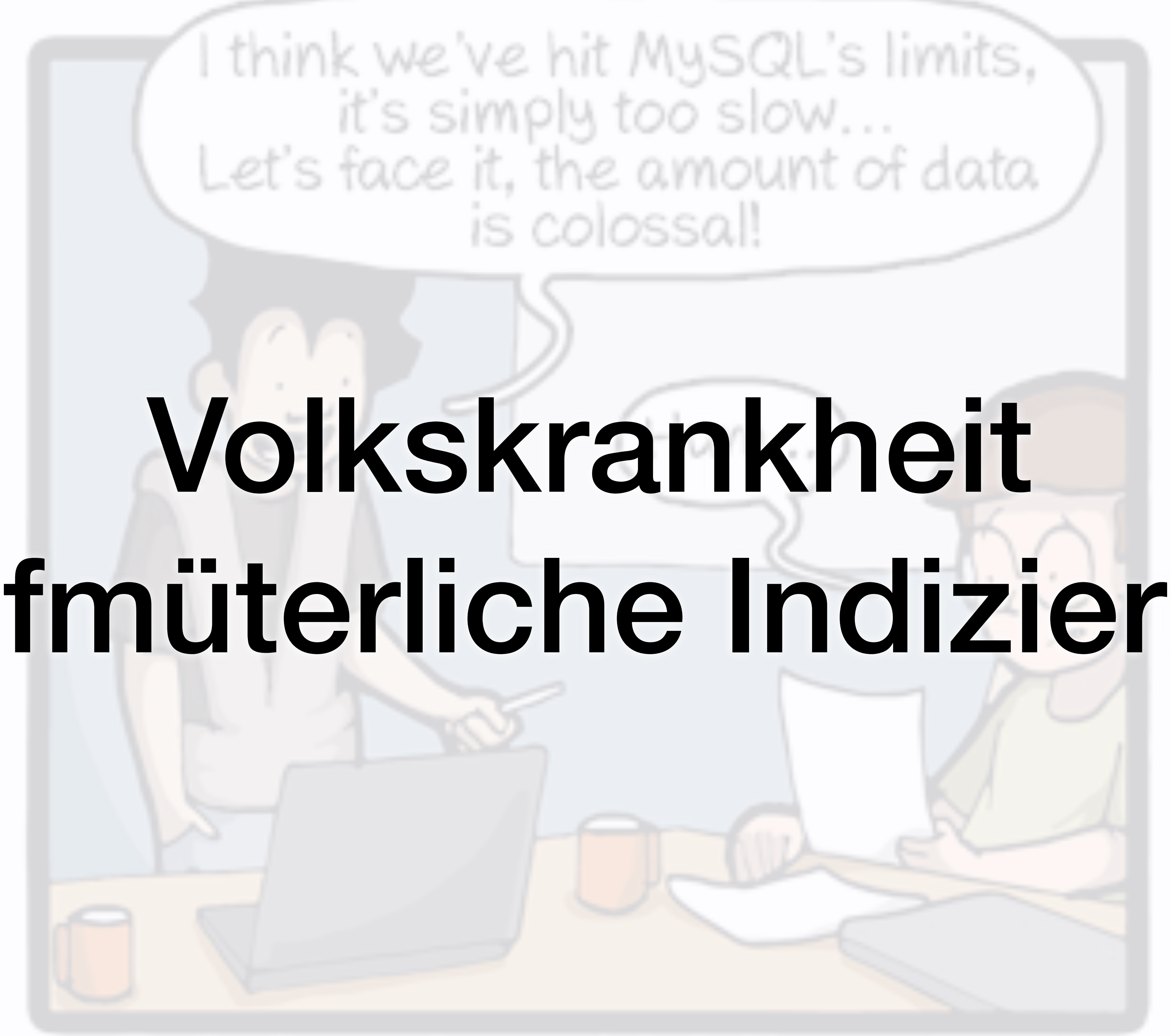
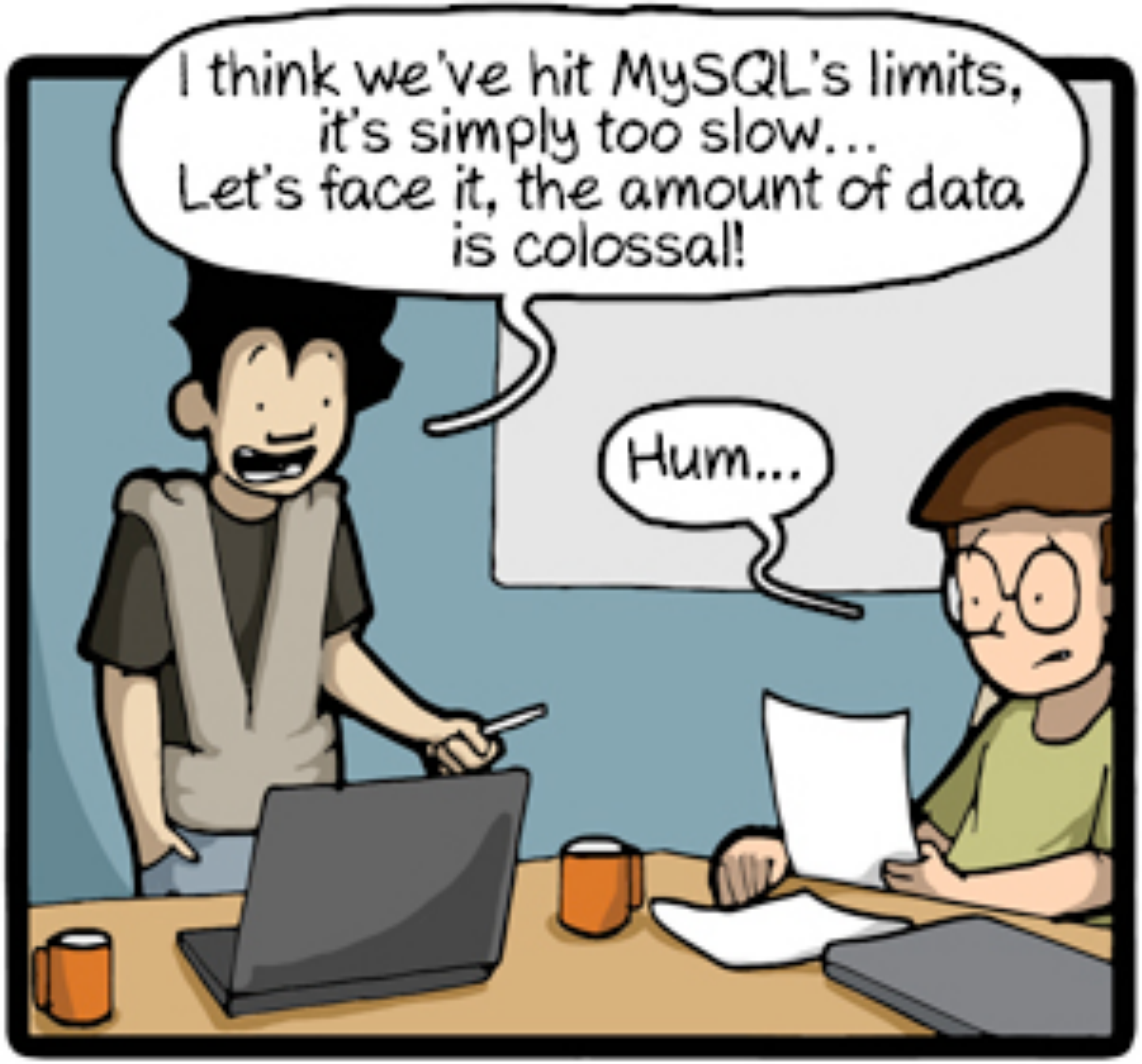

I think we've hit MySQL's limits, it's simply too slow... Let's face it, the amount of data is colossal!

Volkskrankheit “stiefmüterliche Indizierung”

@MarkusWinand

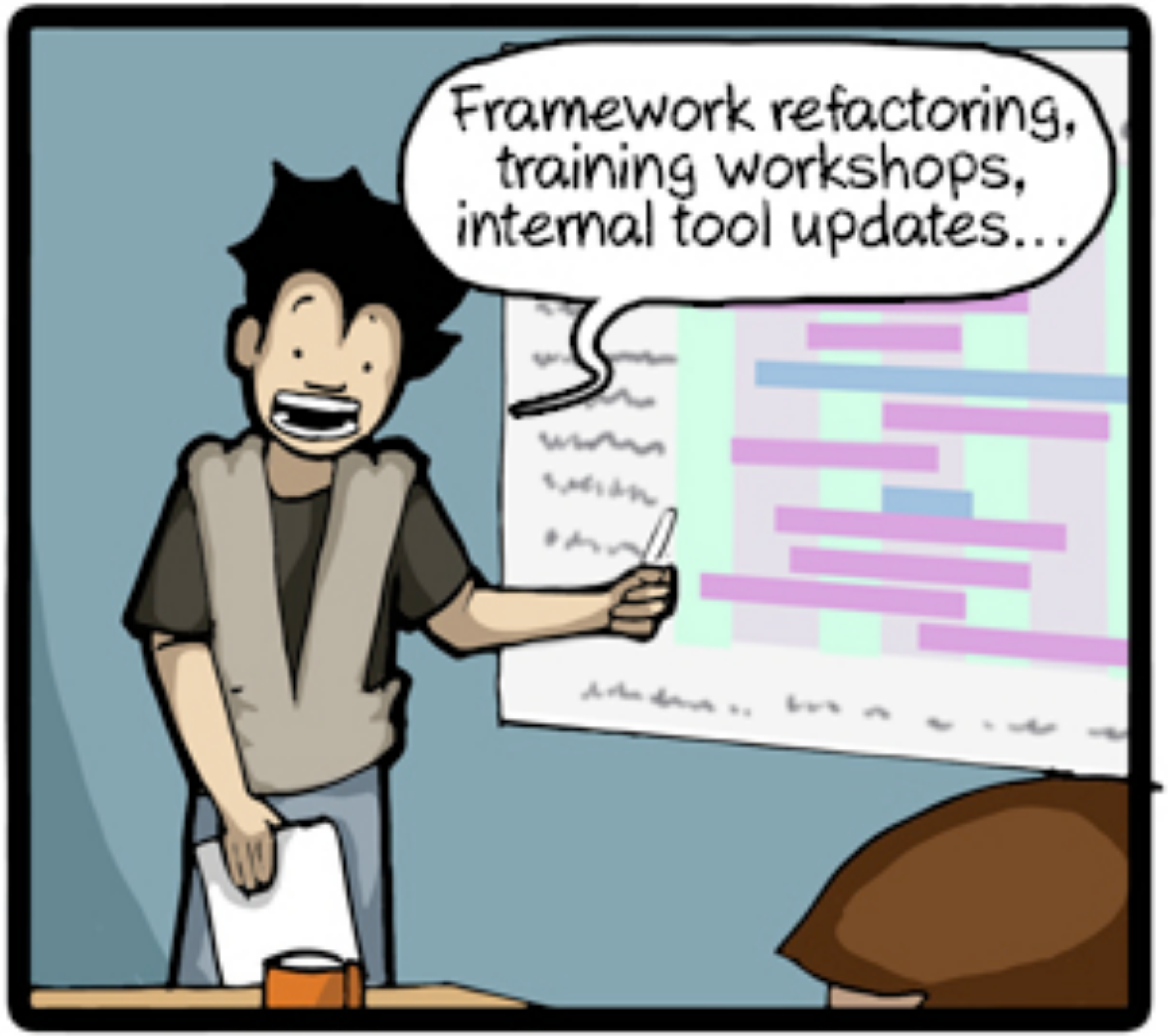
http://www.commitstrip.com/en/2014/06/03/the-problem-is-not-the-tool-itself/


I think we've hit MySQL's limits, it's simply too slow... Let's face it, the amount of data is colossal!
Hum...

http://www.commitstrip.com/en/2014/06/03/the-problem-is-not-the-tool-itself/

So I was thinking we should switch to NoSQL. I've prepared a 3-month battle plan. Here, look!

http://www.commitstrip.com/en/2014/06/03/the-problem-is-not-the-tool-itself/


Framework refactoring, training workshops, internal tool updates...

http://www.commitstrip.com/en/2014/06/03/the-problem-is-not-the-tool-itself/

http://www.commitstrip.com/en/2014/06/03/the-problem-is-not-the-tool-itself/

http://www.commitstrip.com/en/2014/06/03/the-problem-is-not-the-tool-itself/

Here you go. It's called an index, and it just made your queries 20 times faster...
Dude, missing indexes? Seriously...
CommitStrip.com

http://www.commitstrip.com/en/2014/06/03/the-problem-is-not-the-tool-itself/

Takeaway #1: Pandemic Scale

It affects you!

(Symbolic image; not real data)

http://upload.wikimedia.org/wikipedia/commons/c/c7/2009_world_subdivisions_flu_pandemic.png

Takeaway #2: Caused by Success

Takeaway #3: It’s Not Your Fault

http://simpsonswiki.com/wiki/File:I_Didn%27t_Do_It!_Volume_III.png

The Problem

Index/Query Mismatch

The Problem: Index/Query Mismatch

> "A very common cause of performance problems is <u>lack of proper indexes</u> or the use of <u>queries that are not using existing indexes</u>."
>
> —Buda Consulting
>
> http://www.budaconsulting.com/Portals/52677/docs/top_5_tech_brief.pdf

The Problem: Index/Query Mismatch

"A **very common** cause of performance problems is lack of proper indexes or the use of queries that are not using existing indexes."

—Buda Consulting

http://www.budaconsulting.com/Portals/52677/docs/top_5_tech_brief.pdf

Quantifying the Problem

Quantifying the Problem

Percona White Paper:

Reasons of performance problems that caused production downtime:

http://www.percona.com/files/white-papers/causes-of-downtime-in-mysql.pdf

Quantifying the Problem

Percona White Paper:

Reasons of performance problems that caused production downtime:

38% bad SQL

http://www.percona.com/files/white-papers/causes-of-downtime-in-mysql.pdf

Quantifying the Problem

Percona White Paper:

Reasons of performance problems that caused production downtime:

38% bad SQL

15% schema and indexing

http://www.percona.com/files/white-papers/causes-of-downtime-in-mysql.pdf

Quantifying the Problem

Survey by sqlskills.com:

Root causes of the last few SQL Server performance problems:

http://www.sqlskills.com/blogs/paul/survey-what-are-the-most-common-causes-of-performance-problems/

Quantifying the Problem

Survey by sqlskills.com:

Root causes of the last few SQL Server performance problems:

27% *T-SQL*

http://www.sqlskills.com/blogs/paul/survey-what-are-the-most-common-causes-of-performance-problems/

Quantifying the Problem

Survey by sqlskills.com:

Root causes of the last few SQL Server performance problems:

27% T-SQL

19% Poor indexing

http://www.sqlskills.com/blogs/paul/survey-what-are-the-most-common-causes-of-performance-problems/

Quantifying the Problem

Craig S. Mullins (strategist and researcher):

„As much as 75% of poor relational performance is caused by "bad" SQL and application code."

Quantifying the Problem

Craig S. Mullins (strategist and researcher):

„As much as 75% of poor relational performance is caused by "bad" SQL and application code."

Noel Yuhanna (Forrester Research):

„The key difficulties surrounding performance continue to be poorly written SQL statements, improper DBMS configuration and a lack of clear understanding of how to tune databases to solve performance issues."

Quantifying the Problem

My observation:

Quantifying the Problem

My observation:

~50% of SQL performance problems are caused by improper index use

The Root Cause

The Root Cause

Indexing is a afterthought…

The Root Cause

Indexing is a afterthought…

...often done by the wrong people

The Root Cause: DBAs are Indexing

How did databases work before SQL?

The Root Cause: DBAs are Indexing

Index use was intrinsically tied to the queries.

The Root Cause: DBAs are Indexing

Example: dBase

The Root Cause: DBAs are Indexing

Example: dBase

Developers had to...

...use indexes explicitly when searching:

```
set index to last_name
find Winand
```

The Root Cause: DBAs are Indexing

Example: dBase

Developers had to...

...use indexes explicitly when searching:

```
set index to last_name
find Winand
```

...take care of index maintenance:

```
set index to last_name, idx2
append
```

The Root Cause: DBAs are Indexing

SQL is an abstraction that only defines the logical view.

The actual SQL implementation takes care of everything else.

The Root Cause: DBAs are Indexing

SQL (language)
has:

SQL Databases (software)
have:

The Root Cause: DBAs are Indexing

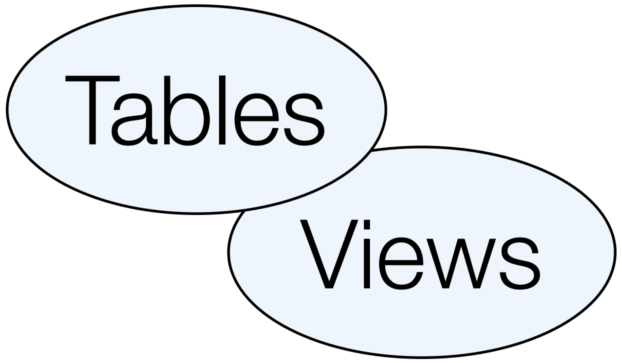
SQL (language)
has:

Tables

SQL Databases (software)
have:

The Root Cause: DBAs are Indexing

SQL (language)
has:



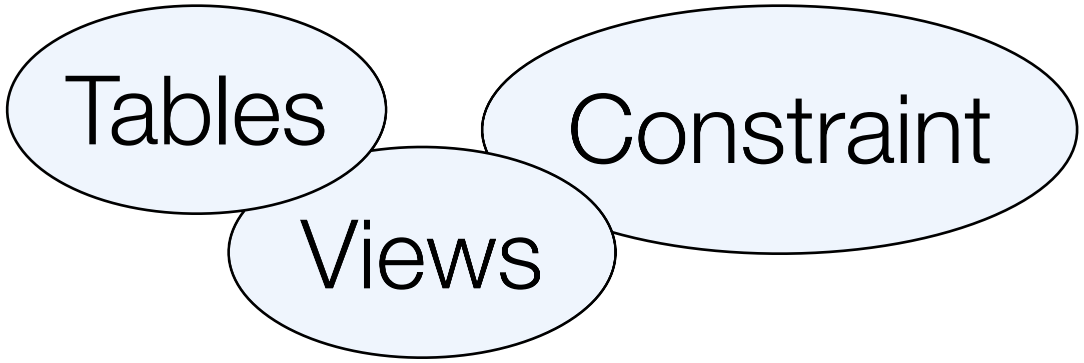
Tables

Views

SQL Databases (software)
have:

The Root Cause: DBAs are Indexing

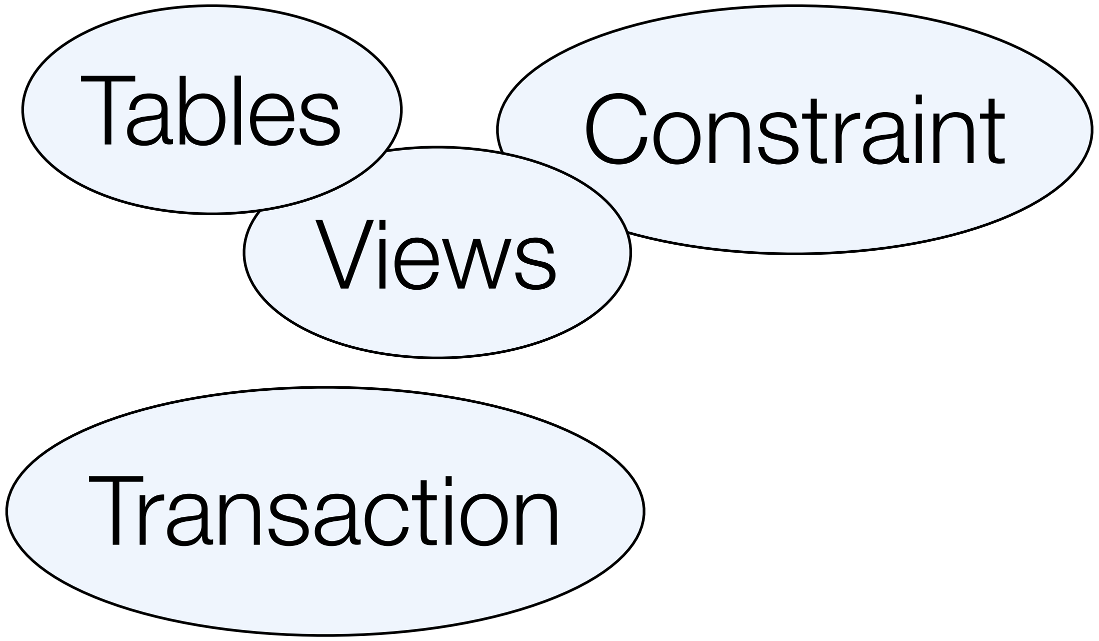
SQL (language)
has:



SQL Databases (software)
have:

The Root Cause: DBAs are Indexing

SQL (language)
has:



Tables

Constraint

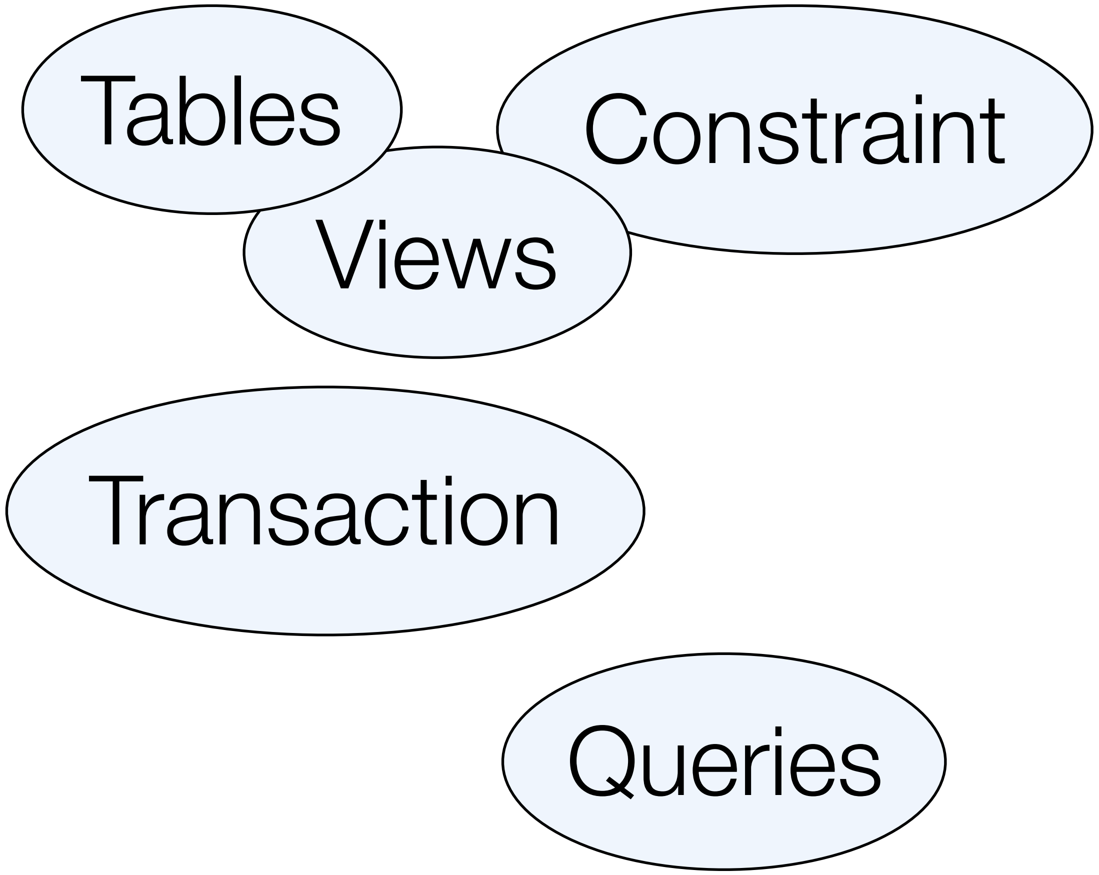
Views

Transaction

SQL Databases (software)
have:

The Root Cause: DBAs are Indexing

SQL (language)
has:



Tables

Constraint

Views

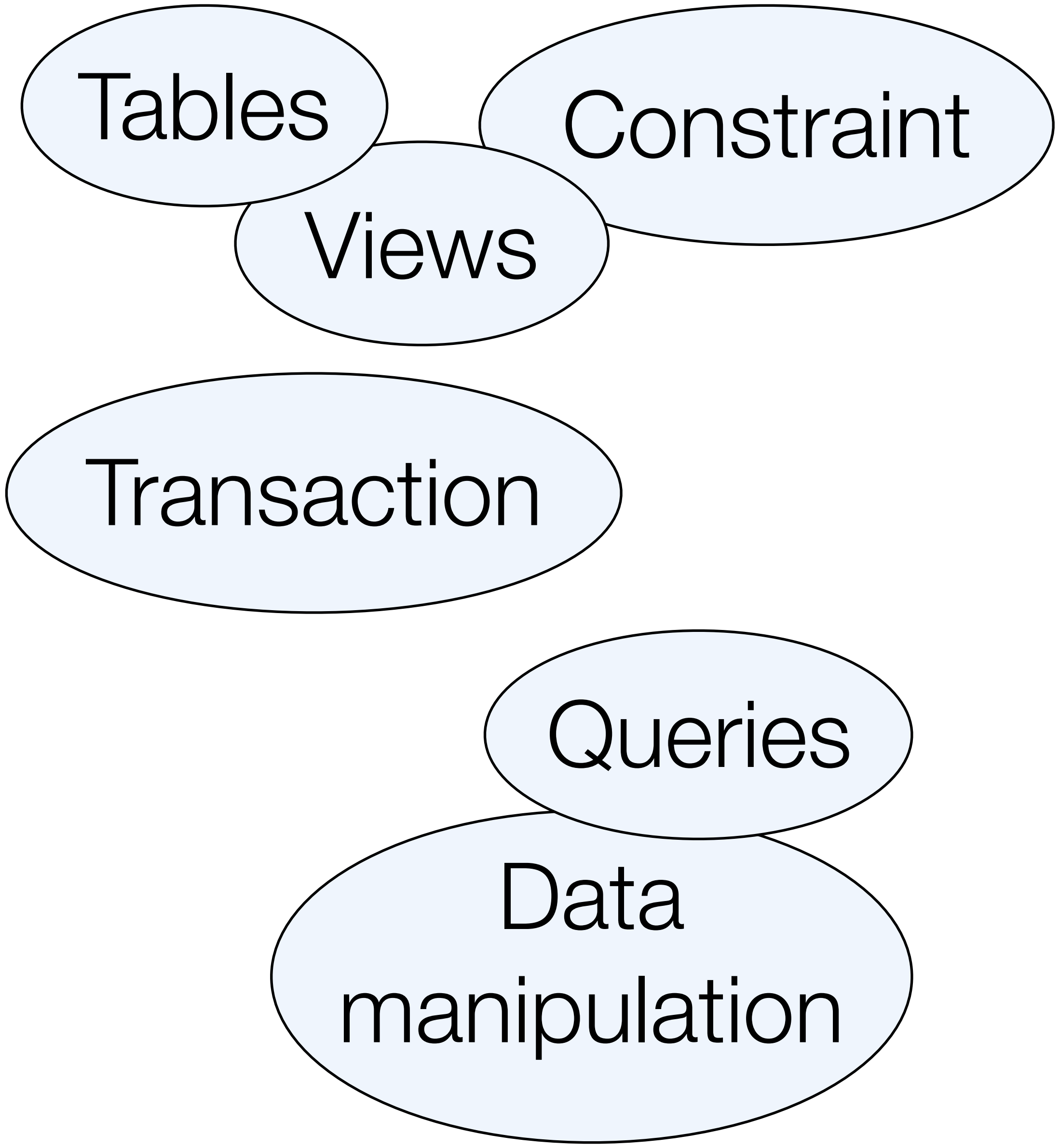
Transaction

Queries

SQL Databases (software)
have:

The Root Cause: DBAs are Indexing

SQL (language)
has:



Tables

Constraint

Views

Transaction

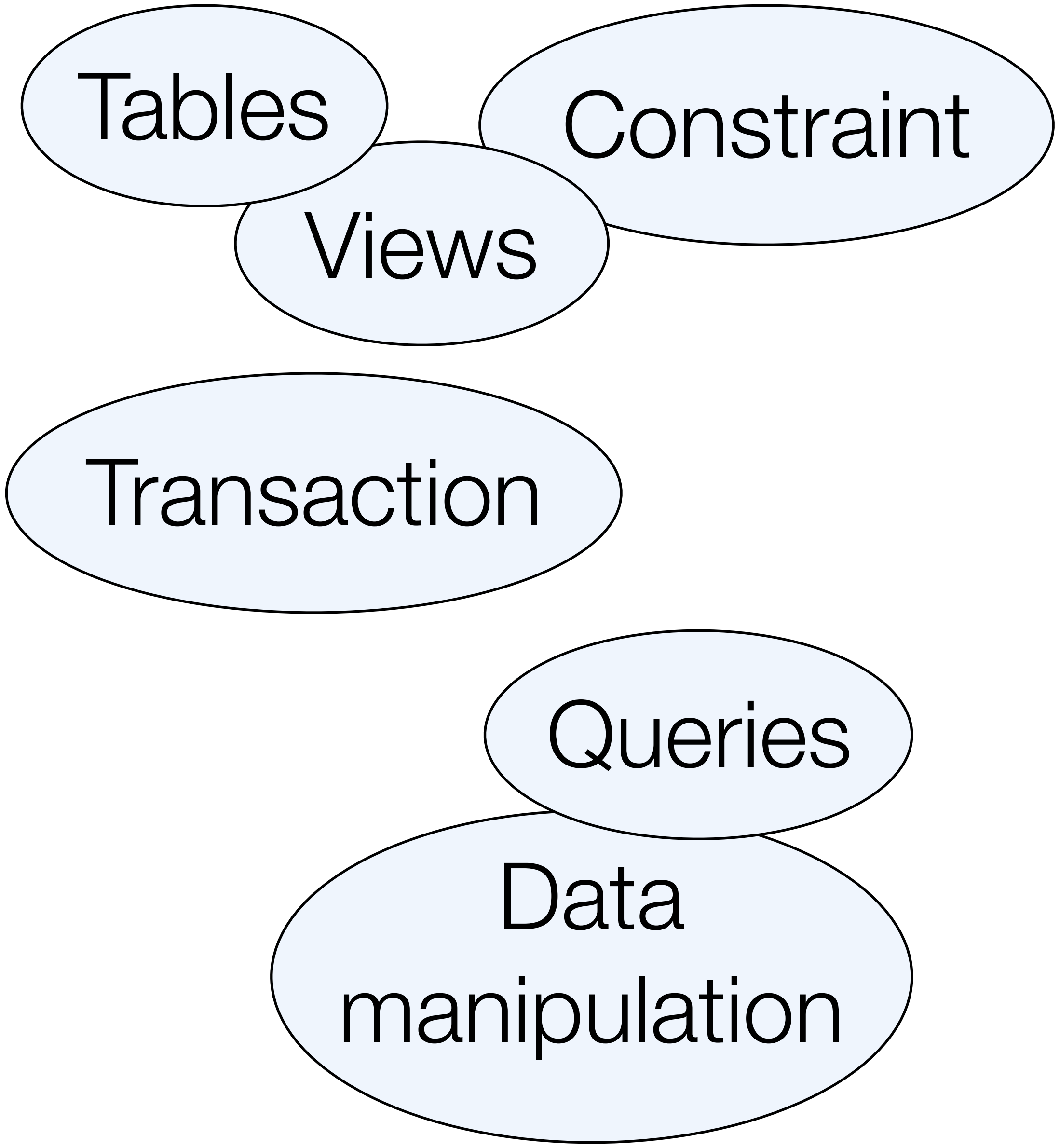
Queries

Data
manipulation

SQL Databases (software)
have:

The Root Cause: DBAs are Indexing

SQL (language) has:



Tables

Constraint

Views

Transaction

Queries

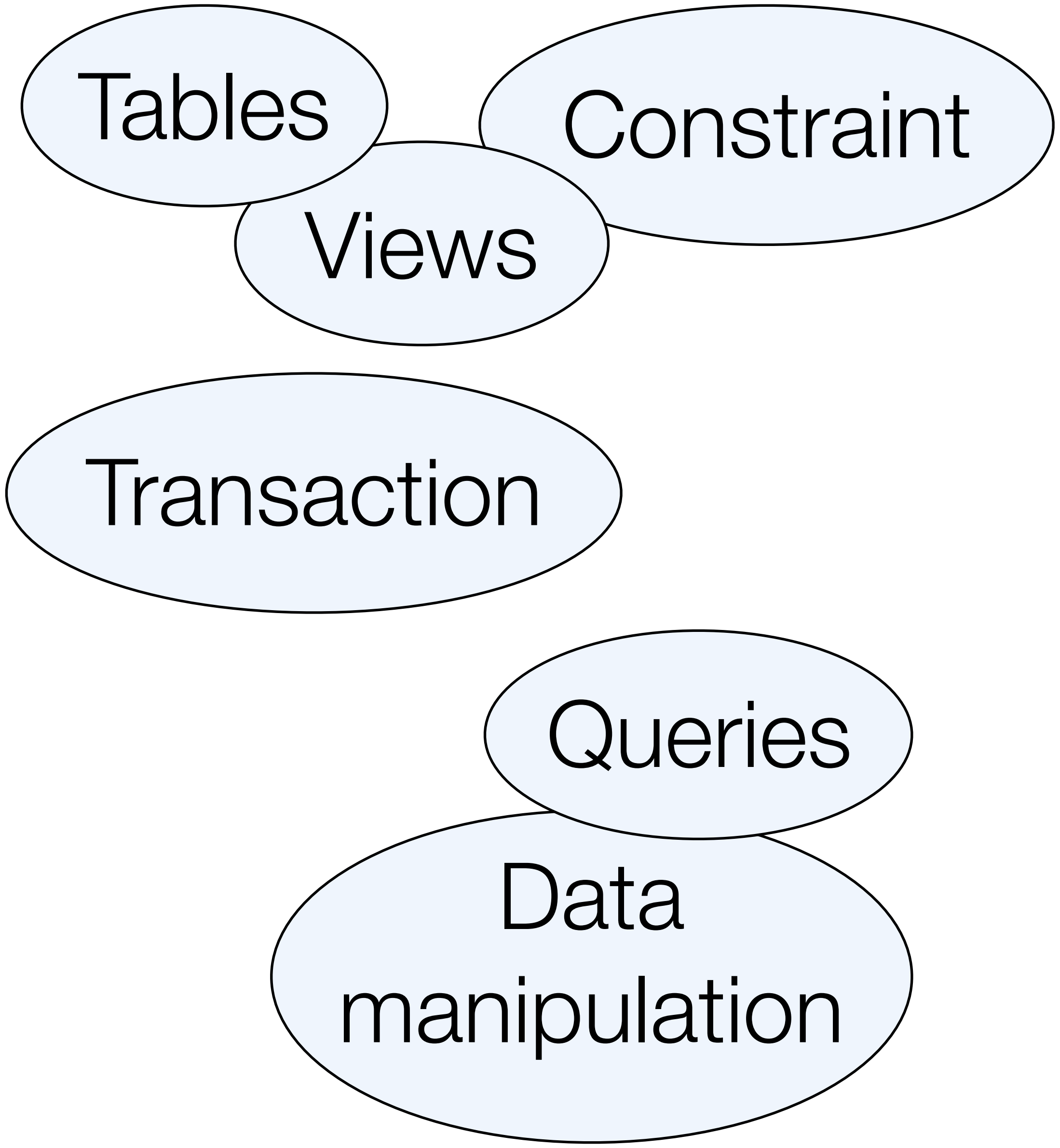
Data manipulation

SQL Databases (software) have:

Storage management

The Root Cause: DBAs are Indexing

SQL (language) has:



Tables

Constraint

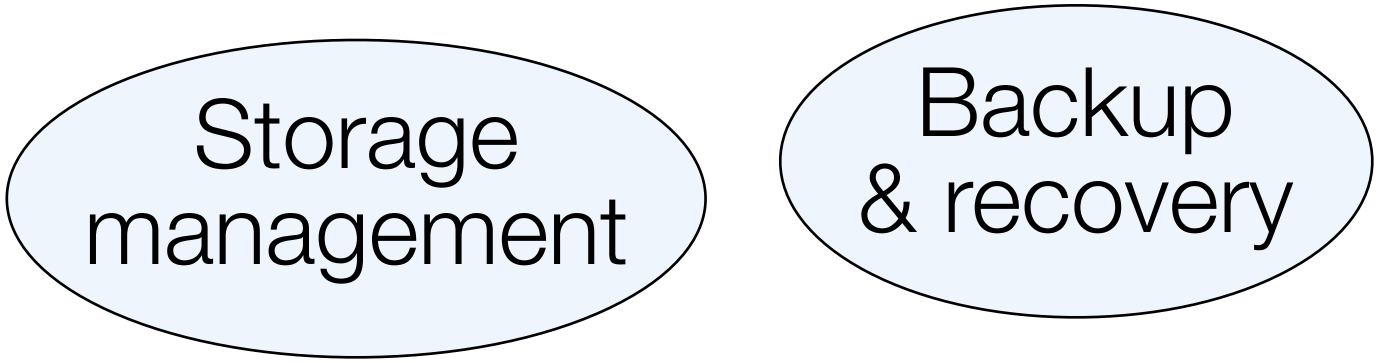
Views

Transaction

Queries

Data manipulation

SQL Databases (software) have:

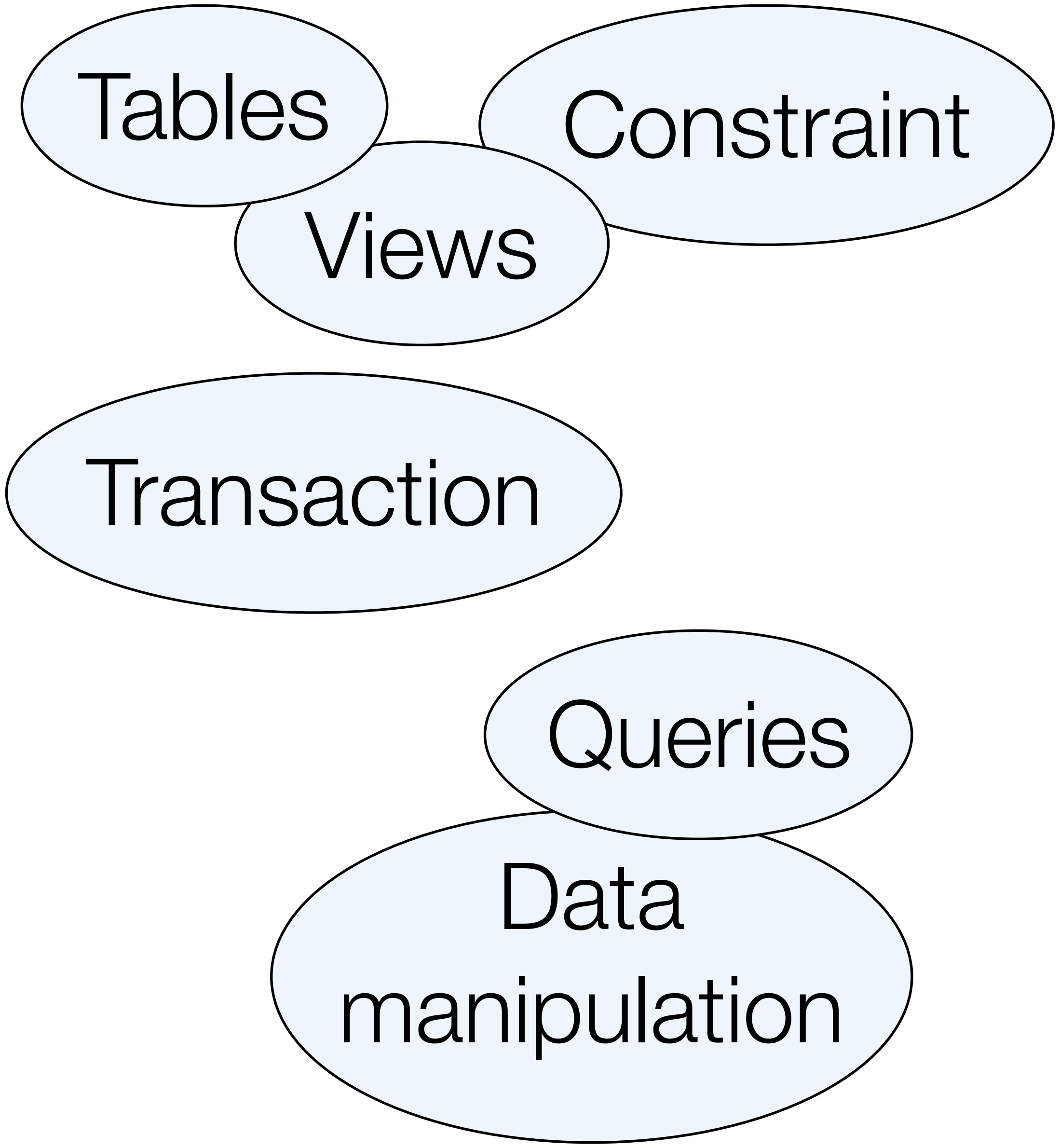


Storage management

Backup & recovery

The Root Cause: DBAs are Indexing

SQL (language) has:



Tables

Constraint

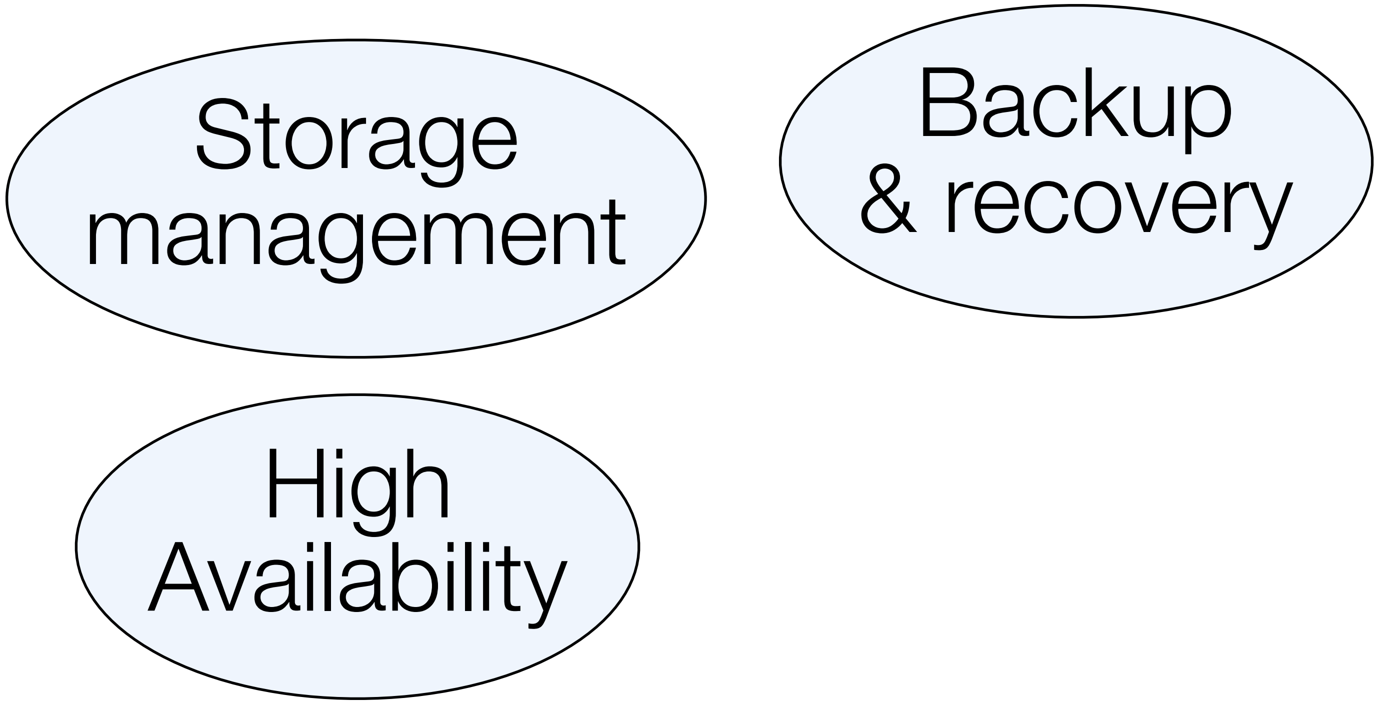
Views

Transaction

Queries

Data manipulation

SQL Databases (software) have:



Storage management

Backup & recovery

High Availability

The Root Cause: DBAs are Indexing

SQL (language) has:

- Tables
- Constraint
- Views
- Transaction
- Queries
- Data manipulation

SQL Databases (software) have:

- Storage management
- Backup & recovery
- High Availability
- Bugs & patches

The Root Cause: DBAs are Indexing

SQL (language) has:

- Tables
- Constraint
- Views
- Transaction
- Queries
- Data manipulation

SQL Databases (software) have:

- Storage management
- Backup & recovery
- High Availability
- Bugs & patches
- Tuning parameters

The Root Cause: DBAs are Indexing

SQL (language) has:

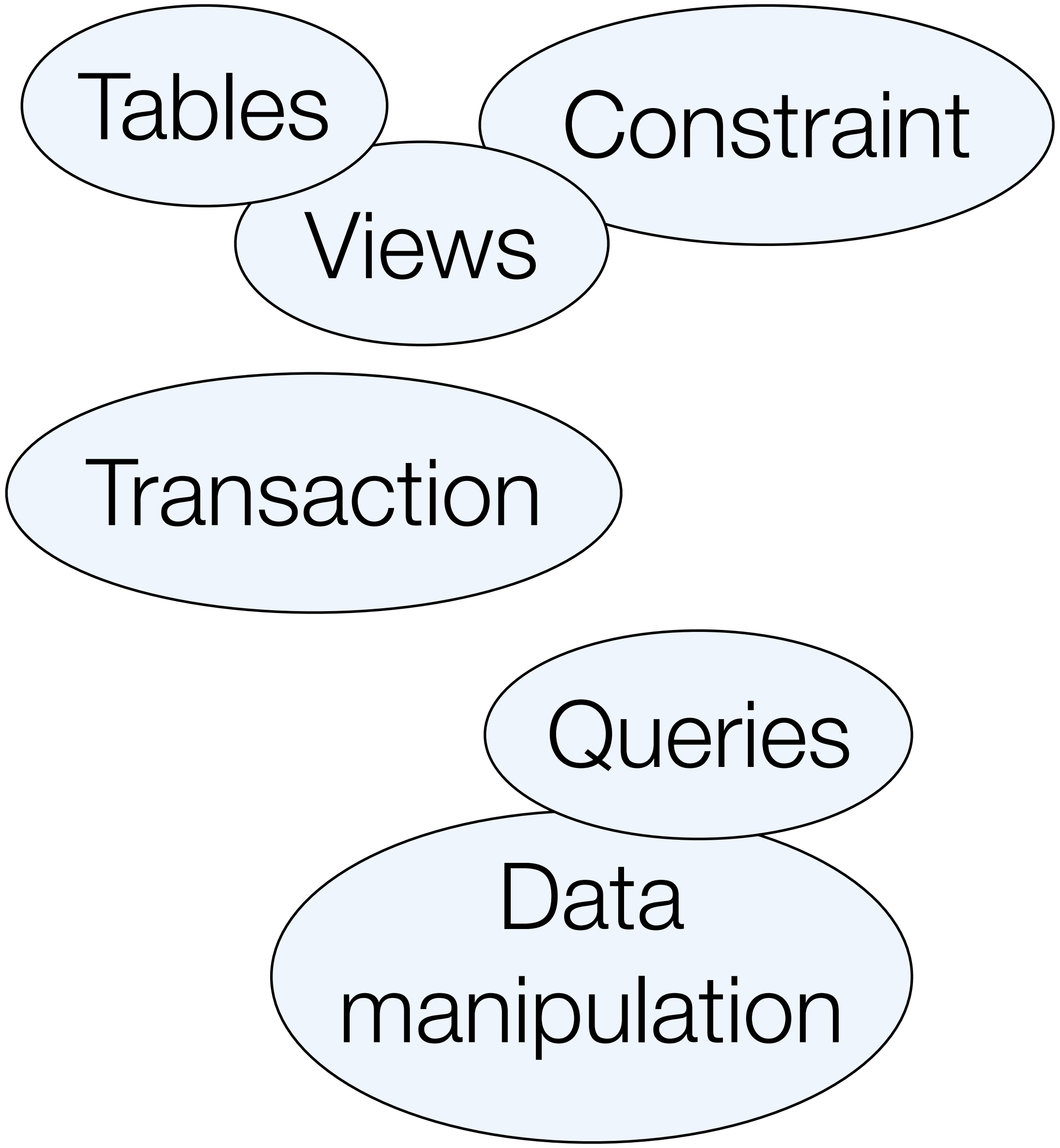
- Tables
- Constraint
- Views
- Transaction
- Queries
- Data manipulation

SQL Databases (software) have:

- Storage management
- Backup & recovery
- High Availability
- Bugs & patches
- Indexes
- Tuning parameters

The Root Cause: DBAs are Indexing

Developers



Tables

Constraint

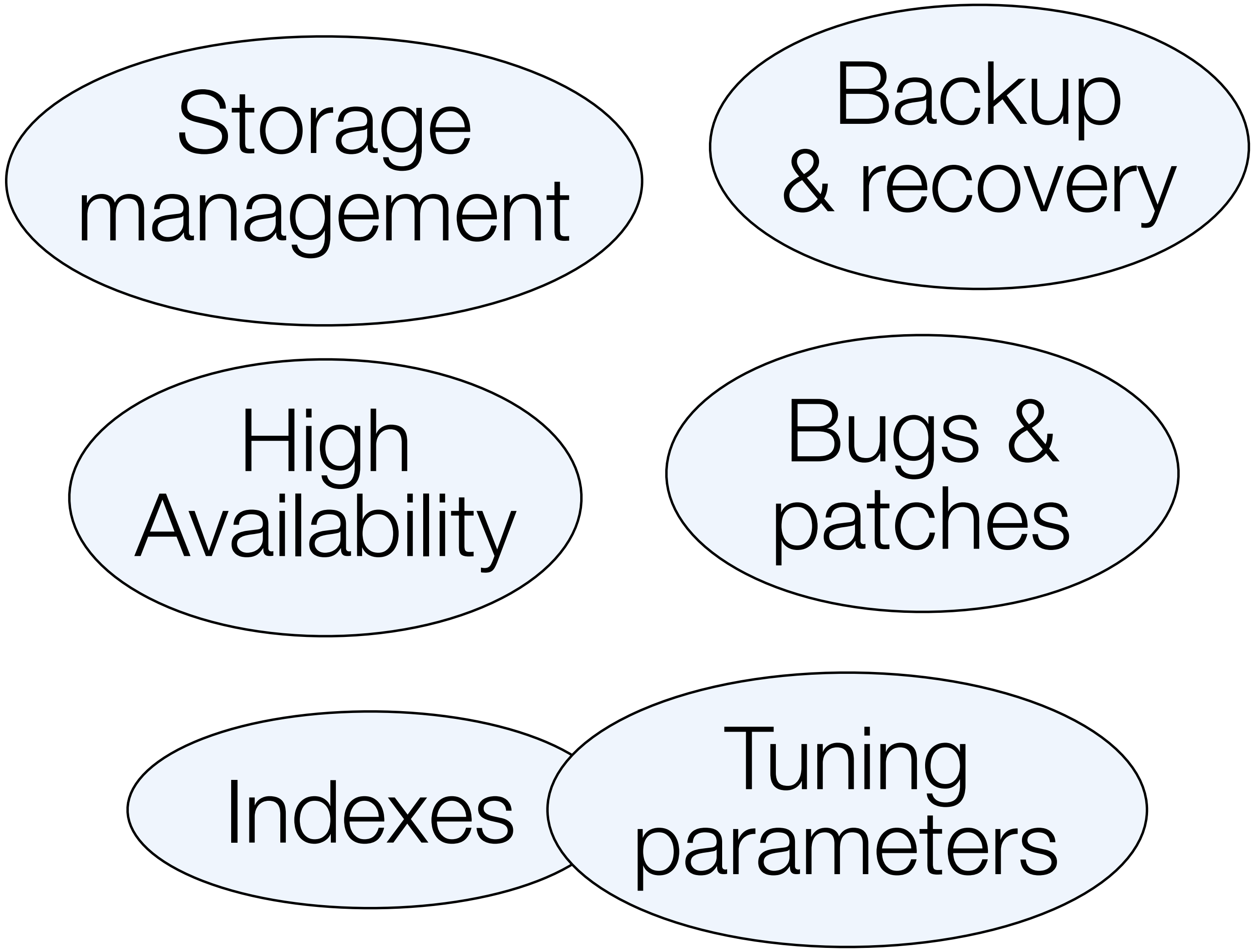
Views

Transaction

Queries

Data manipulation

SQL Databases (software) have:



Storage management

Backup & recovery

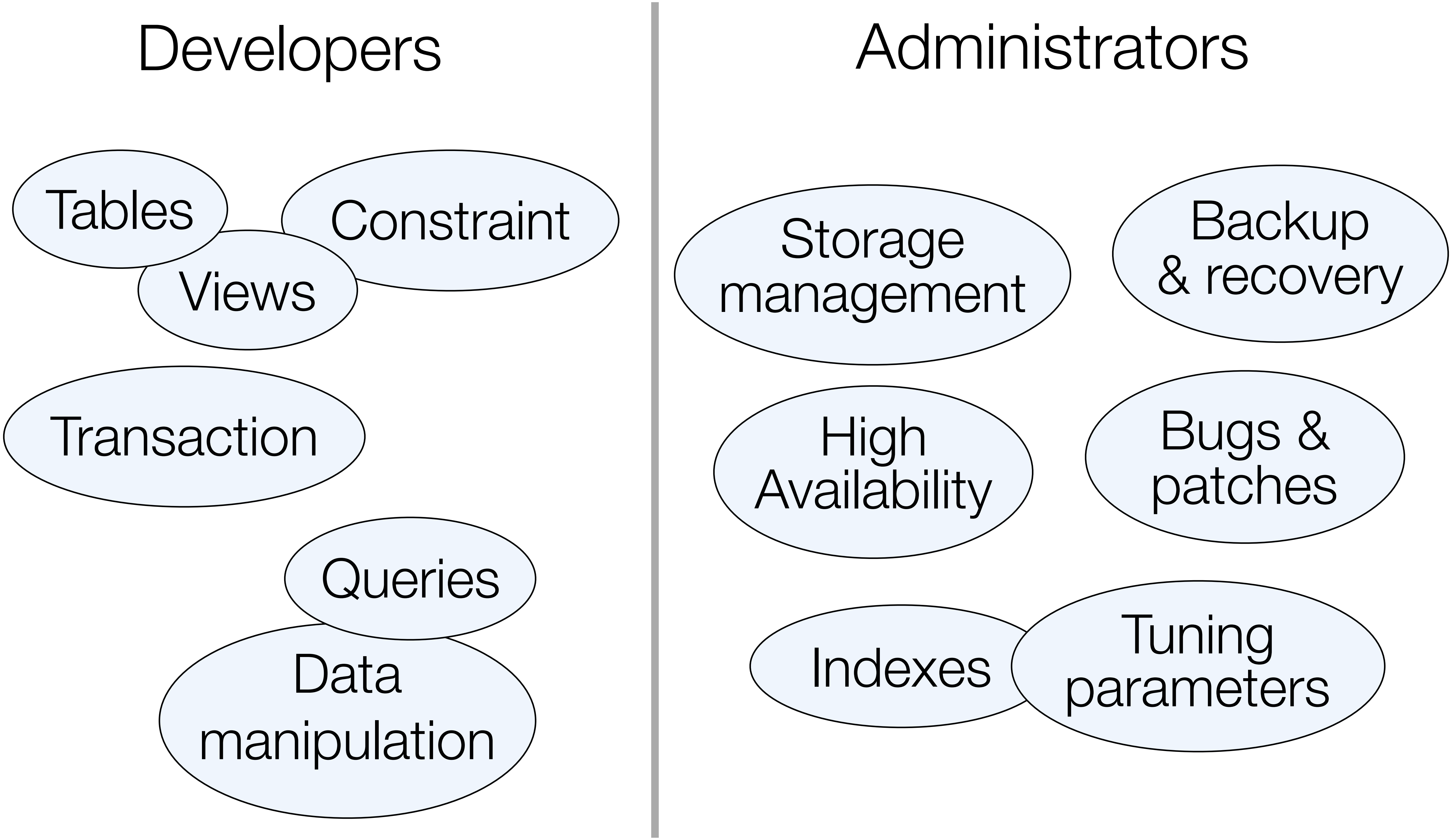
High Availability

Bugs & patches

Indexes

Tuning parameters

The Root Cause: DBAs are Indexing



Developers

Tables

Constraint

Views

Transaction

Queries

Data manipulation

Administrators

Storage management

Backup & recovery

High Availability

Bugs & patches

Indexes

Tuning parameters

The Root Cause: DBAs are Indexing

Today, indexing is often considered a tuning task that belongs to the administrators responsibilities.

The Root Cause: DBAs are Indexing

A misconception that causes new problems:

The Root Cause: DBAs are Indexing

A misconception that causes new problems:

DBAs don’t know
the queries

The Root Cause: DBAs are Indexing

A misconception that causes new problems:

DBAs don’t know
the queries

Have to “investigate”
to find the queries.

The Root Cause: DBAs are Indexing

A misconception that causes new problems:

DBAs don't know the queries

Have to "investigate" to find the queries.

by G-10gian82
deviantart.com

The Root Cause: DBAs are Indexing

A misconception that causes new problems:

DBAs don't know the queries

Have to "investigate" to find the queries.

It is time consuming and almost always incomplete.

by G-10gian82
deviantart.com

The Root Cause: DBAs are Indexing

A misconception that causes new problems:

DBAs don’t know
the queries

Have to “investigate”
to find the queries.

It is time consuming and
almost always incomplete.

The Root Cause: DBAs are Indexing

A misconception that causes new problems:

DBAs don’t know the queries

Have to “investigate” to find the queries.

It is time consuming and almost always incomplete.

DBAs can’t change the queries

The Root Cause: DBAs are Indexing

A misconception that causes new problems:

DBAs don’t know the queries

Have to “investigate” to find the queries.

It is time consuming and almost always incomplete.

DBAs can’t change the queries

Can make the index match the query.

The Root Cause: DBAs are Indexing

A misconception that causes new problems:

DBAs don't know the queries

Have to "investigate" to find the queries.

It is time consuming and almost always incomplete.

DBAs can't change the queries

Can make the index match the query.

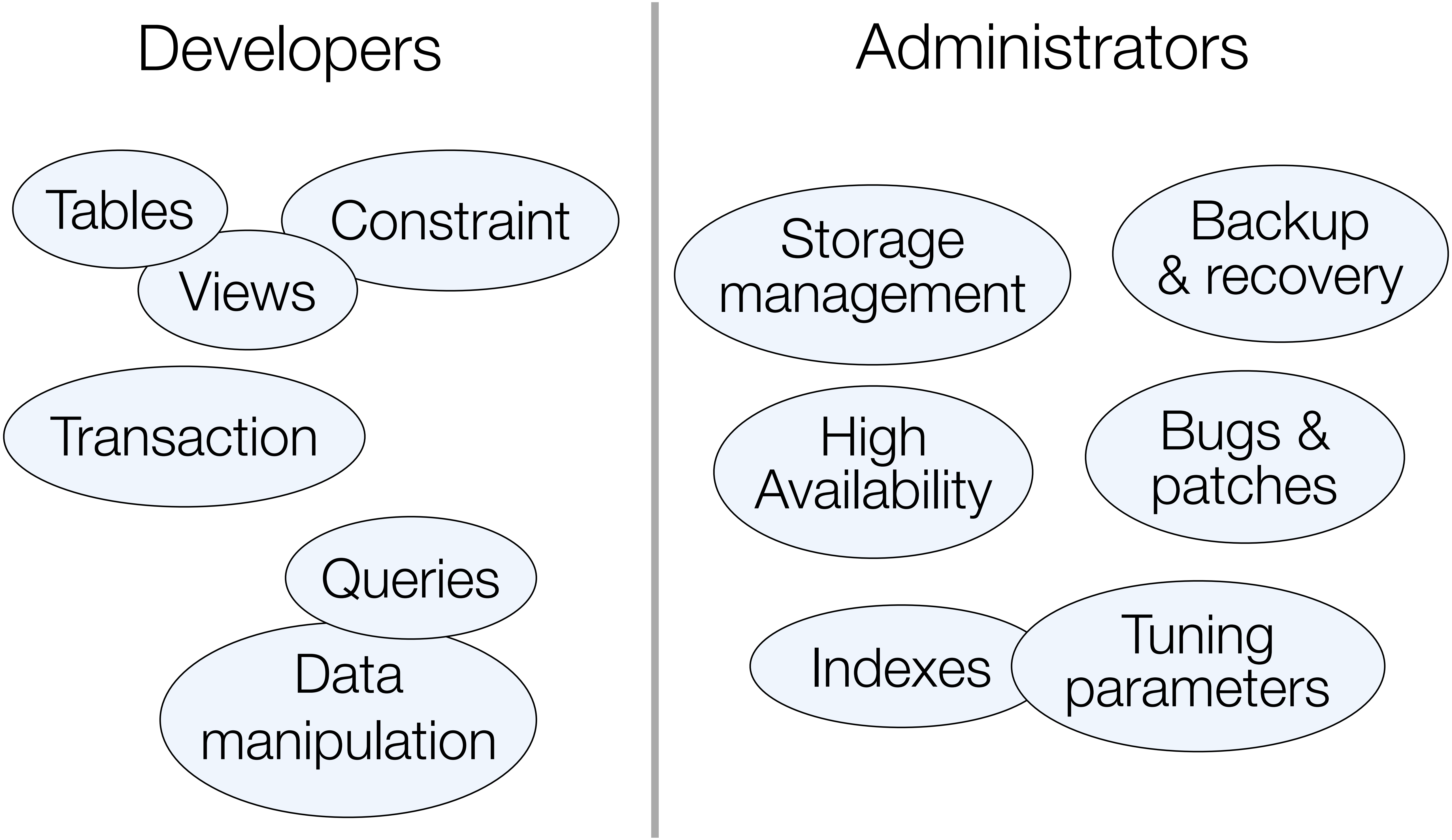
Can't make the query match the index!

The Solution

The Solution

Indexing is a
Development Task

The Solution: It’s a Dev Task



Developers

Tables

Constraint

Views

Transaction

Queries

Data manipulation

Administrators

Storage management

Backup & recovery

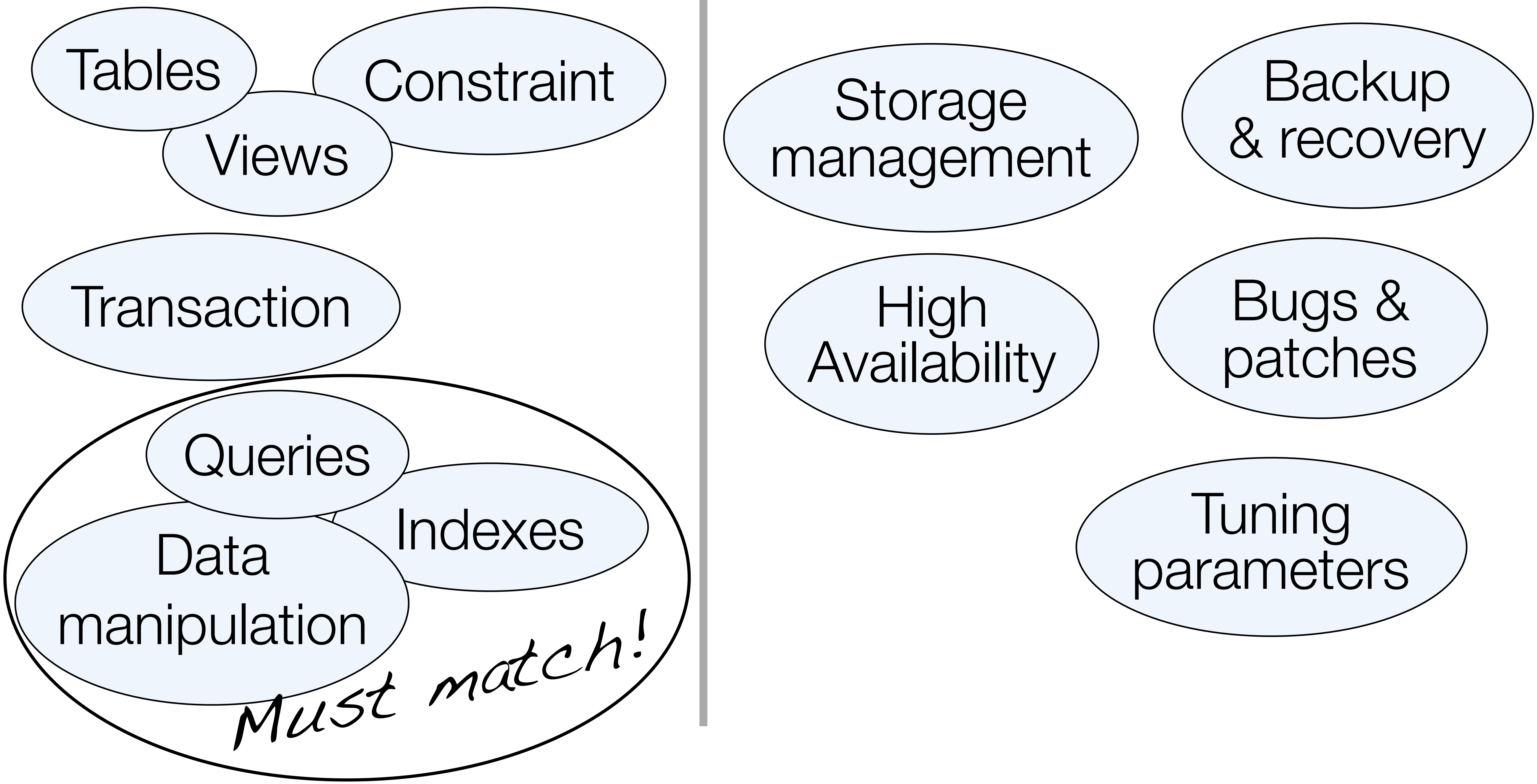
High Availability

Bugs & patches

Indexes

Tuning parameters

The Solution: It’s a Dev Task



Developers

Tables

Constraint

Views

Transaction

Queries

Indexes

Data manipulation

Must match!

Administrators

Storage management

Backup & recovery

High Availability

Bugs & patches

Tuning parameters

Another Problem: It’s not Taught

Another Problem: It’s not Taught

Indexes are not part of the pure SQL (language) literature because indexes are not part of the SQL standard.

Another Problem: It’s not Taught

Indexes are not part of the pure SQL (language) literature because indexes are not part of the SQL standard.

11 SQL books analyzed: only **1.0%** of the pages are about indexes (70 out of 7330 pages).

Another Problem: It’s not Taught

Indexes are not part of the pure SQL (language) literature because indexes are not part of the SQL standard.

11 SQL books analyzed: only **1.0%** of the pages are about indexes (70 out of 7330 pages).

Examples:

Oracle SQL by Example: **2.0%** (19/960)
Beginning DBs with PostgreSQL: **0.8%** (5/664)
Learning SQL: **3.3%** (11/336 — highest rate in class)

Another Problem: It's not Taught

Proper index usage is sometimes covered in database tuning books but is always buried between hundreds of pages of HW, OS and DB parameterization topics.

Another Problem: It's not Taught

Proper index usage is sometimes covered in database tuning books but is always buried between hundreds of pages of HW, OS and DB parameterization topics.

14 database administration books analyzed: **6%** of the pages are about indexes (395 out of 6568 pages).

Another Problem: It's not Taught

Proper index usage is sometimes covered in database tuning books but is always buried between hundreds of pages of HW, OS and DB parameterization topics.

14 database administration books analyzed: **6%** of the pages are about indexes (395 out of 6568 pages).

Examples:

- Oracle Performance Survival Guide: **5.2%** (38/730)
- High Performance MySQL: **8%** (55/684)
- PostgreSQL 9 High Performance: **5.8%** (27/468)
- SQL Server 2012 Query Perf. Tuning: **19.6%** (98/499)

Another Problem: It's not Taught

Proper index usage is sometimes covered in database tuning books but is always buried between hundreds of pages of HW, OS and DB parameterization topics.

14 database administration books analyzed: **6%** of the pages are about indexes (395 out of 6568 pages).

Examples:

Oracle Performance Survival Guide: **5.2%** (38/730)
High Performance MySQL: **8%** (55/684)
PostgreSQL 9 High Performance: **5.8%** (27/468)
SQL Server 2012 Query Perf. Tuning: **19.6%** (98/499)

Another Problem: It’s not Taught

Another Problem: It’s not Taught

Consequence:

Developers don’t know how to use indexes properly.

Another Problem: It's not Taught

Consequence:

Developers don't know how to use indexes properly.

Results of the 3-minute online quiz:

https://use-the-index-luke.com/3-minute-test

5 questions: each about a specific
index usage pattern.

Non-representative!

3-Minute Quiz: Indexing Skills

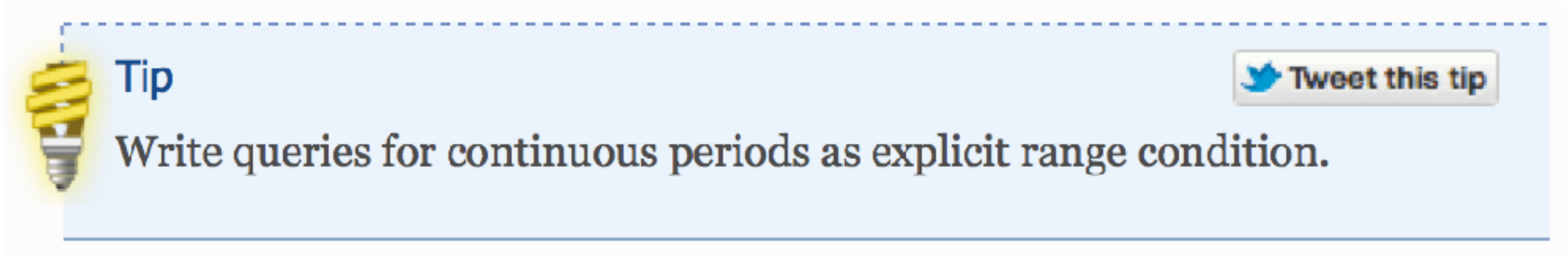
Q1: Good or Bad? *(Function use)*

```
CREATE INDEX tbl_idx ON tbl (date_column);

SELECT text, date_column
  FROM tbl
 WHERE EXTRACT(YEAR FROM date_column) = 2018;
```

3-Minute Quiz: Indexing Skills

Q1: Good or Bad? *(Function use)*

```
CREATE INDEX tbl_idx ON tbl (date_column);

SELECT text, date_column
  FROM tbl
 WHERE EXTRACT(YEAR FROM date_column) = 2018;
```



Write queries for continuous periods as explicit range condition.

http://use-the-index-luke.com/sql/where-clause/obfuscation/dates

3-Minute Quiz: Indexing Skills

```
...WHERE EXTRACT(YEAR FROM date_column) = 2018

      Seq Scan on tbl (rows=365)
         Rows Removed by Filter: 49635
      Total runtime: 118.796 ms
```

```
...WHERE date_column >= '2018-01-01'
       AND date_column <  '2019-01-01'

     Index Scan using tbl_idx on tbl (rows=365)
      Total runtime: 0.430 ms
```

(Above: simplified PostgreSQL execution plans when selecting 365 rows out of 50000)

3-Minute Quiz: Indexing Skills

```
...WHERE EXTRACT(YEAR FROM date_column) = 2018
```

```
Seq Scan on tbl (rows=365)
   Rows Removed by Filter: 49635
Total runtime: 118.796 ms
```

```
...WHERE               mn >= '2018-01-01'
     AND               mn <   '2019-01-01'
```

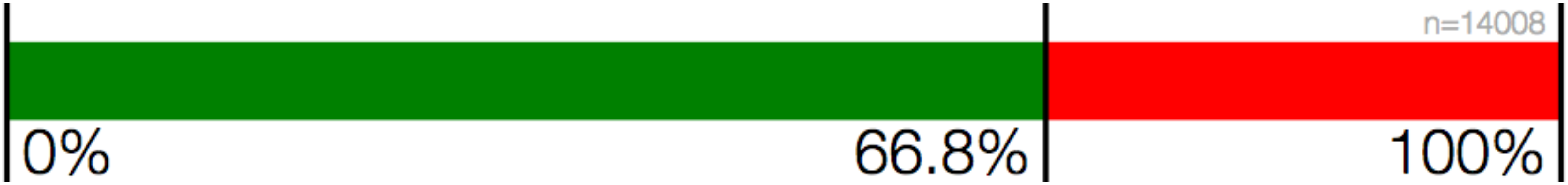
270x faster

```
Index Scan using tbl_idx on tbl (rows=365)
Total runtime: 0.430 ms
```

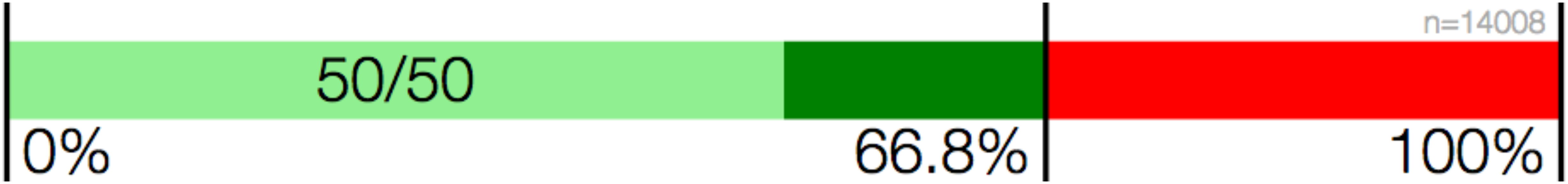
(Above: simplified PostgreSQL execution plans when selecting 365 rows out of 50000)

3-Minute Quiz: Q1 — Results

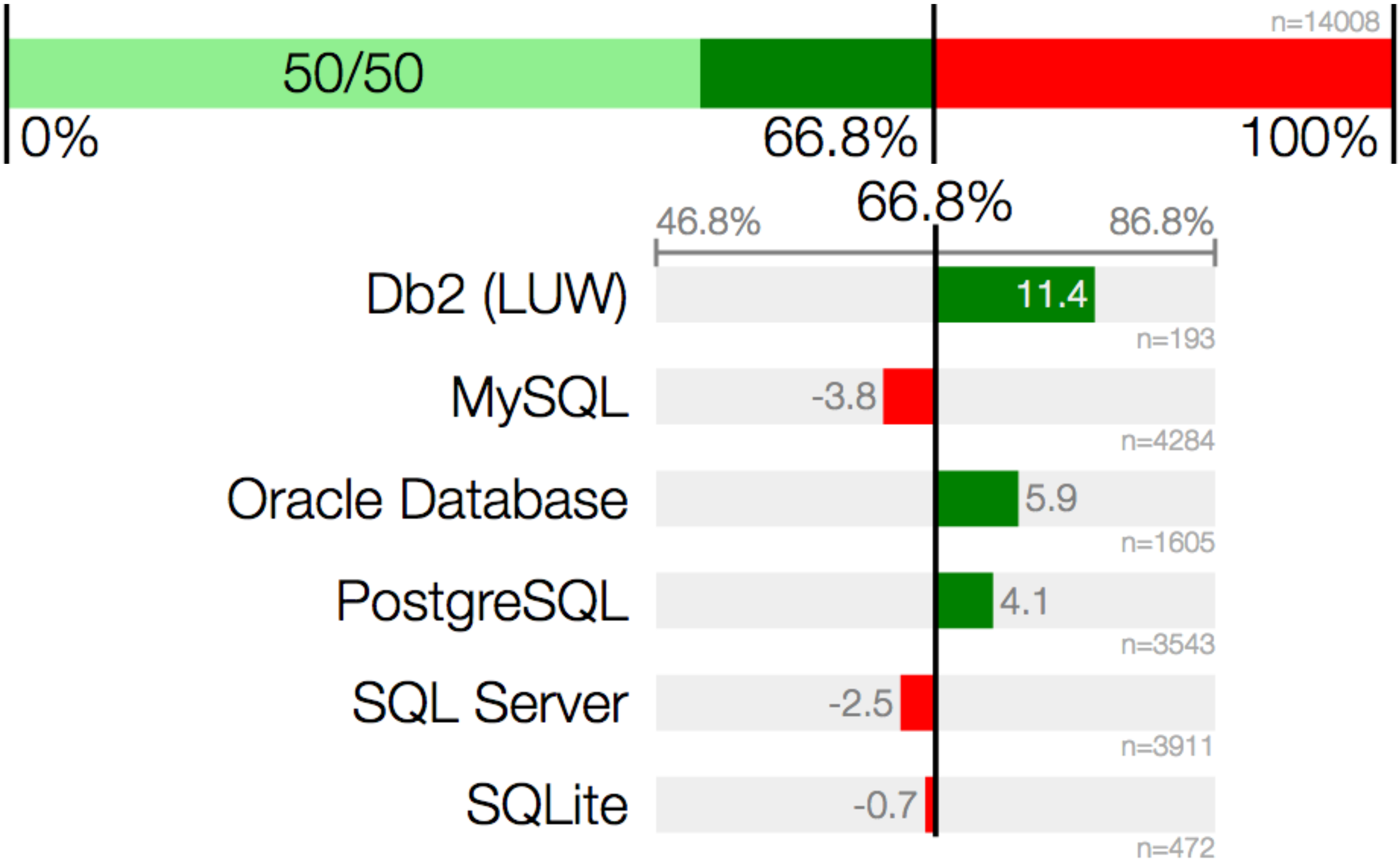
3-Minute Quiz: Q1 — Results



3-Minute Quiz: Q1 — Results



3-Minute Quiz: Q1 — Results



3-Minute Quiz: Indexing Skills

Q2: Good or Bad? *(Indexed Top-N, no IOS)*

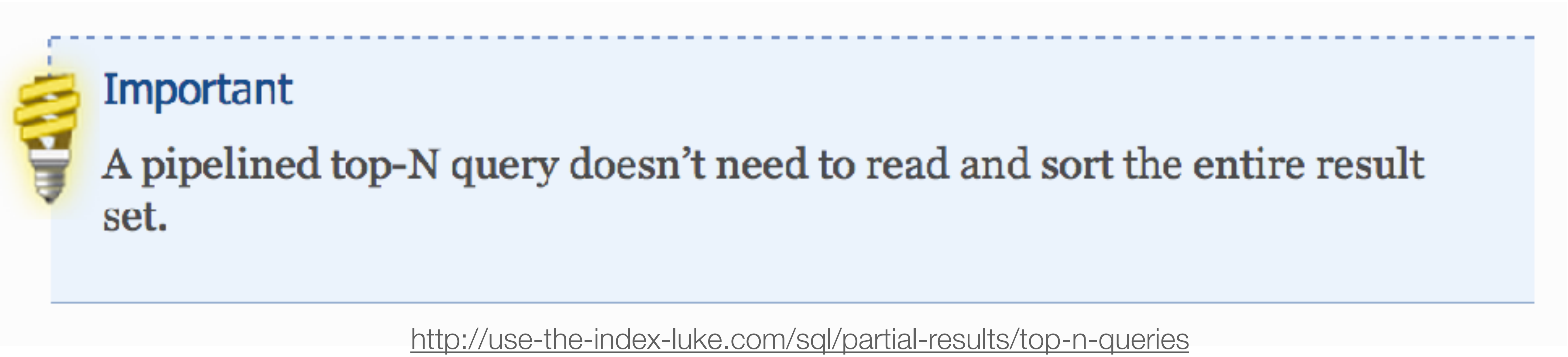
```
CREATE INDEX tbl_idx ON tbl (a, date_col);

SELECT id, a, date_col
  FROM tbl
 WHERE a = ?
 ORDER BY date_col DESC
 LIMIT 1;
```

http://use-the-index-luke.com/sql/partial-results/top-n-queries

3-Minute Quiz: Indexing Skills

Q2: Good or Bad? *(Indexed Top-N, no IOS)*

```
CREATE INDEX tbl_idx ON tbl (a, date_col);

SELECT id, a, date_col
  FROM tbl
 WHERE a = ?
 ORDER BY date_col DESC
 LIMIT 1;
```



Important

A pipelined top-N query doesn't need to read and sort the entire result set.

http://use-the-index-luke.com/sql/partial-results/top-n-queries

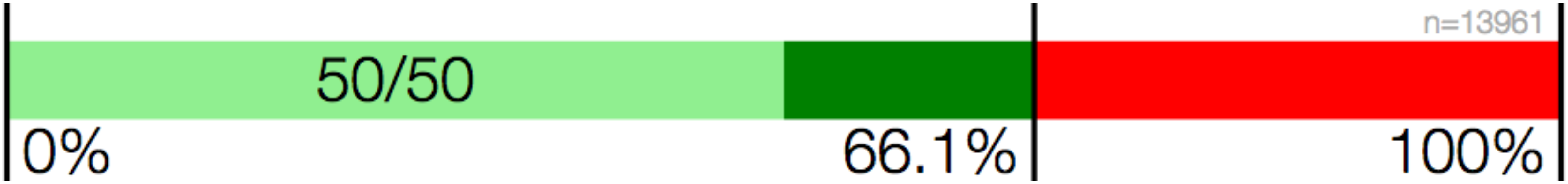
3-Minute Quiz: Indexing Skills

It is already the most optimal solution (not considering index-only scan).

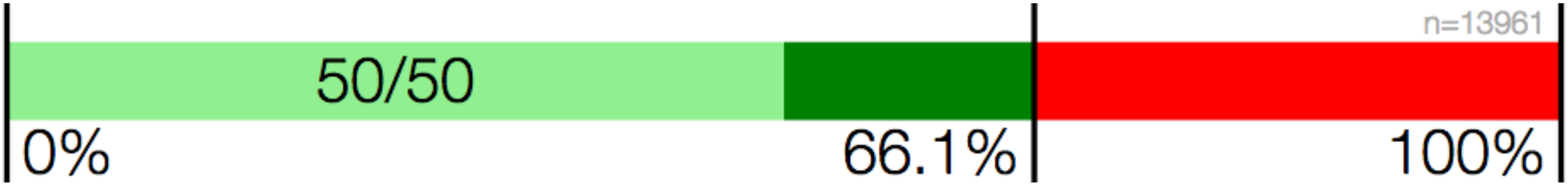
```
Limit (rows=1)
  -> Index Scan Backward using tbl_idx on tbl  (rows=1)
     Index Cond: (a = 123::numeric)
Total runtime: 0.053 ms
```

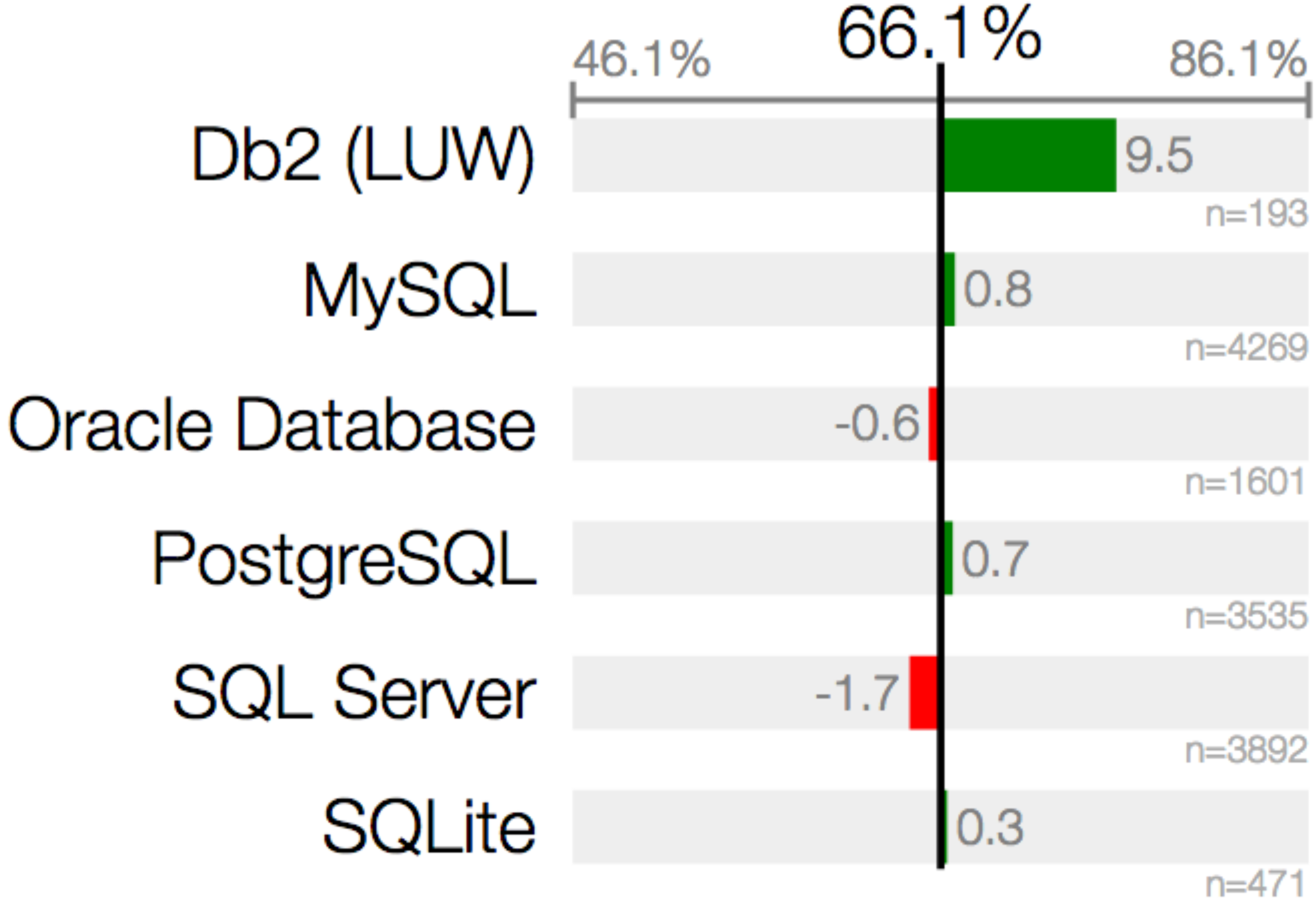
As fast as a primary key lookup because it can never return more than one row.

3-Minute Quiz: Q2 — Results



3-Minute Quiz: Q2 — Results





3-Minute Quiz: Indexing Skills

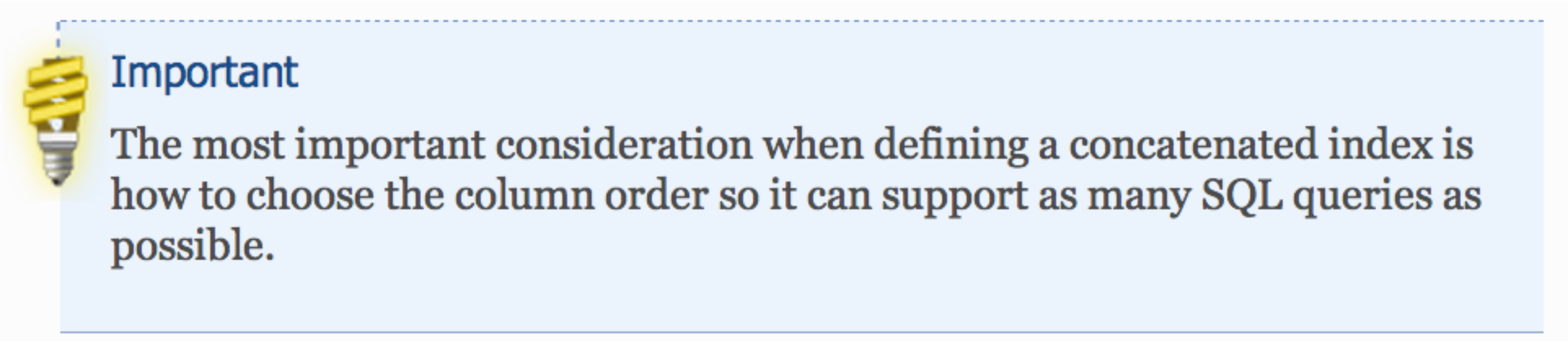
Q3: Good or Bad? *(Column order)*

```
CREATE INDEX tbl_idx ON tbl (a, b);

SELECT id, a, b FROM tbl
 WHERE a = ? AND b = ?;

SELECT id, a, b FROM tbl
 WHERE b = ?;
```

3-Minute Quiz: Indexing Skills

Q3: Good or Bad? *(Column order)*

```
CREATE INDEX tbl_idx ON tbl (a, b);

SELECT id, a, b FROM tbl
 WHERE a = ? AND b = ?;

SELECT id, a, b FROM tbl
 WHERE b = ?;
```



Important

The most important consideration when defining a concatenated index is how to choose the column order so it can support as many SQL queries as possible.

http://use-the-index-luke.com/sql/where-clause/the-equals-operator/concatenated-keys

3-Minute Quiz: Indexing Skills

As-is only one query can use the index (a,b):

```
...WHERE a = ? AND b = ?;
```

```
Bitmap Heap Scan on tbl  (rows=6)
 -> Bitmap Index Scan on tbl_idx (rows=6)
    Index Cond: ((a = 123) AND (b = 1))
Total runtime: 0.055 ms
```

```
...WHERE b = ?;
```

```
Seq Scan on tbl (rows=5142)
  Rows Removed by Filter: 44858
Total runtime: 29.849 ms
```

3-Minute Quiz: Indexing Skills

Change the index to (b, a) so both can use it:

```
...WHERE a = ? AND b = ?;
```

```
Bitmap Heap Scan on tbl  (rows=6)
 -> Bitmap Index Scan on tbl_idx (rows=6)
    Index Cond: ((a = 123) AND (b = 1))
Total runtime: 0.056 ms
```

```
...WHERE b = ?;
```

```
Bitmap Heap Scan on tbl (rows=5142)
 -> Bitmap Index Scan on tbl_idx  (rows=5142)
    Index Cond: (b = 1::numeric)
Total runtime: 6.932 ms
```

3-Minute Quiz: Indexing Skills

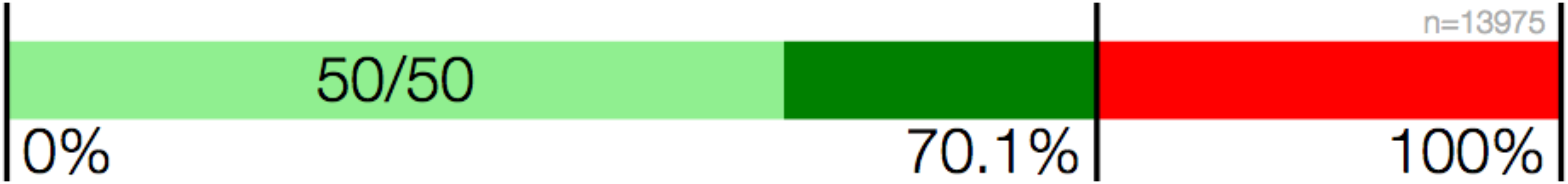
Change the index to (b, a) so both can use it:

```
...WHERE a = ? AND b = ?;

      Bitmap Heap Scan on tbl  (rows=6)
       -> Bitmap Index Scan on tbl_idx (rows=6)
          Index Cond: ((a = 123) AND (b = 1))
      Total runtime: 0.056 ms
```

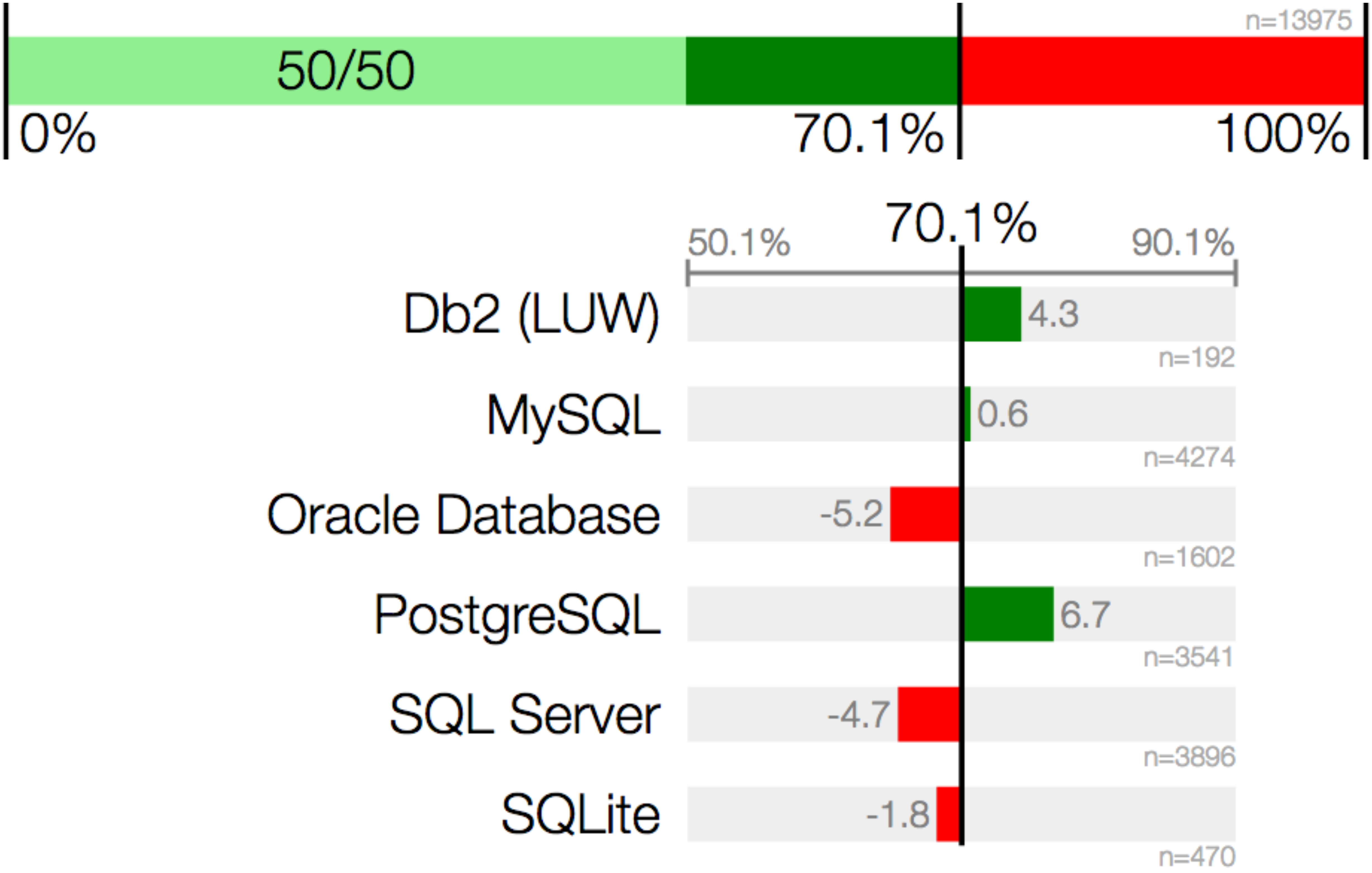
```
...WHERE b = ?;

      Bitmap Heap Scan on tbl (rows=5
       -> Bitmap Index Scan on tbl_idx (rows=5142)
          Index Cond: (b = 1::numeric)
      Total runtime: 6.932 ms
```

4x faster

3-Minute Quiz: Q3 — Results



3-Minute Quiz: Q3 — Results



3-Minute Quiz: Indexing Skills

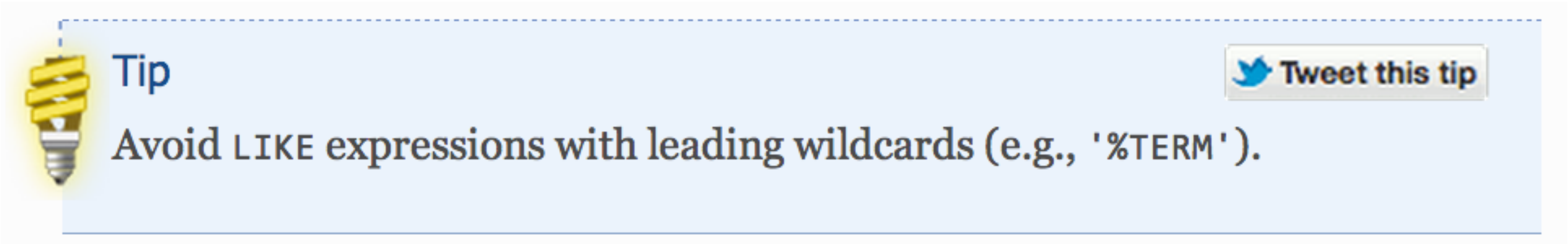
Q4: Good or Bad? *(Indexing LIKE)*

```
CREATE INDEX tbl_idx
    ON tbl (text);
SELECT id, text
  FROM tbl
 WHERE text LIKE 'TJ%';
```

3-Minute Quiz: Indexing Skills

Q4: Good or Bad? *(Indexing LIKE)*

```
CREATE INDEX tbl_idx
    ON tbl (text);

SELECT id, text
  FROM tbl
 WHERE text LIKE 'TJ%';
```

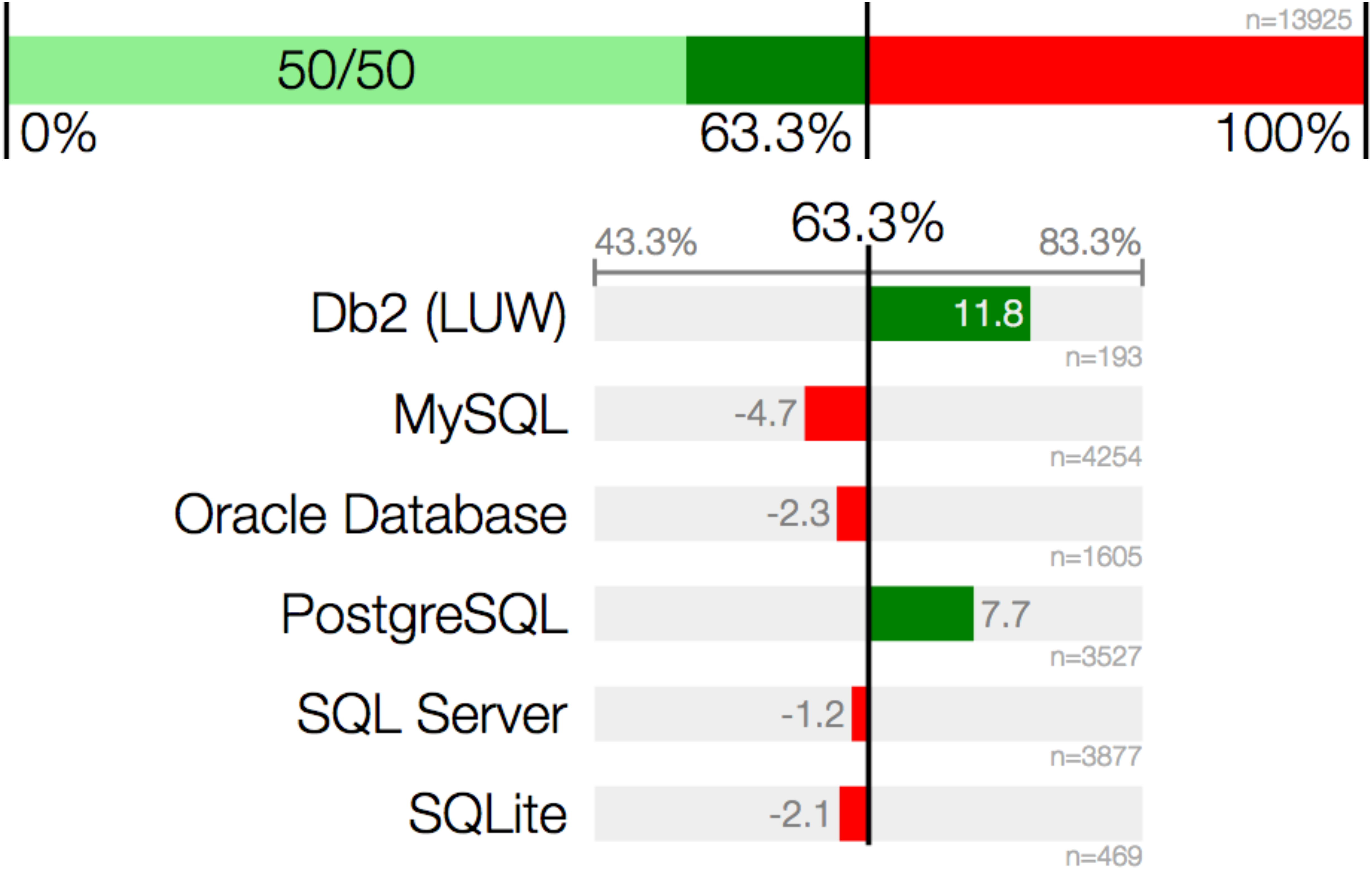


Tip

Avoid `LIKE` expressions with leading wildcards (e.g., `'%TERM'`).

http://use-the-index-luke.com/sql/where-clause/searching-for-ranges/like-performance-tuning

3-Minute Quiz: Q4 — Results



3-Minute Quiz: Indexing Skills

Q5: How will performance change? *(IOS)*

```
CREATE INDEX tbl_idx
          ON tbl (a, date_column);


SELECT date_column
     , count(*)
  FROM tbl
 WHERE a = ?
 GROUP BY date_column;



       (~3 rows)
```

3-Minute Quiz: Indexing Skills

Q5: How will performance change? *(IOS)*

```
CREATE INDEX tbl_idx
          ON tbl (a, date_column);
```

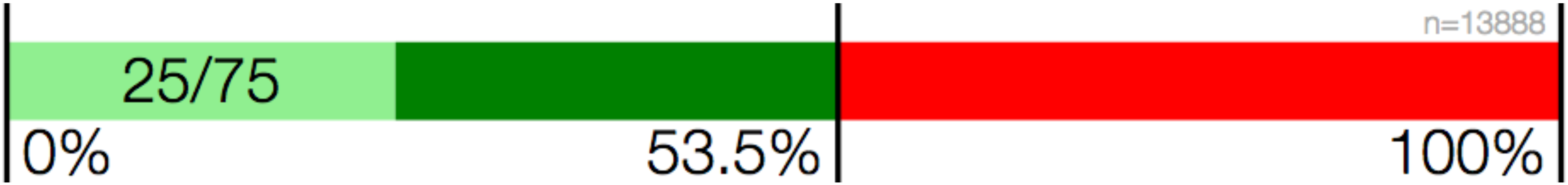
```
SELECT date_column
     , count(*)
  FROM tbl
 WHERE a = ?
 GROUP BY date_column;
```

(~3 rows)

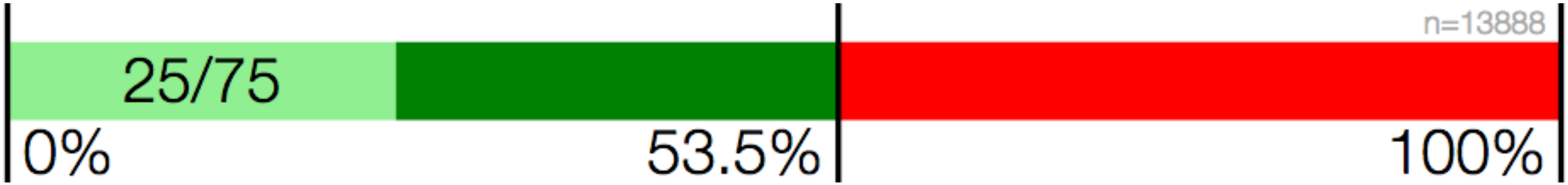
```
SELECT date_column
     , count(*)
  FROM tbl
 WHERE a = ?
   AND b = ?
 GROUP BY date_column;
```

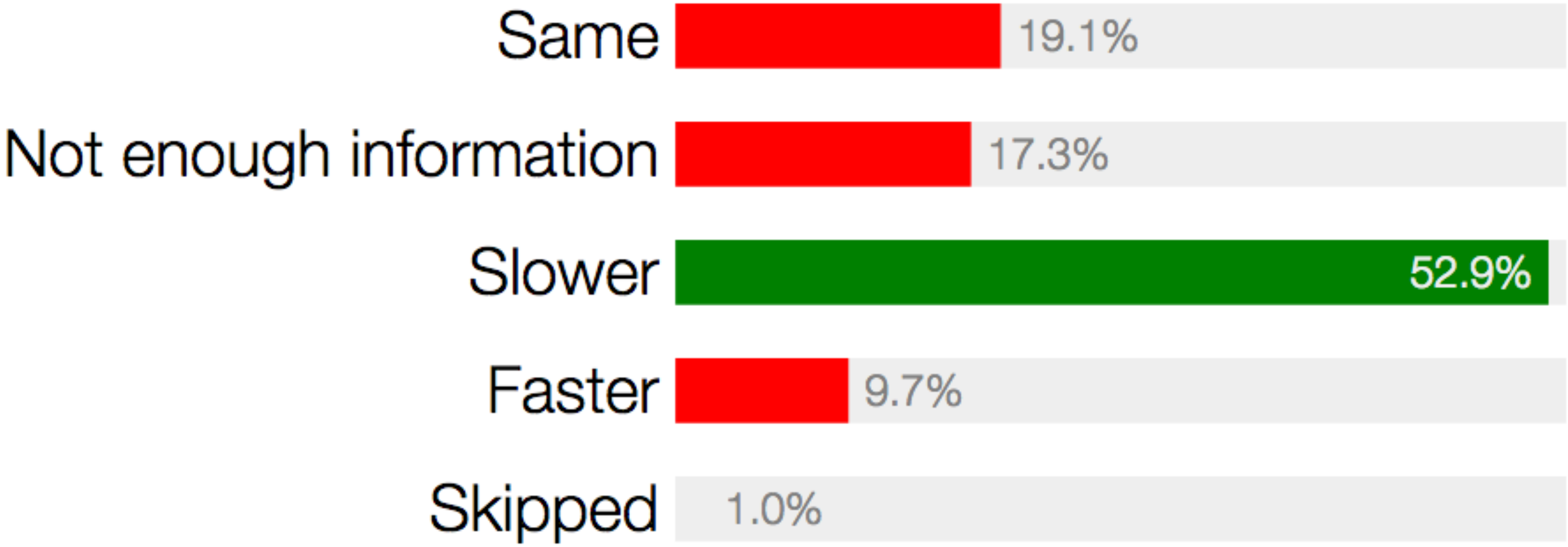
(~1 rows)

3-Minute Quiz: Q4 — Results

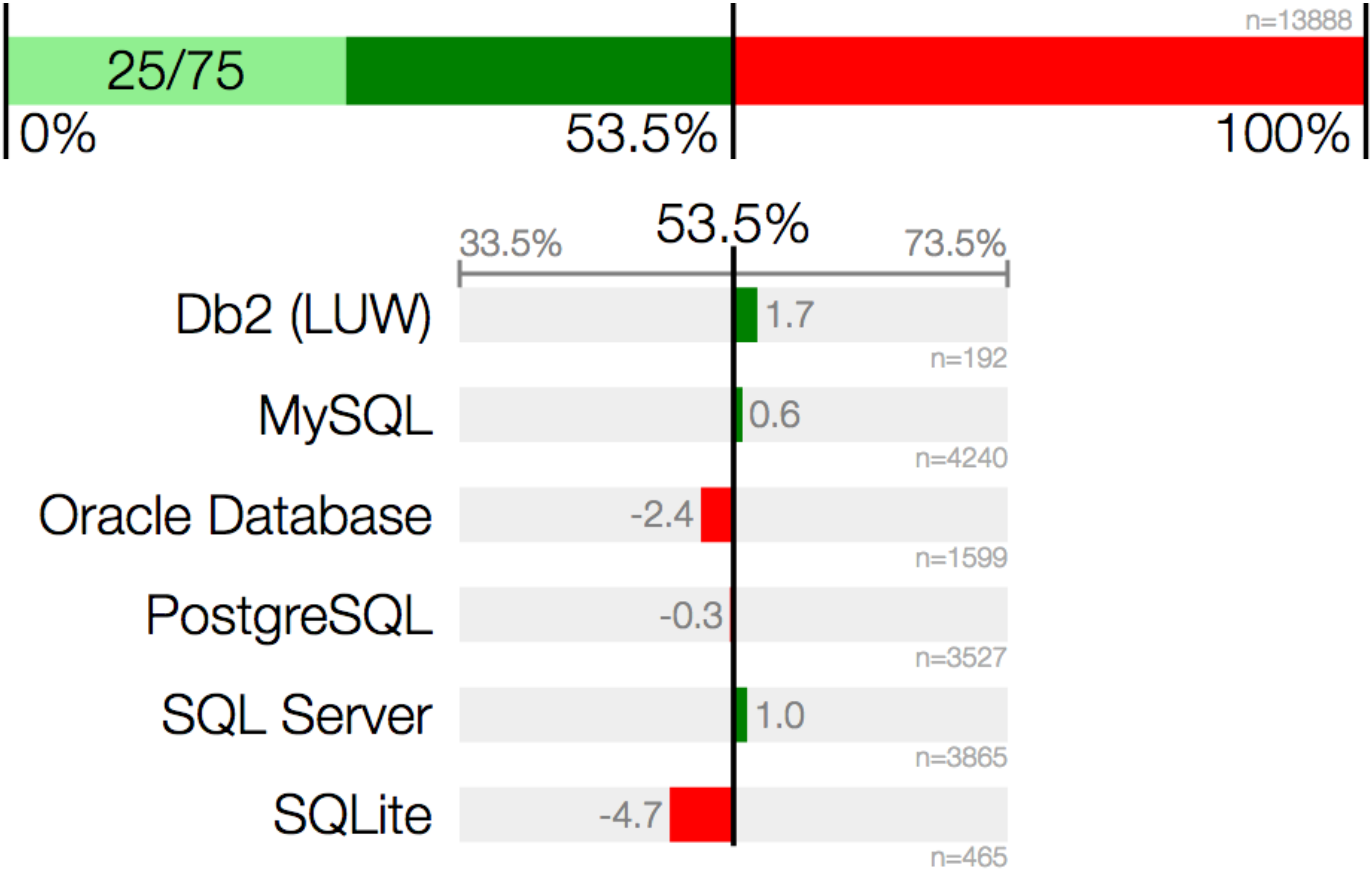


3-Minute Quiz: Q4 — Results





3-Minute Quiz: Q4 — Results



3-Minute Quiz: Indexing Skills

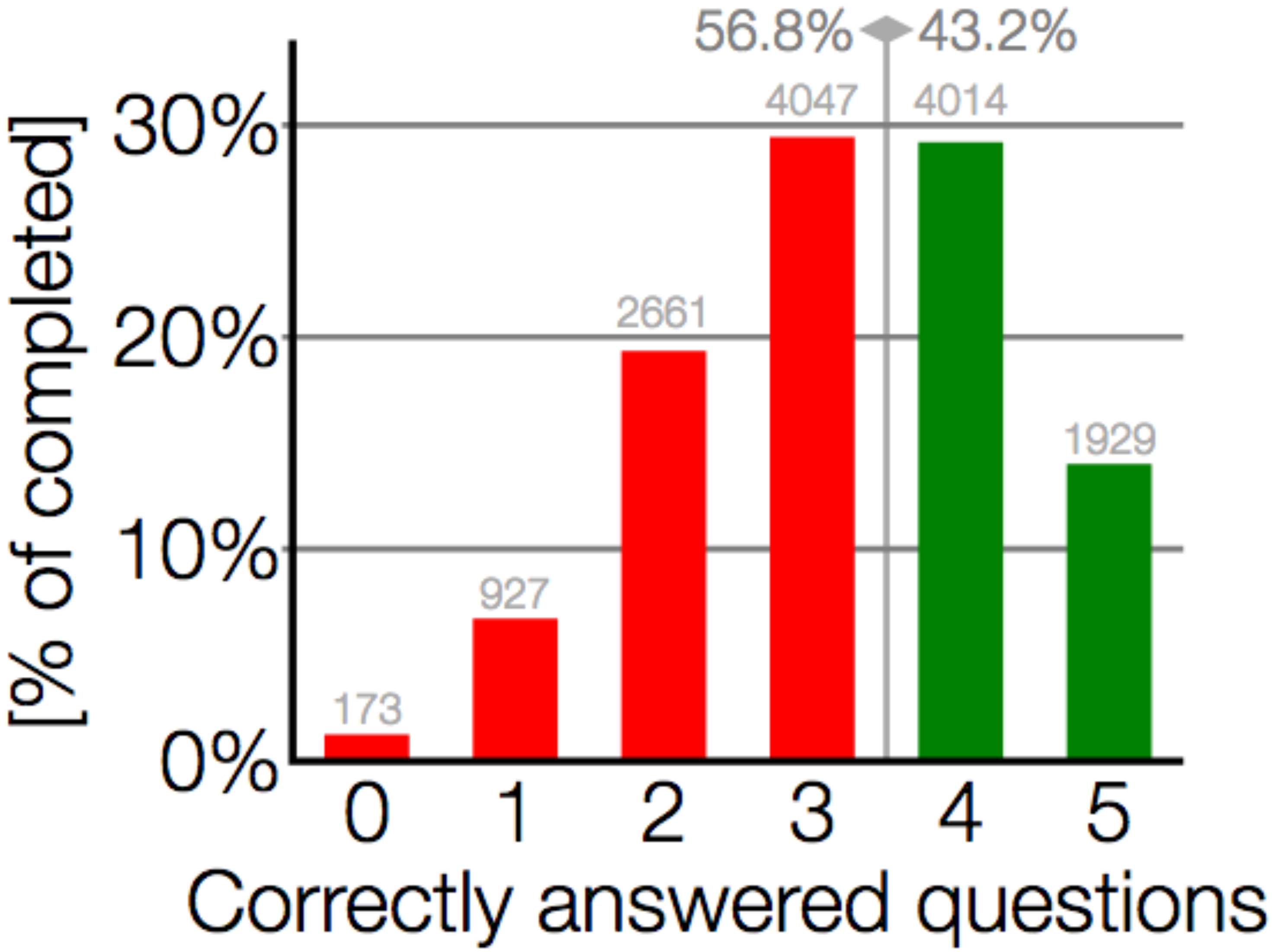
Original query could do an index-only scan ("covering index"), new query not.

```
GroupAggregate (actual rows=3 loops=1)
-> Index Only Scan using tbl_idx on tbl (actual rows=3 loops=1)


GroupAggregate (actual rows=1 loops=1)
Group Key: date_column
-> Sort (actual rows=1 loops=1)
   -> Bitmap Heap Scan on tbl (actual rows=1 loops=1)
      Rows Removed by Filter: 2
      -> Bitmap Index Scan on tbl_idx (actual rows=3 loops=1)
```

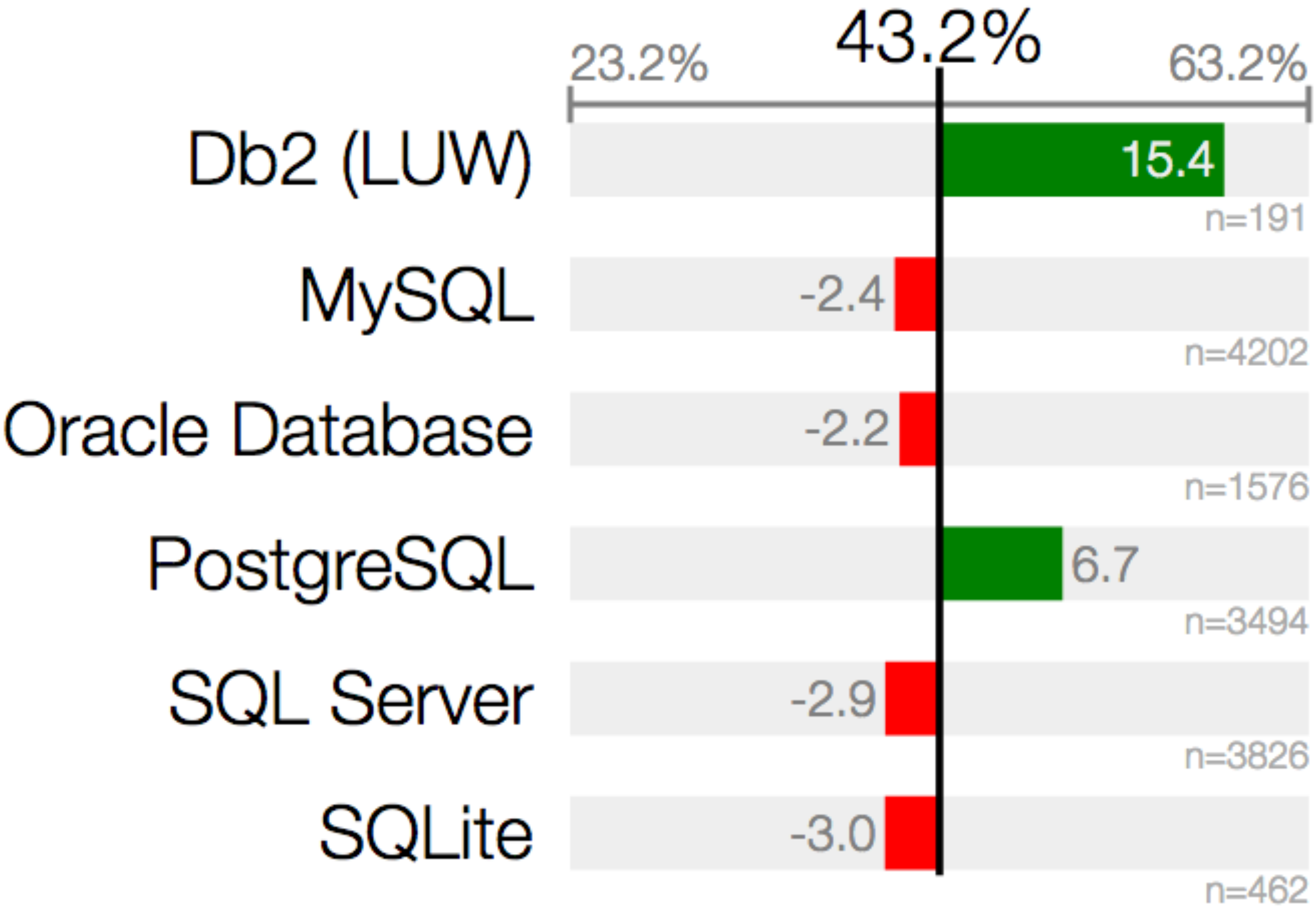
http://use-the-index-luke.com/blog/2014-01/unreasonable-defaults-primary-key-clustering-key

3-Minute Quiz: How many pass it?

3-Minute Quiz: How many pass it?



3-Minute Quiz: How many pass it?



Indexes: The Neglected All-Rounder

Everybody knows indexing is important for performance, yet nobody takes the time to learn and apply is properly.

Indexes: The Neglected All-Rounder

Index details are hardly known.

➡ “Details” like column-order or equality vs. range conditions must be **learned and understood**.

Indexes: The Neglected All-Rounder

Index details are hardly known.

➡ “Details” like column-order or equality vs. range conditions must be learned and understood.

Only one index capability is used: finding data quickly

➡ Indexes have three capabilities (powers): finding data, clustering data, and sorting data.

Indexes: The Neglected All-Rounder

Index details are hardly known.

➡ "Details" like column-order or equality vs. range conditions must be <u>learned and understood</u>.

Only one index capability is used: finding data quickly

➡ Indexes have <u>three capabilities (powers)</u>: finding data, clustering data, and sorting data.

Indexing is done from single query perspective.

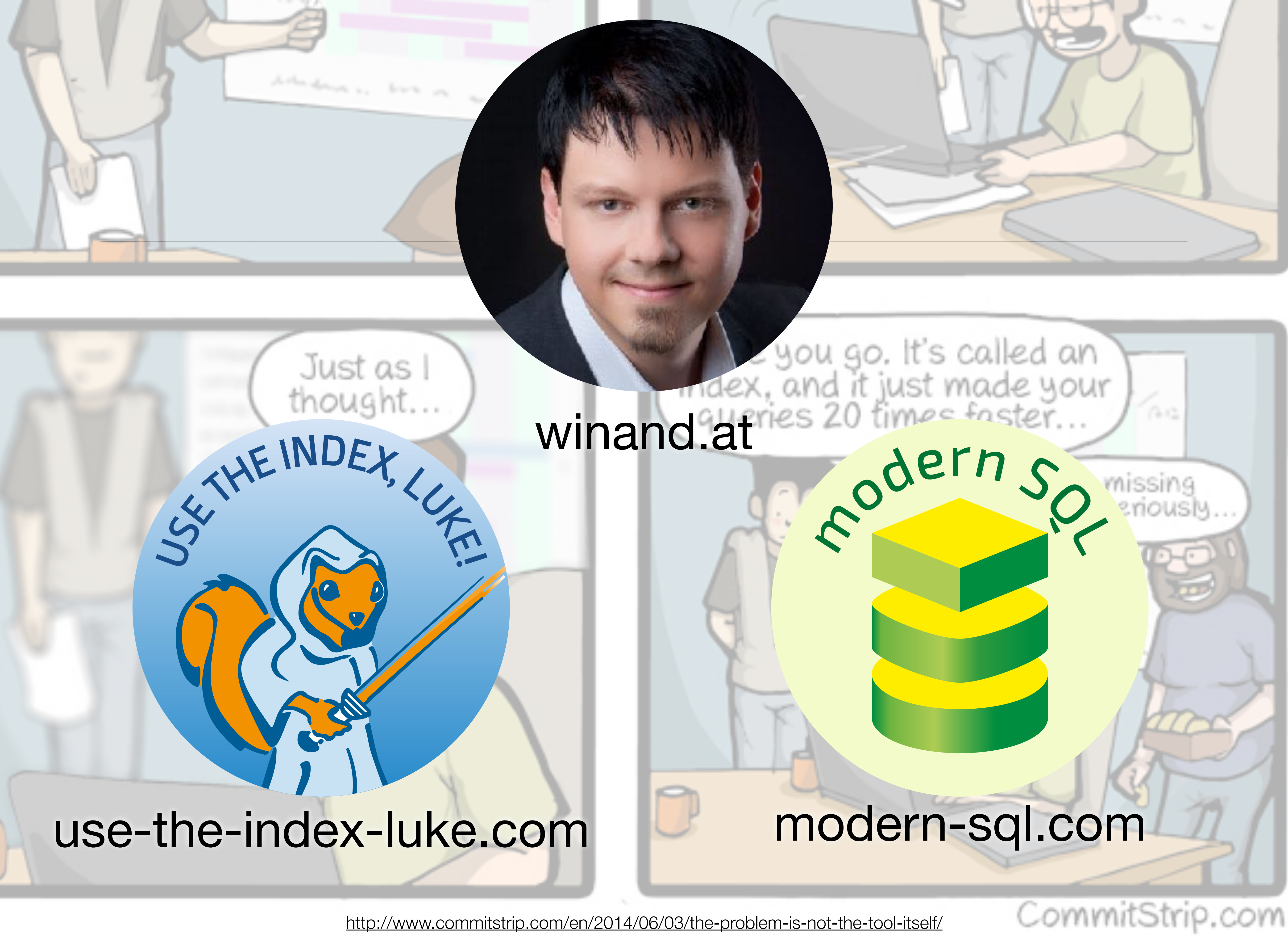
➡ Should be done from application perspective (considering all queries). It's a <u>design</u> task!

Indexes: The Neglected All-Rounder

Are you just adding indexes

or

are you designing indexes?



winand.at

use-the-index-luke.com

modern-sql.com

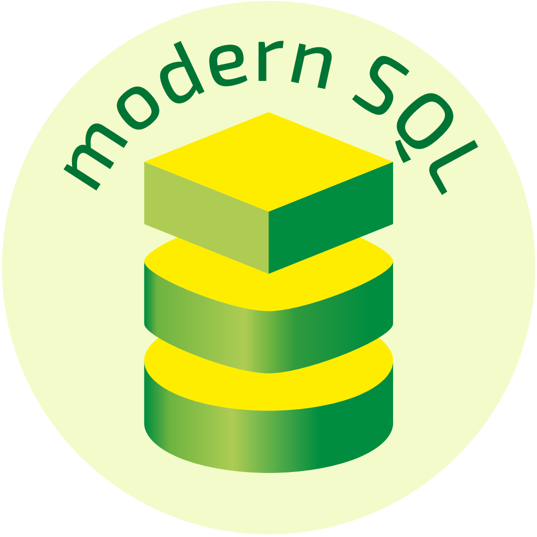
http://www.commitstrip.com/en/2014/06/03/the-problem-is-not-the-tool-itself/

Bitte geben Sie uns jetzt Ihr Feedback!

Volkskrankheit "stiefmütterliche Indizierung"

Markus Winand

use-the-index-luke.com



modern-sql.com