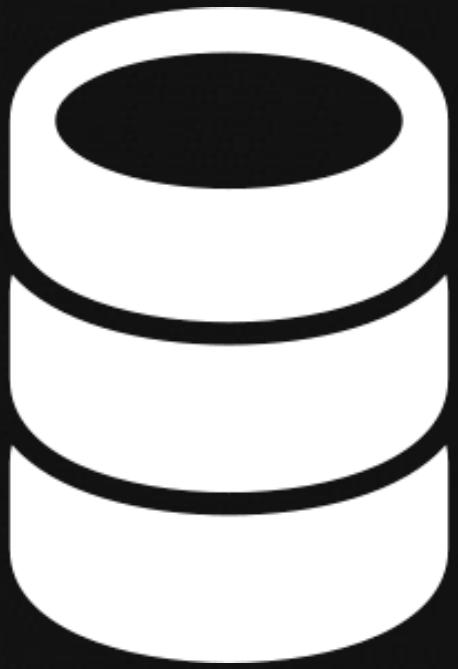


Crash Course on the History of Database Systems

Adapted from *"What Goes Around Comes Around,"*
by Hellerstein & Stonebraker

http://bit.ly/cmu-db-fall2013

@andy_pavlo

Administrivia

- Everyone should have gotten mailing list notification.
  - *Speaker sign up.*
- If you don't want to take this for credit, please drop soon.
  - *You can still hang out.*
  - *We won't judge.*

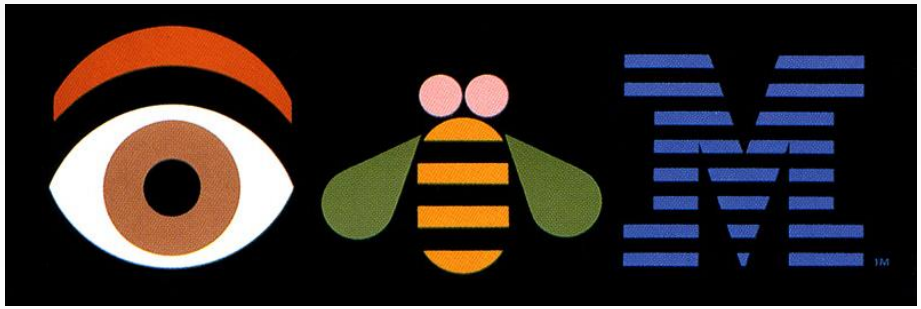
Why?

History Repeats Itself

- Old database issues are still relevant today.
- The "SQL vs. NoSQL" debate is reminiscent of "Relational vs. CODASYL" debate.
- Many of the ideas in today's database systems are not new.

1960s – IBM IMS

- First database system.
- Hierarchical data model.
- Programmer-defined physical storage format.
- Tuple-at-a-time queries.



Hierarchical Data Model

Duplicate Data

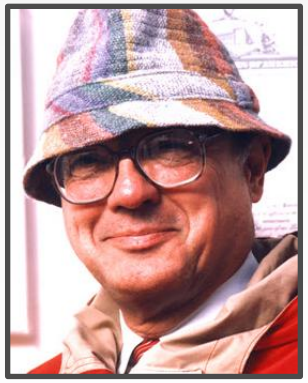
No Independence

PART
(pno,pname,psize,qty,price)

1001, 'Battery Pack', Large, 500, $100

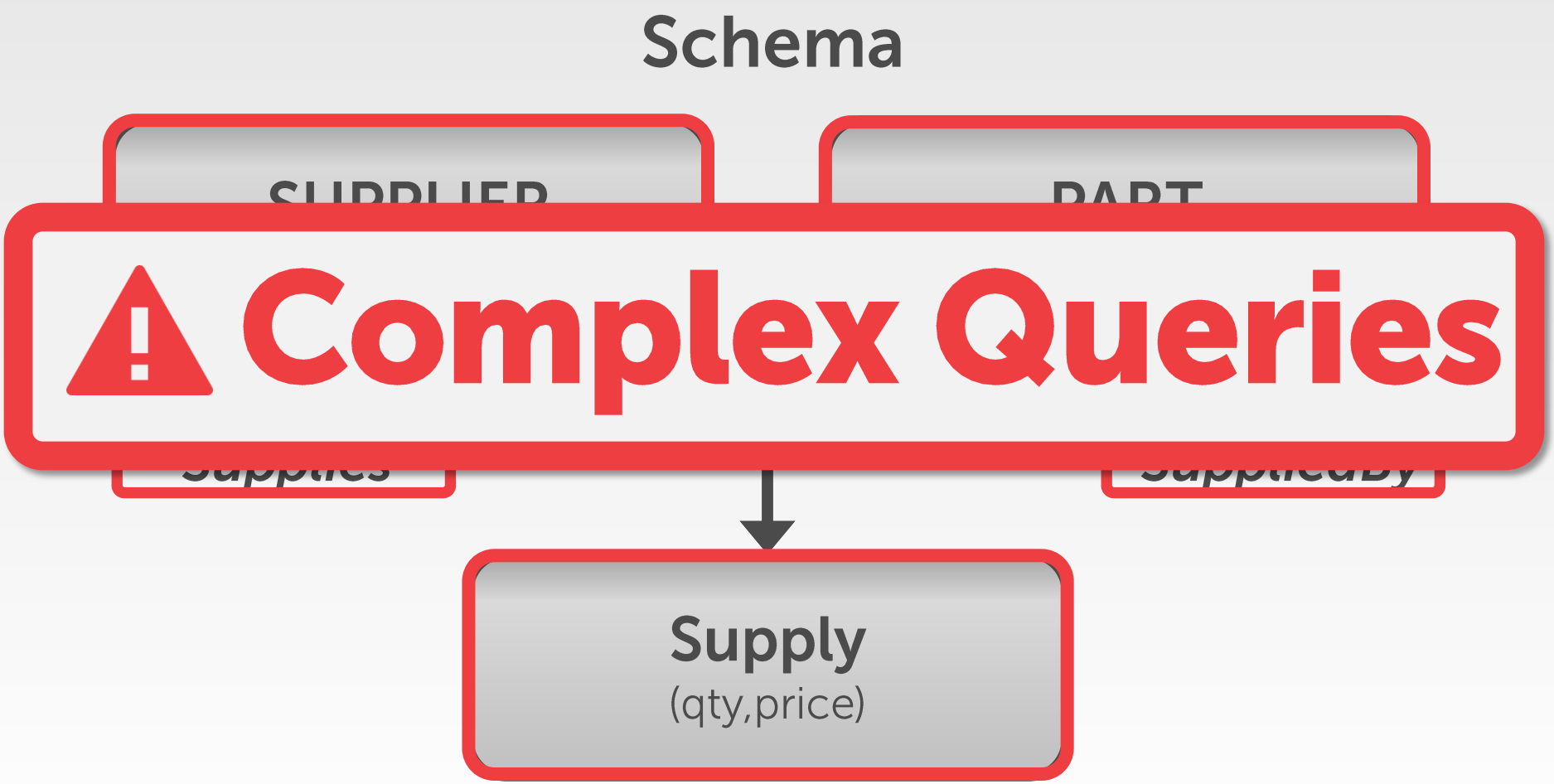
1970s – CODASYL

- COBOL people got together and proposed a standard.
- Network data model.
- Tuple-at-a-time queries.



Bachman

Network Data Model



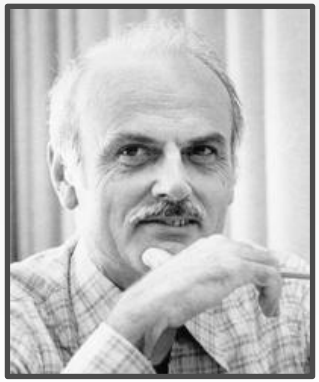
Stonebraker Lessons

- Physical and logical data independence are good.
- Tree-based data models are too restrictive.
- Record-at-a-time forces the programmer to do manual query optimization.

Source: Kristen LeFerve, November 2009, http://bit.ly/13M2Mfb

1970s – Relational Model

- Codd saw the maintenance overhead for IMS/Codasyl.
- Proposes database abstraction based on tables.



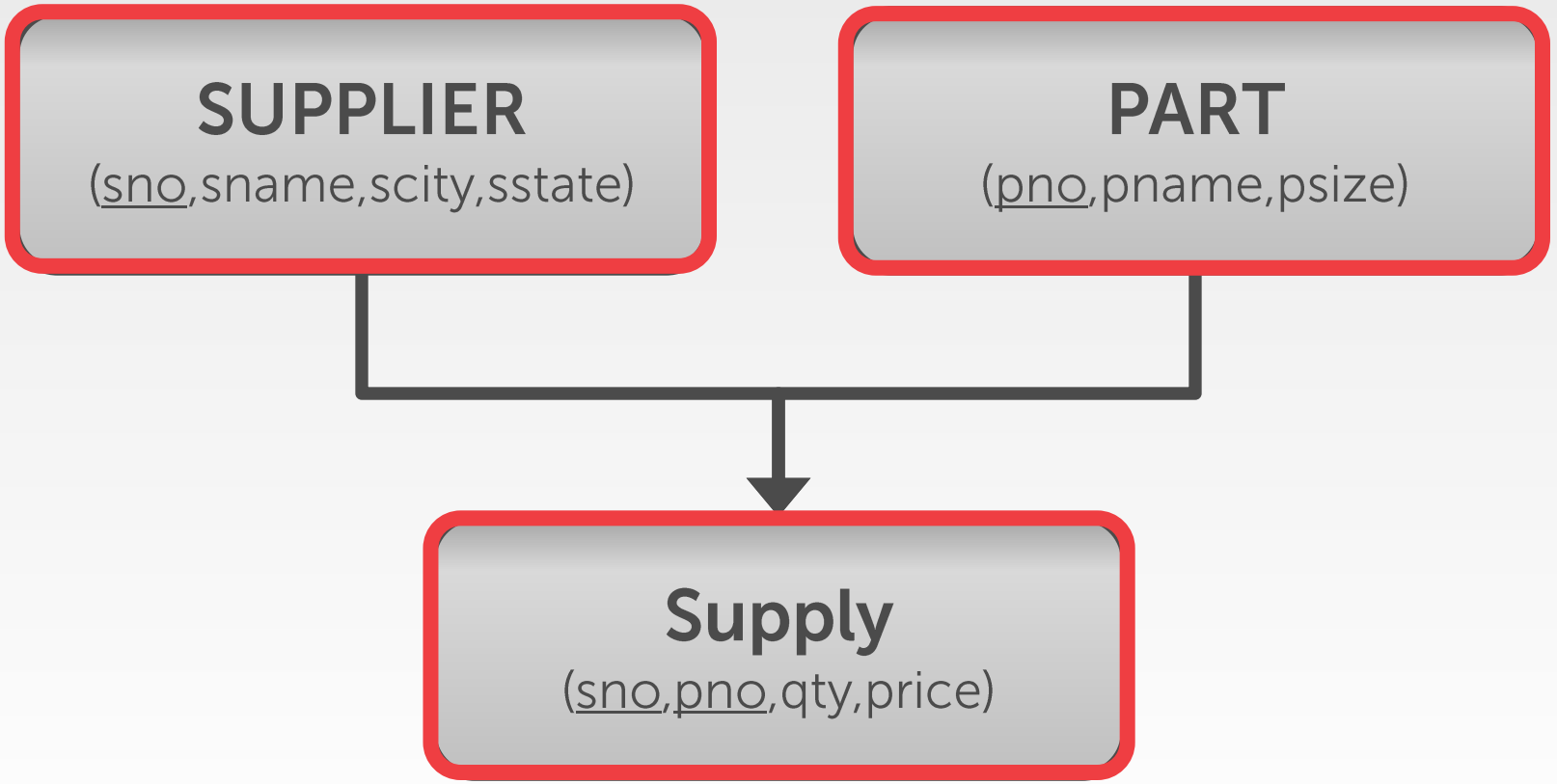
Codd

Relational Model

- Store database in simple data structures (i.e., tables).
- Access it through high-level language (i.e., SQL).
- Physical storage left up to implementation.

Relational Data Model

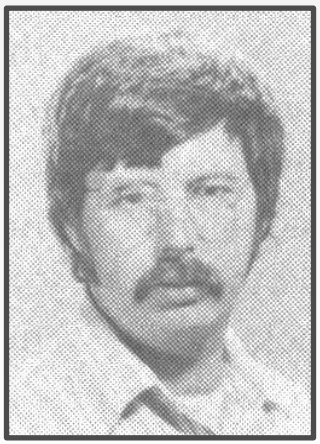
Schema



1970s – Relational Model

- System R – IBM Research
- INGRES – Berkeley
- Oracle – Larry Ellison

Gray



Stonebraker

Ellison

1980s – Relational Model

- IBM comes out with DB2.
- SQL becomes the standard.
- Oracle wins marketplace.
- Stonebraker creates Postgres.

Stonebraker

Stonebraker Lessons

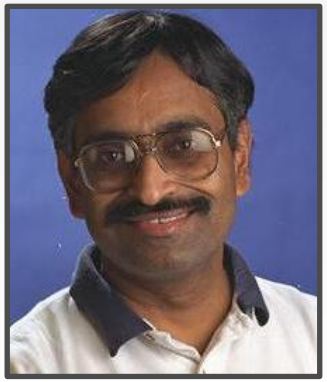
- Set-at-a-time interface offers better physical data independence.
- Database system optimizer is better than manual tuning.

Source: Kristen LeFerve, November 2009, http://bit.ly/13M2Mfb

1980s – Distributed DBs

- SDD-1 – CCA
- System R* – IBM Research
- Gamma – Univ. of Wisconsin
- NonStop SQL – Tandem


Bernstein

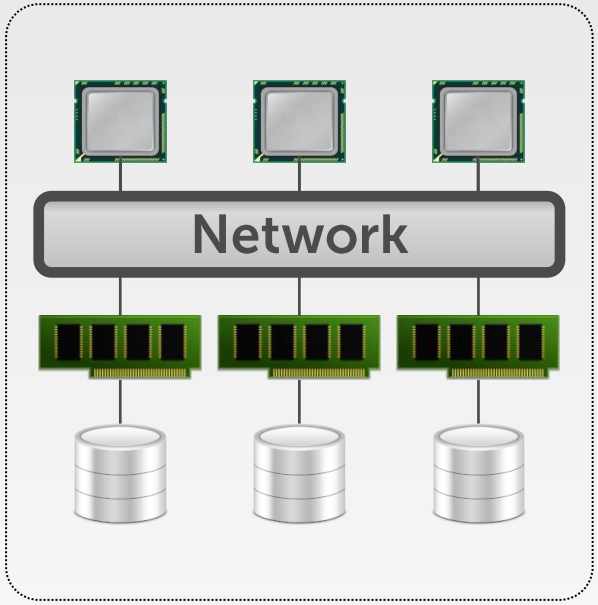

Mohan


DeWitt

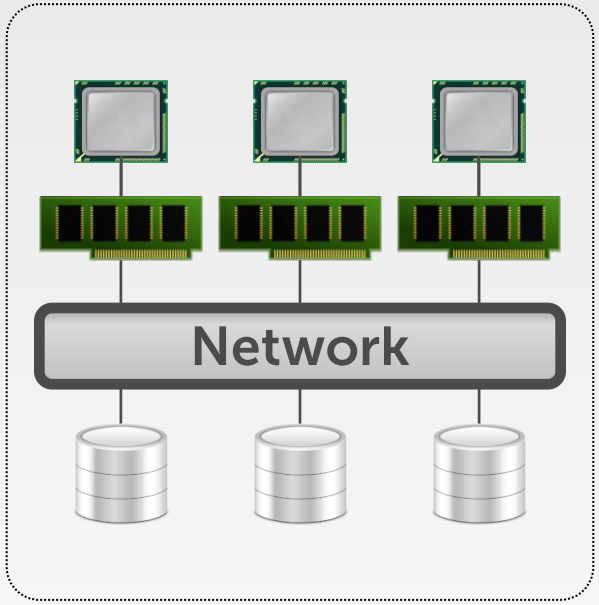

Gray

Quick Detour

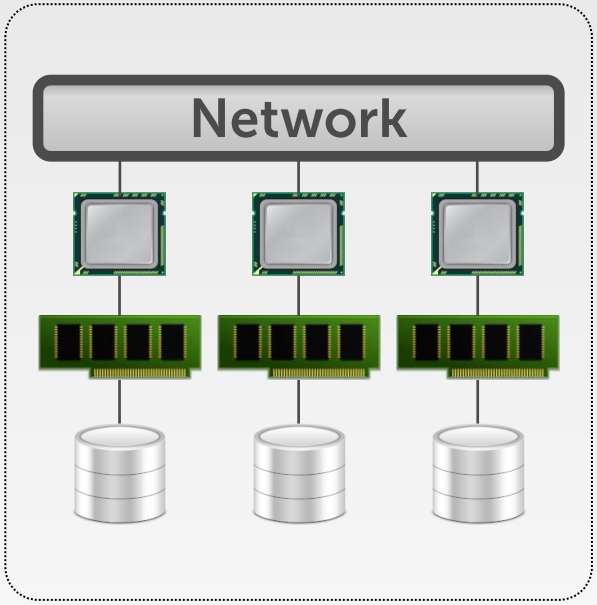
Database Architectures



Shared Memory



Shared Disk

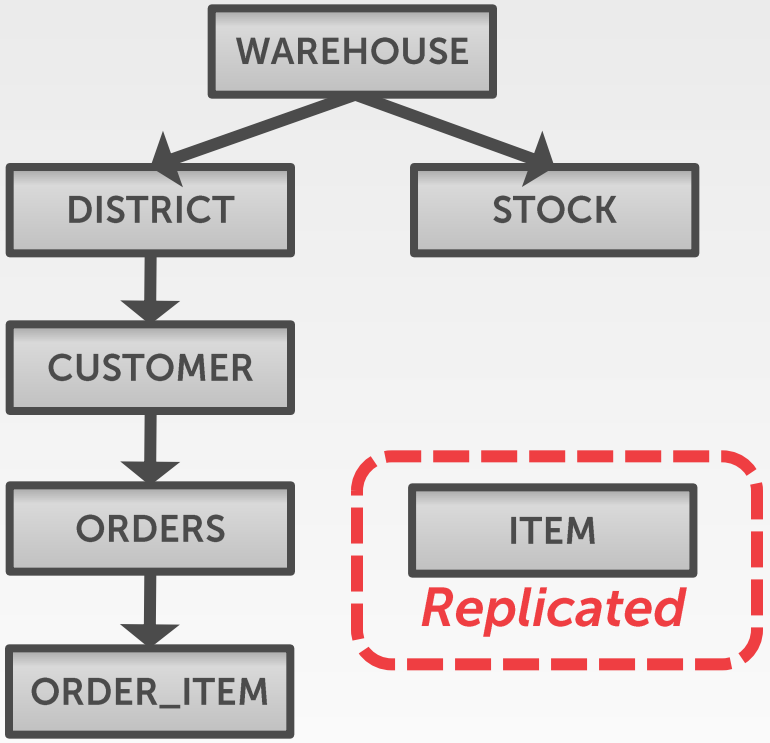


Shared Nothing

Database Partitioning

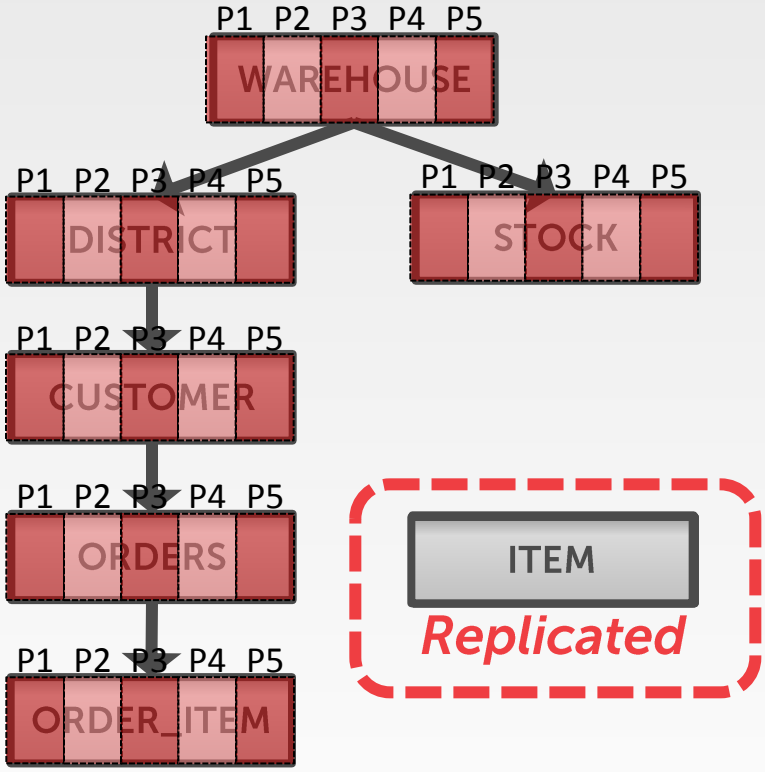
Schema

Schema Tree

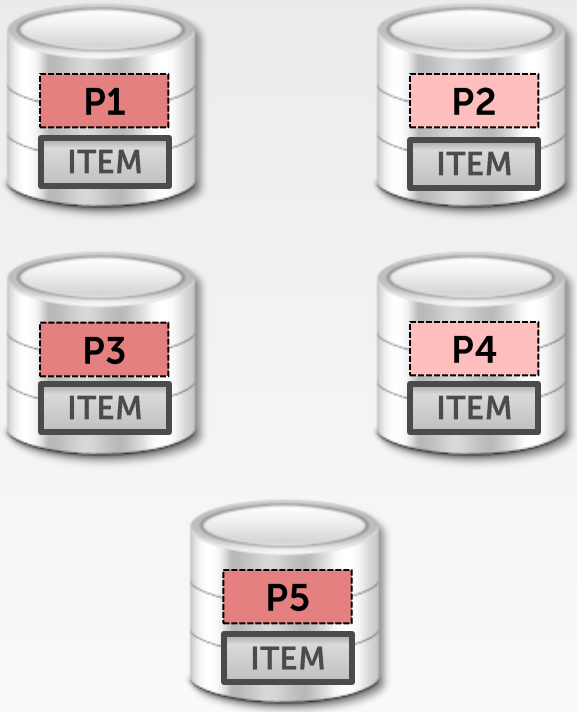


Database Partitioning

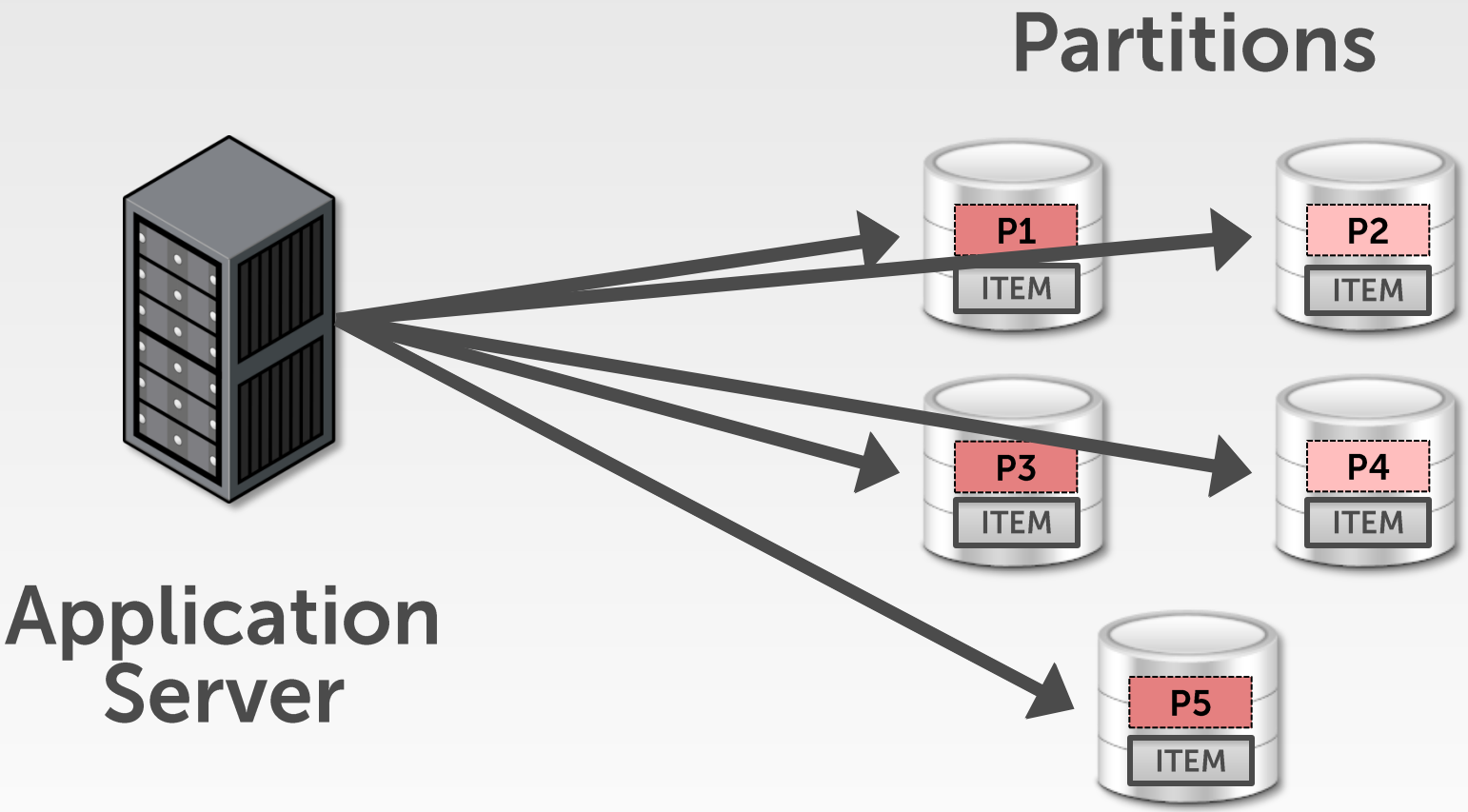
Schema Tree



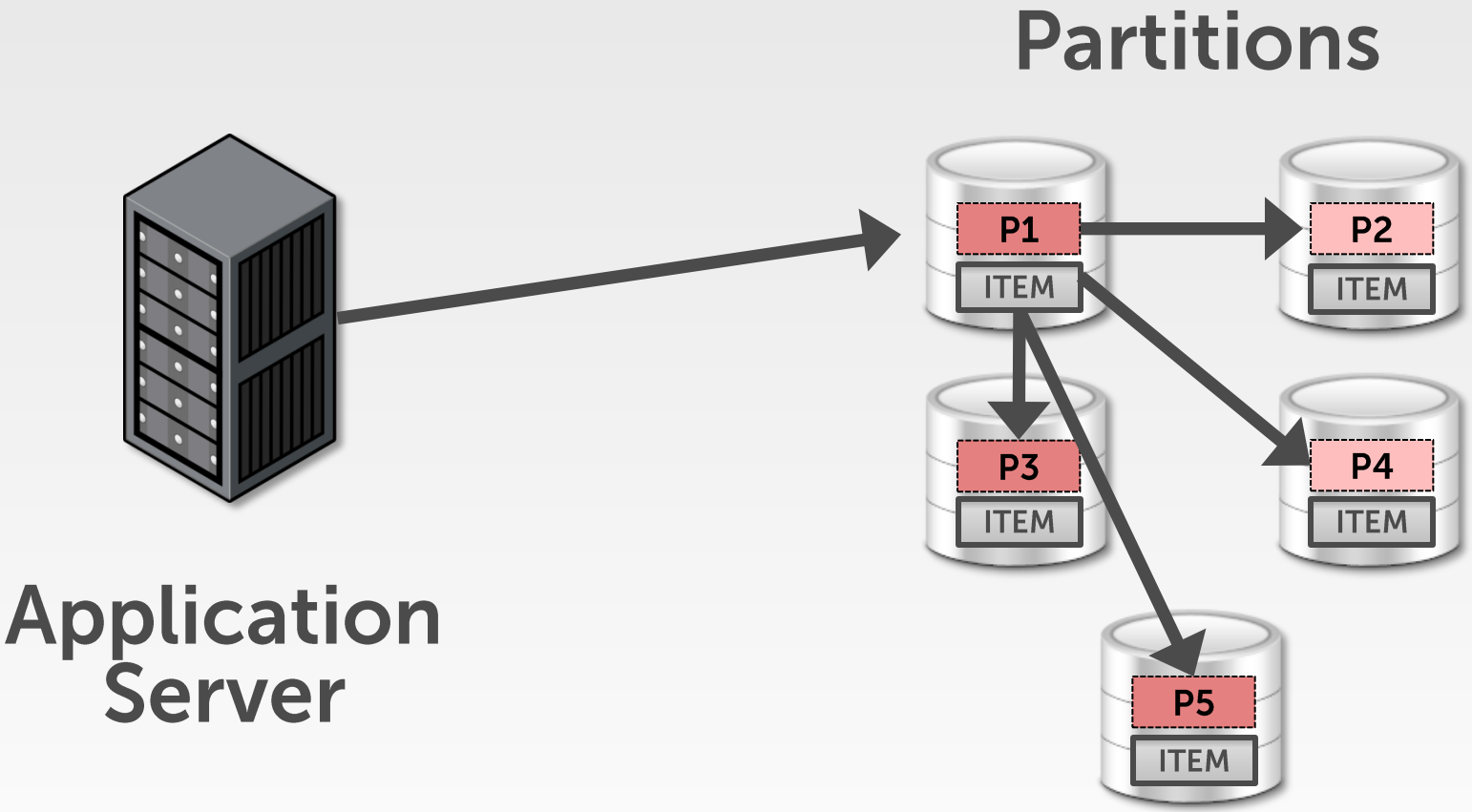
Partitions



Distributed OLTP



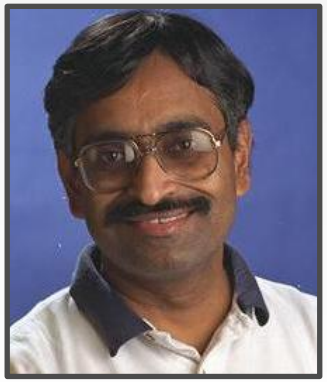
Distributed OLAP



1980s – Distributed DBs

- SDD-1 – CCA
- System R* – IBM Research
- Gamma – Univ. of Wisconsin
- NonStop SQL – Tandem


Bernstein


Mohan


DeWitt


Gray

1980s – OO Databases

- Avoid "relational-object impedance mismatch."
- Tight coupling between objects and database.

Zdonik

Object-Oriented Model

Application Code

```
class Student {
```

Schema

STUDENT_PHONE
(sid, phone)

⚠ Too Much Work

id	name	email
1001	Tone Loc	funky@medina.com

sid	phone
1001	444-444-4444
1001	555-555-5555

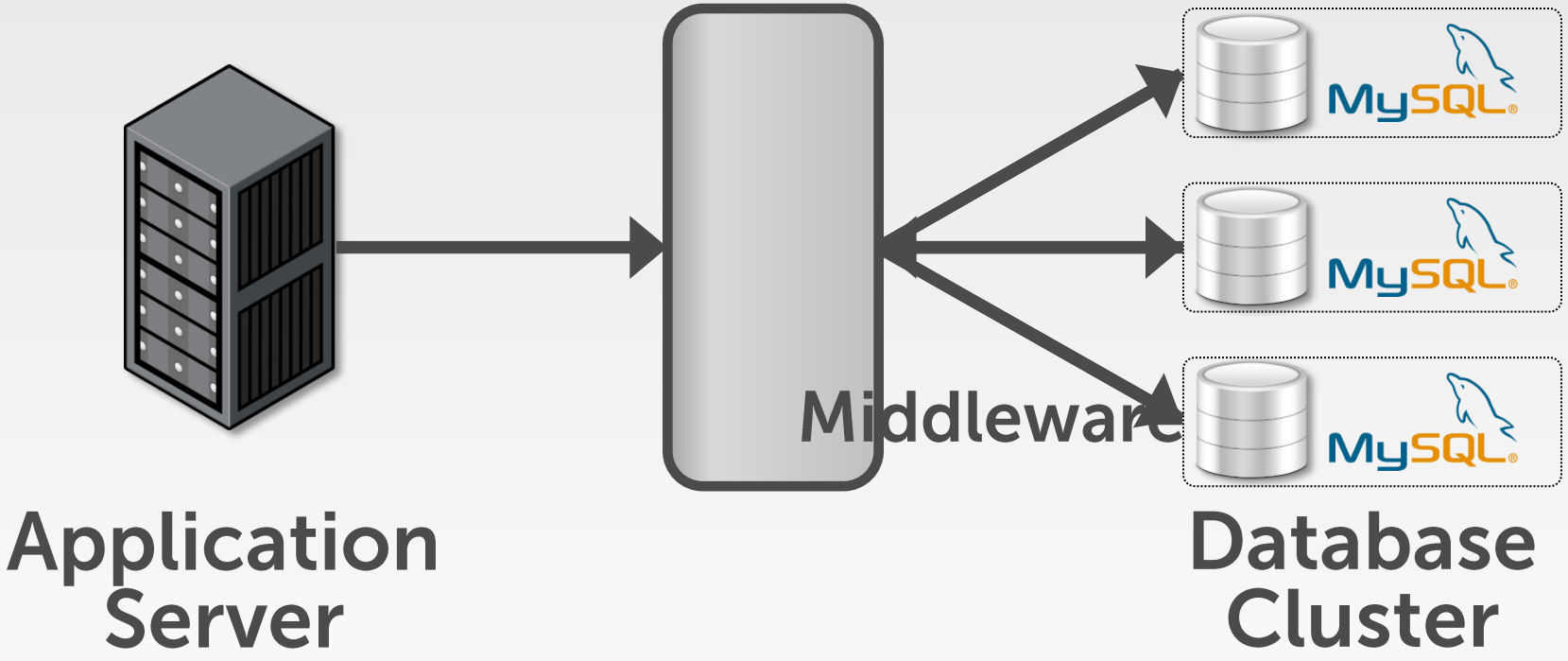
1990s – Boring Days

- Microsoft forks Sybase and creates SQL Server.
- MySQL is written as a replacement for mSQL.
- Postgres gets SQL support.

2000s – Internet Boom

- All the big players were heavyweight and expensive.
- Open-source databases were missing important features.
- Custom scale-out middleware.
  - *Examples: eBay, Facebook*

Middleware Approach



2000s – Data Warehouses

- Rise of the special purpose data warehouse DBMSs.
  - *Distributed / Shared-Nothing*
  - *Relational / SQL*
  - *Alternative storage models.*
  - *Examples: Vertica, Greenplum, Aster Data, Netezza, ParAccel*

2000s – NoSQL

- Focus on high-availability & high-scalability:
  - *–Schemaless ("Schema Last")*
  - *–Not ACID*
  - *–Custom APIs instead of SQL.*

2000s – NoSQL

- Alternative data models:
  - *Column-family (Cassandra, HBase)*
  - *Document (MongoDB, CouchDB)*
  - *Key-value (Riak, Dynamo)*
  - *Graph (Neo4j, FlockDB)*
- Usually open-source.
- "A" + "P" in CAP Theorem

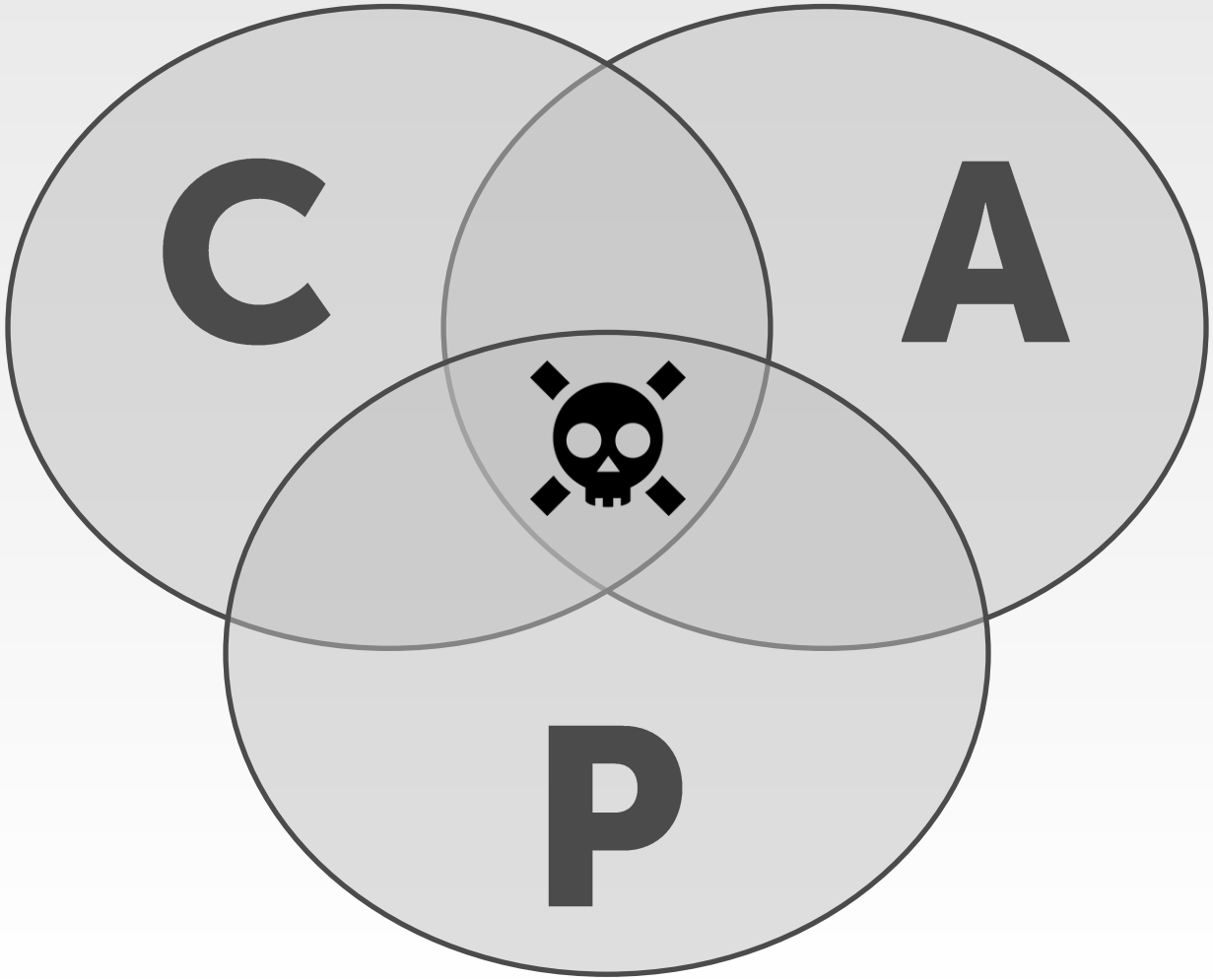
Quick Detour

CAP Theorem

- Proposed by Eric Brewer that it is impossible for a distributed system to always be:
  - *Consistent*
  - *Always Available*
  - *Network Partition Tolerant*
- Proved in 2002.

Brewer

Consistency
Availability
Partition Tolerant



2000s – NoSQL

- Alternative data models:
  - *Column-family (Cassandra, HBase)*
  - *Document (MongoDB, CouchDB)*
  - *Key-value (Riak, Dynamo)*
  - *Graph (Neo4j, FlockDB)*
- Usually open-source.
- "A" + "P" in CAP Theorem

2010s – NewSQL

- Provide same performance of NoSQL without giving up ACID
  - *Relational / SQL*
  - *Distributed (Mostly)*
- Usually closed-source.

2010s – NewSQL

- Different solutions:
  - *Specialized OLTP (H-Store, VoltDB)*
  - *Distributed MVCC (NuoDB)*
  - *Custom Hardware (Clustrix, Spanner)*
  - *Relaxed Consistency (MemSQL, SQLFire)*
  - *Middleware (ScaleBase, dbShards)*

Observations

- Problems outlined in DeWitt paper are still relevant today:
  - *Mixing Workloads.*
  - *Database Design.*
  - *On-Line Reorganization.*

Observations

- Innovations come from both industry and academia.
- IBM was the vanguard during 1970-1980s.
- Google is current trendsetter.

Next Week

- Distributed Transactions
- Consensus Protocols