



# Self Service Data Exploration with Apache Drill

```
{ "name" : "Aditya Kishore", "github" : "adityakishore", "twitter" : "@adiore" }
```

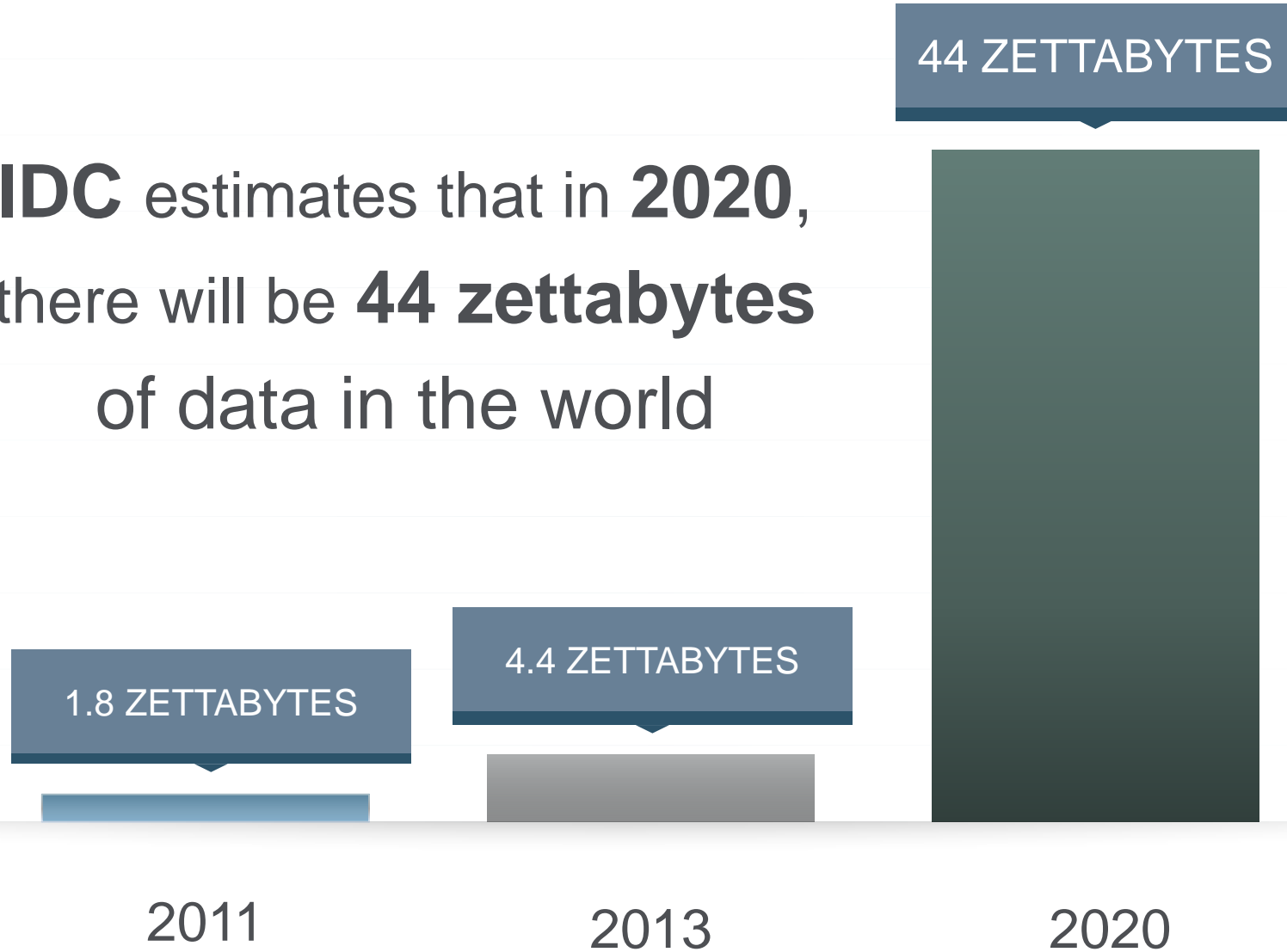
November 19, 2014 – Portland Big Data User Group



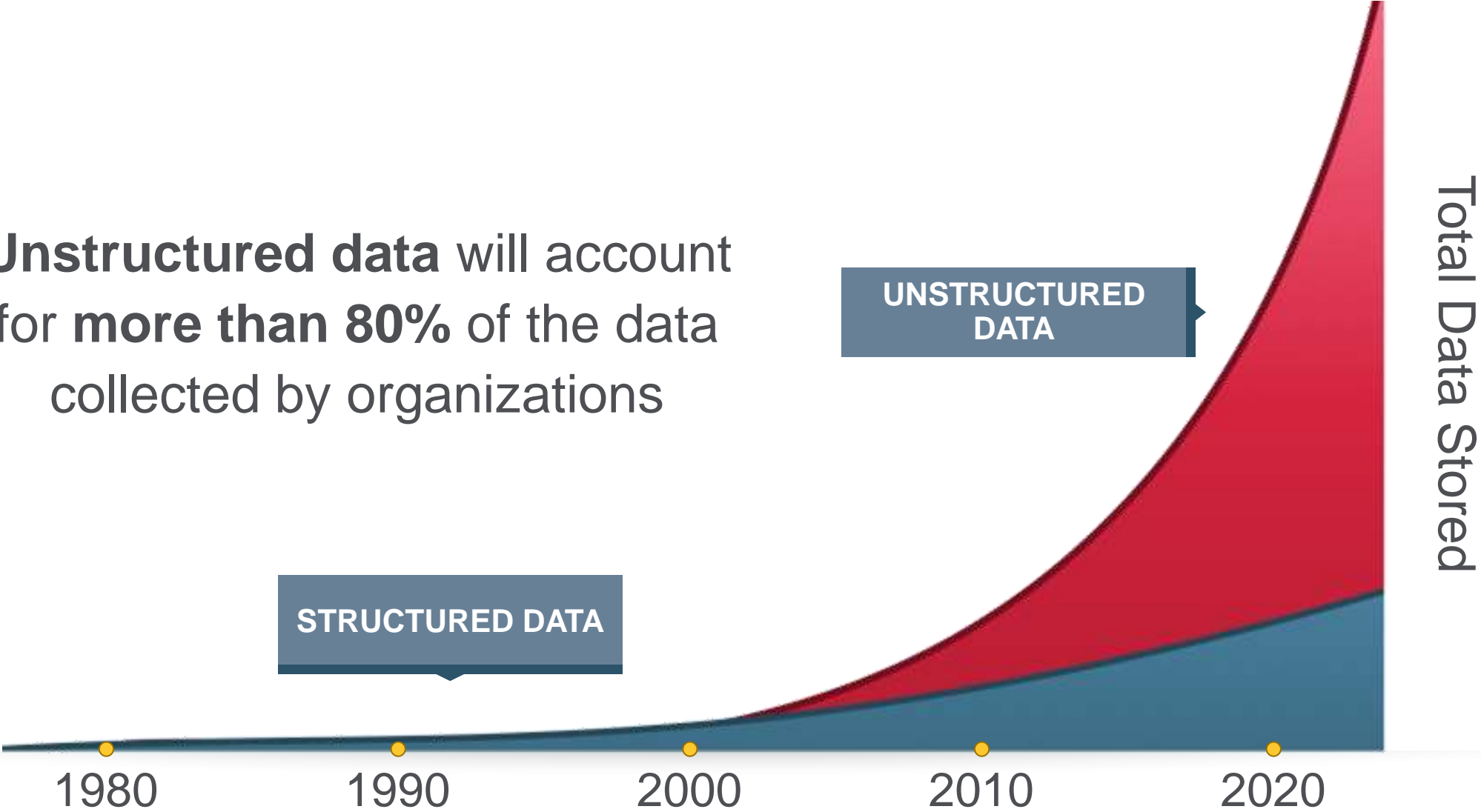
**Data is doubling in  
size every two years**



**IDC** estimates that in **2020**, there will be **44 zettabytes** of data in the world

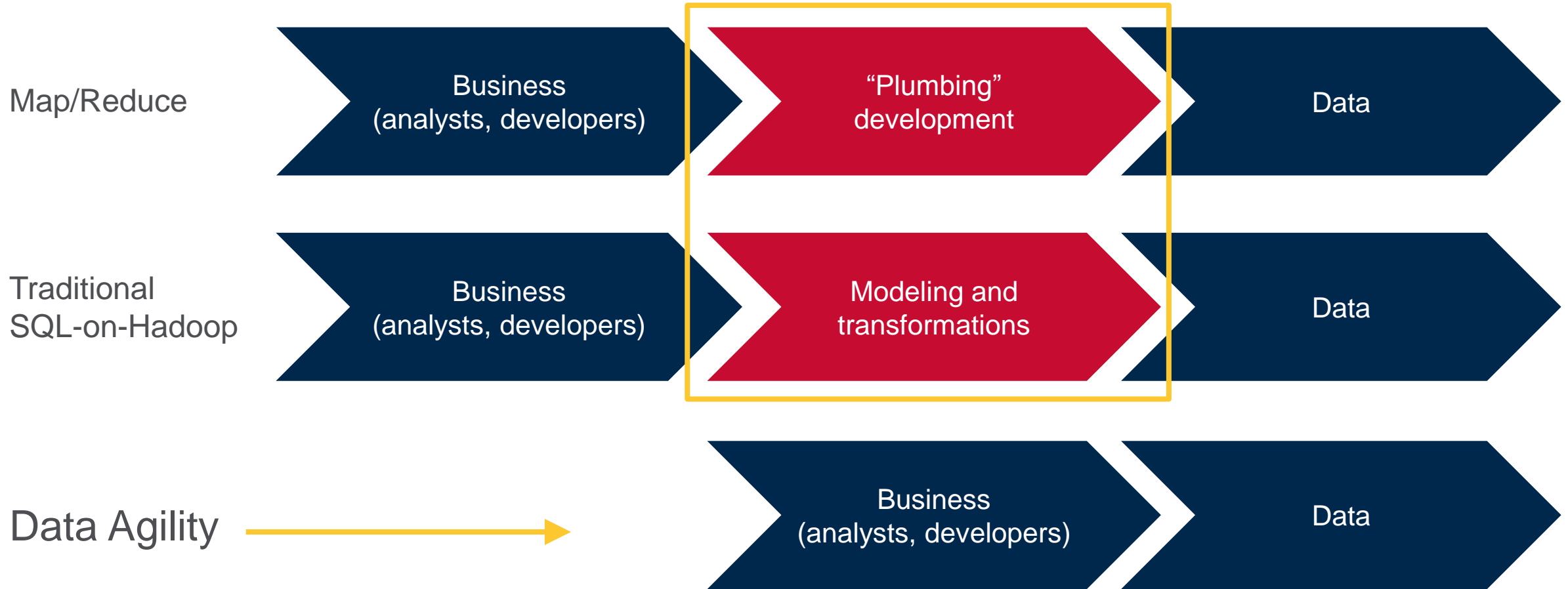


**Unstructured data** will account for **more than 80%** of the data collected by organizations



# Distance to Data

Existing approaches  
require a middleman (IT)



# SQL in a NoSchema World

## WANT

- SQL
- BI (Tableau, MicroStrategy, etc.)
- Low latency
- Scalability

## DON'T WANT

- Create and maintain schemas on:
  - HDFS (Parquet, JSON, etc.)
  - HBase
  - MongoDB
- Transform or copy data



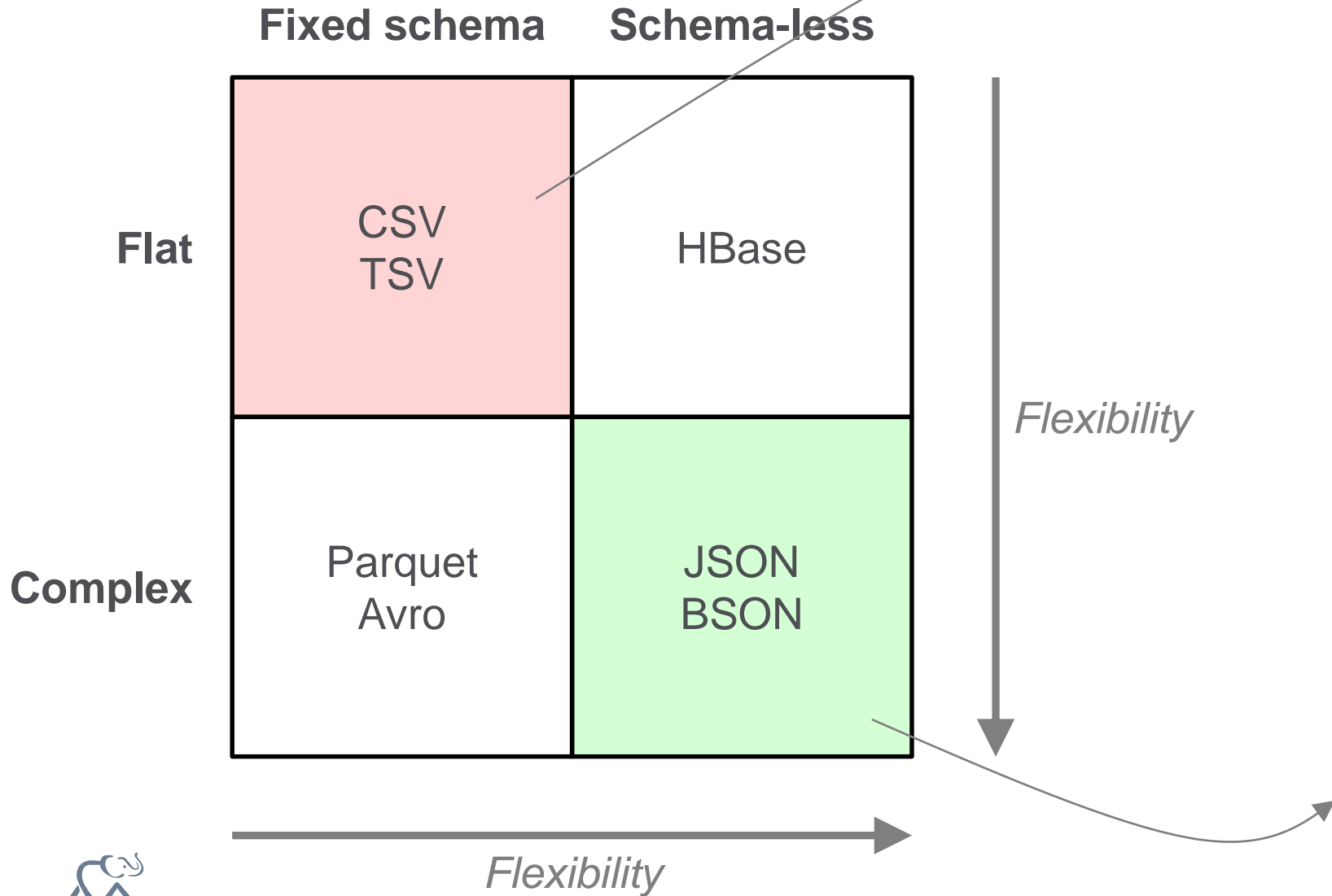
# APACHE DRILL



- Schema-free scale-out query engine for Hadoop and NoSQL
- Low latency
- Extreme ease of use
- Industry-standard APIs: ANSI SQL, ODBC/JDBC, RESTful APIs



# Drill's Data Model is Flexible



RDBMS/SQL-on-Hadoop table

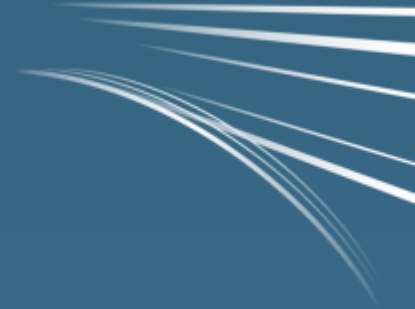
| Name     | Gender | Age |
|----------|--------|-----|
| Michael  | M      | 6   |
| Jennifer | F      | 3   |
|          |        |     |

Apache Drill table

```
{  
  name: {  
    first: Michael,  
    last: Smith  
  },  
  hobbies: [ski, soccer],  
  district: Los Altos  
}  
{  
  name: {  
    first: Jennifer,  
    last: Gates  
  },  
  hobbies: [sing],  
  preschool: CCLC  
}
```







# Architecture



# High Level Architecture

- Cluster of commodity servers
  - Daemon (`drillbit`) on each node
- ZooKeeper maintains ephemeral cluster membership information
  - `drillbit` uses ZooKeeper to find other `drillbits` in the cluster
  - Client uses ZooKeeper to find `drillbits`
- Built-in, optimistic query execution engine. Does not require a particular storage or execution system (MapReduce, Spark, Tez)
  - Better performance and manageability
- Data processing unit is *columnar record batches*
  - Enables schema flexibility with negligible performance impact



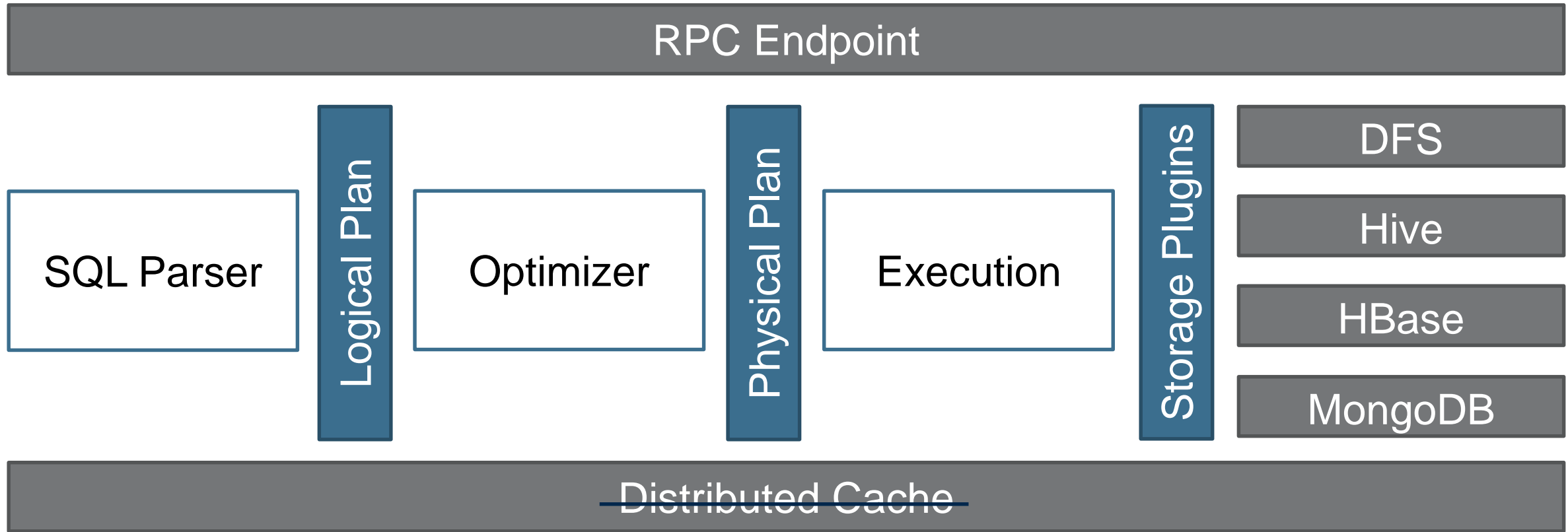
# Drill Maximizes Data Locality



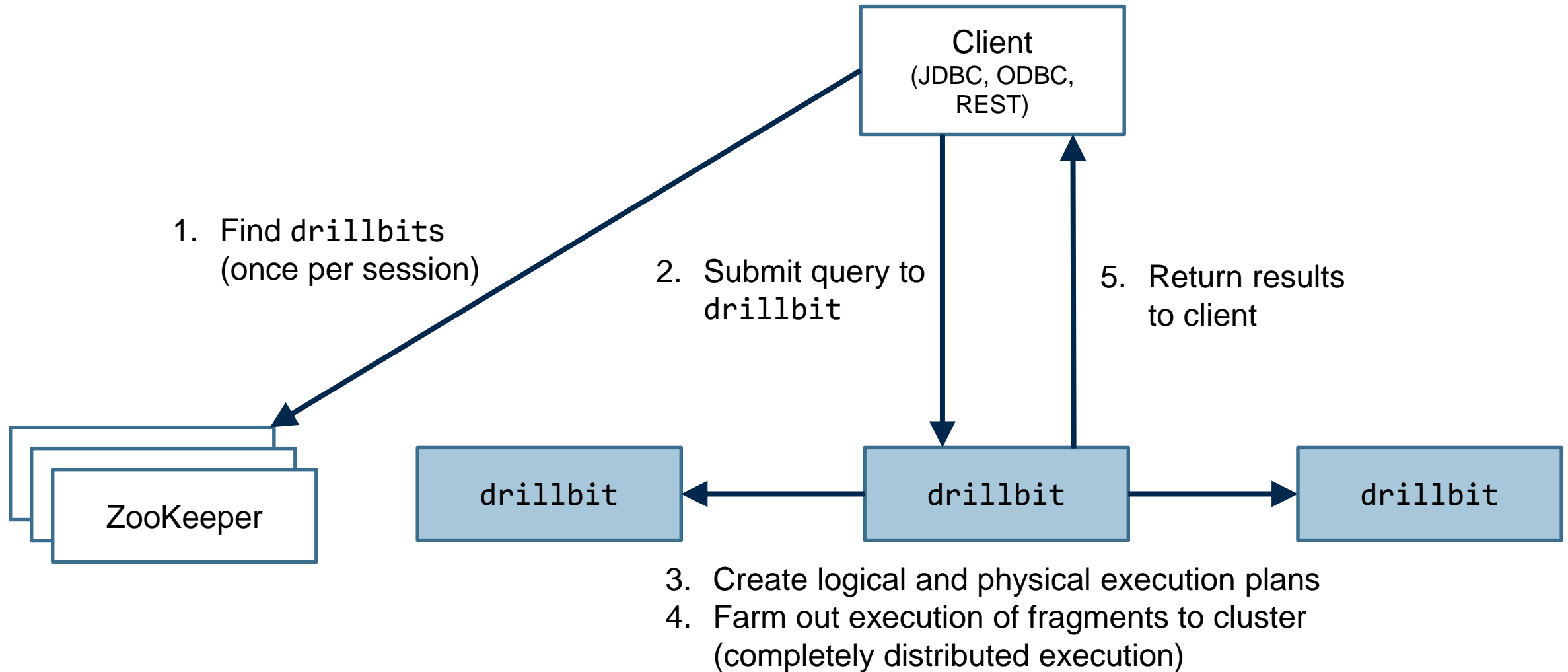
| Data Source      | Best Practice                                                                  |
|------------------|--------------------------------------------------------------------------------|
| HDFS or MapR-FS  | drillbit on each DataNode                                                      |
| HBase or MapR-DB | drillbit on each RegionServer                                                  |
| MongoDB          | drillbit on each mongod node (when using replicas, run it on the replica node) |



# Core Modules within drillbit



# SELECT \* Query Execution



\* CTAS (CREATE TABLE AS SELECT) queries include steps 1-4

# Example: Analyzing Real-World Data



# Demo Plan

1. Run Drill
2. Configure DFS and HBase storage plugins
3. Explore the data
  - Basics
  - Complex data
  - Views



# Run Drill





# Run Drill in Embedded Mode (sqlline)

```
$ tar xf apache-drill-0.6.0.tar.gz
$ cd apache-drill-0.6.0
$ bin/sqlline -u jdbc:drill:zk=local
> SELECT name, user_id, yelping_since, votes, review_count
   FROM dfs.`/E:/drill/data/yelp/user.json`
   LIMIT 1;
```

You can now access the Web UI:  
<http://localhost:8047>

```
+-----+-----+-----+-----+-----+
|  name  | user_id | yelping_since | votes | review_count |
+-----+-----+-----+-----+-----+
| Lee    | qtrmBGNqCvupHMHL_bKFgQ | 2012-02      | {"funny":1,"useful":5,"cool":0} | 6 |
+-----+-----+-----+-----+-----+
1 row selected (0.417 seconds)
```

- drillbit (Drill daemon) starts automatically in embedded mode
- No ZooKeeper in embedded mode (hence zk=local)
- Can't use BI clients (JDBC/ODBC) in embedded mode



# Or Run Drill in Distributed Mode...

- Make sure ZooKeeper (zkServer) is running:

```
$ zkServer start
```

- Define the Drill cluster name and ZooKeeper nodes in `conf/drill-override.conf`
- Start drillbit:

```
$ bin/drillbit.sh start
```

- Access the Web UI: <http://localhost:8047>
- Connect a client to the cluster (e.g., `sqlline`):

```
$ bin/sqlline -u jdbc:drill:zk=localhost:2181
```

- Clients (like `sqlline`) connect to ZooKeeper to discover the cluster nodes
- If you have multiple Drill clusters registered in one ZooKeeper ensemble, specify the desired cluster in the JDBC connection string:  
`jdbc:drill:zk=localhost:2181/drill/<clustername>`
- Not sure if ZooKeeper is running? Run `telnet localhost 2181` and make sure it connects



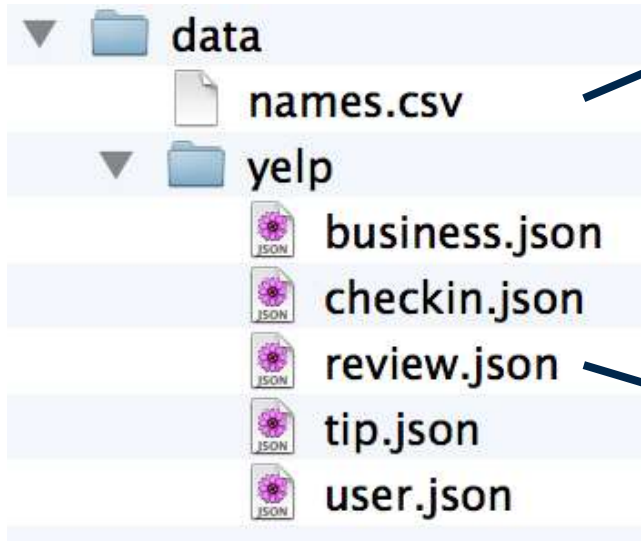
# Configure Storage Plugins



# Explore the Data: Basics



# Inventory: DFS Files



|   | A         | B            | C          | D            | E           |
|---|-----------|--------------|------------|--------------|-------------|
| 1 | Name      | years.appear | count.male | count.female | prob.gender |
| 2 | Aaban     | 5            | 42         | 0            | Male        |
| 3 | Aabha     | 2            | 0          | 12           | Female      |
| 4 | Aabid     | 1            | 5          | 0            | Male        |
| 5 | Aabriella | 1            | 0          | 5            | Female      |
| 6 | Aadam     | 21           | 161        | 0            | Male        |
| 7 | Aadan     | 7            | 91         | 0            | Male        |
| 8 | Aadarsh   | 12           | 128        | 0            | Male        |
| 9 | Aaden     | 12           | 3474       | 5            | Male        |

```
{  
  "votes": {"funny": 0, "useful": 2, "cool": 1},  
  "user_id": "Xqd0DzHaiyRqVH3WRG7hzg",  
  "review_id": "15SdjuK7DmYqUAj6rjGowg",  
  "stars": 5,  
  "date": "2007-05-17",  
  "text": "dr. goldberg offers everything ...",  
  "type": "review",  
  "business_id": "vcNAWiLM4dR7D2nwwJ7nCA"  
}
```



# Using Storage Plugins and Workspaces

Storage plugin

Workspace

Path relative to workspace

```
➤ SELECT * FROM dfs.root.`/E:/drill/data/yelp/review.json`;
➤ SELECT * FROM dfs.yelp.`review.json` LIMIT 1;
➤ USE dfs.yelp;
➤ SELECT * FROM `review.json` LIMIT 1;
➤ SELECT * FROM hbase.users LIMIT 1;
```

| Storage Plugin | Workspace | Table                      |
|----------------|-----------|----------------------------|
| dfs            | Path      | Path relative to workspace |
| mongo          | Database  | Collection                 |
| hive           | Database  | Table                      |
| hbase          | Namespace | Table                      |



# Most Common User Names (MongoDB)

```
> SELECT name, COUNT(*) AS users
   FROM dfs.yelp.`user.json`
   GROUP BY name
   ORDER BY users DESC LIMIT 10;
```

| name     | users |
|----------|-------|
| David    | 2453  |
| John     | 2378  |
| Michael  | 2322  |
| .....    |       |
| .....    |       |
| Jessica  | 1463  |
| Jason    | 1457  |
| Michelle | 1439  |
| Brian    | 1436  |

```
10 rows selected (9.348 seconds)
```



# Cities with the Most Reviews

```
> SELECT state, city, COUNT(*) AS reviews
   FROM dfs.yelp.`business.json`
   GROUP BY state, city
   ORDER BY reviews DESC LIMIT 10;
```

| state | city       | reviews |
|-------|------------|---------|
| NV    | Las Vegas  | 12021   |
| AZ    | Phoenix    | 7499    |
| AZ    | Scottsdale | 3605    |
| EDH   | Edinburgh  | 2804    |
| AZ    | Mesa       | 2041    |
| AZ    | Tempe      | 2025    |
| NV    | Henderson  | 1914    |
| AZ    | Chandler   | 1637    |
| WI    | Madison    | 1630    |
| AZ    | Glendale   | 1196    |





# Explore the Data: Complex Data



# business.json (1)

```
{  
  "business_id": "4bEj0yTaDG24SY5TxsaUNQ",  
  "full_address": "3655 Las Vegas Blvd S\nThe Strip\nLas Vegas, NV 89109",  
  "hours": {  
    "Monday": {"close": "23:00", "open": "07:00"},  
    "Tuesday": {"close": "23:00", "open": "07:00"},  
    "Friday": {"close": "00:00", "open": "07:00"},  
    "Wednesday": {"close": "23:00", "open": "07:00"},  
    "Thursday": {"close": "23:00", "open": "07:00"},  
    "Sunday": {"close": "23:00", "open": "07:00"},  
    "Saturday": {"close": "00:00", "open": "07:00"}  
  },  
  "open": true,  
  "categories": ["Breakfast & Brunch", "Steakhouses", "French", "Restaurants"],  
  "city": "Las Vegas",  
  "review_count": 4084,  
  "name": "Mon Ami Gabi",  
  "neighborhoods": ["The Strip"],  
  "longitude": -115.172588519464,
```



# business.json (2)

```
"state": "NV",
"stars": 4.0,
"attributes": {
  "Alcohol": "full_bar",
  "Noise Level": "average",
  "Has TV": false,
  "Attire": "casual",
  "Ambience": {
    "romantic": true,
    "intimate": false,
    "touristy": false,
    "hipster": false,
    "classy": true,
    "trendy": false,
    "casual": false
  },
  "Good For": {"dessert": false, "latenight": false, "lunch": false,
    "dinner": true, "breakfast": false, "brunch": false},
}
```



# Which Places Are Open Right Now (22:00)?

```
> SELECT name, b.hours
   FROM dfs.yelp.`business.json` b
   WHERE b.hours.Saturday.`open` < '22:00' AND
          b.hours.Saturday.`close` > '22:00'
   LIMIT 1;
```

```
+-----+-----+
| name | hours |
+-----+-----+
| Chang Jiang Chinese Kitchen |
{"Tuesday":{"close":"22:00","open":"11:00"},"Friday":{"close":"22:30","open":"11:00"},"Monday":{"
close":"22:00","open":"11:00"},"Wednesday":{"close":"22:00","open":"11:00"},"Thursday":{"close":"
22:00","open":"11:00"},"Sunday":{"close":"21:00","open":"16:00"},"Saturday":{"close":"22:30","ope
n":"11:00"}} |
+-----+-----+
1 row selected (0.013 seconds)
```



# It's 10pm in Vegas and I Want Good Hummus!

```
> SELECT name, stars, b.hours.Wednesday, categories
   FROM dfs.yelp.`business.json` b
   WHERE b.hours.Wednesday.`open` < '22:00' AND
         b.hours.Wednesday.`close` > '22:00' AND
         REPEATED CONTAINS(categories, 'Mediterranean') AND
         city = 'Las Vegas'
   ORDER BY stars DESC
   LIMIT 1;
```

```
+-----+-----+-----+-----+
| name | stars | EXPR$2 | categories |
+-----+-----+-----+-----+
| Marrakech Moroccan Restaurant | 4.0 | {"close":"23:00","open":"17:30"} |
["Mediterranean","Middle Eastern","Moroccan","Restaurants"] |
+-----+-----+-----+-----+
1 row selected (2.185 seconds)
```



# Flatten Repeated Values

```
> SELECT name, categories
   FROM dfs.yelp.`business.json` LIMIT 2;
```

```
+-----+-----+
| name   | categories |
+-----+-----+
| Eric Goldberg, MD | ["Doctors","Health & Medical"] |
| Pine Cone Restaurant | ["Restaurants"] |
| Deforest Family Restaurant | ["American (Traditional)","Restaurants"] |
+-----+-----+
```

```
> SELECT name, FLATTEN(categories) AS categories
   FROM dfs.yelp.`business.json` LIMIT 3;
```

```
+-----+-----+
| name   | categories |
+-----+-----+
| Eric Goldberg, MD | Doctors      |
| Eric Goldberg, MD | Health & Medical |
| Pine Cone Restaurant | Restaurants |
+-----+-----+
```



# Most and Least Common Business Categories

```
> SELECT category, COUNT(*) AS businesses
   FROM (SELECT name, FLATTEN(categories) AS category
         FROM dfs.yelp.`business.json`)
   GROUP BY category ORDER BY businesses DESC;
```

```
+-----+-----+
| category | businesses |
+-----+-----+
| Restaurants | 14303 |
| ..... |
| Firewood | 1 |
+-----+-----+
```

715 rows selected (3.439 seconds)

```
> SELECT name, categories FROM dfs.yelp.`business.json`
   WHERE true AND REPEATED_CONTAINS(categories, 'Australian');
```

```
+-----+-----+
| name | categories |
+-----+-----+
| The Australian AZ | ["Bars", "Burgers", "Nightlife", "Australian", "Sports Bars", "Restaurants"] |
+-----+-----+
```

# Explore the Data: Views





# Create a view combining business and reviews datasets.

```
> CREATE OR REPLACE VIEW dfs.tmp.BusinessReviews AS
  SELECT b.name, b.stars, r.votes.funny,
         r.votes.useful, r.votes.cool, r.`date`
  FROM dfs.yelp.`business.json` b, dfs.yelp.`review.json` r
  WHERE r.business_id = b.business_id;
```

```
+-----+-----+
|      ok      | summary |
+-----+-----+
| true         | View 'BusinessReviews' created successfully in 'dfs.tmp' schema |
+-----+-----+
```

```
> SELECT COUNT(*) AS Total FROM dfs.tmp.BusinessReviews;
```

```
+-----+
|      Total      |
+-----+
|      1125458    |
+-----+
```



# Materialized Views AKA Tables

```
> ALTER SESSION SET `store.format` = 'parquet';  
  
> CREATE TABLE dfs.tmp.BusinessReviewsTbl AS  
  SELECT b.name, b.stars, r.votes.funny funny,  
         r.votes.useful useful, r.votes.cool cool, r.`date`  
  FROM dfs.yelp.`business.json` b, dfs.yelp.`review.json` r  
  WHERE r.business_id = b.business_id;
```

| Fragment | Number of records written |
|----------|---------------------------|
| 1_0      | 176448                    |
| 1_1      | 192439                    |
| 1_2      | 198625                    |
| 1_3      | 200863                    |
| 1_4      | 181420                    |
| 1_5      | 175663                    |



# Participate

- Learn: [incubator.apache.org/drill/](http://incubator.apache.org/drill/)
- Download: [incubator.apache.org/drill/download/](http://incubator.apache.org/drill/download/)
- Ask questions: [drill-user@incubator.apache.org](mailto:drill-user@incubator.apache.org)
- Engage on Twitter: [@ApacheDrill](https://twitter.com/ApacheDrill)
- Contact me: [adi@apache.org](mailto:adi@apache.org)



# Thank You

Aditya Kishore, Software Engineer

@mapr



maprtech

mapr-technologies



MapRTechnologies

aditya@mapr.com



maprtech

