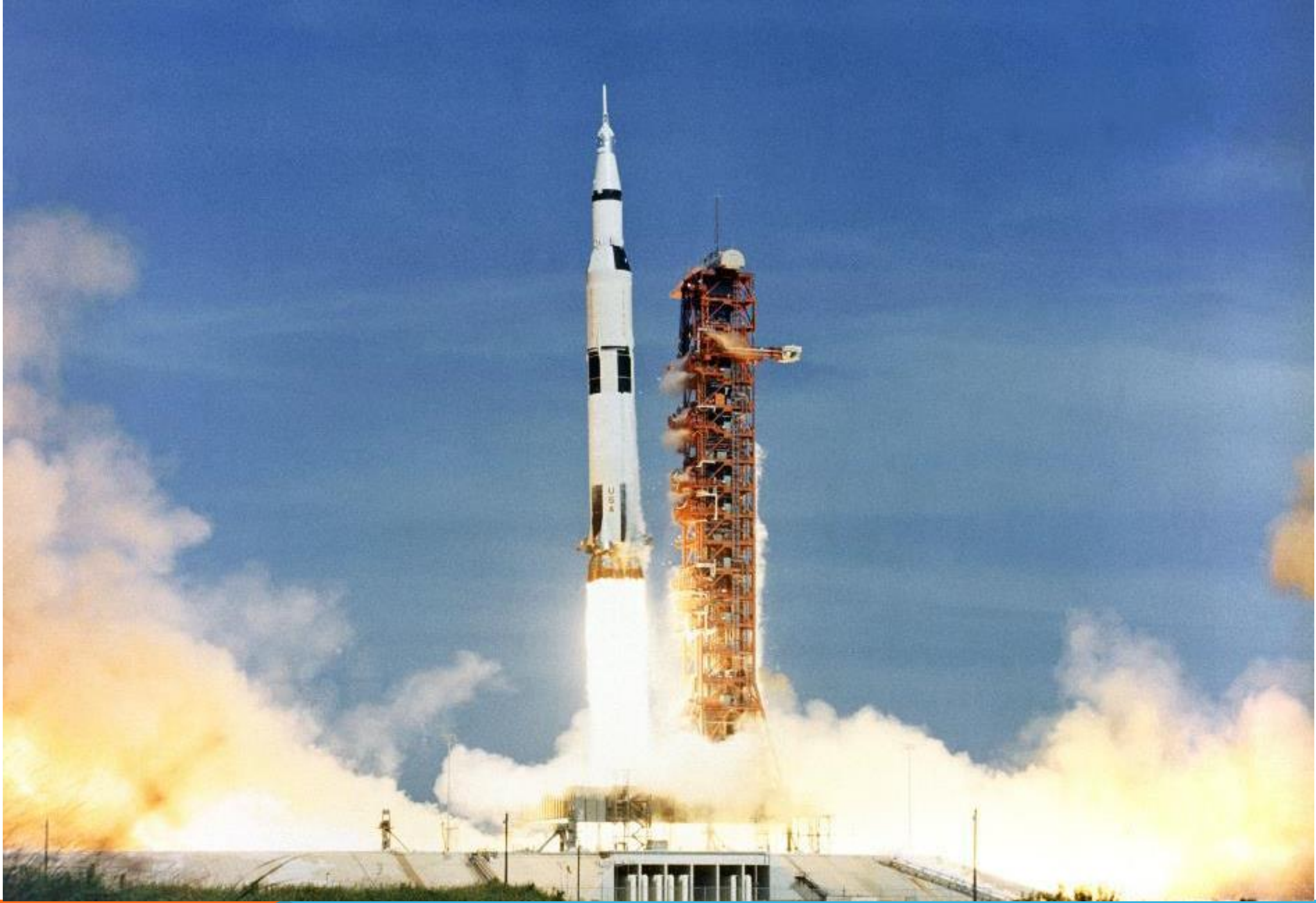
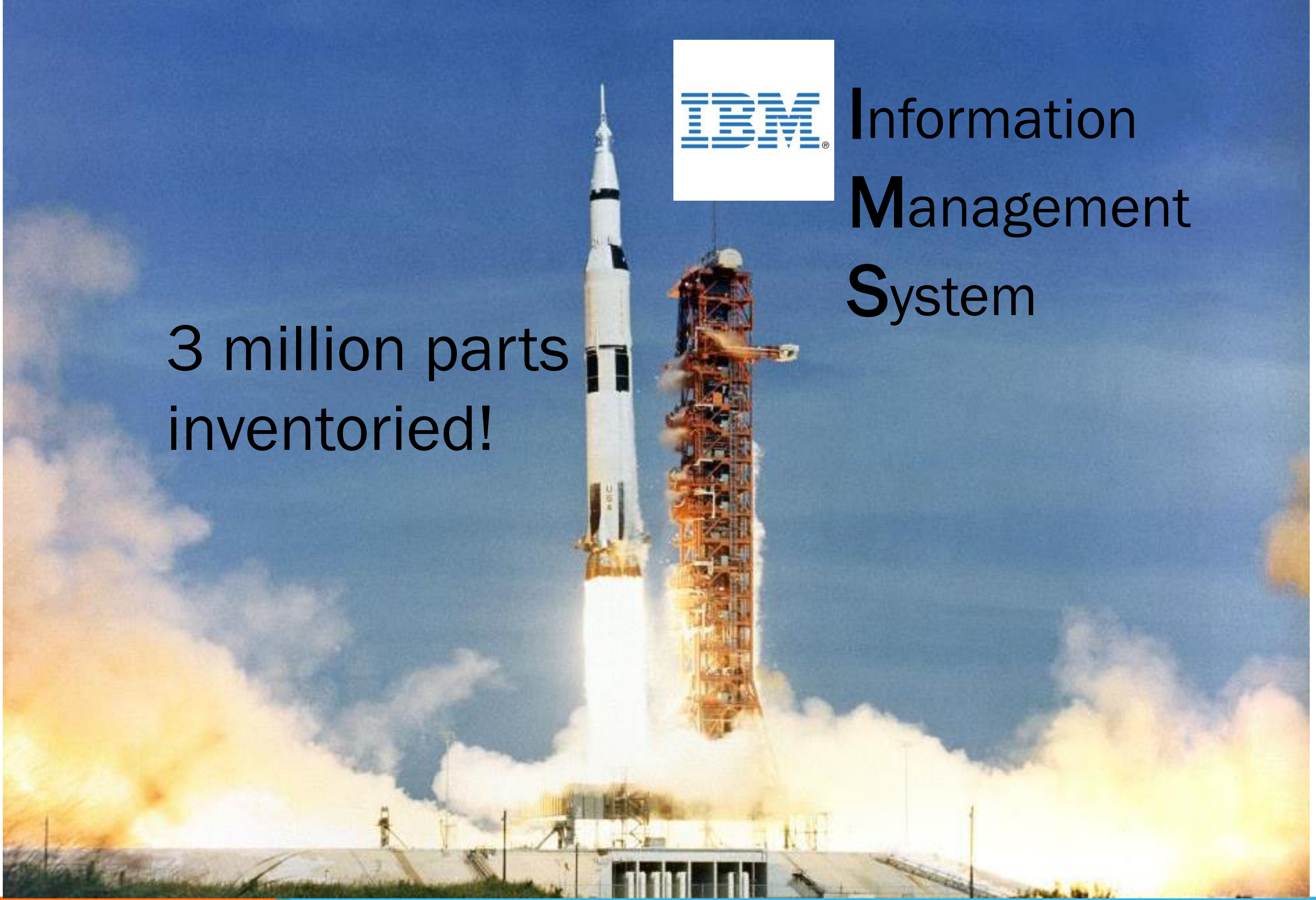
DATABASE INTRO

JOHN JERNIGAN 8/17/2020


USA

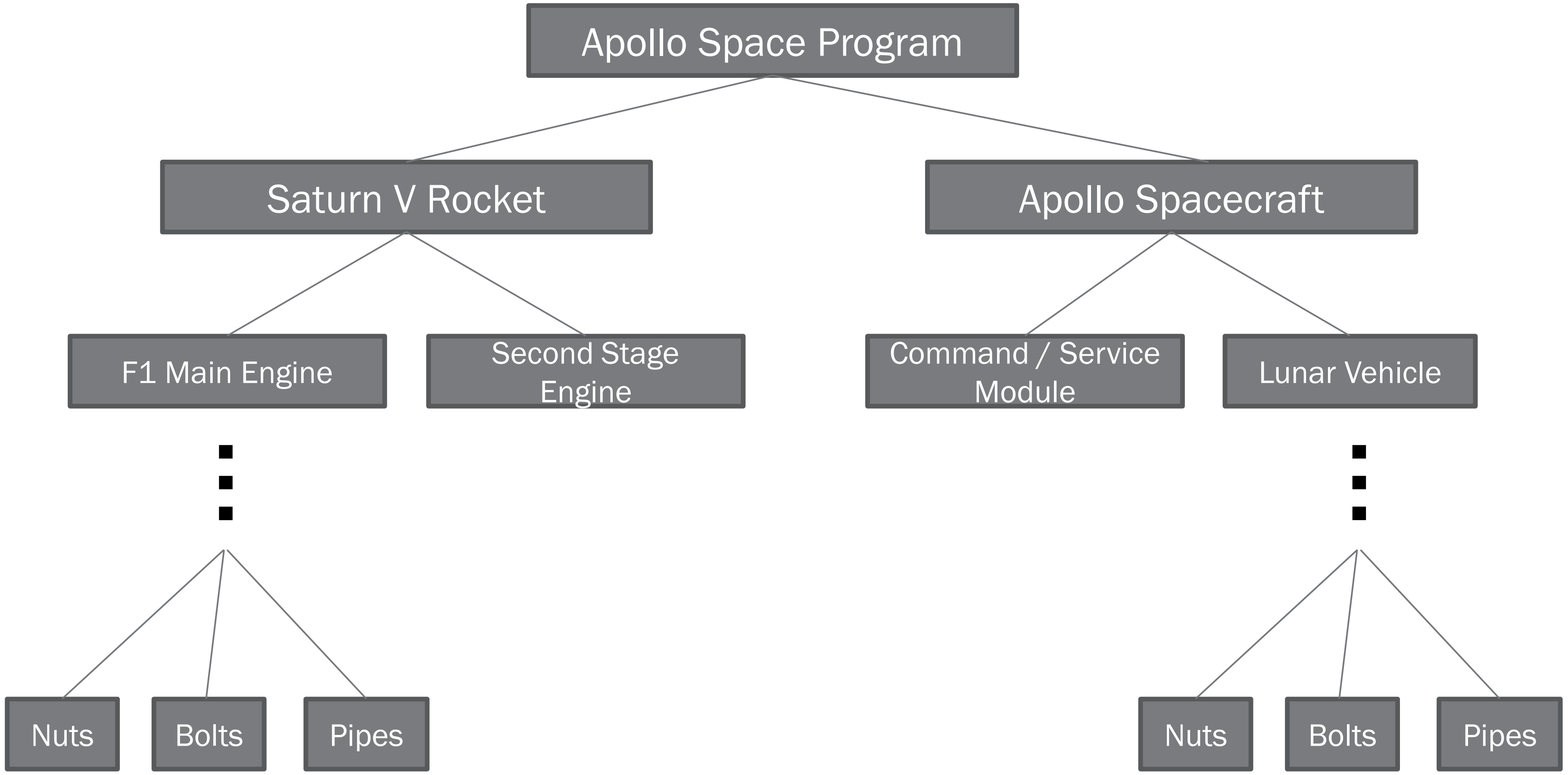


Information
Management
System

3 million parts inventoried!

1966: First Database Management System (DBMS)

IMS had a hierarchical "tree" structure, internally



Tree structures are fast and efficient!

Hierarchical Database Disadvantages:

- Developers MUST understand internal database structure
- Which means you really don’t want to fundamentally alter the database
- Because that will break your existing applications that use it

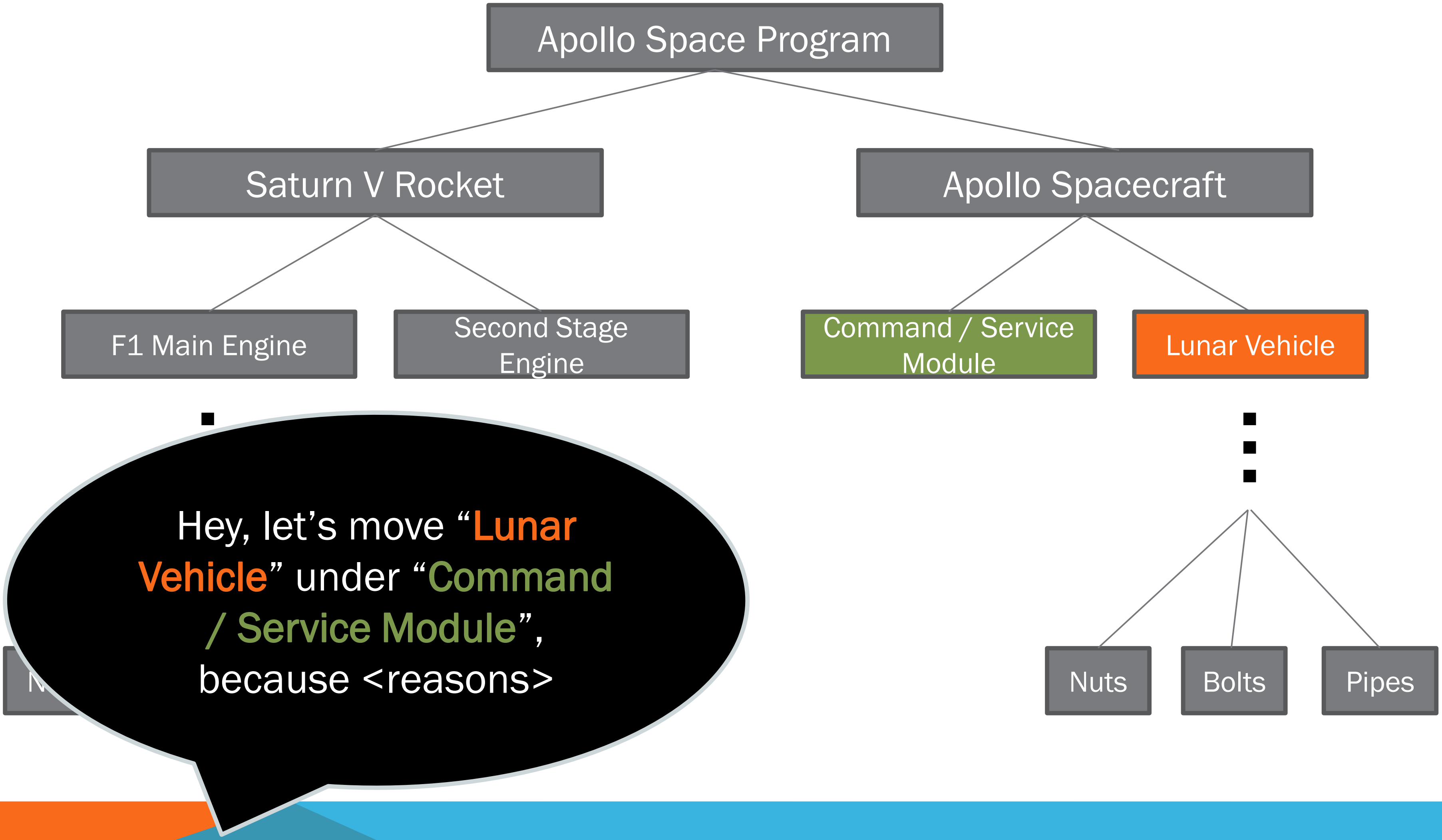
Hierarchical Database Disadvantages:

- Developers MUST understand internal database structure
- Which means you really don’t want to fundamentally alter the database
- Because that will break your existing applications that use it

Hierarchical Database Disadvantages:

- Developers MUST understand internal database structure
- Which means you really don't want to fundamentally alter the database
- Because that will break your existing applications that already use it

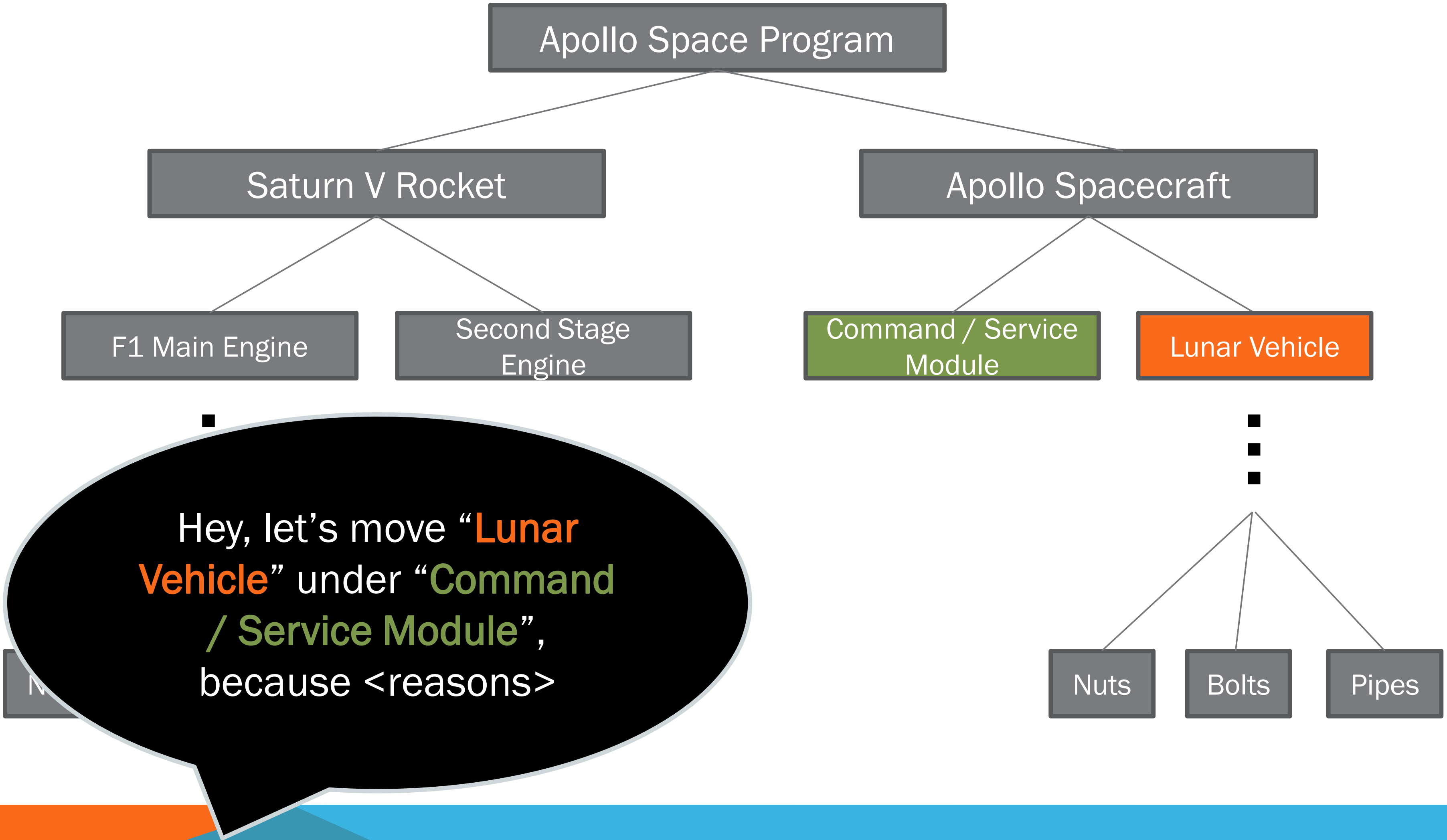
You Really Don't Want to Fundamentally Alter a Hierarchical Database



Hierarchical Database Disadvantages:

- Developers MUST understand internal database structure
- Which means you really don’t want to fundamentally alter the database
- Because that will break your existing applications that use the database

You Really Don't Want to Fundamentally Alter a Hierarchical Database



But *what if* there were a database model…

But *what if* there were a database model…

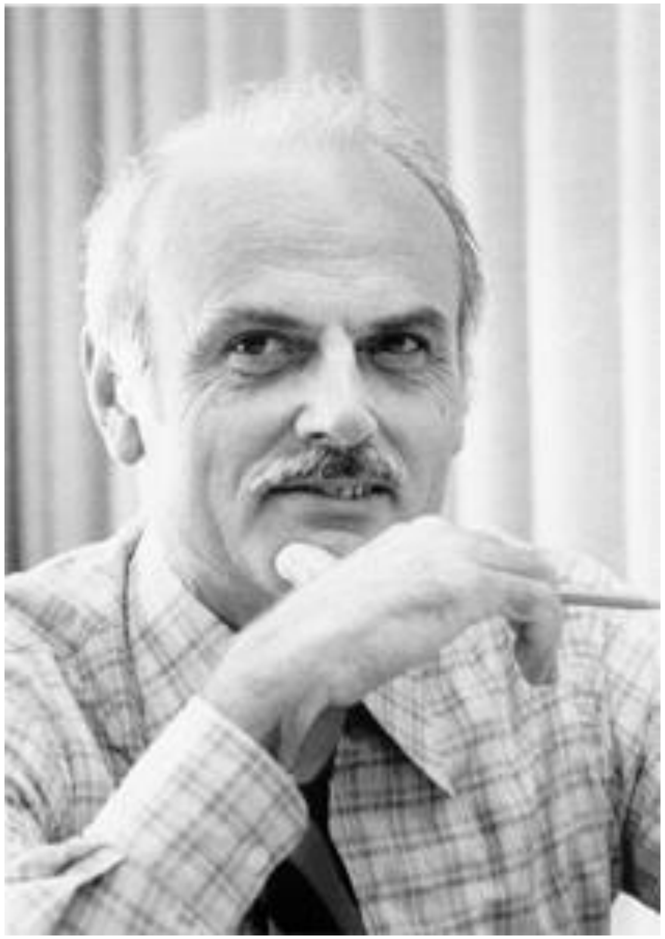
…where expensive database applications *continued to work*…

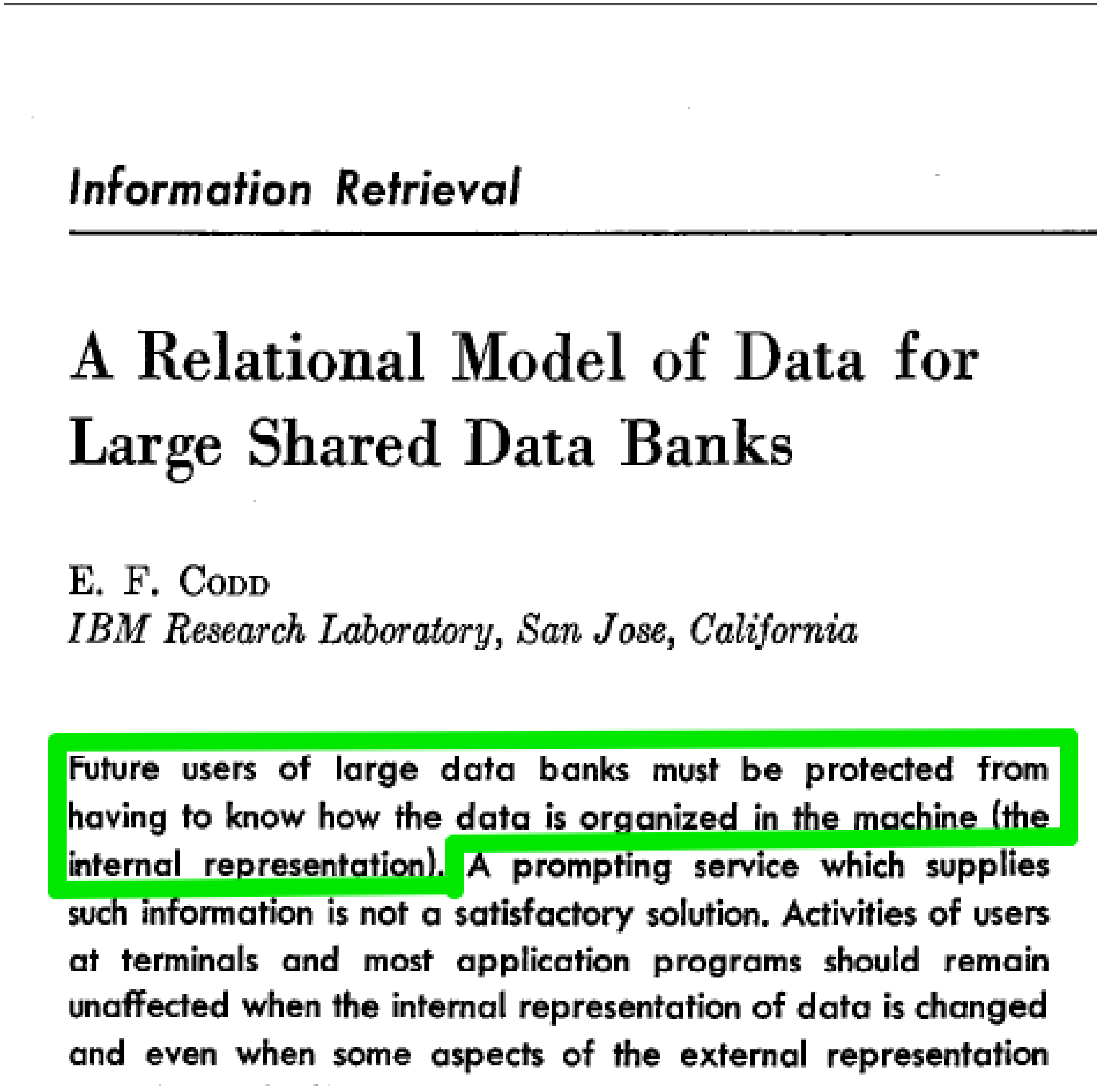
But *what if* there were a database model…

…where expensive database applications *continued to work…*

…even if the internal database structure were altered?

In 1970, Dr. Edgar Codd invents: "*Relational* model for database management"





Information Retrieval

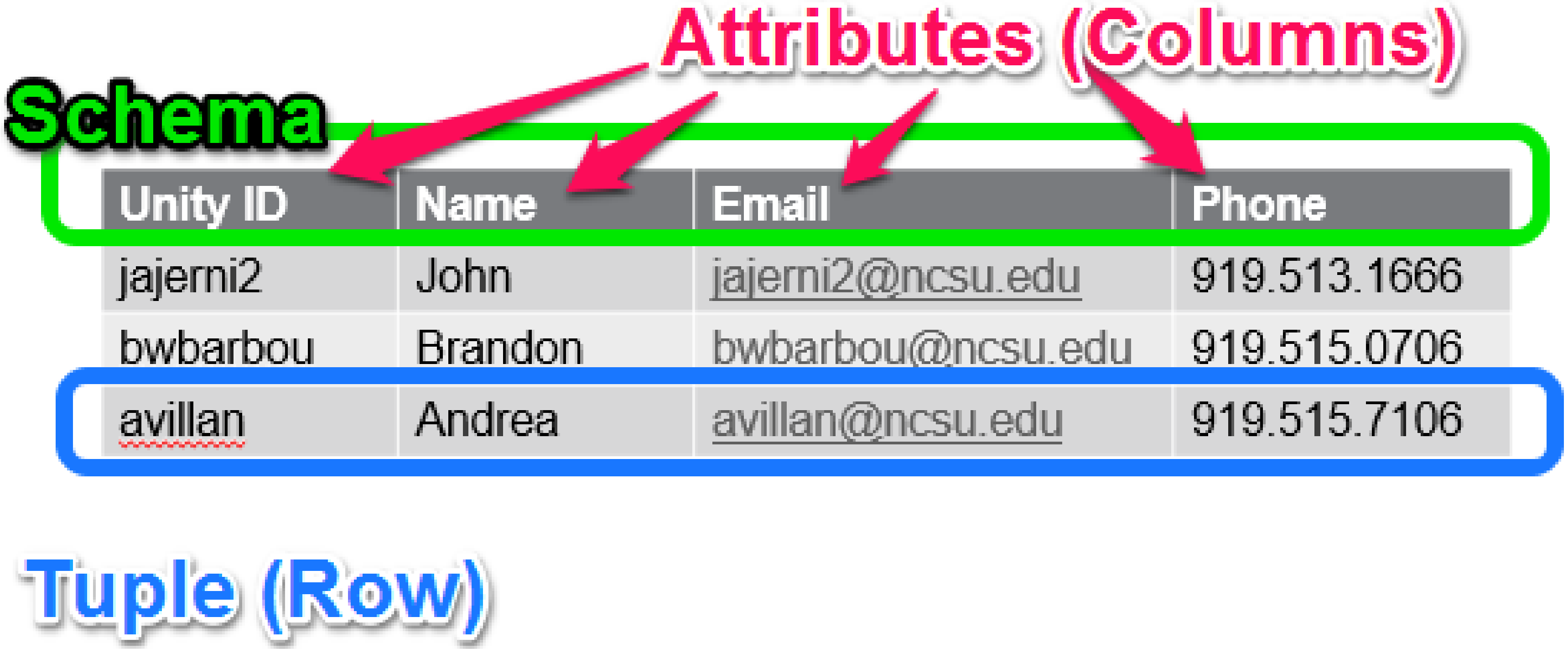
A Relational Model of Data for Large Shared Data Banks

E. F. Codd
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation

Communications of the ACM
Volume 13 Issue 6, June 1970

Relational Databases Use Table Structures, Not Trees


Attributes (Columns)
Schema
Tuple (Row)

Unity ID	Name	Email	Phone
jajerni2	John	jajerni2@ncsu.edu	919.513.1666
bwbarbou	Brandon	bwbarbou@ncsu.edu	919.515.0706
avillan	Andrea	avillan@ncsu.edu	919.515.7106

Use Structured Query Language (SQL) to Interface with Database

Unity ID	Name	Email	Phone
jajerni2	John	jajerni2@ncsu.edu	919.513.1666
bwbarbou	Brandon	bwbarbou@ncsu.edu	919.515.0706

mysql> SELECT Name,Email FROM staff WHERE ‘Unity ID’=‘jajerni2’;

```
+---------+--------------------+
| Name    | Email              |
+---------+--------------------+
| John    | jajerni2@ncsu.edu  |
+---------+--------------------+
```

Creating the First Relational Databases…

Creating the First Relational Databases…

Ed Oates Bob Miner Larry Ellison

Creating the First Relational Databases…

ORACLE®

Some Top Relational Database Management Systems

- Oracle
- Microsoft SQL Server
- MySQL   (free, open-source; see: MariaDB)
- Postgres (free, open-source)

Some Top Relational Database Management Systems

- Oracle
- Microsoft SQL Server
- MySQL (free, open-source; see: MariaDB)
- Postgres (free, open-source)

Some Top Relational Database Management Systems

- Oracle
- Microsoft SQL Server
- MySQL (free, open-source; see: MariaDB)
- Postgres (free, open-source)

Some Top Relational Database Management Systems

- Oracle
- Microsoft SQL Server
- MySQL (free, open-source; see: MariaDB)
- PostgreSQL (free, open-source)

Special Mention of Popular Database:

SQLite

- Stripped-down
- Bare-essentials
- Still powerful (also: free)

What’s Wrong with Relational Databases?

What’s Wrong with Relational Databases?

ORACLE DATABASE

PostgreSQL

MySQL

- Not much

What’s Wrong with Relational Databases?

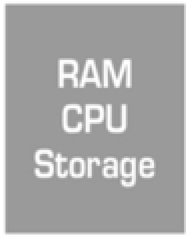
- Not much
- Until you start dealing with Big Data

CLASSIC "BIG DATA" DEFINITION

HTTP://WWW.SLIDESHARE.NET/UWESEILER/INTRODUCTION-TO-THE-HADOOP-ECOSYSTEM-25557364

Database Too Slow?

Scale Resources Vertically



Database Too Slow?

Scale Resources Vertically

RAM

CPU

Storage

Database Too Slow?

Scale Resources Vertically

RAM

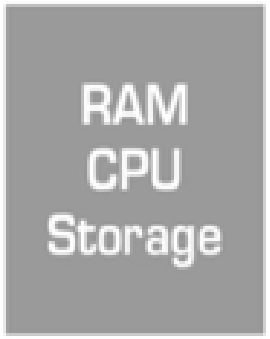
CPU

Storage

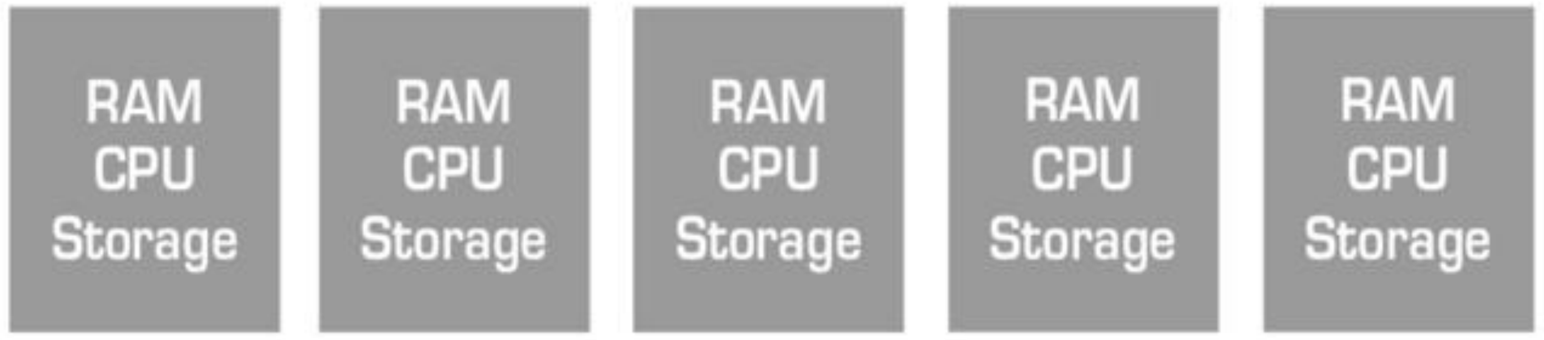
What Can Go Wrong With Relational Databases:

Poor Performance

Or … Scale Resources Horizontally



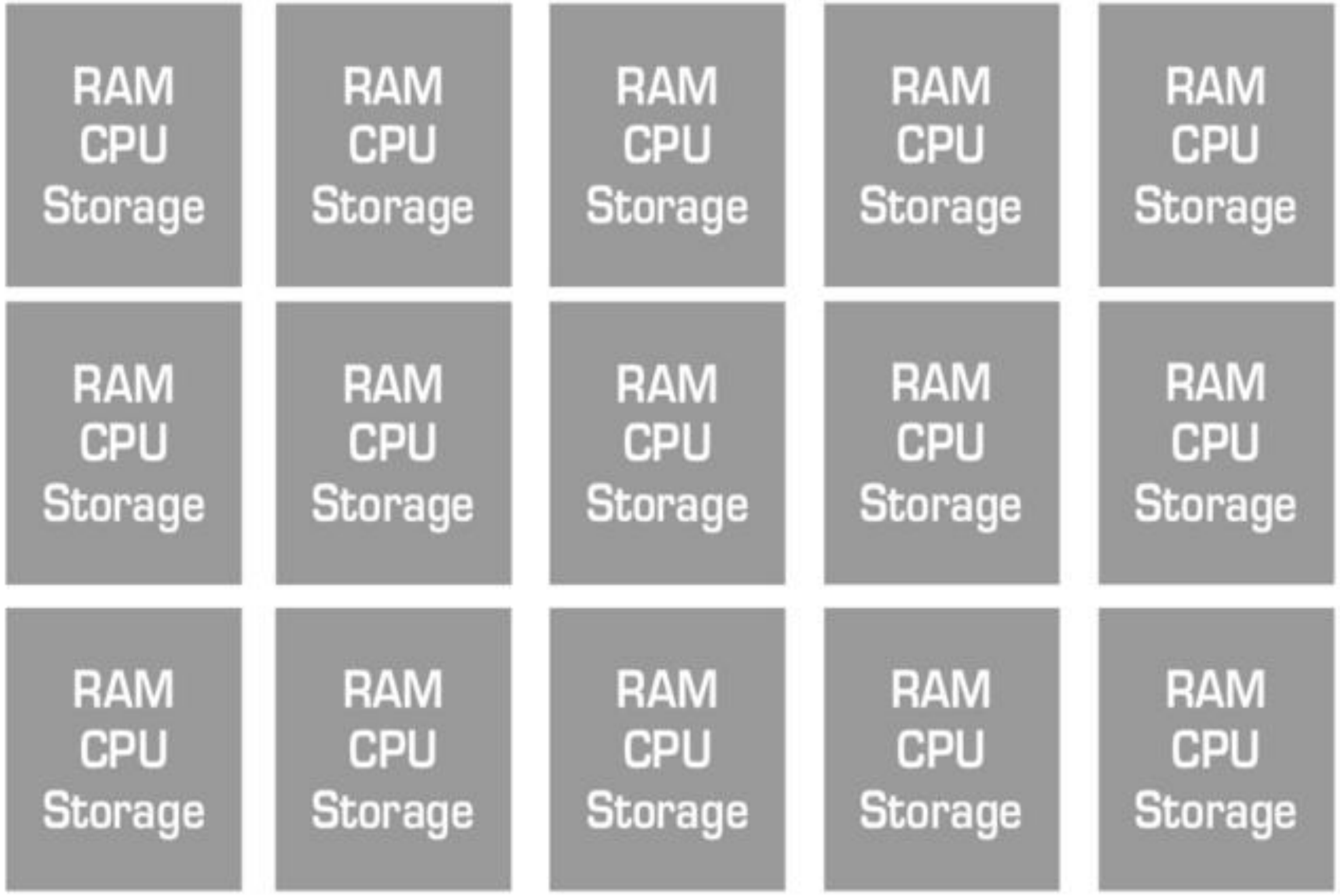
Or … Scale Resources Horizontally



What Can Go Wrong With Relational Databases:

Poor Performance

Or … Scale Resources Horizontally



What Can Go Wrong With Relational Databases:

Poor Performance

Scaling Vertically: Limited by physics, and state-of-the-art

Scaling Horizontally: Works! But … presents new problems

What Can Go Wrong With Relational Databases:

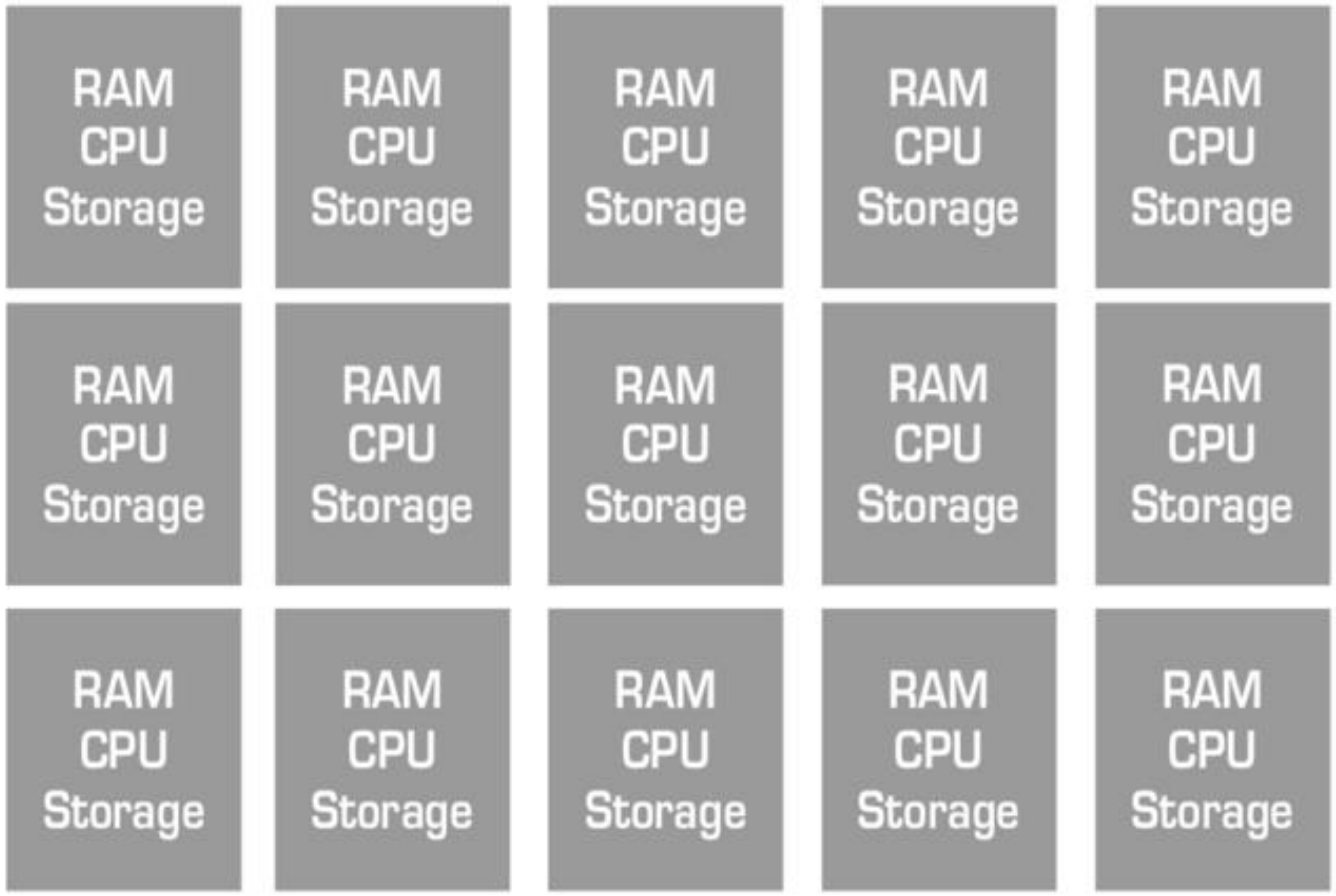
Poor Performance

Scaling Vertically: Limited by physics, and state-of-the-art

Scaling Horizontally: Works! But …
presents new problems

What Can Go Wrong With Relational Databases: **Poor Performance**

Consider the synchronization issues when updating the database simultaneously on multiple nodes



What Can Go Wrong With Relational Databases: **Poor Performance**

Engineers attempted to build <u>**better performing**</u> databases. They were called…

NoSQL - A misleading term

Have you heard the term NoSQL?

NoSQL - A misleading term

NoSQL?

It *seems* to mean
“not a relational database”

NoSQL - A misleading term

A better way of interpreting it:

NoSQL - A misleading term

Think: Solving Big Data database challenges with *application-specific solutions*.

NoSQL - A misleading term

Think: Solving Big Data database challenges with *application-specific solutions*.

Working with big JSON data?
Try:

CouchDB
relax

NoSQL - A misleading term

Think: Solving Big Data database challenges with *application-specific solutions*.

Working with big key-value pairs?
Try:

redis

NoSQL - A misleading term

Think: Solving Big Data database challenges with *application-specific solutions*.

Working with big graph data?
Try:

neo4j

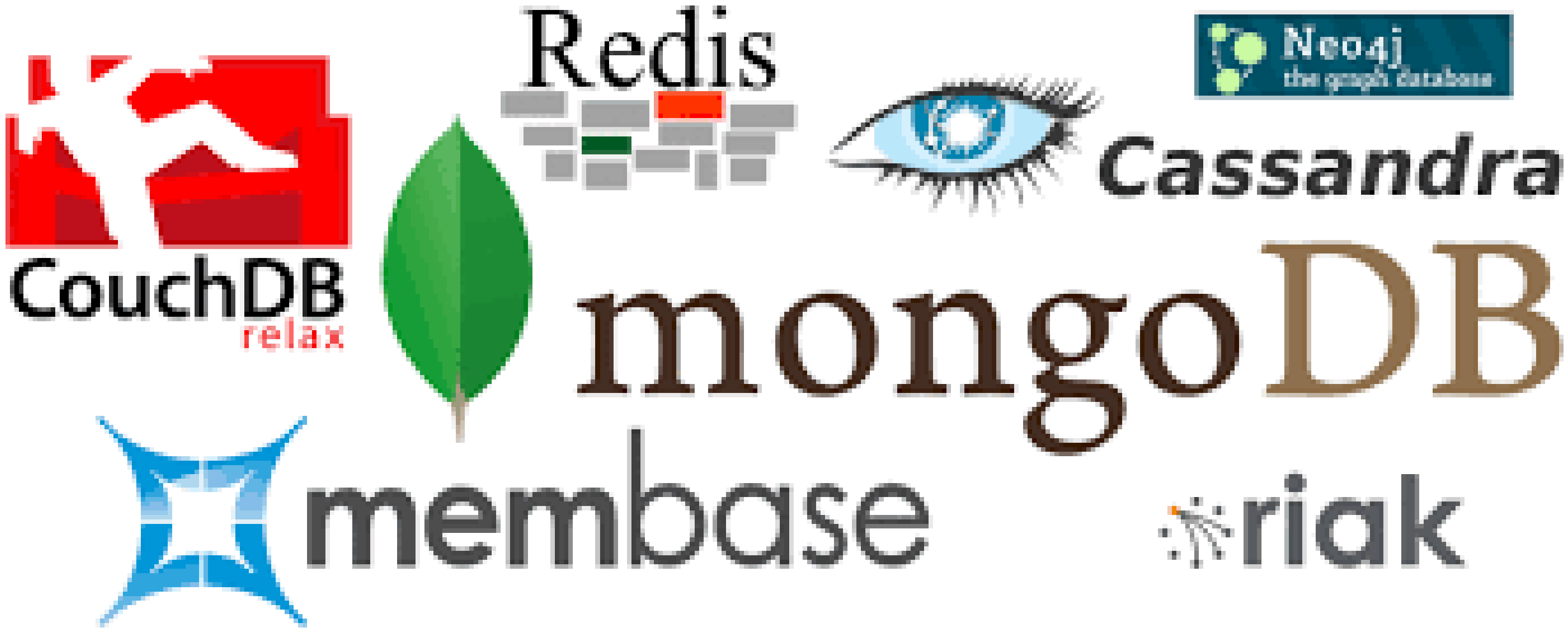
ORACLE DATABASE

PostgreSQL

MySQL

NoSQL - A misleading term

Think: Solving Big Data database challenges with *application-specific solutions*.



Recap: The Big Picture of Relational Databases and NoSQL

Recap: The Big Picture of Relational Databases and NoSQL

IBM IMS ⇨ **Relational** Database Management Systems (RDBMS) ⇨ **NoSQL** Database Management Systems

Recap: The Big Picture of Relational Databases and NoSQL

Gartner “Strategic Planning Assumption”:

By 2017, all leading operational DBMSs will offer multiple data models, relational and NoSQL, in a single DBMS platform.

https://www.gartner.com/doc/reprints?id=1-2PMFPEN&ct=151013&st=sb

RDBMS

NoSQL

RDBMS + NoSQL = *Flying Car*

PRACTICUM TIP: USE INDEXING

If you have a SAS7BDAT dataset…

PRACTICUM TIP: USE INDEXING

If you have a SAS7BDAT dataset…

Or your data is inside MySQL / Postgres / MSSQL…

PRACTICUM TIP: USE INDEXING

If you have a SAS7BDAT dataset…

Or your data is inside MySQL / Postgres / MSSQL…

You NEED to CONSIDER where to

create indexes for performance purposes.

PRACTICUM TIP: USE INDEXING

✓ *Dramatically* decrease query times!

E.g. 1 hour→ 2.4 seconds

I have seen this… that is a <u>1500x</u> speedup.

PRACTICUM TIP: USE INDEXING

- ✓ *Dramatically* decrease query times!
- ✓ *Speed up* time to sort data!

E.g. 1 hour → 2.4 seconds

I have seen this… that is a <u>1500x</u> speedup.

PRACTICUM TIP: USE INDEXING

- Indexes are created *<u>per-column </u>(attribute)*

e.g.

- Employee Number
- Purchase Price
- Transaction Date

Index them all!

PRACTICUM TIP: USE INDEXING

- Indexes are created *per-column (attribute)*

- Indexes can hugely speed up your queries!

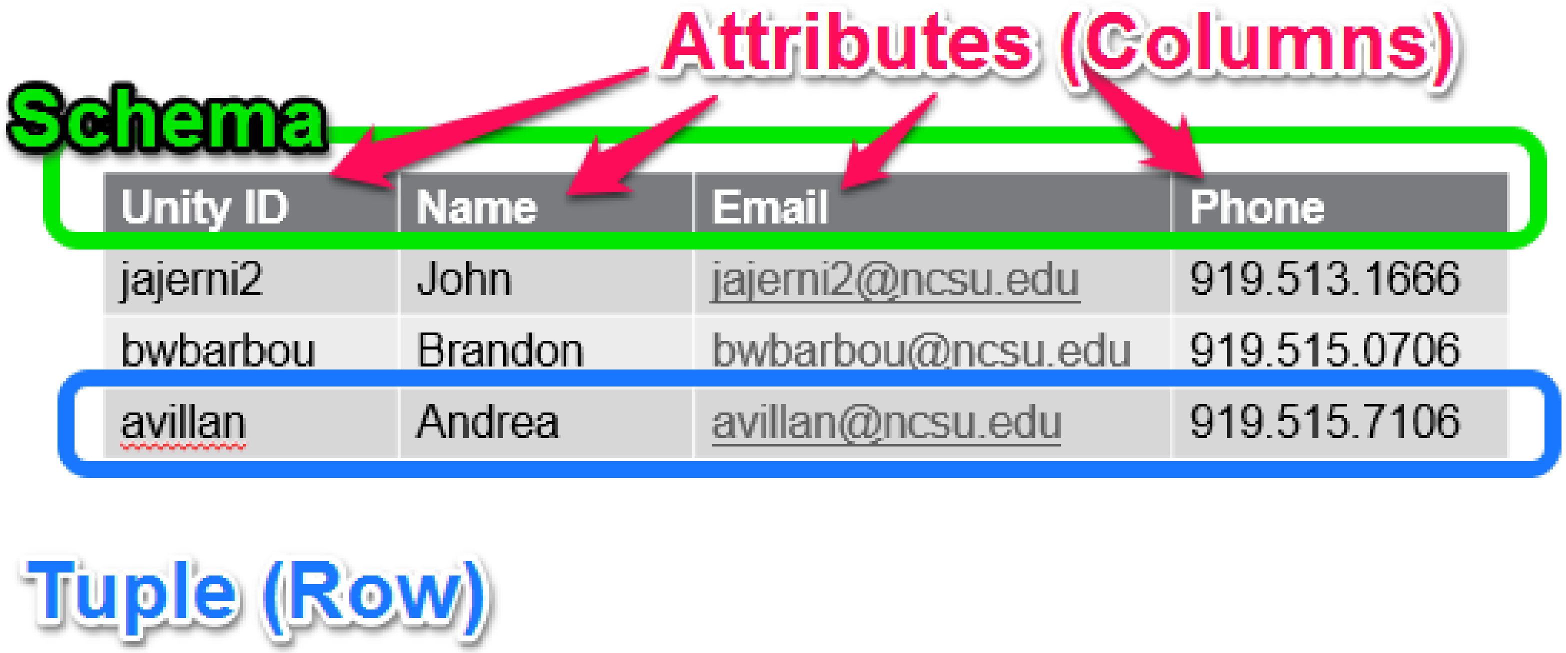
PRACTICUM TIP: USE INDEXING

- Indexes are created *per-column* (*attribute*)
- Indexes can hugely speed up your queries!
- Index updates are ***performance overhead*** ***when data is added to database*** which is a tradeoff to consider.

MSA Students:	Don’t worry about this!
Real-World Production Systems:	Worry about this!

Consider Which Columns You are Querying Against

E.g. will you be searching the table by Name? Unity ID? All columns?

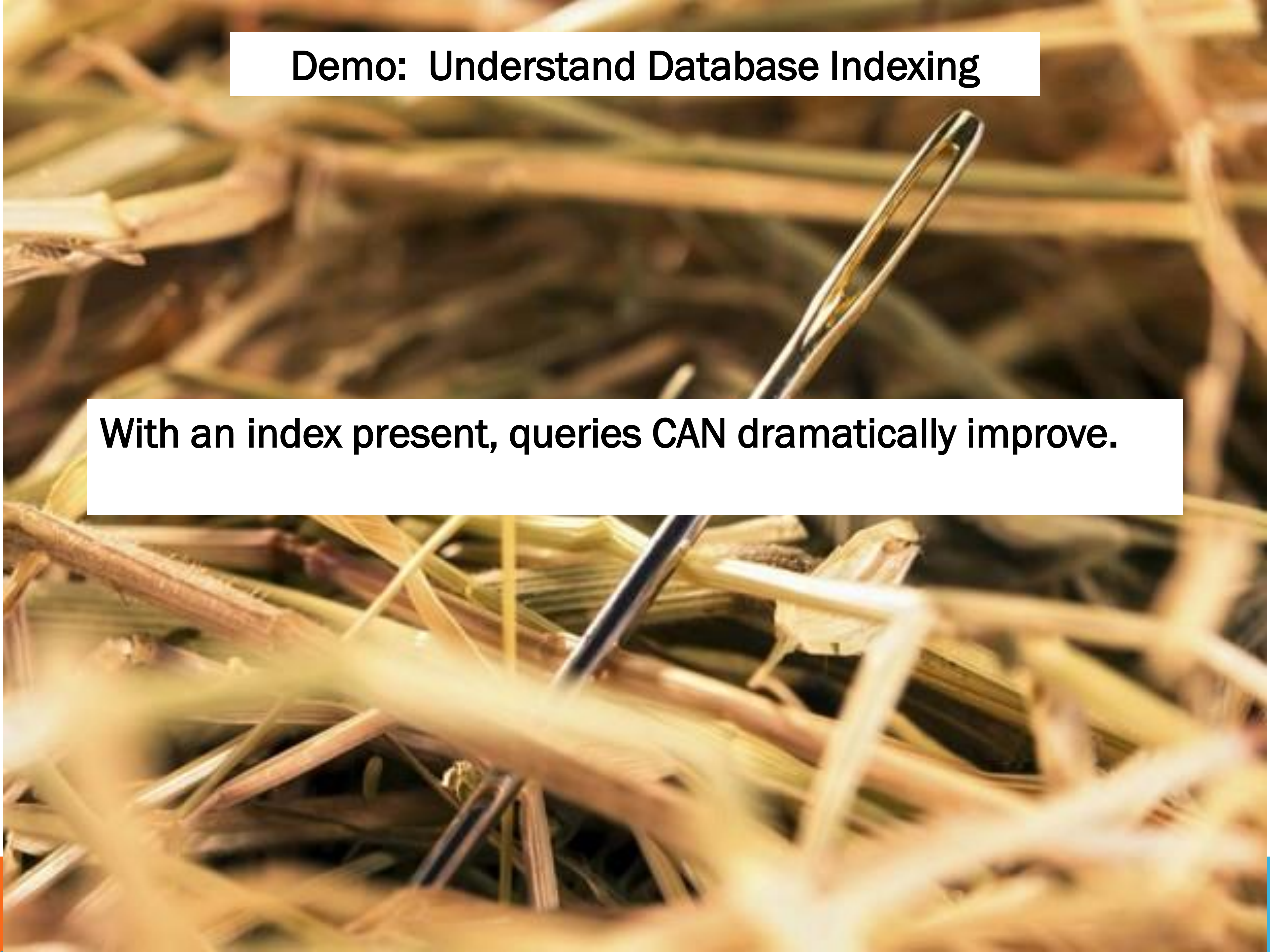

Attributes (Columns)
Schema
Tuple (Row)

Unity ID	Name	Email	Phone
jajerni2	John	jajerni2@ncsu.edu	919.513.1666
bwbarbou	Brandon	bwbarbou@ncsu.edu	919.515.0706
avillan	Andrea	avillan@ncsu.edu	919.515.7106

Demo: Understand Database Indexing

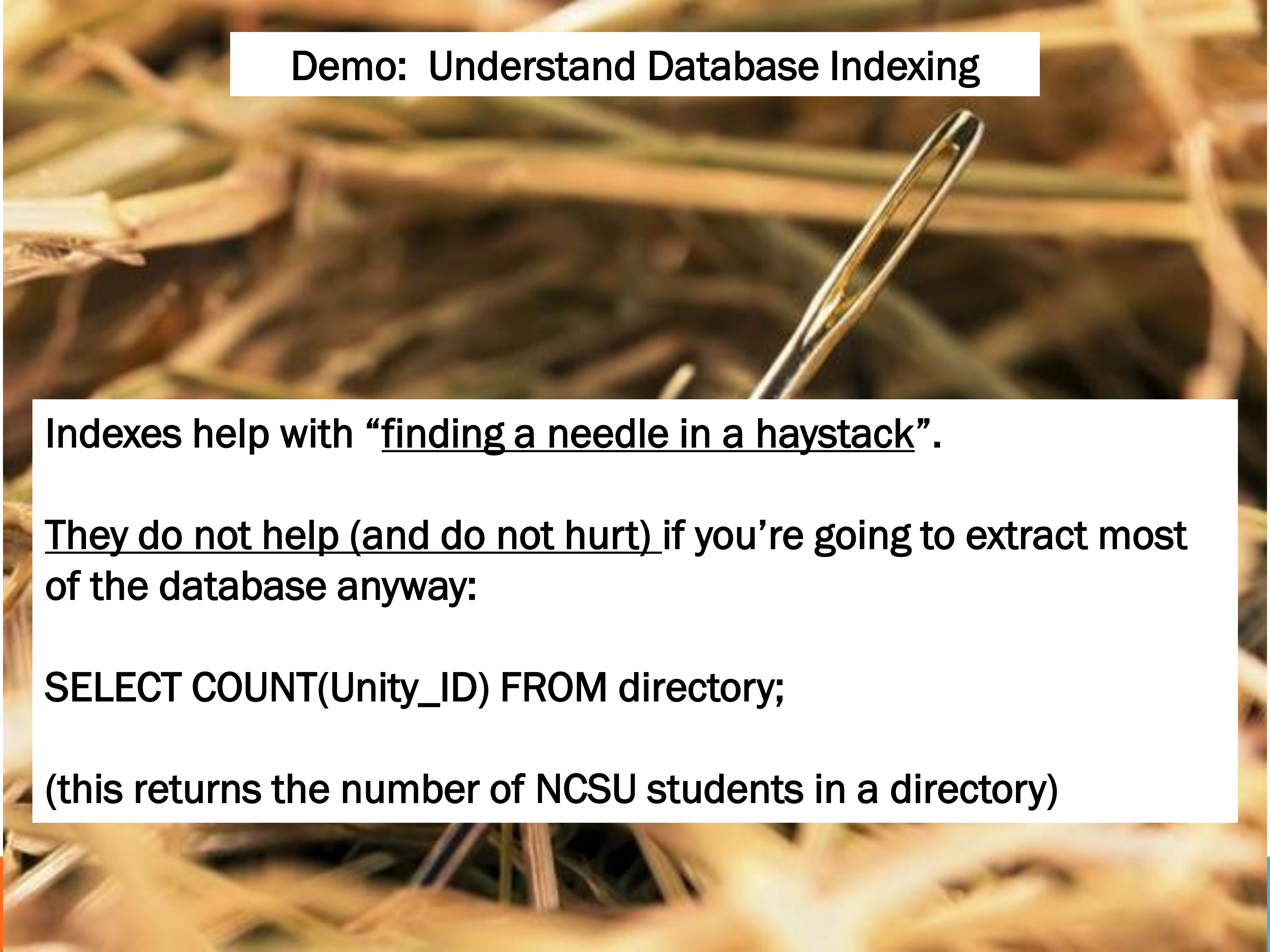
Demo: Understand Database Indexing

Demo: Understand Database Indexing



With an index present, queries CAN dramatically improve.

Demo: Understand Database Indexing



Indexes help with "finding a needle in a haystack".

They do not help (and do not hurt) if you're going to extract most of the database anyway:

SELECT COUNT(Unity_ID) FROM directory;

(this returns the number of NCSU students in a directory)

Index Your Database Columns!

As a data scientist…

… if you use SQL queries in a database…

Index Your Database Columns!

As a data scientist…

… if you use SQL queries in a database…

… which includes proc sql on a SAS7BDAT file …

… or SQL with PostgreSQL …

Index Your Database Columns!

As a data scientist…

… if you use SQL queries in a database…

… which includes proc sql on a SAS7BDAT file …

… or SQL with PostgreSQL …

… learn how to turn on indexing for any relevant data columns.

Index Your Database Columns!

As a data scientist…

… if you use SQL queries in a database…

… which includes proc sql on a SAS7BDAT file …

… or SQL with PostgreSQL …

… learn how to turn on indexing for any relevant data columns.

(e.g. index “date” columns for transactional data)

Index Your Database Columns!

As a data scientist…

… if you use SQL queries in a database…

… which includes proc sql on a SAS7BDAT file …

… or SQL with PostgreSQL …

… learn how to turn on indexing for any relevant data columns.

(e.g. index "date" columns for transactional data)

This will save you time whenever you're looking for a small bit of data in a large dataset…

… but it's up to YOU to turn on indexing.

For more info on indexing…

- SAS Indexes: http://www2.sas.com/proceedings/sugi29/123-29.pdf

  Also: SAS Programming 3 book, section 3-5

- SQLite Indexes: http://www.sqlitetutorial.net/sqlite-index/

- PostgreSQL Indexes:

  https://www.postgresql.org/docs/current/static/indexes.html

- General Info: https://en.wikipedia.org/wiki/Database_index

Questions