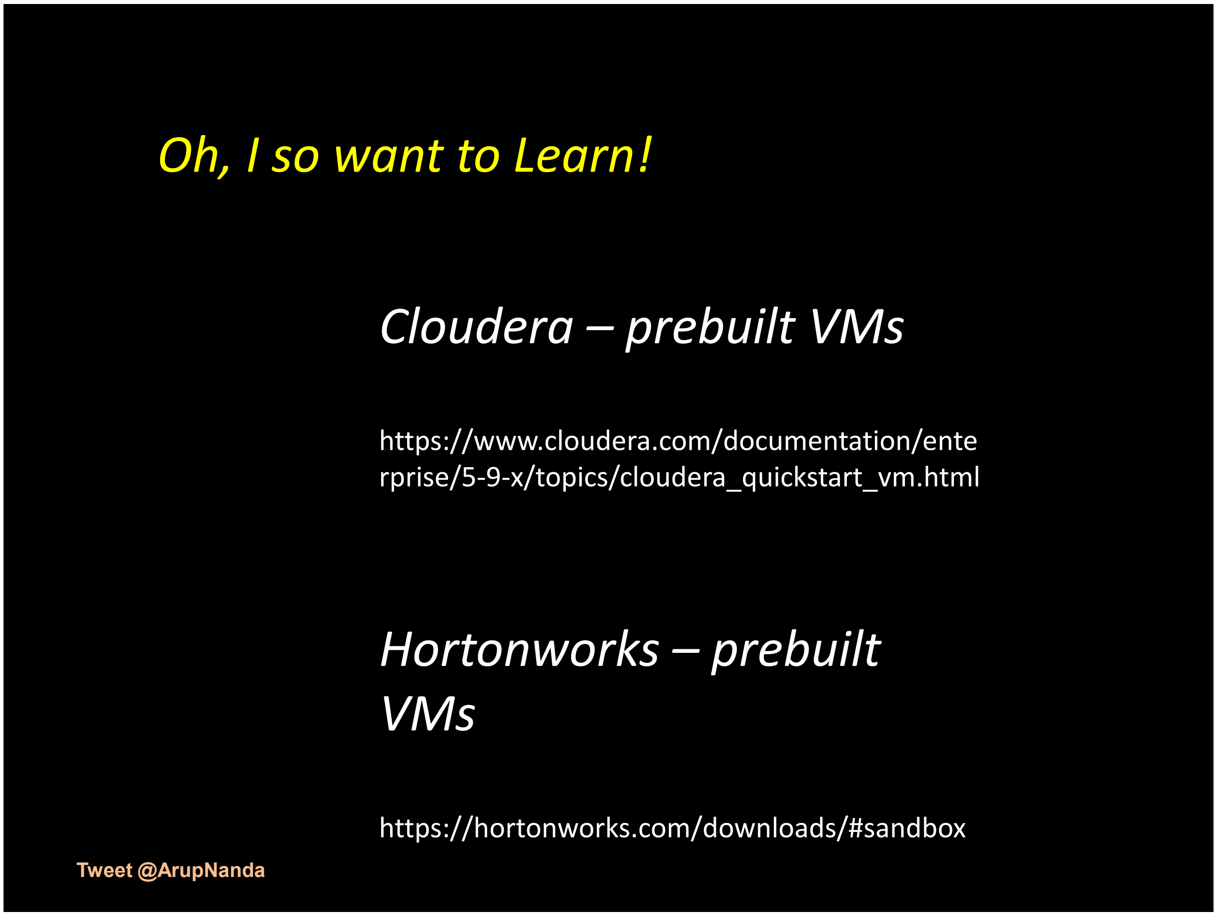
Spark

Core

Can use Java, Python or Scala



- Divide and conquer is the key
- Non-shared division of data is important
  - Local access
  - Redundancy
- Hadoop is a framework
  - You have to write the programs
- Big data is batch-oriented
- Hive is SQL-like
- Pig Latin is a 4GL-like scripting language
- Spark uses memory



Oh, I so want to Learn!

Cloudera – prebuilt VMs

https://www.cloudera.com/documentation/enterprise/5-9-x/topics/cloudera_quickstart_vm.html

Hortonworks – prebuilt VMs

https://hortonworks.com/downloads/#sandbox

Thanks!

arup.blogspot.com @ArupNanda