DATABASE IMAGE CONTENT EXPLORER: CARVING DATA THAT DOES NOT OFFICIALLY EXIST

James Wagner, Alexander Rasin, Jonathan Grier

Overview

- Background (Database Carving and Database Storage)
- DICE
- Reconstructing 3 Flavors of Deleted Data
- Experiments
- Conclusion and Future Work

Database Carving

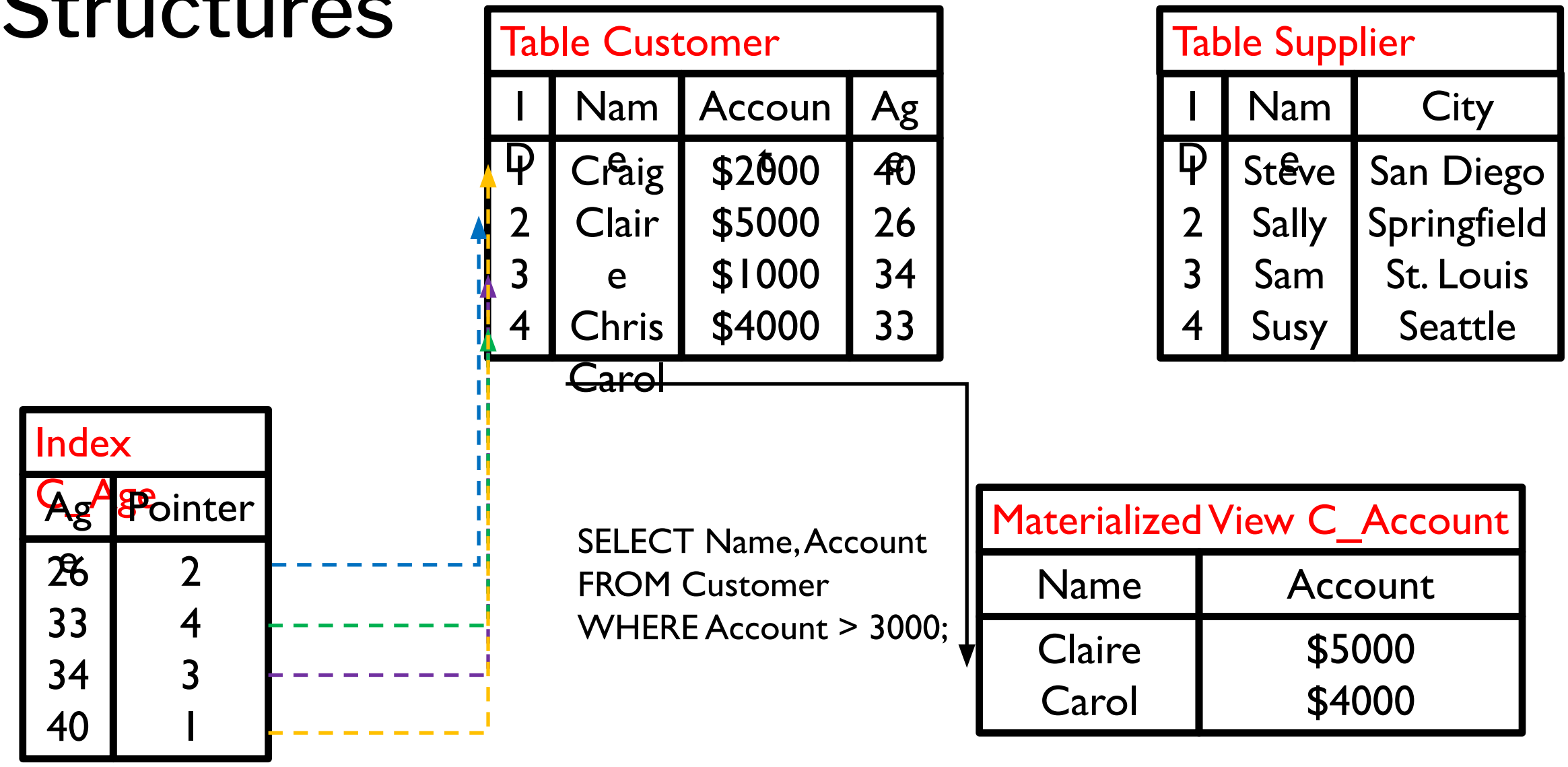
Challenges

- Database specific storage models
- Database files do not have headers
- Record reconstruction
- Values are encoded with metadata

Database Carving is Reconstruction, not Recovery

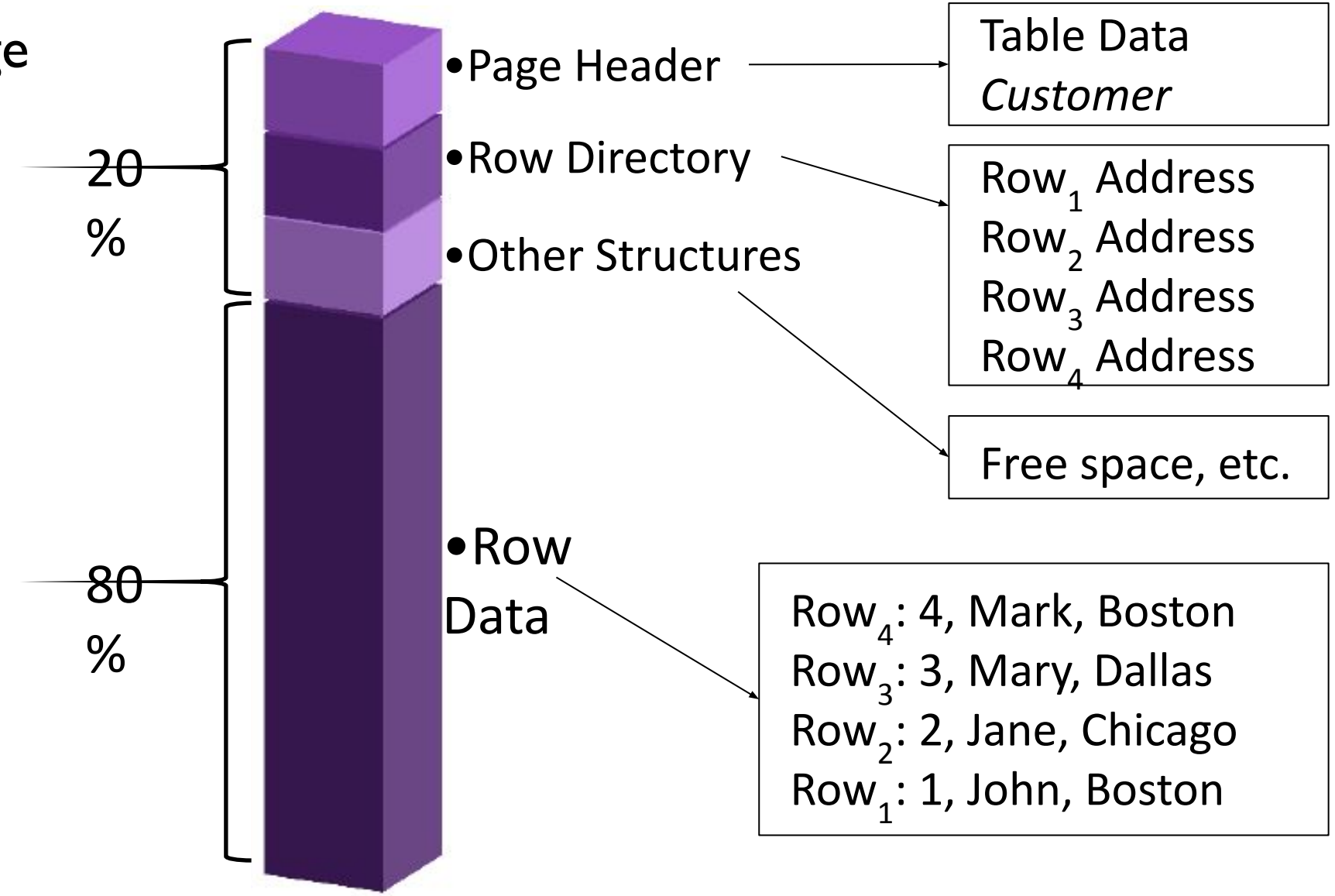
- Doesn't rely on working database
- Evidence not found in backups
- Corrupt disk, deleted files, read-only

Database Structures

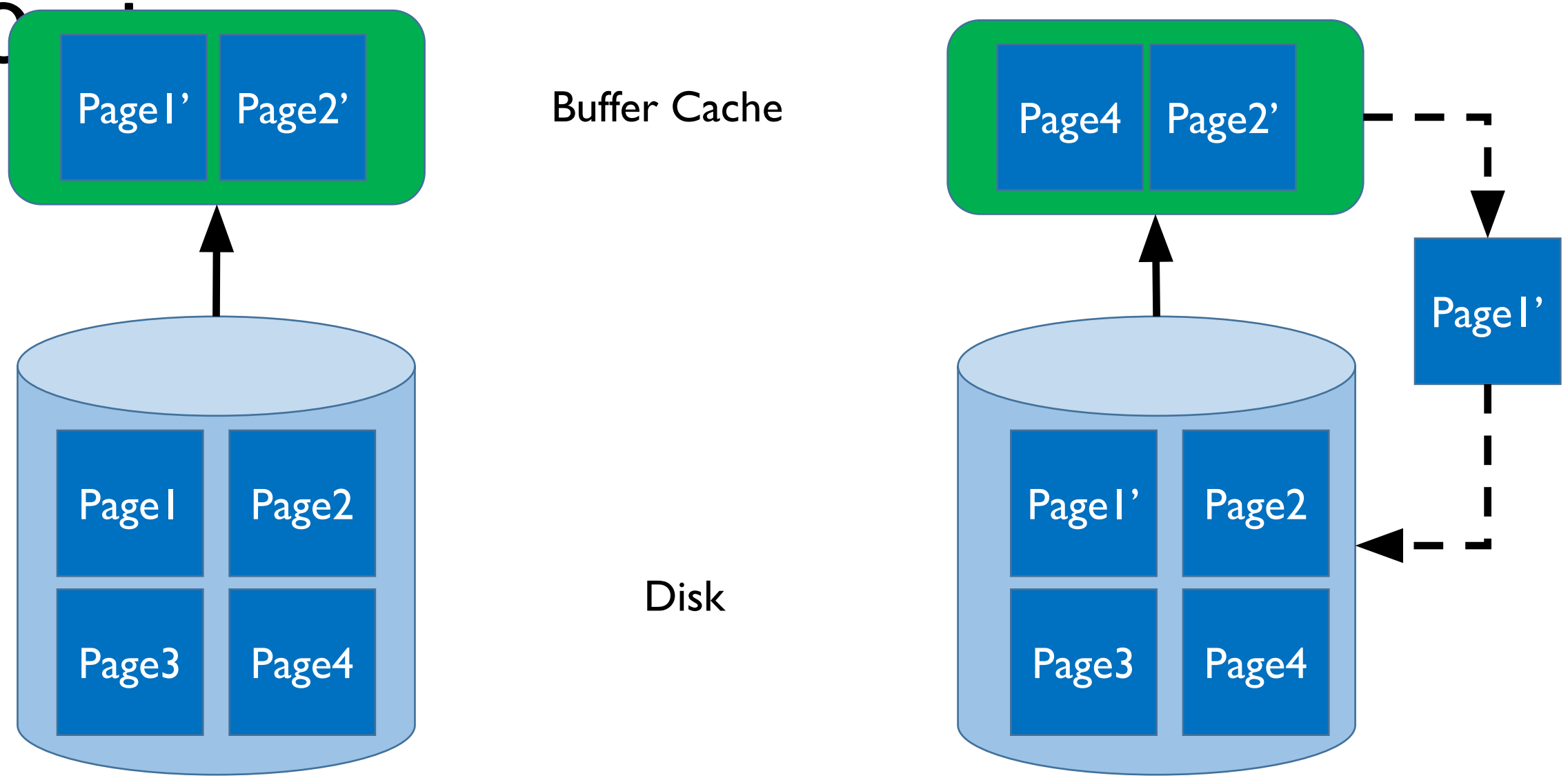


Database Storage: Pages

- Minimum Unit of Storage
- Memory Management
- Supported DBMSes:
  - DB2
  - Oracle
  - SQL Server
  - PostgreSQL
  - MySQL
  - SQLite
  - Firebird
  - Apache Derby



Database Buffer C



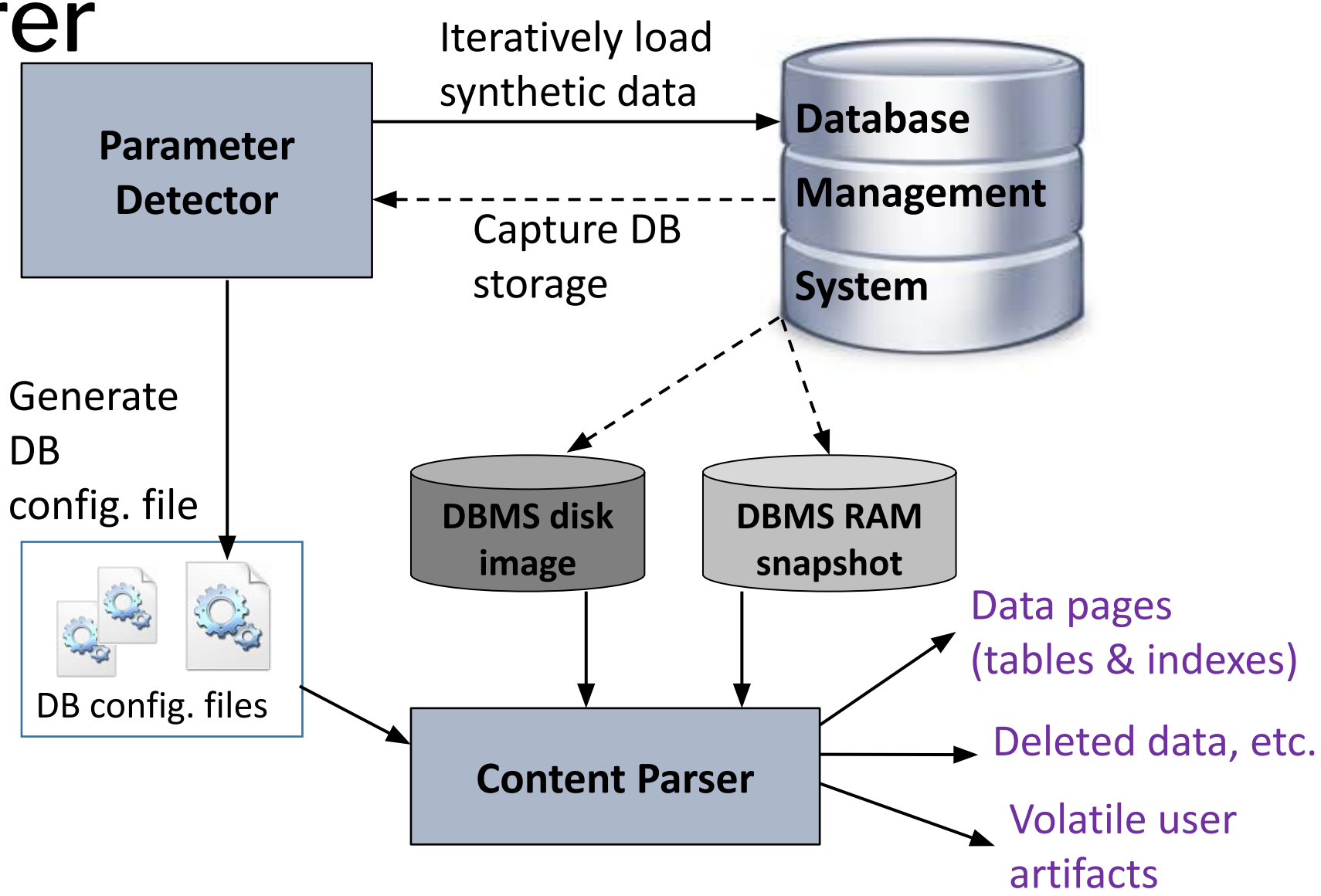
Query1 modifies Page1 and Page2

Query2 reads Page4

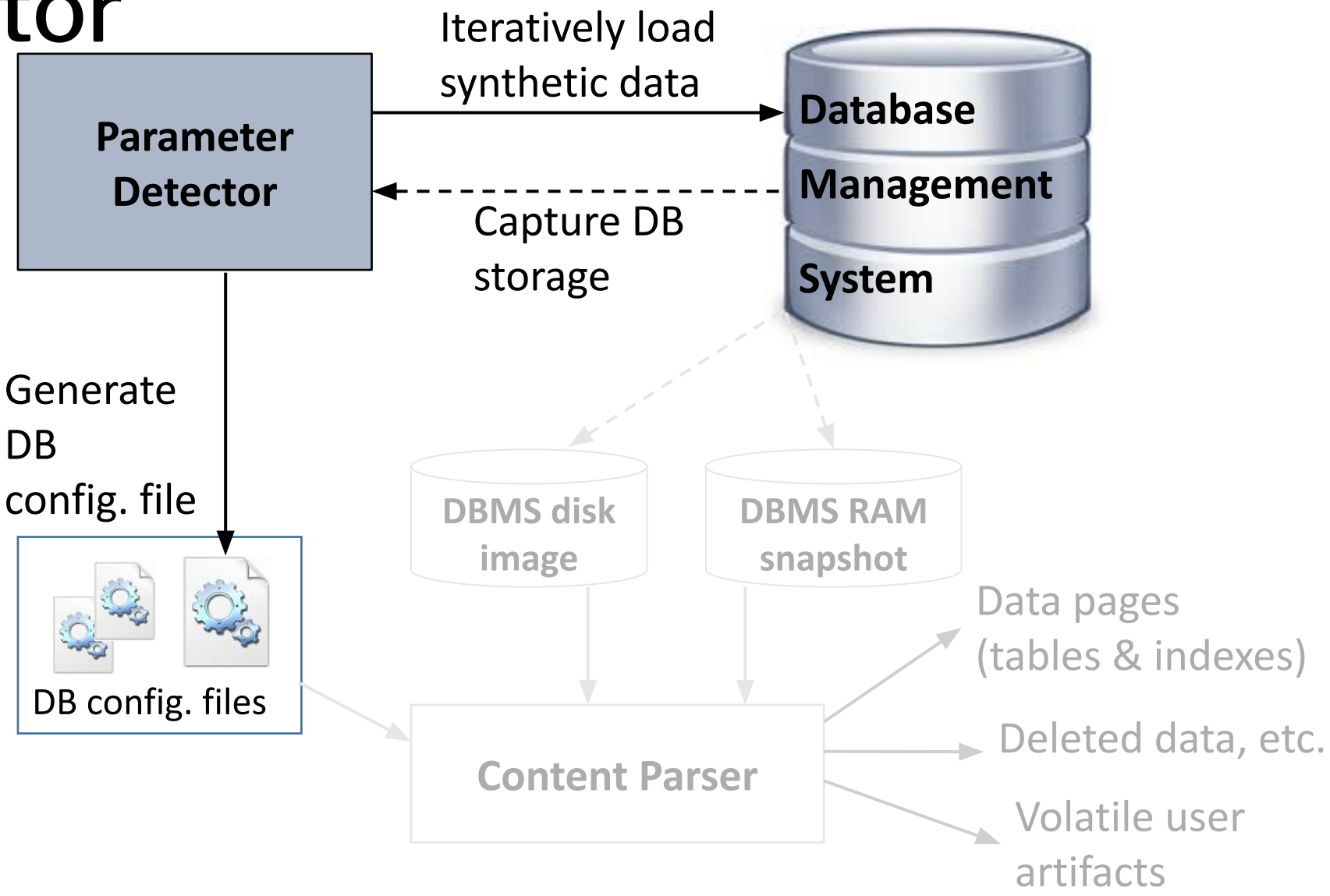
Overview

- Background (Database Carving and Database Storage)
- DICE
- Reconstructing 3 Flavors of Deleted Data
- Experiments
- Conclusion and Future Work

DICE: Database Image Content Explorer



DICE: Parameter Collector



Parameter Collection: Datatypes

- How do you automate it? Example: Integers

  PostgreSQL(4 bytes): $(256^0 * B_1) + (256^1 * B_2) + (256^2 * B_3) + (256^3 * B_4)$

  1 = 1, 0, 0, 0

  256 = 0, 1, 0, 0

  257 = 1, 1, 0, 0

  Oracle: Uses zero compression.

  4 = 3, 192, 5

  40 = 3, 192, 41

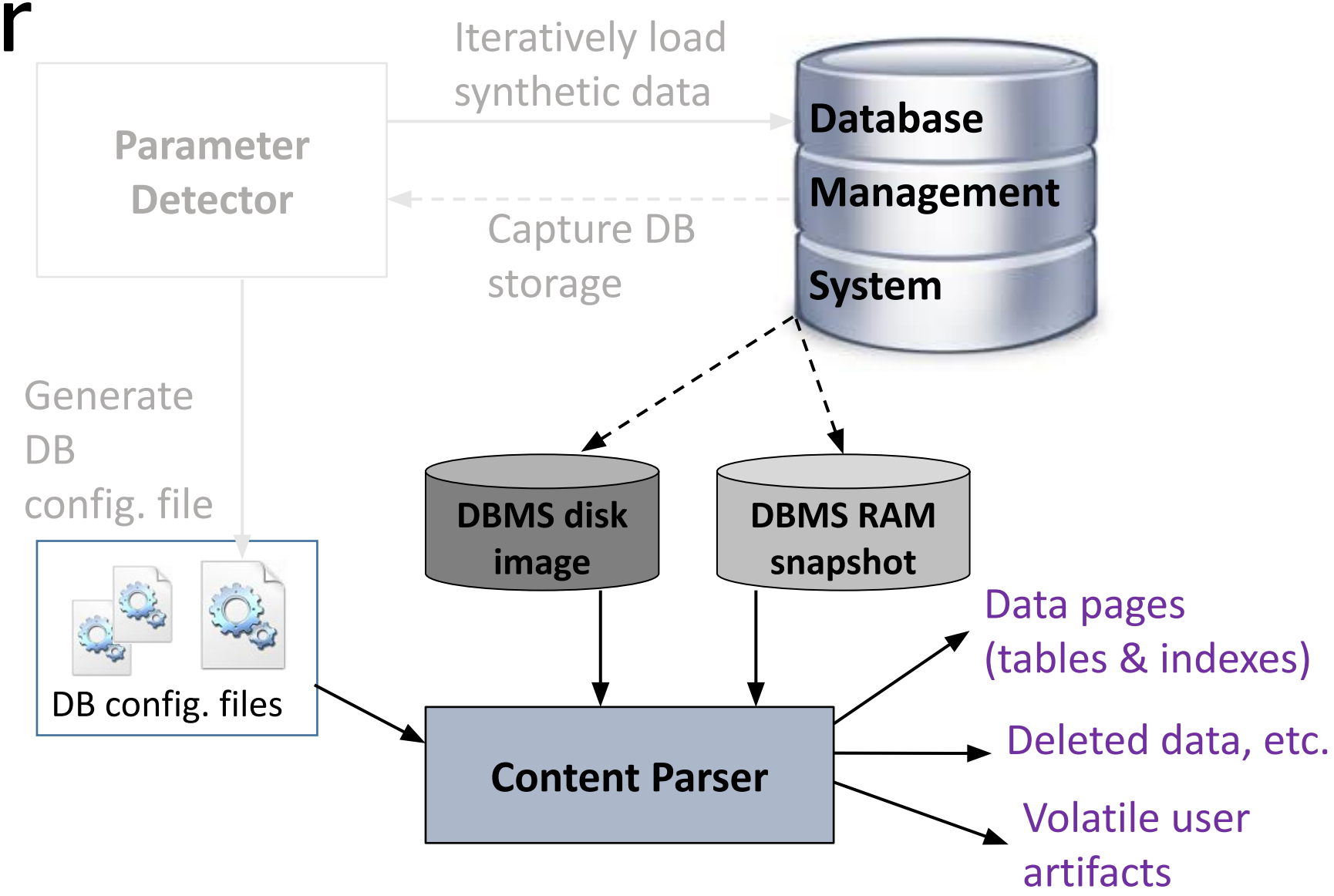
  400 = 3, 193, 5

  440 = 4, 193, 5, 41

- Datatype Detection. Example: PostgreSQL Integer of String?

ASCII	.	J	a	y
Decimal	3	74	97	121

DICE: Content Parser



Parsing Example: PostgreSQL

Parameter	Value
Raw Data Delimiter	2, 9, 24
Raw Data Position	4
Number Storage	4
Number Method	PSQL
String X	2
String Y	3

$$\frac{\text{String Length}}{2} = \frac{39-3}{2} = 18$$

ASCII	Decimal
	0
	2
	9
	24
	0
+	43
	6
	0
	0
'	39
C	67
u	117
s	115
t	116

ASCII	Decimal
o	111
m	109
e	101
r	114
#	35
0	48
0	48
0	48
0	48
0	48
1	49
5	53
7	55
9	57

C1 = 1579(Integer)

C2 = Customer#000001579(String)

Overview

- Background (Database Carving and Database Storage)
- DICE
- Reconstructing 3 Flavors of Deleted Data
- Experiments
- Conclusion and Future Work

Deleted Data

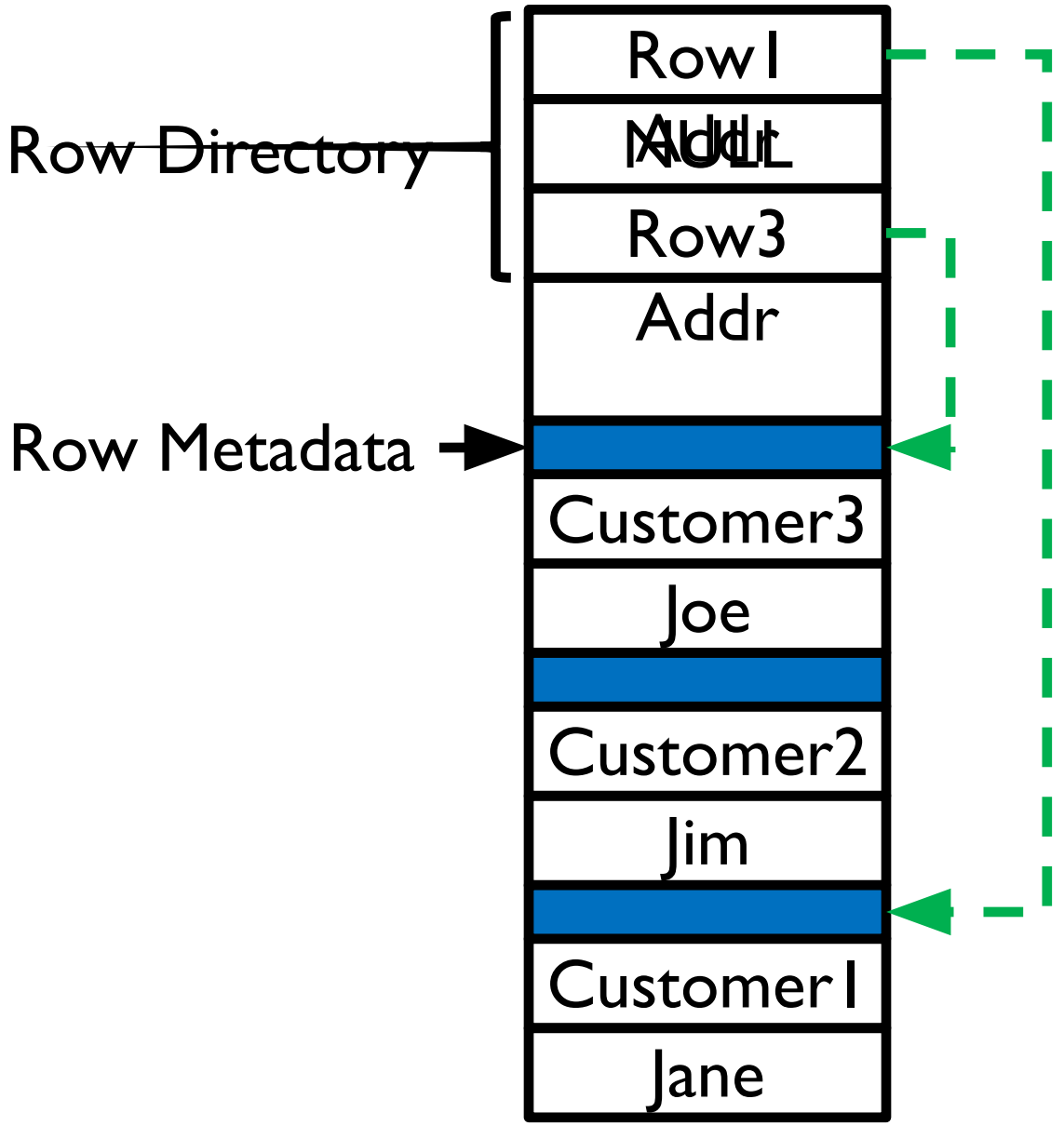
- No longer recoverable by the DBMS
- A delete only marks data, not overwrite.
- Unallocated storage
- Three types of deleted data DICE can reconstruct:
    1. Rows
    2. Pages
    3. Values

Deleted Data: Rows(1/4)

- Rows are the smallest unit of deletion or insertion in a RDMBS
- Three possible page alterations:
    1. Page Header
    2. Row Directory
    3. Row Data
- The page header checksum verifies data integrity

Deleted Data: Rows(2/4)

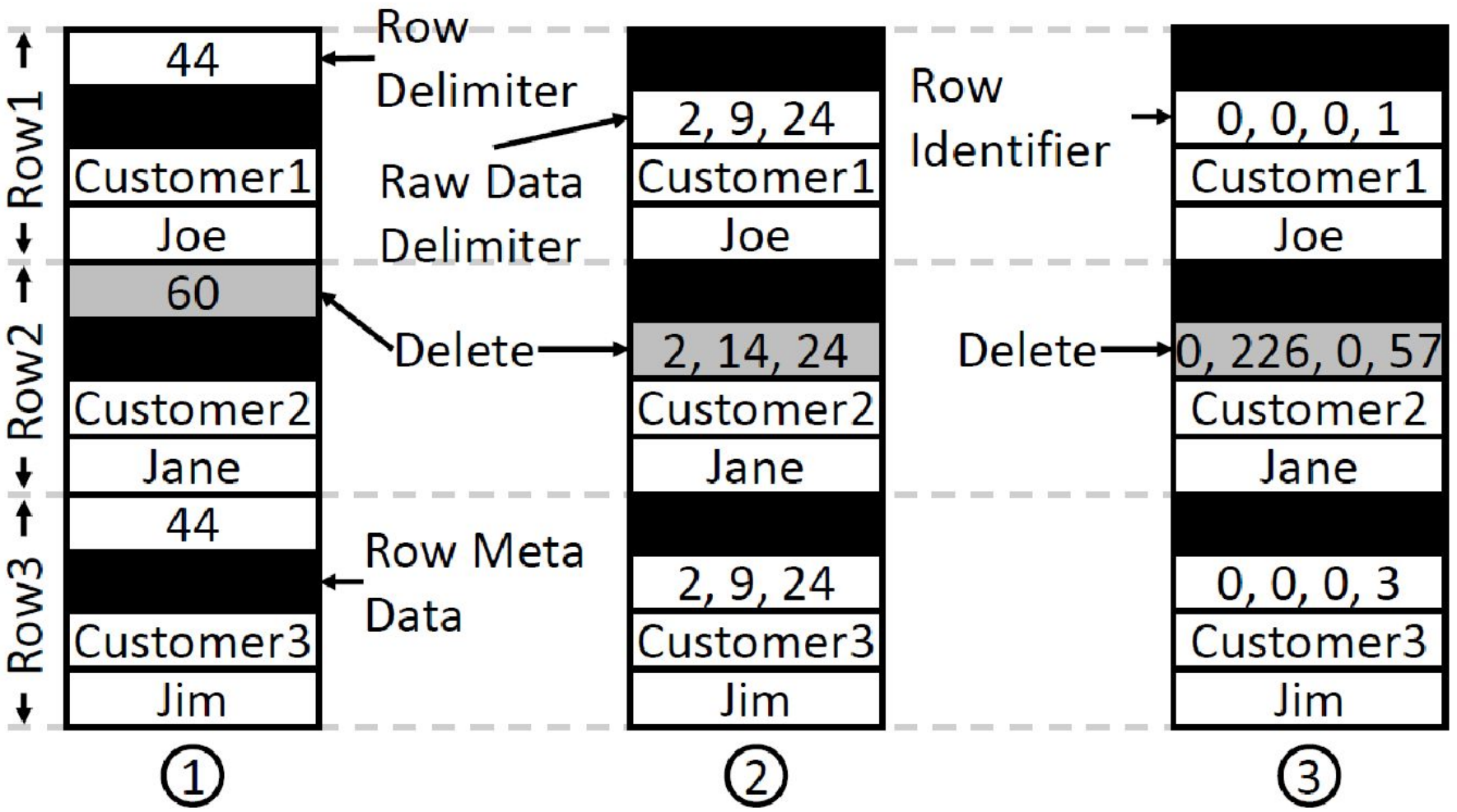
- Row directory address overwrite
- DB2 and SQL Server



Deleted Data: Rows(3/4)

- Row metadata flag

1. MySQL or Oracle
2. PostgreSQL
3. SQLite



*DB2 and SQL Server mark a deletion in the row directory

Deleted Data: Rows(4/4)

Updated Rows

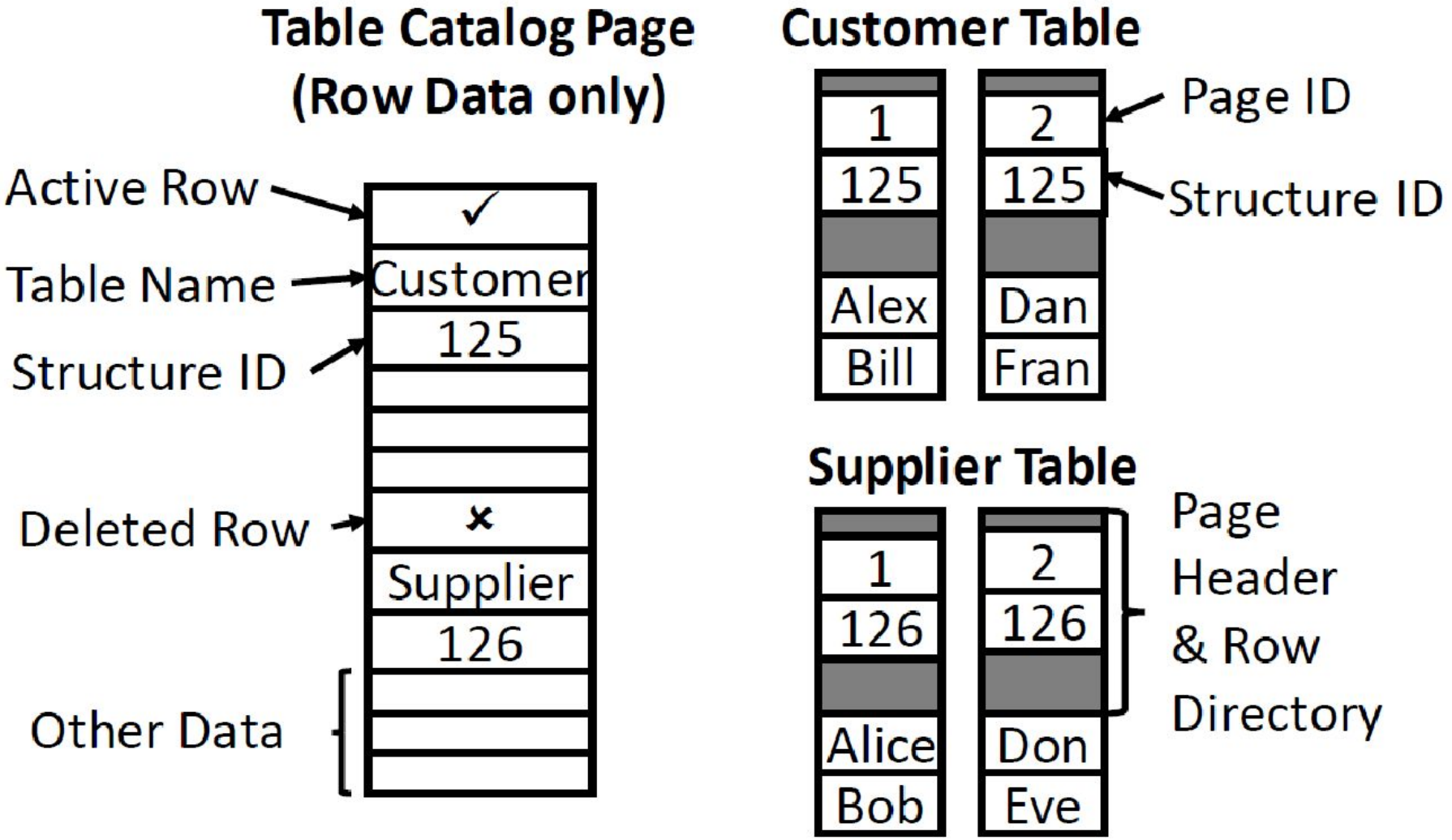
- In-place or DELETE + INSERT
- A new row can overwrite an row of equal or smaller size

Transactional Effects

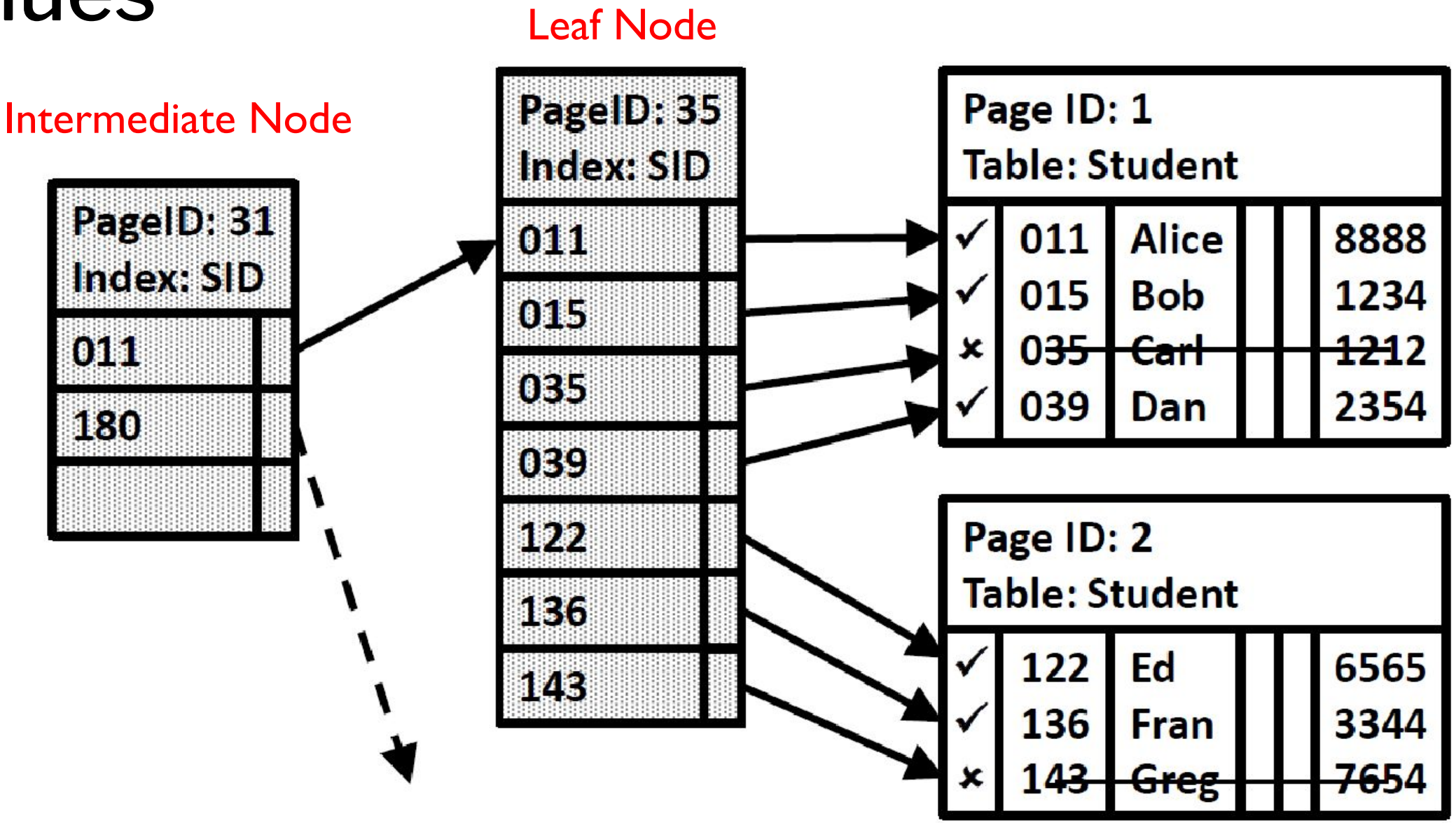
- Failed transactions: User perspective vs. storage
- Undone insert looks like a deleted row in the page

Deleted Data: Pages

- Drop Table
- System Tables



Deleted Data: Index Values



Overview

- Background (Database Carving and Database Storage)
- Related Work (Carving Data and DICE)
- Reconstructing 3 Flavors of Deleted Data
- Experiments
- Conclusion and Future Work

Experiment 1: Reconstructing Deleted Rows

Introduction

- Lifetime of deleted rows
- Representative DBs
  - Oracle - Percent page utilization (39%)
  - SQL Server – Overwrite if there's space

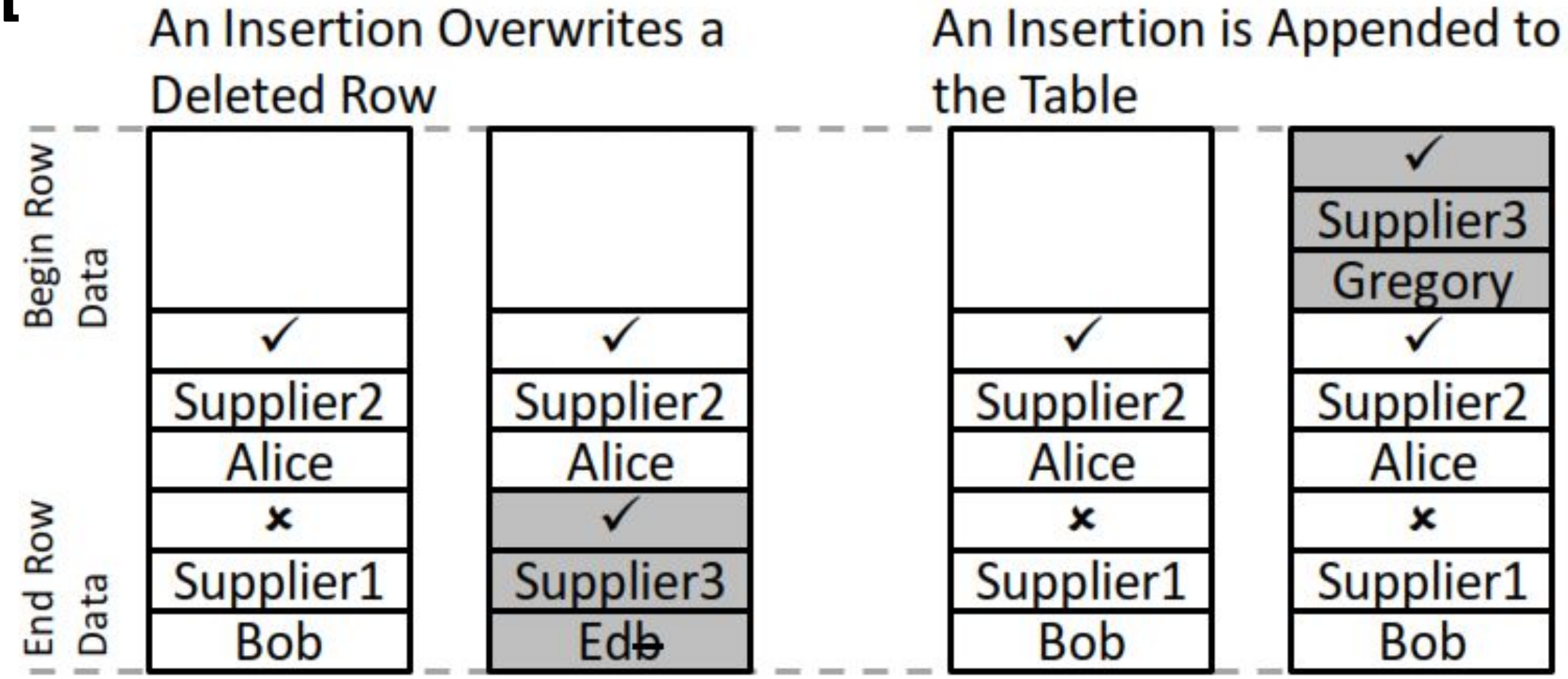
Setup

- 2 tables: 20K random sized rows
- 85 rows/page, 236 pages
- Random & contiguous deletes
- Inserted rows were random size

		Oracle		SQLServer	
	Action	T1(Rand)	T2(Cont)	T1(Rand)	T2(Cont)
Step 1	Delete 1K	1000	1000	1000	1000
Step 2	Rows	1000	8	416	354
Step 3	Insert 1K Rows	1000	8	394	12

Insert 1K Rows

Experiment 1: Example



Experiment 2: Aborted Transaction

Purpose

- Show that data inserted by aborted transactions still exists in storage

Procedure

- T0: Start with an unmodified table
- T1: Insert 1000 rows (~12 pages)
- T2: Abort the inserts
- T3: Flush buffer cache

Aborted Insert Location

Step	Disk	RAM
T0	∅	∅
T1	∅	✔
T2	∅	□
T3	□	□*

∅ Row does not exist

✔ Row is marked as active

□ Row is marked as deleted

*Flush cache command does not overwrite pages

Experiment 3: Table Rebuild

Introduction

- Identify what's behind after a table rebuild
- Representative DB

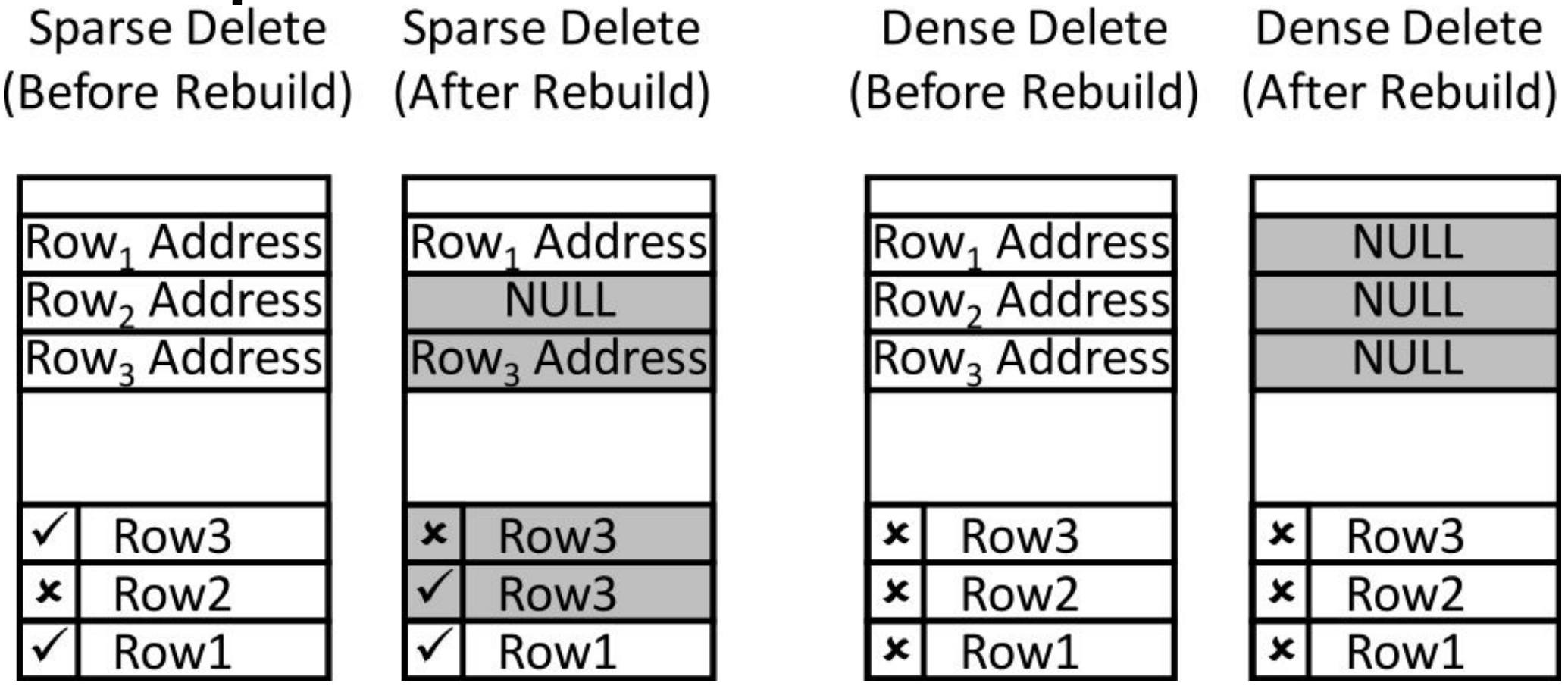
  PostgreSQL – Defragmentation command

Setup

- 2 tables: 20K random sized rows
- 85 rows/page, 236 pages
- Random & contiguous deletes

	Before Rebuild		After Rebuild	
Row Type	T1(Rand)	T2(Cont)	T1(Rand)	T2(Cont)
Deleted	1000	1000	16	854
Active Duplicates	0	0	1134	182

Experiment 3: Example



Conclusion

- DICE can reconstruct data that is outside of the user's view
- Three types of data in unallocated space: rows, pages, and values

Future Work

- Make DICE output user friendly
- Connect DICE output and other forensic tool output for meta-querying
- Audit query log consistency with disk and RAM

Questions

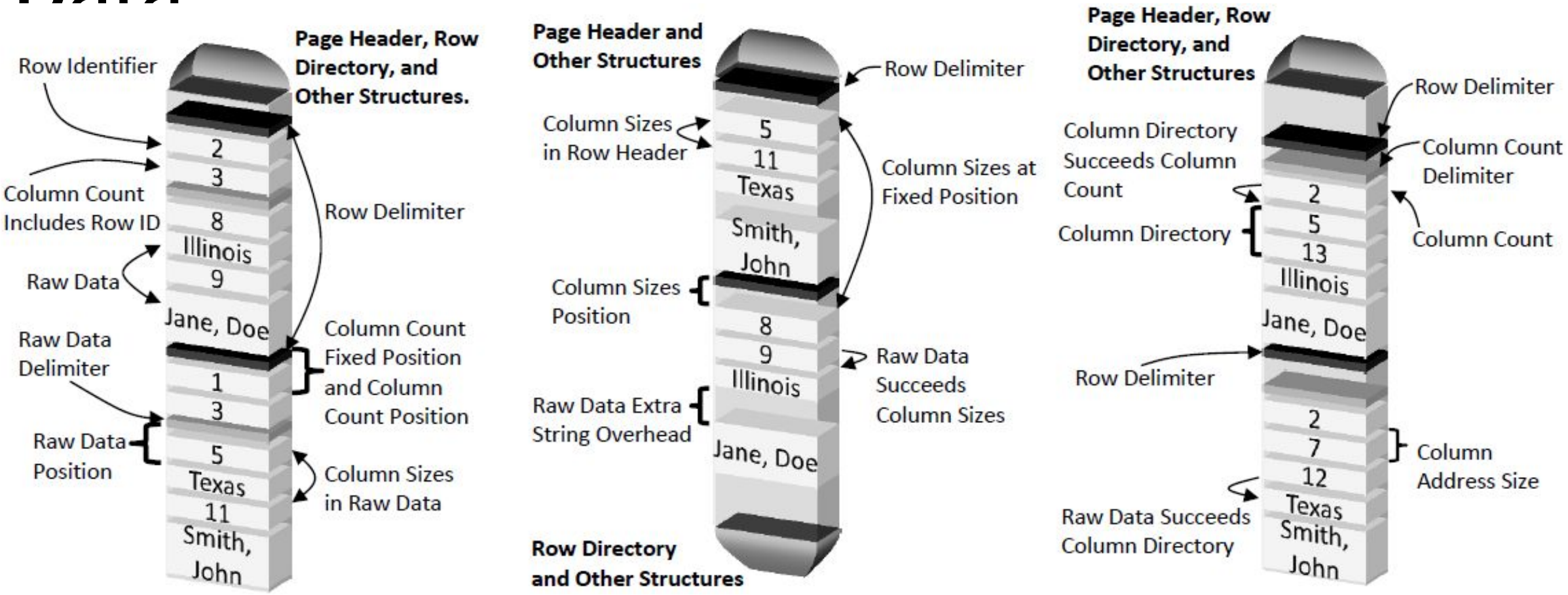
Generalizing Parameters

	Oracle	PostgreSQL	SQLite	Firebird	DB2	SQLServer	MySQL	Apache Derby
Structure Identifier	Yes	No	Yes					No
Unique Page ID	Yes							No
Row Dir. Sequence	Top-to-bottom insertion					Bottom-to-top insertion		
Row Identifier	No	Yes	No				Yes	
Column Count	Yes		No		Yes		No	Yes
Column Sizes	Yes				No		Yes	
Column Directory	No				Yes		No	
Numbers w/Strings	Yes				No		Yes	

5 | 10John Smith | 11 | 5Texas

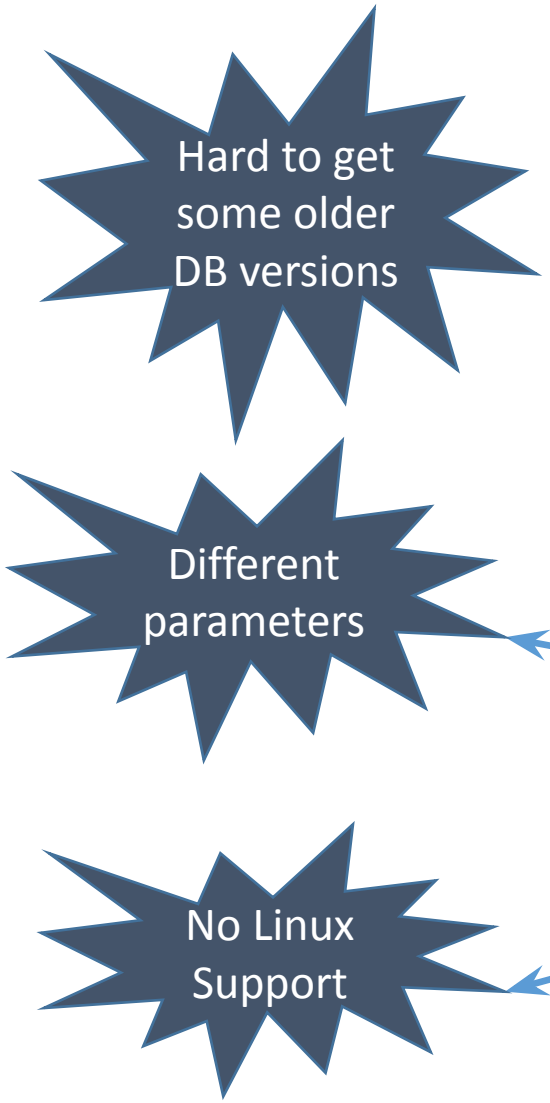
1&3&5&15 || 5 | 11 | John Smith | Texas

Parameter Collection: Row Data



Example: MySQL primary key storage

DBMSes



DBMS Version	Testing OS	Buffer Size(MB)	Page Size(KB)
Apache Derby 10.10	Linux	400	4
Apache Derby 10.5	Linux	400	4
DB2 Express-C 10.5	Linux	400	4
Firebird 2.5.1	Linux	400	8
Firebird 2.1.7	Windows	400	8
MySQL Server 5.1.73	Linux	800	16
MySQL Server 5.6.1	Windows	800	16
Oracle 11g R2	Windows	800	8
Oracle 12c R1	Windows	1200	8
PostgreSQL 7.3	Linux	400	8
PostgreSQL 8.4	Linux	400	8
PostgreSQL 9.3	Windows	800	8
SQLite 3.8.6	Linux	2	1
SQLite 3.8.7	Windows	2	1
SQLServer 2008 Enterprise	Windows (~~Linux~~)	800	8