The Evolution of Open Source Databases

Past, Present and Future

Ivan Zoratti

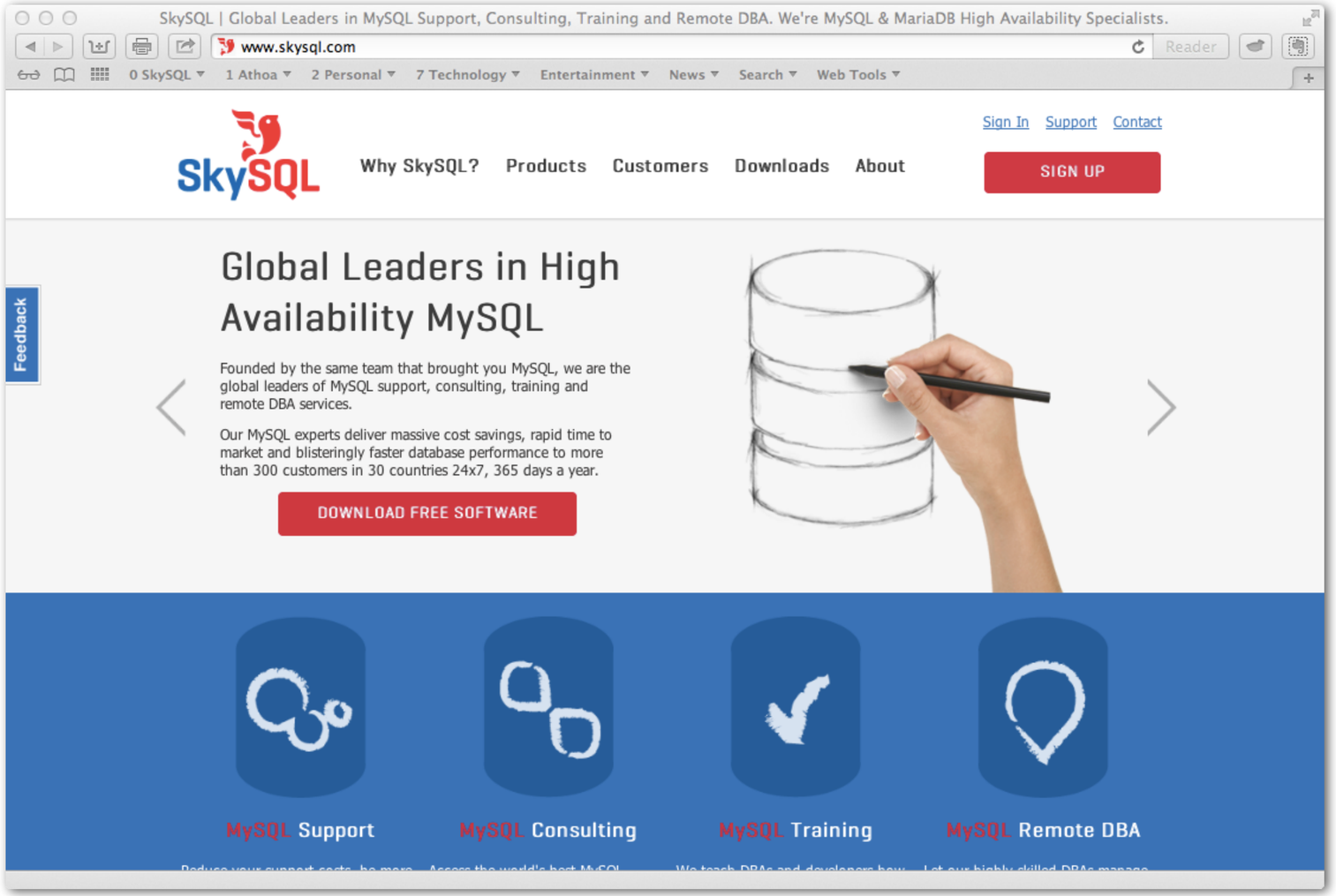
V1310.01

Who is Ivan

?

SkySQL

- Leading provider of open source databases, services and solutions
- Home for the founders and the original developers of the core of MySQL
- The creators of MariaDB, the drop-in, innovative replacement of MySQL



The Past

Databases and DBMSs

- Database

  "An organized collection of data"

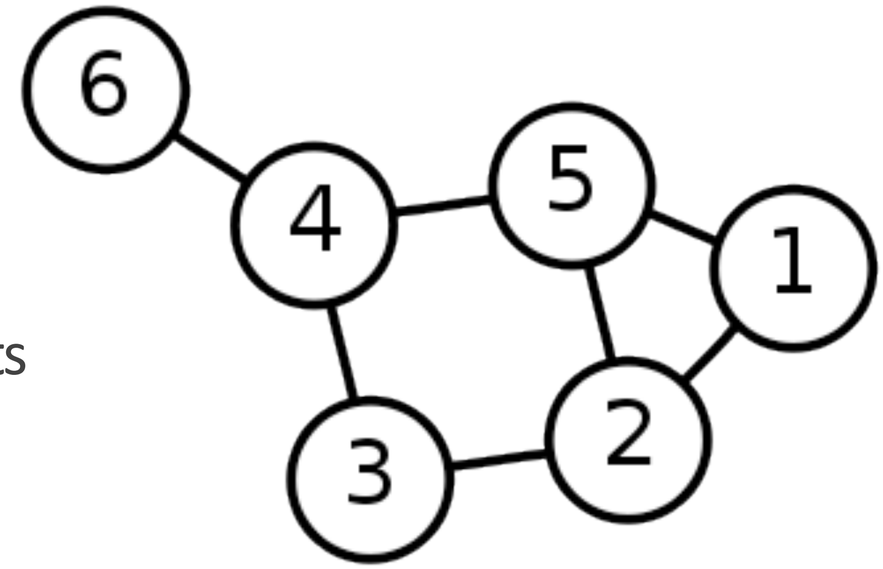
  "Databases are created to operate large quantities of information by inputting, storing, retrieving, and managing that information"

- DBMS - DataBase Management System

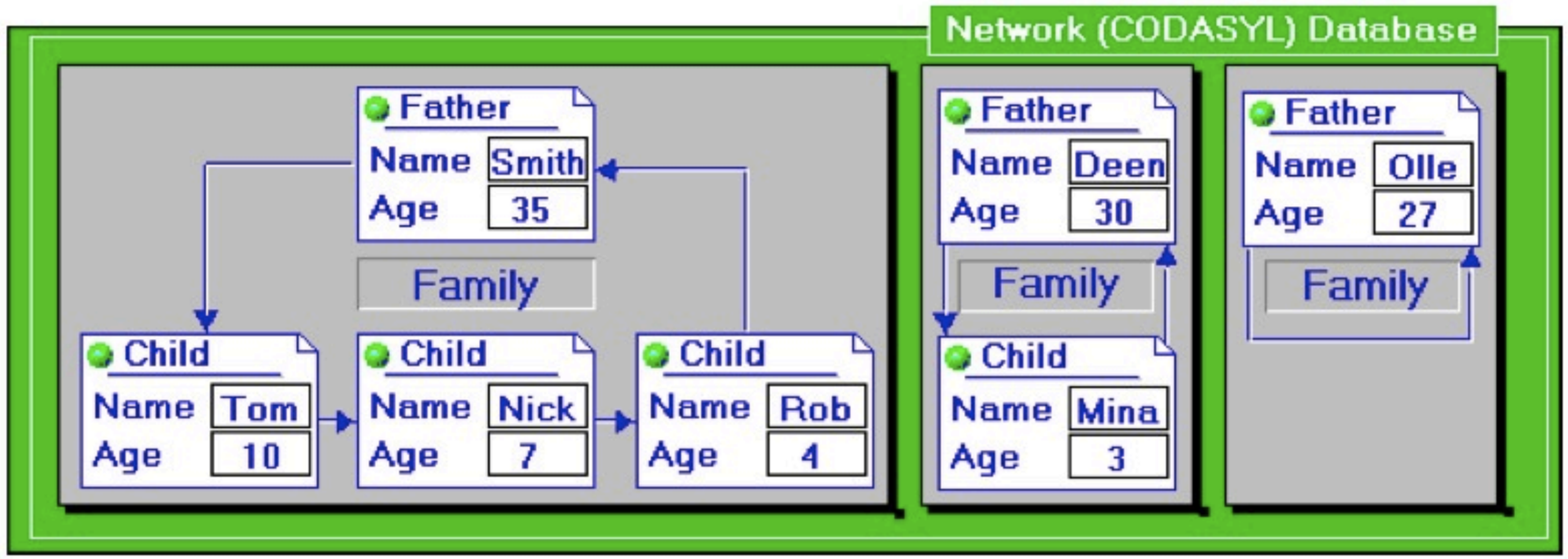
  "A suite of computer software providing the interface between users and a database or databases"

Wikipedia - http://en.wikipedia.org/wiki/Database

The early days

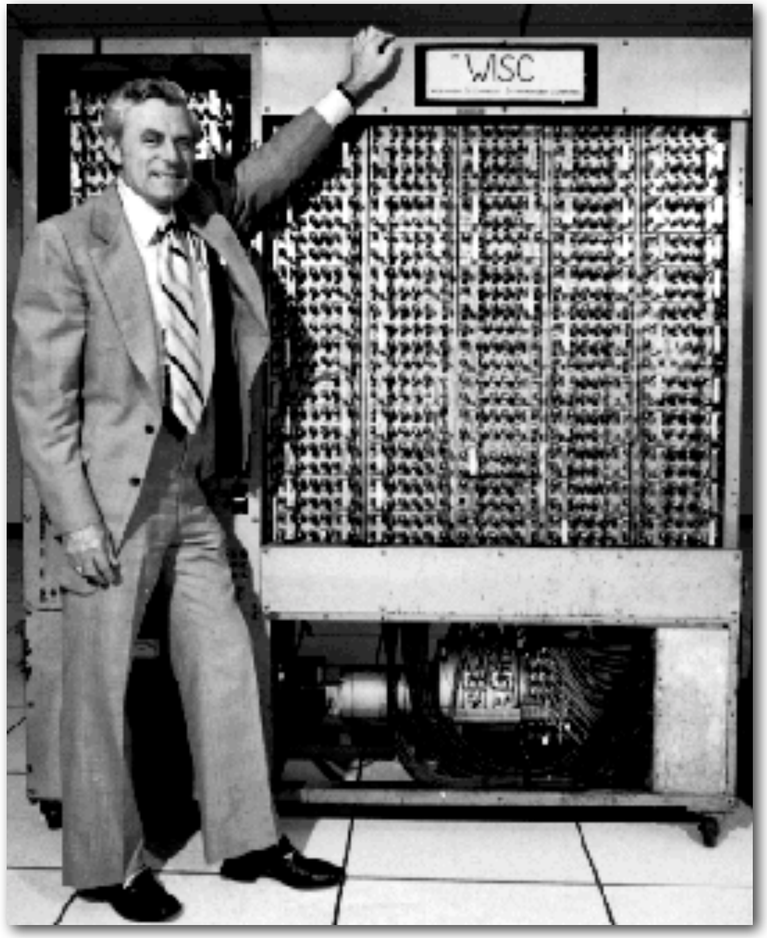


- Navigational Databases
  - Objects are found by following references from other objects
- Network and Hierarchical Databases
  - Data is organized into a network or a tree-like structure

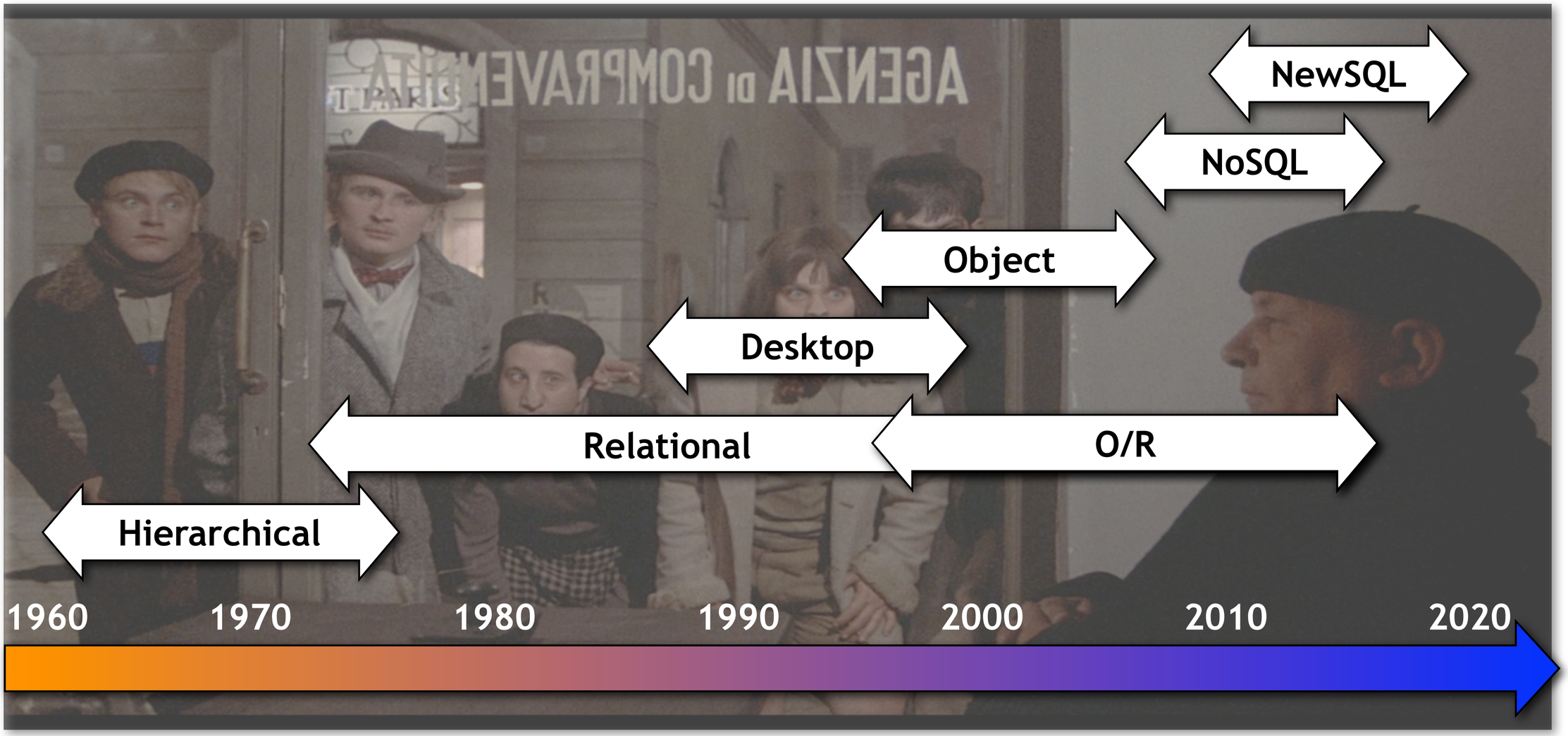


The beginning of a relationship...

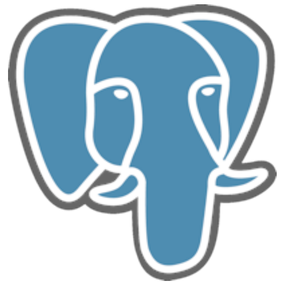
- 1970 - Edgar Codd starts working on the relational model
- 1973 - Ingres project at Berkeley University
- 1974 - System R (DB/2 precursor)
- 1977 - Larry Ellison founds Software Development Labs
- 1979 - Oracle v2 becomes relational and implements SQL
- 1980 - Informix is established as Relational Database Systems - the first RDBMS is based on ISAM
- 1984 - Mark Hoffman and Bob Epstein found Sybase to exploit the capabilities of client/server architectures
- 1985 - Michael Stonebraker starts the POSTGRES project
- 1992 - Micosoft starts JET (Joint Engine Technology)
- 1995 - MySQL v1 from MySQL AB
- 2000 - Richard Hipp designs SQLite



I remember...


NewSQL
NoSQL
Object
Desktop
Relational
O/R
Hierarchical
1960
1970
1980
1990
2000
2010
2020

Open source RDBMSs



- PostgreSQL License
  - Similar to the MIT license
- Open source versions:
  - Greenplum
- Commercial versions:
  - EnterpriseDB Postgres Plus
  - Netezza
  - Red Hat Database

- GPL v2 License
- Open source versions:
  - MariaDB Server
  - Percona Server
- Commercial versions:
  - Oracle MySQL Enterprise
  - InfiniDB
  - Infobright
  - ScaleDB

- Public domain
- Operating Systems
  - OS X
  - IOS
  - Windows Phone
- Web Browsers
  - Firefox
  - Chrome
- Others
  - Thunderbird

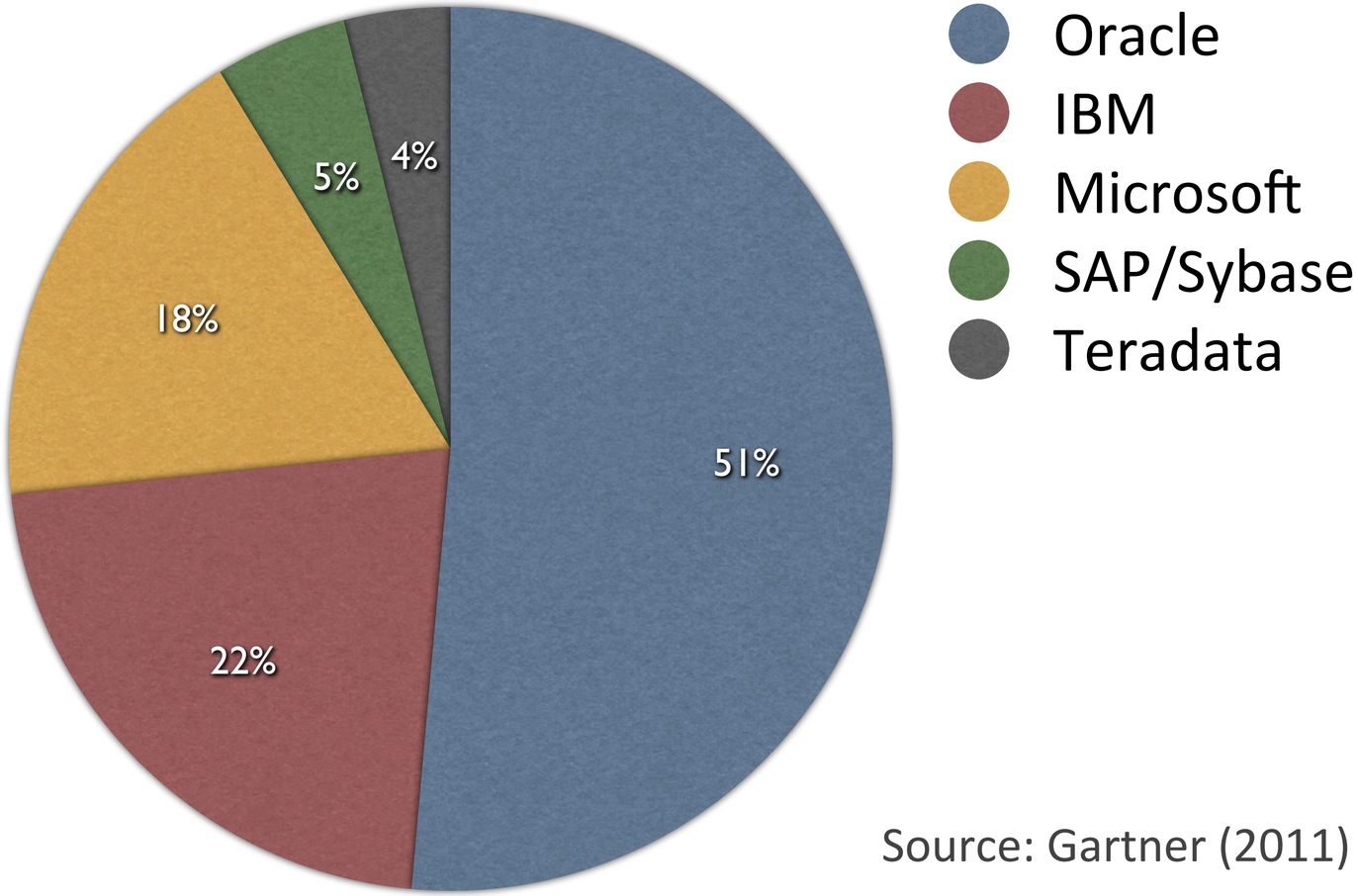
The Present

Online vs. Enterprise

The Top 10 web sites

1. Google
2. Facebook
3. YouTube
4. Yahoo!
5. Baidu.com
6. Wikipedia
7. QQ.com
8. Linkedin
9. Windows Live
10. Twitter

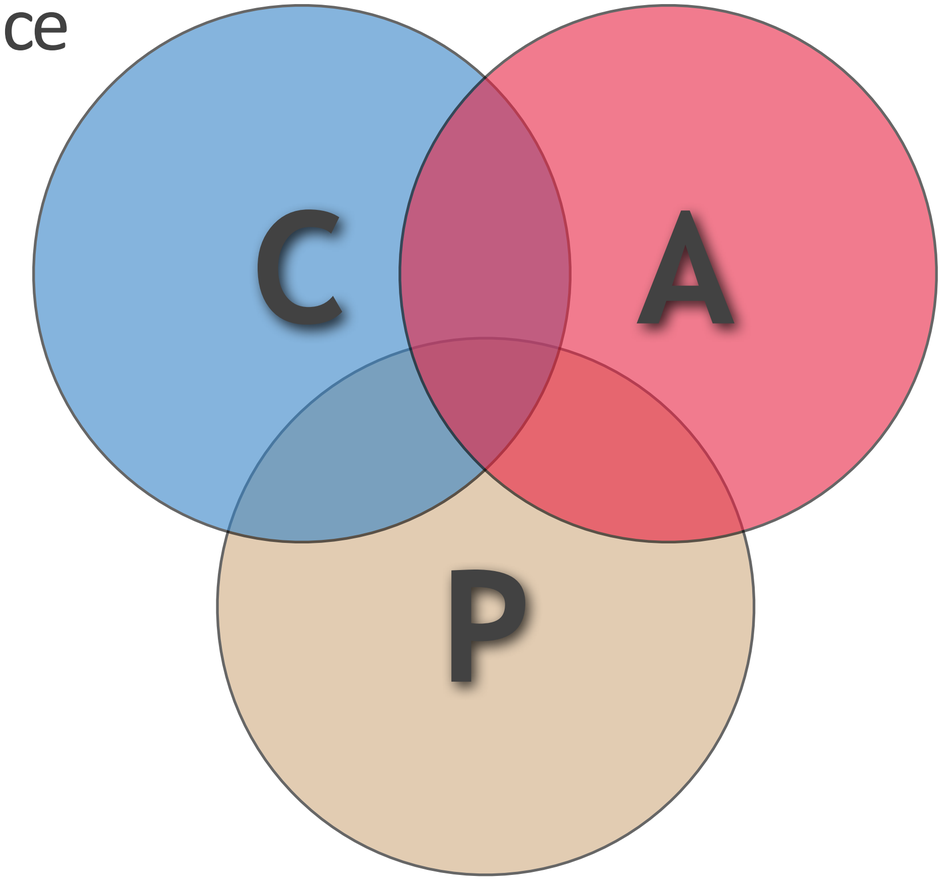
www.alexa.com



Source: Gartner (2011)

Understanding the CAP theorem

- Consistency / Availability / Partition Tolerance
- CA
  - Synchronous Replication
  - Two Phase Commit
  - MySQL, ACID/RDBMSs
- CP
  - MongoDB, HBase, Redis, MemcacheD
- AP
  - Cassandra, Riak, CouchDB



The advent of NoSQL

- First seen in 1998, but really active since 2009
- SQL is complex, boring and clunky
- Not everybody needs ACID compliance
- Relational databases are:
  - Slow
  - Not scalable
  - Not optimized for rich data
  - Difficult to change and maintain

NoSQL vs SQL

NoSQL

- Schema-less (or dynamic schema)
- Dynamic horizontal scaling
- Good to store and retrieve a great quantity of data
- Great Flexibility
- Full ACID not required - “BASE is better”
  Basically Available, Soft state, Eventually consistent
- Objects: Collections, Documents, Fields
- NoSQL DBs:
  Key/Value
  BigTable
  Document
  Graph

SQL

- Rigid Schema design
- Static or no horizontal scaling
- Good to store and retrieve data that has relationship between the elements
- Pretty inflexible
- ACID as a given
  - Atomic, Consistent, Isolated, Durable
- Objects: Tables, Rows, Columns
- SQL DBs:
  - Row-based
  - Columnar
  - Object Relational

NoSQL players

Key/Value Stores

redis

Cassandra

riak

memcached

Document Store

mongoDB

Apache CouchDB

Column Store

APACHE HBASE

Graph DB

Neo4j

SkySQL

mySQL

```
SELECT
    Dim1, Dim2,
    SUM(Measure1) AS MSum,
    COUNT(*) AS RecordCount,
    AVG(Measure2) AS MAvg,
    MIN(Measure1) AS MMin
    MAX(CASE
      WHEN Measure2 < 100
      THEN Measure2
    END) AS MMax
FROM DenormAggTable
WHERE (Filter1 IN ('A','B'))
    AND (Filter2 = 'C')
    AND (Filter3 > 123)
GROUP BY Dim1, Dim2
HAVING (MMin > 0)
ORDER BY RecordCount DESC
LIMIT 4, 8
```

MongoDB

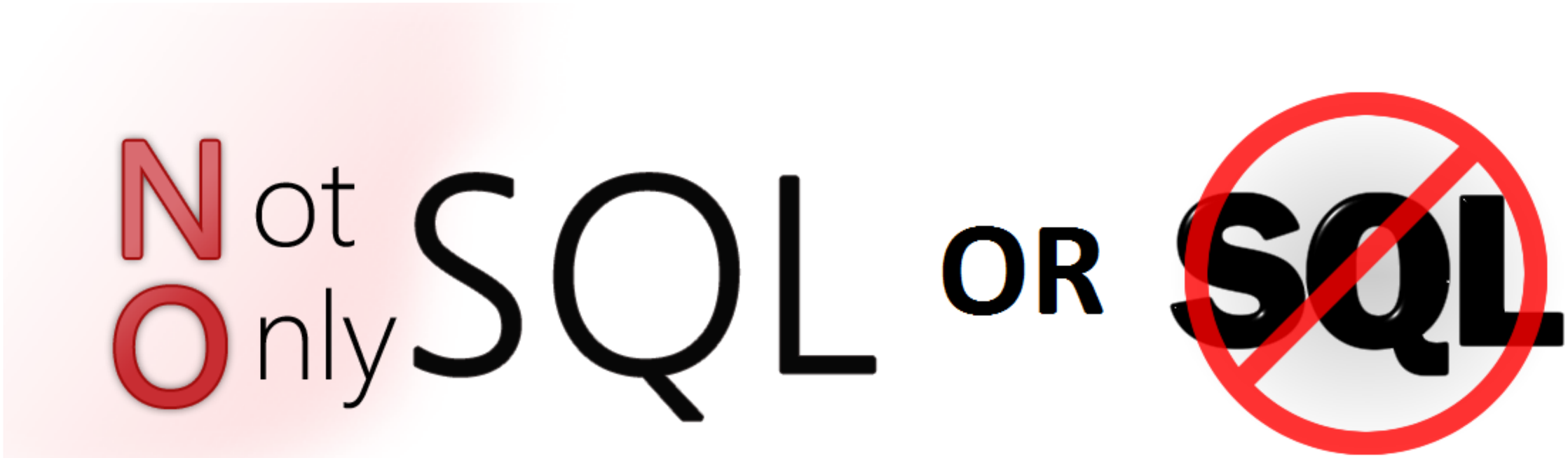
```
db.runCommand({
mapreduce: "DenormAggCollection",
query: {
    filter1: { '$in': [ 'A', 'B' ] },
    filter2: 'C',
    filter3: { '$gt': 123 }
  },
map: function() { emit(
    { d1: this.Dim1, d2: this.Dim2 },
    { msum: this.measure1, recs: 1, mmin: this.measure1,
      mmax: this.measure2 < 100 ? this.measure2 : 0 }
  );},
reduce: function(key, vals) {
    var ret = { msum: 0, recs: 0, mmin: 0, mmax: 0 };
    for(var i = 0; i < vals.length; i++) {
      ret.msum += vals[i].msum;
      ret.recs += vals[i].recs;
      if(vals[i].mmin < ret.mmin) ret.mmin = vals[i].mmin;
      if((vals[i].mmax < 100) && (vals[i].mmax > ret.mmax))
        ret.mmax = vals[i].mmax;
    }
    return ret;
  },
finalize: function(key, val) {
    val.mavg = val.msum / val.recs;
    return val;
  },
out: 'result1',
verbose: true
});
db.result1.
  find({ mmin: { '$gt': 0 } }).
  sort({ recs: -1 }).
  skip(8).
  limit(4);
```

1. Grouped dimension columns are pulled out as keys in the map function, reducing the size of the working set.
2. Measures must be manually aggregated.
3. Aggregates depending on record counts must wait until finalization.
4. Measures can use procedural logic.
5. Filters have an ORM/ActiveRecord-looking style.
6. Aggregate filtering must be applied to the result set, not in the map/reduce.
7. Ascending: 1; Descending: -1

Revision 3, Created 2010-02-10
Rick Osborne, rickosborne.org

The Future

Here is what is going on in the NoSQL world

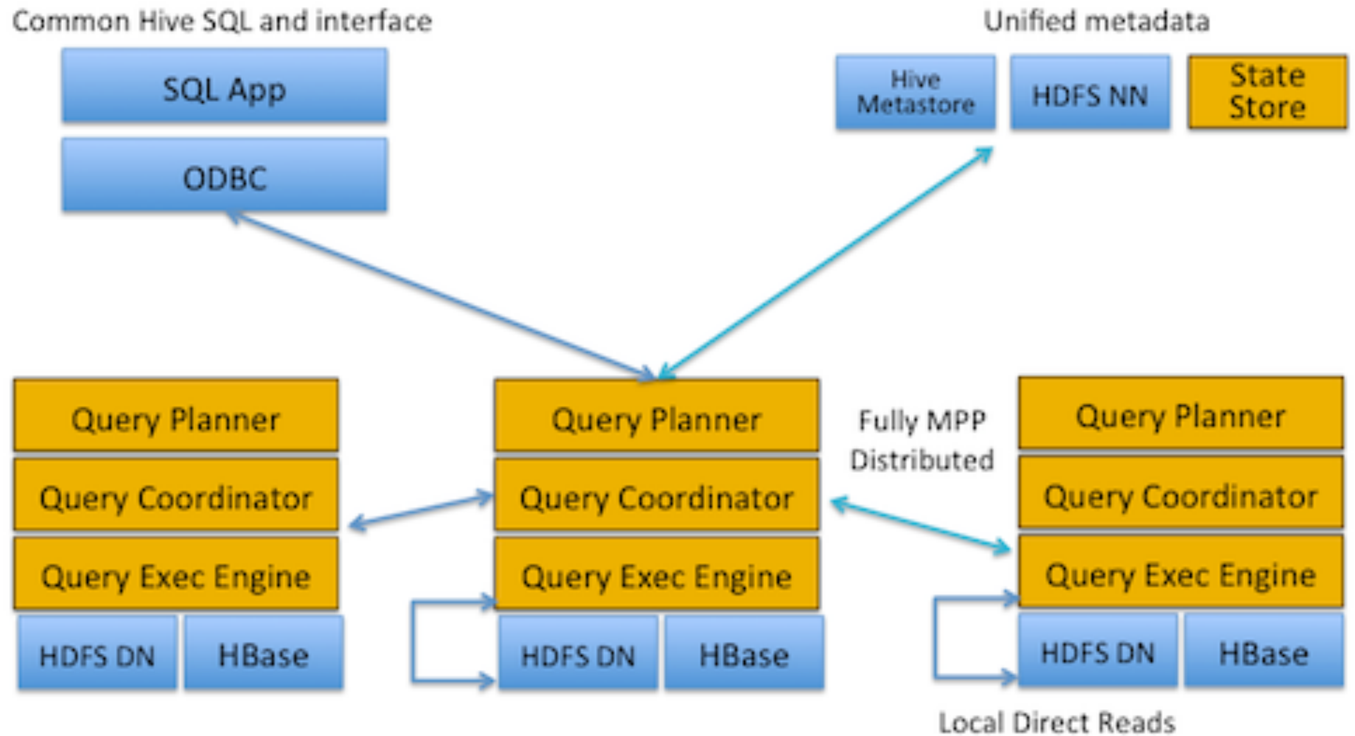


Here is what is going on in the NoSQL world

Cassandra

- New in CQL with Cassandra 2.0
  - Lightweight transaction for INSERT and UPDATE with optimistic locking
- Initial support for triggers
- Prepared statements and cursors
- User authentication

cloudera IMPALA



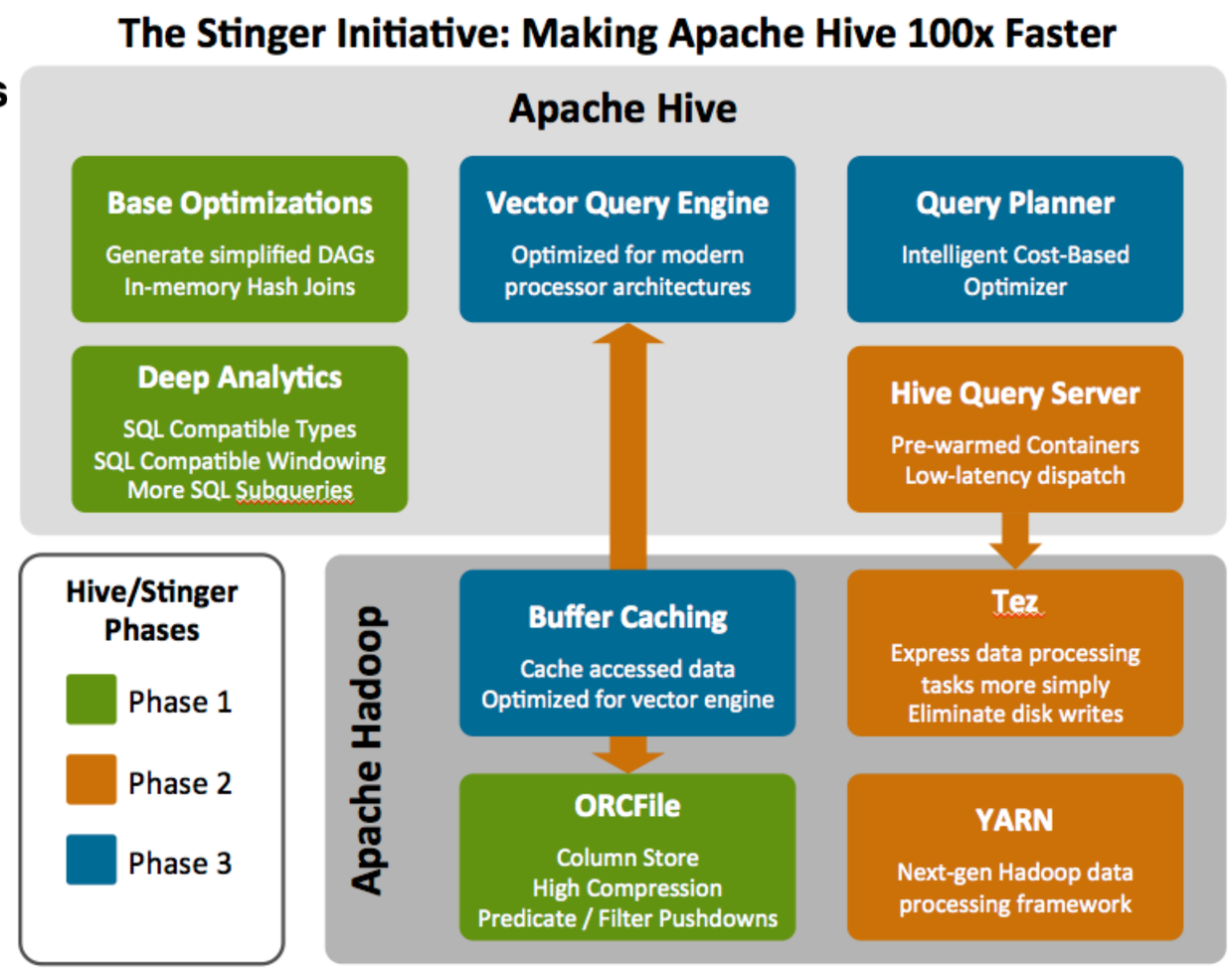
Here is what is going on in the NoSQL world

mongoDB

Tokutek

- Transactions with MVCC and ACID reliability with TokuMX
- Fractal indexing
- Replication
- Compression

Hortonworks



Welcome to the Cloud

- Some cloud-focused DB players
  - Herokypostgres - Open source - PostgreSQL
  - Clustrix - Closed source - MySQL client compatible
  - GenieDB - Closed source - MySQL client compatible
  - Amazon DynamoDB - Closed source - From the Dynamo project
  - Amazon RDS/MySQL - Open Source - Pure MySQL
- Cloud lock-in / Service lock-in / Vendor lock-in

...and welcome to NewSQL

- NoSQL performance for OLTP systems requiring ACID features
  - Shared nothing
  - Sharding
  - In-memory storage
  - Local and geographical replication
- NewSQL players
  - NuoDB
  - VoltDB
  - Clustrix
  - Spanner/Google F1
  - Translattice
  - MariaDB

...and welcome to NewSQL

MariaDB 10

- Multi-source Replication
- Handler Socket
- Dynamic Columns
- Text Search Engines
  - Sphinx
- Sharding Engines
  - Spider
  - ScaleDB
- Integration Engines
  - Cassandra
  - CONNECT
- Optimistic Locking / Sync Replication
  - Galera
- RESTFul API

Risks and old tricks

- Vendors who Open Source DB compatibility, but they are closed source
- Vendors who start open sources, then they move to a closed source license
- Only DBs available from non-commercial organisations seem to be safe

Databases backed by foundations

- Apache Software Foundation
  - Accumulo - Key/value written in Java, based on Google BigTable (on top of Hadoop)
  - Cassandra - Key/valye written in Java
  - Derby - RDBMS written in Java
  - HBase - Big Data columnar engine written in Java
- MariaDB Foundation
  - MariaDB - MySQL compatible, GPL v2, written in C, with multiple storage engine options
- SPI - Software in the Public Interest
  - Postgresql

Watch what is coming

- Flash Storage and SSD optimized databases
- NoSQL and SQL integration > NewSQL as a standard
  - Sharding
  - Eventual Consistency as an option
  - Complex queries + in-DB logic vs. performance
- Data has changed, databases must handle the new **and** the old data
- Interoperability among multiple databases
- More Big Data features
  - Columnar storage
  - Document handling and indexing
- Databases as part of the IaaS layer
  - Or what we used to say in the old days: The database is the operating system

Pictures courtesy of:

- www.interflora.co.uk
- www.bubblenews.com
- www.fullhdwpp.com
- www.hortonworks.com
- www.cloudera.com

Thank You!

www.skysql.com

ivan@skysql.com
izoratti.blogspot.com
www.slideshare.net/izoratti