

Database System Internals

Lecture 28

Column-store DBMSs

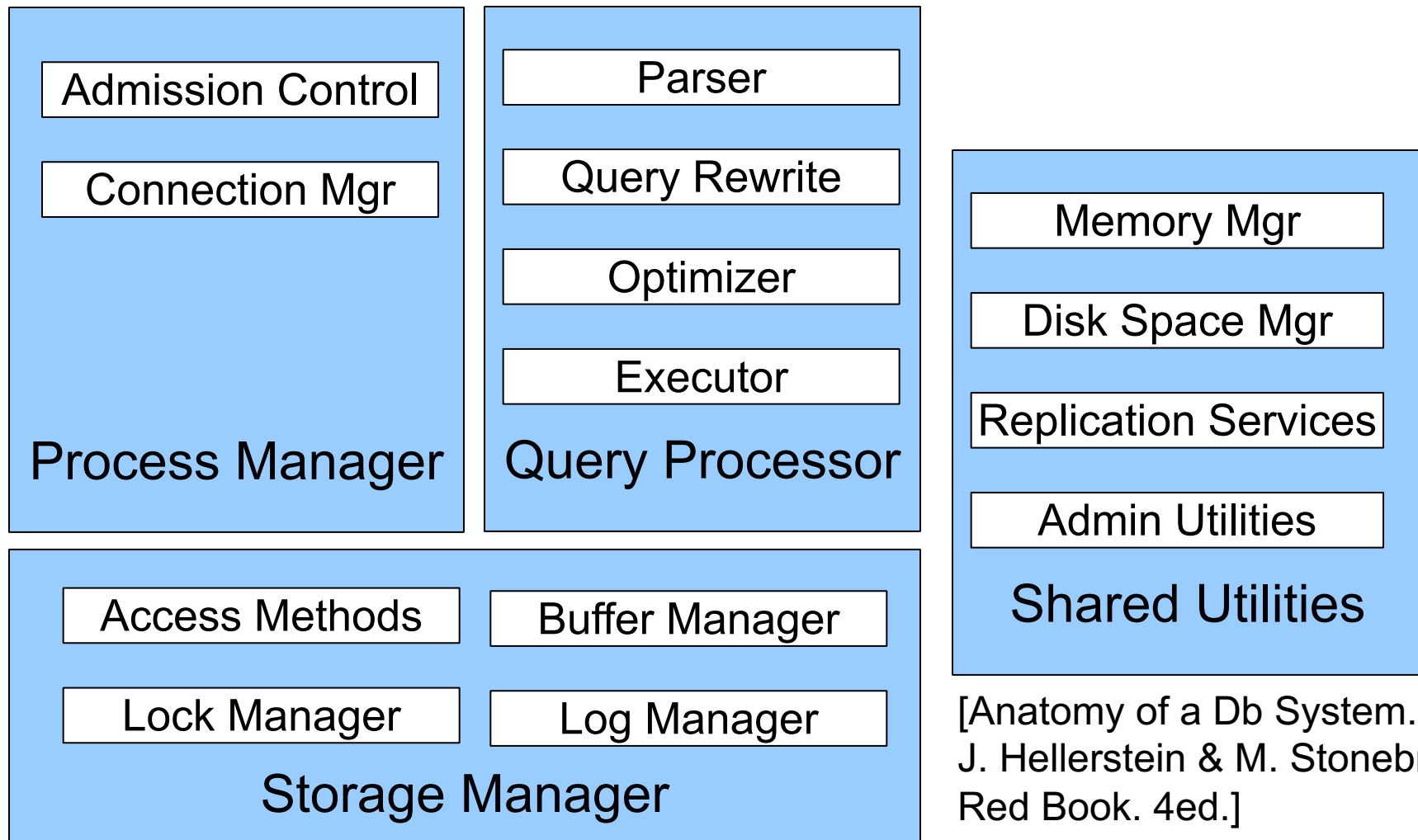
References

- Ailamaki et al. *Weaving Relations for Cache Performance*, VLDB'2001
- Daniel Abadi, Peter Boncz, Stavros Harizopoulos, Stratos Idreos, Samuel Madden. *The Design and Implementation of Modern Column-Oriented Database Systems* Foundations and Trends® in Databases 2012

Column Store Popularity Gain

- C-store ideas and research since 1970's
- **2004:** C-store research prototype at MIT
 - Started by Mike Stonebraker
 - Lead graduate student Daniel Abadi
- **2005:** Vertica founded by M. Stonebraker & A. Palmer
- **2011:** Vertica acquired by HP
- **2012:** As of VLDB'12 paper, 500 production deployments of Vertica, three over a PB in size
- **2013:** All **major DB vendors** include some column-store implementation

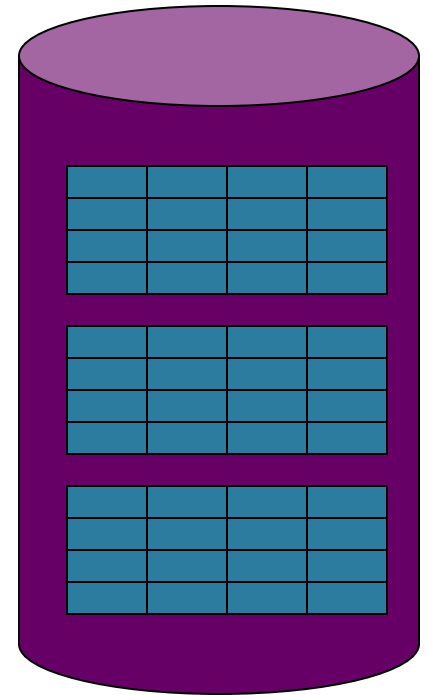
DBMS Architecture



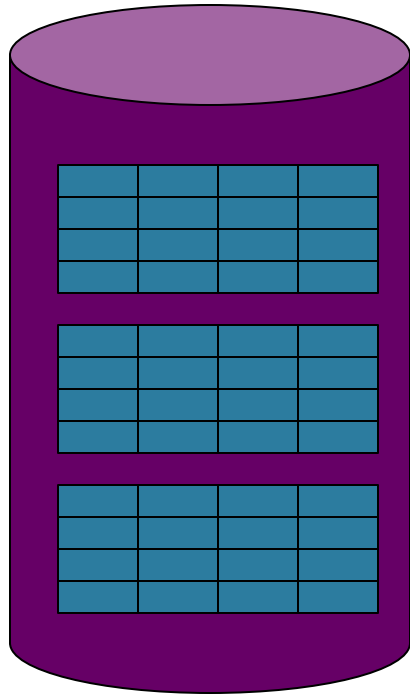
[Anatomy of a Db System.
J. Hellerstein & M. Stonebraker.
Red Book. 4ed.]

Working with Pages

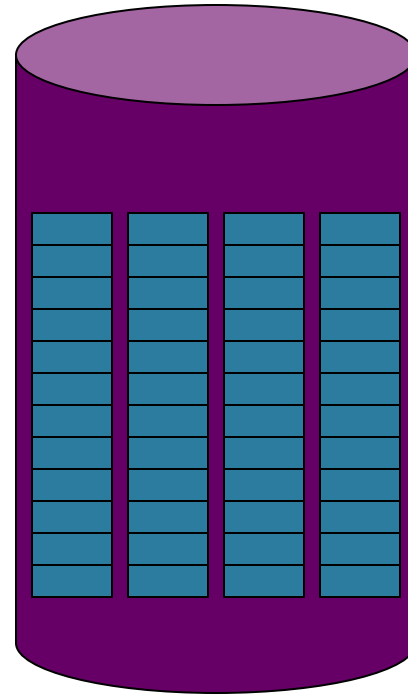
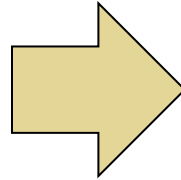
- Reading/writing to/from **disk**
 - Seeking takes a long time!
 - Reading sequentially is fast
- To simplify **buffer manager**, want to cache a collection of same-sized objects
- Solution: **Read/write pages** of data
 - A page should correspond to a disk block



From Row-Store to Column-Store



Rows stored
contiguously on disk
(+ tuple headers)



Columns stored
contiguously on disk
(no headers needed)

C-Store Illustration

Row-based
(4 pages)

Page {

A	1
A	2
A	2
A	2
B	2
B	4
C	4
C	4

Column-based
(4 pages)

A	1
A	2
A	2
A	2
B	2
B	4
C	4
C	4

} Page

C-Store also
avoids large
tuple headers

Column-Oriented Databases

- Main idea:
 - **Physical storage**: complete vertical partition; each column stored separately: R.A, R.B, R.A
 - **Logical schema**: remains the same R(A,B,C)
- Main advantage:
 - **Improved transfer rate**: disk to memory, memory to CPU, better cache locality

Basic Trade-Off

- Row stores
 - Quick to update entire tuple (1 page IO)
 - Quick to access a single tuple
- Column stores
 - Avoid reading unnecessary columns
 - Better compression
- Entire system needs a different design
 - Not only storage manager
 - To achieve high performance

An Intermediate Format: PAX

- PAX = Partition Attributes Across
- Addresses memory access bottleneck (not the disk bottleneck)

From Row to Column Storage (Initial Designs - 1985)

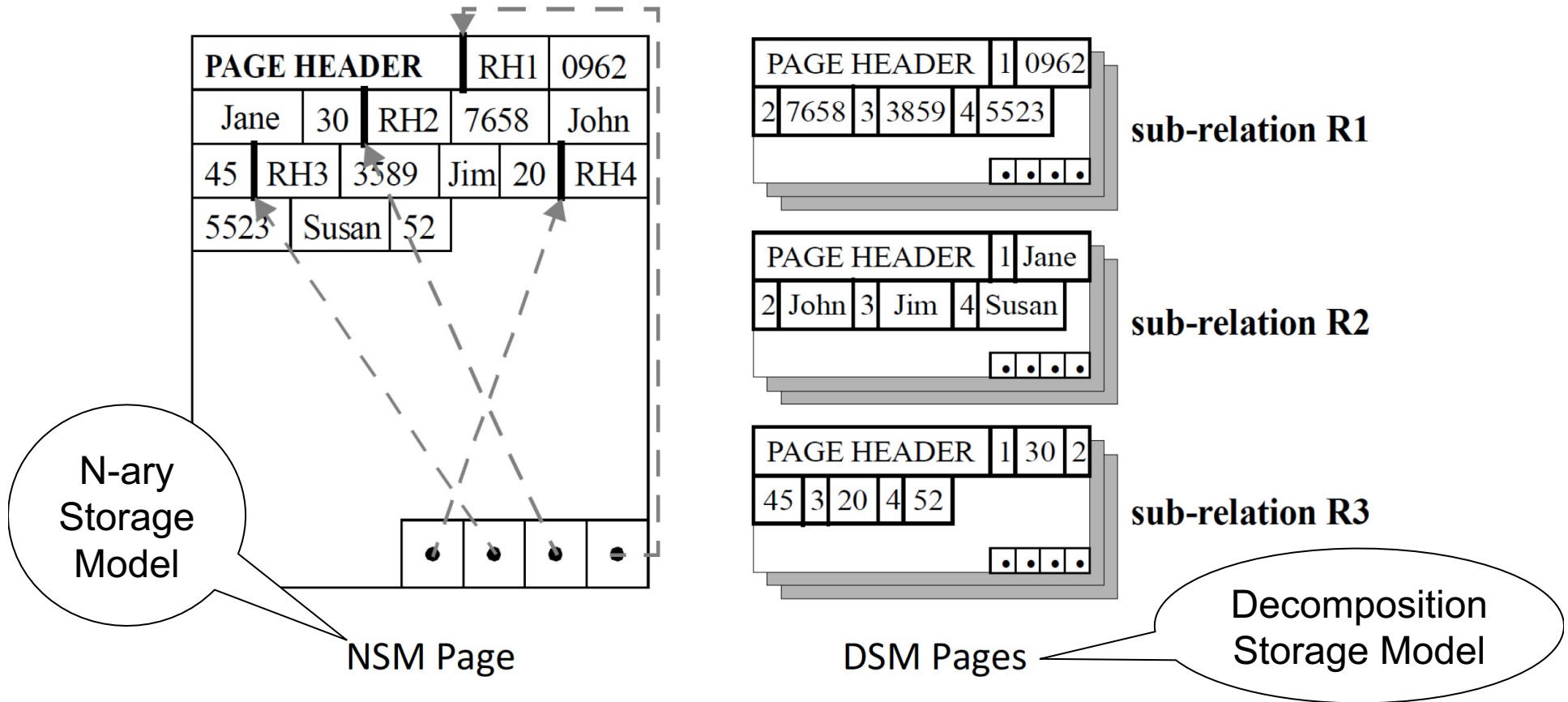
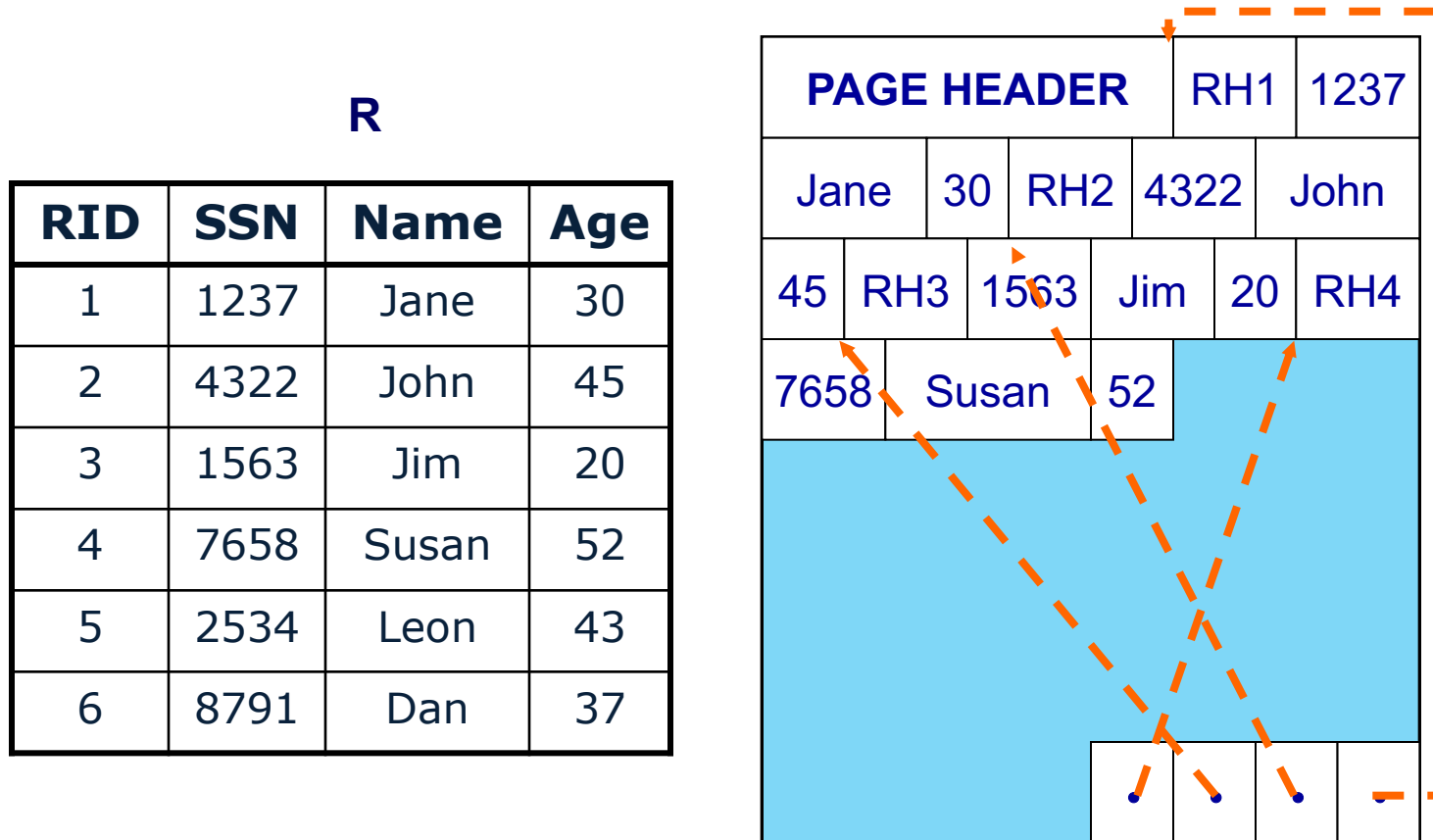


Figure 2.1: Storage models for storing database records inside disk pages: NSM (row-store) and DSM (a predecessor to column-stores). Figure taken from [5].

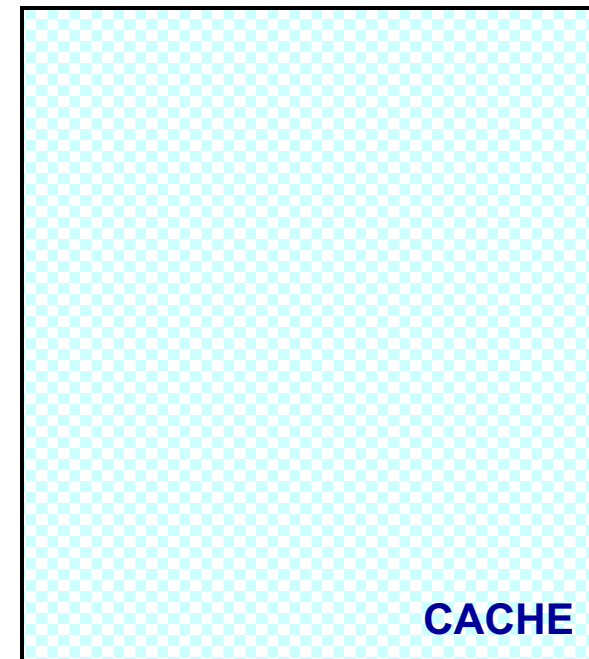
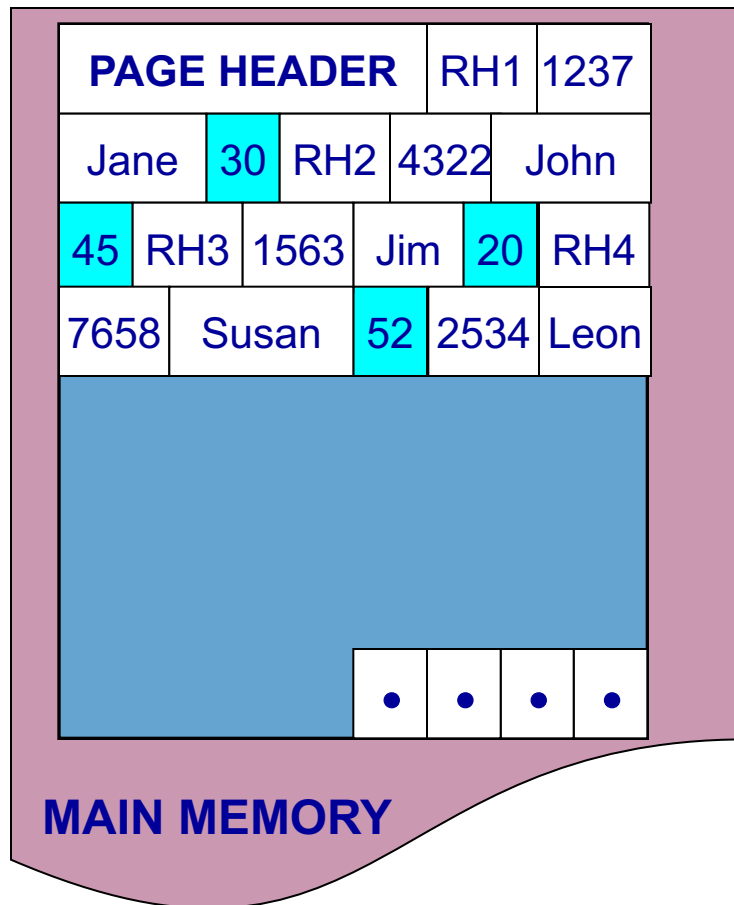
Current Scheme: Slotted Pages

Formal name: NSM (N-ary Storage Model)



- ❑ Records are stored sequentially
- ❑ Offsets to start of each record at end of page

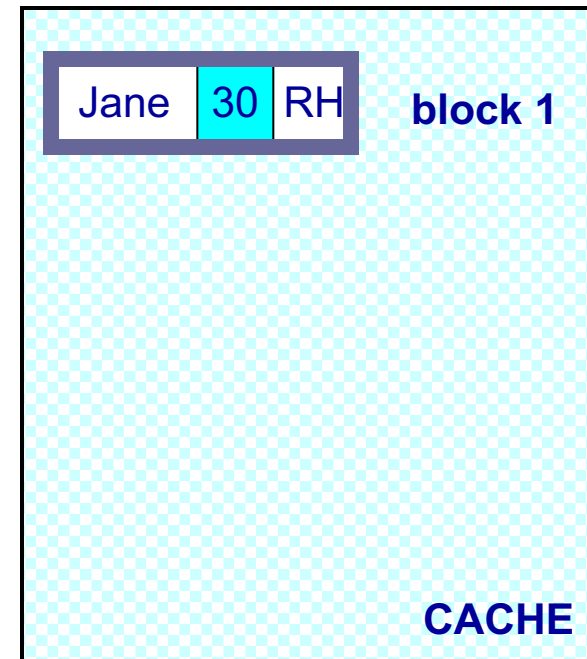
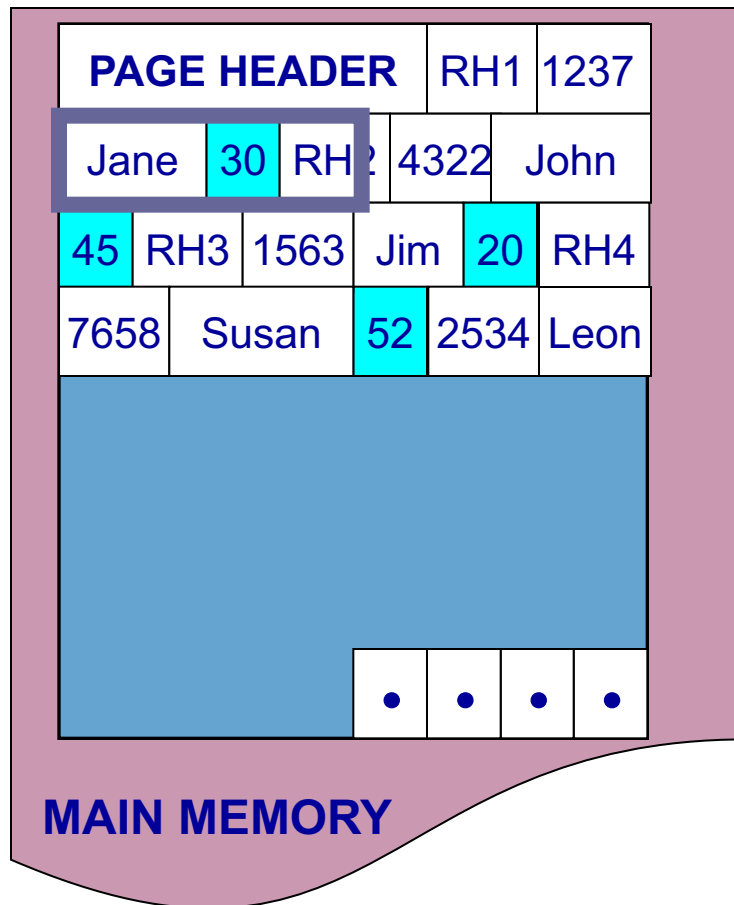
Predicate Evaluation using NSM



*select name
from R
where age > 50*

NSM pushes non-referenced data to the cache

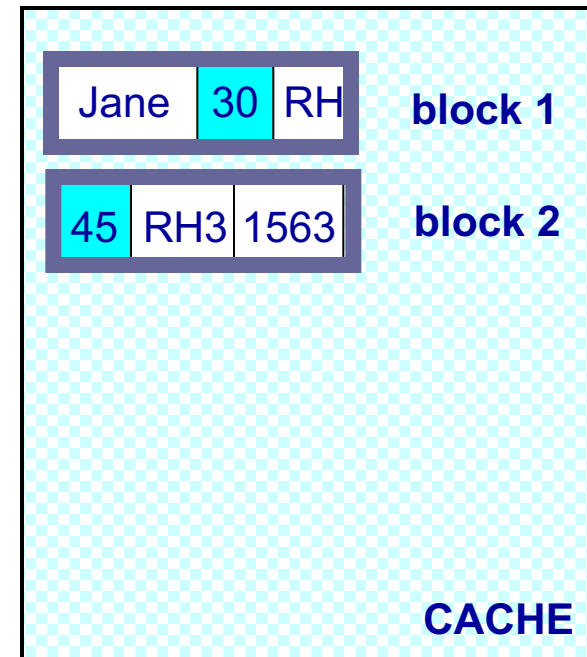
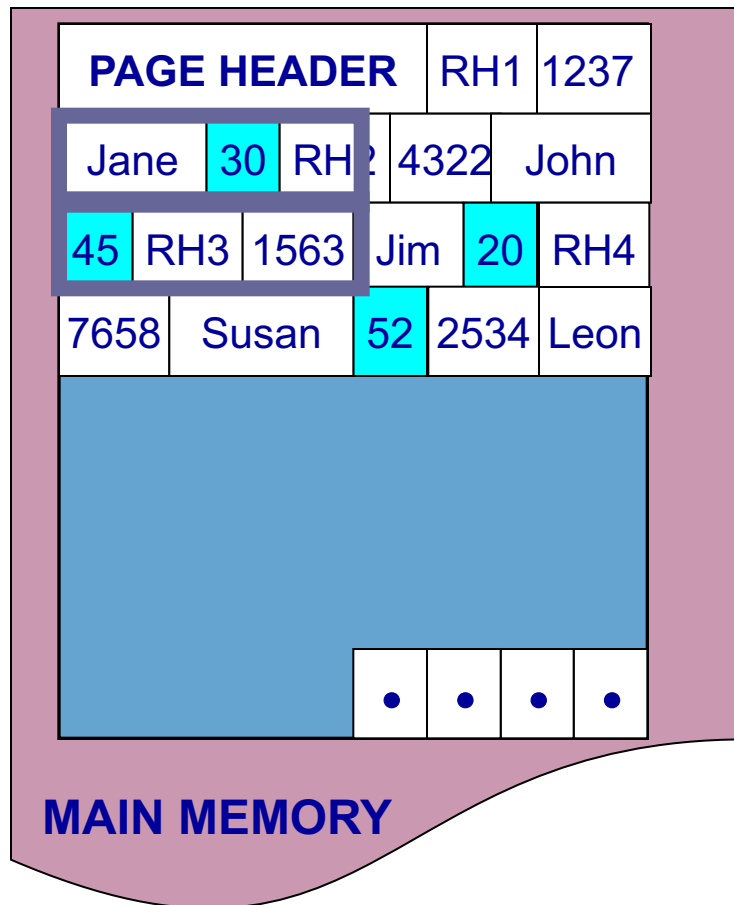
Predicate Evaluation using NSM



*select name
from R
where age > 50*

NSM pushes non-referenced data to the cache

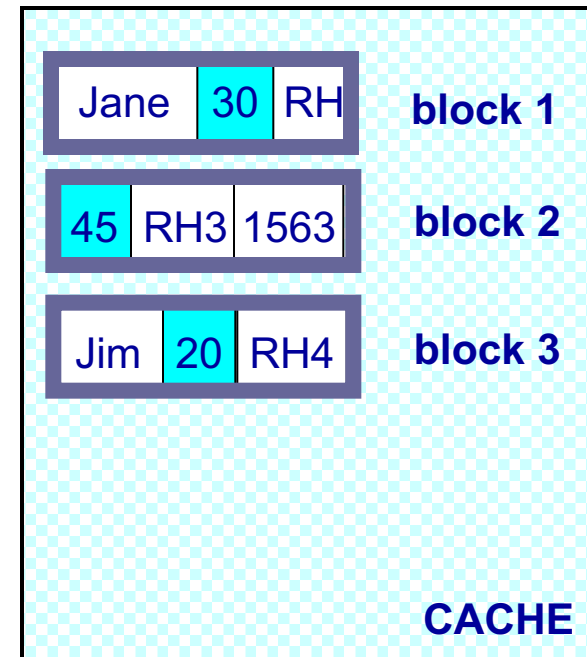
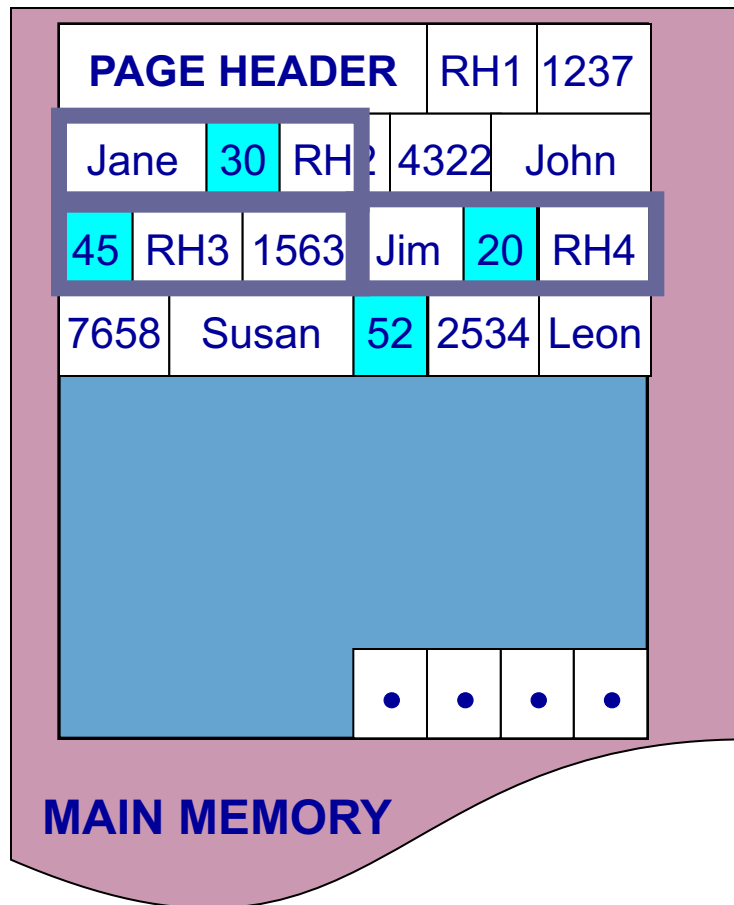
Predicate Evaluation using NSM



*select name
from R
where age > 50*

NSM pushes non-referenced data to the cache

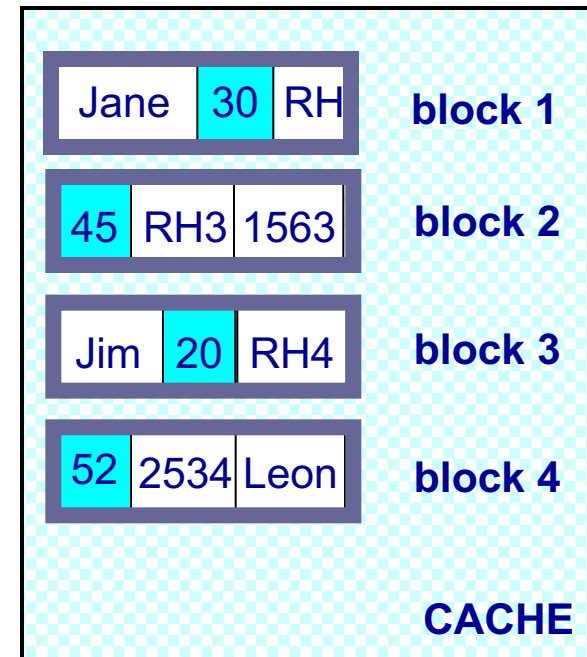
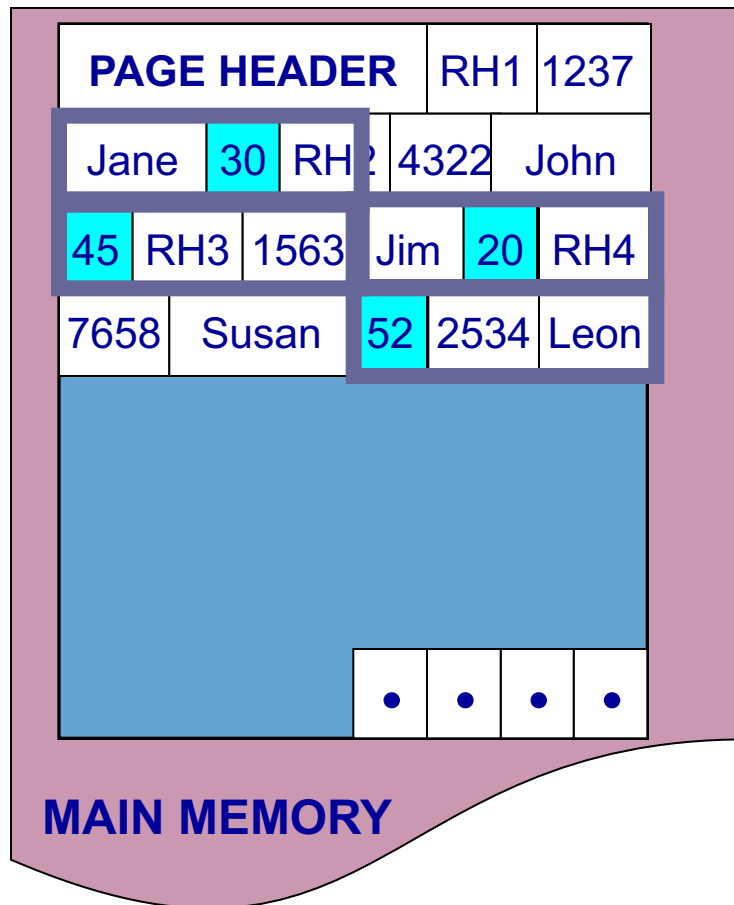
Predicate Evaluation using NSM



*select name
from R
where age > 50*

NSM pushes non-referenced data to the cache

Predicate Evaluation using NSM



*select name
from R
where age > 50*

NSM pushes non-referenced data to the cache

Need New Data Page Layout

- Eliminates unnecessary memory accesses
- Improves inter-record locality
- Keeps a record's fields together
- Does not affect I/O performance

and, most importantly, is...

low-implementation-cost, high-impact

Partition Attributes Across (PAX)

NSM PAGE

PAGE HEADER				RH1	1237	
Jane		30	RH2	4322	John	
45	RH3		1563	Jim	20	RH4
7658		Susan		52		
					•	•
					•	•

PAX PAGE

PAGE HEADER				1237	4322				
1563		7658							
Jane		John		Jim		Susan			

Partition data *within* the page for spatial locality

Partition Attributes Across (PAX)

NSM PAGE

PAGE HEADER				RH1	1237
Jane	30	RH2	4322	John	
45	RH3	1563	Jim	20	RH4
7658	Susan		52		
				.	.
				.	.

PAX PAGE

PAGE HEADER				1237	4322
1563	7658				
Jane	John	Jim	Susan		
				.	.
30	52	45	20		

Partition data *within* the page for spatial locality

Partition Attributes Across (PAX)

NSM PAGE

PAGE HEADER			RH1	1237		
Jane	30	RH2	4322	John		
45	RH3	1563	Jim	20		
7658	Susan	52				
			.	.		
			.	.		
			.	.		
			.	.		

PAX PAGE

PAGE HEADER				1237	4322
1563	7658				
Jane	John	Jim	Susan		
				.	.
30	52	45	20		

Partition data *within* the page for spatial locality

Partition Attributes Across (PAX)

NSM PAGE

PAGE HEADER				RH1	1237
Jane	30	RH2	4322	John	
45	RH3	1563	Jim	20	RH4
7658	Susan	52			
				.	.
				.	.

PAX PAGE

PAGE HEADER				1237	4322
1563	7658				
Jane	John	Jim	Susan		
30	52	45	20		

Partition data *within* the page for spatial locality

Partition Attributes Across (PAX)

NSM PAGE

[illegible]

PAX PAGE

PAGE HEADER				1237	4322
1563	7658				
Jane	John	Jim	Susan		
				.	.
30	52	45	20	.	.

Partition data *within* the page for spatial locality

Partition Attributes Across (PAX)

NSM PAGE

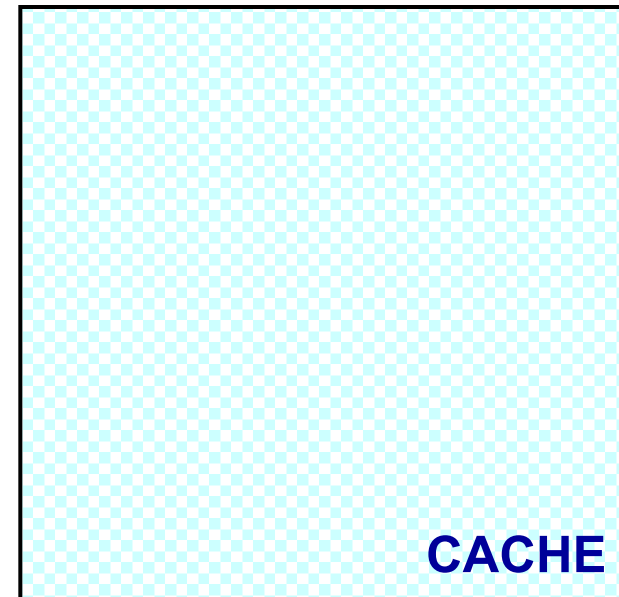
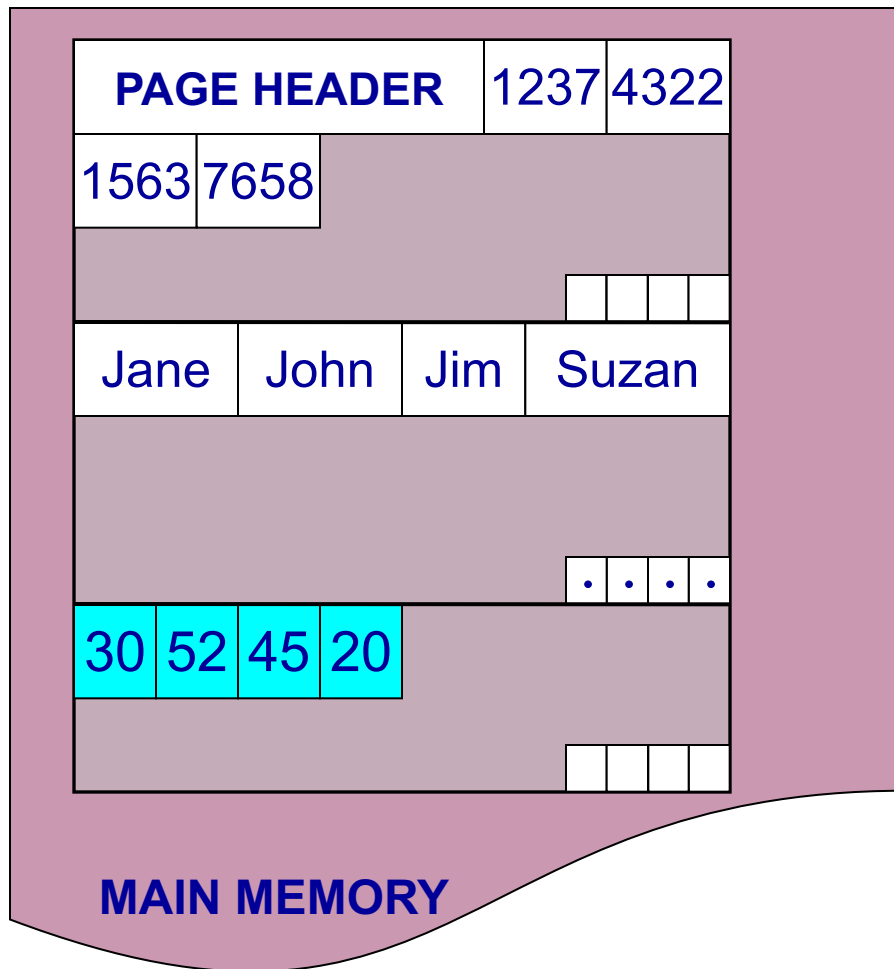
[illegible]

PAX PAGE

PAGE HEADER			1237	4322				
1563	7658							
Jane			John	Jim	Susan			
					.	.	.	.
30	52	45	20					

Partition data *within* the page for spatial locality

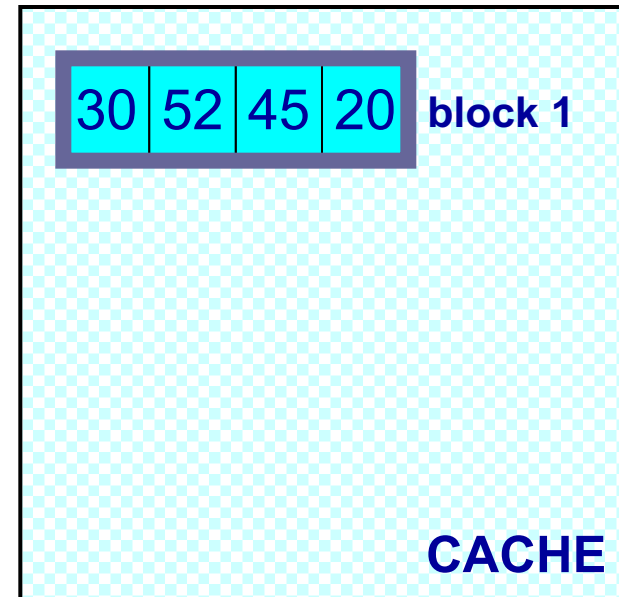
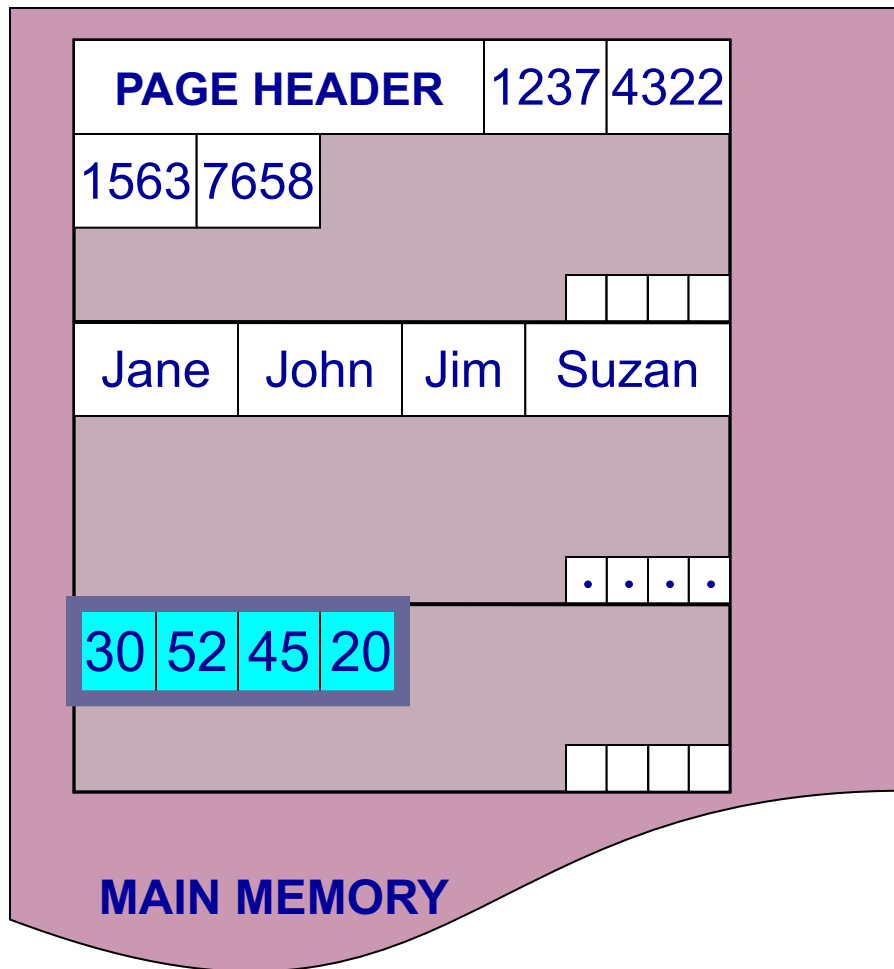
Predicate Evaluation using PAX



*select name
from R
where age > 50*

Fewer cache misses, low reconstruction cost

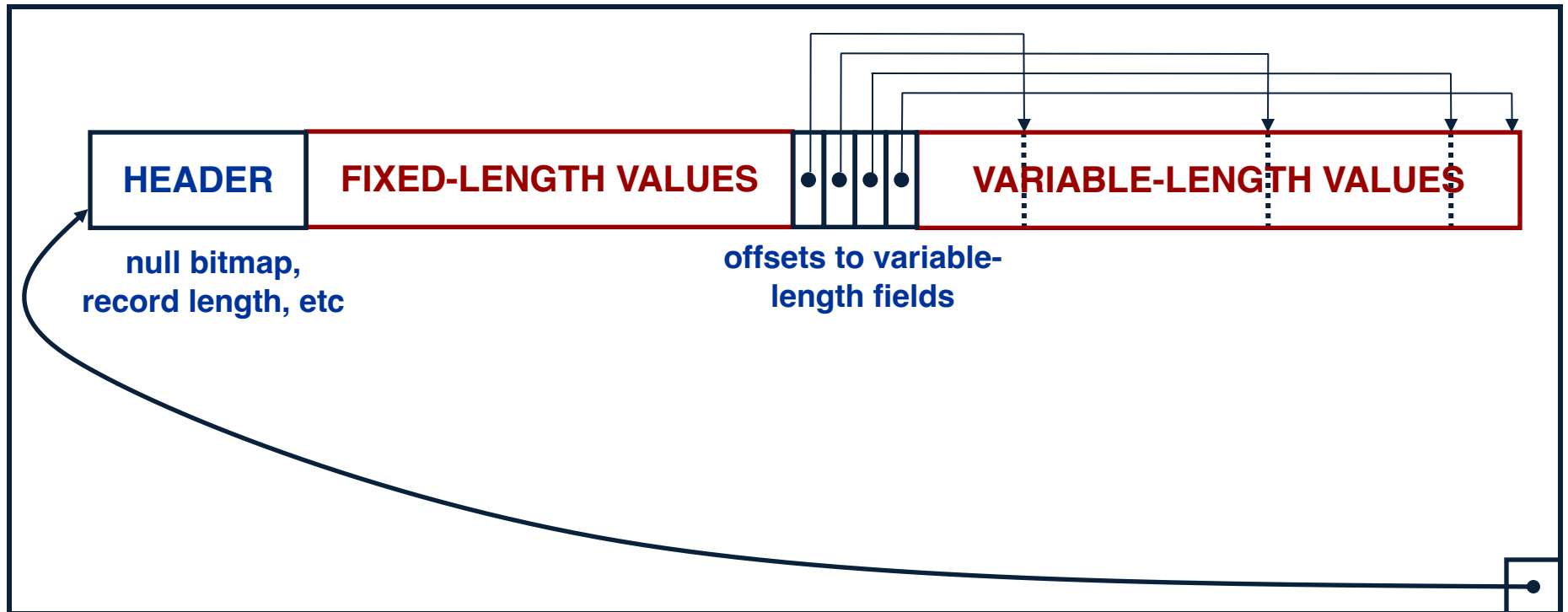
Predicate Evaluation using PAX



*select name
from R
where age > 50*

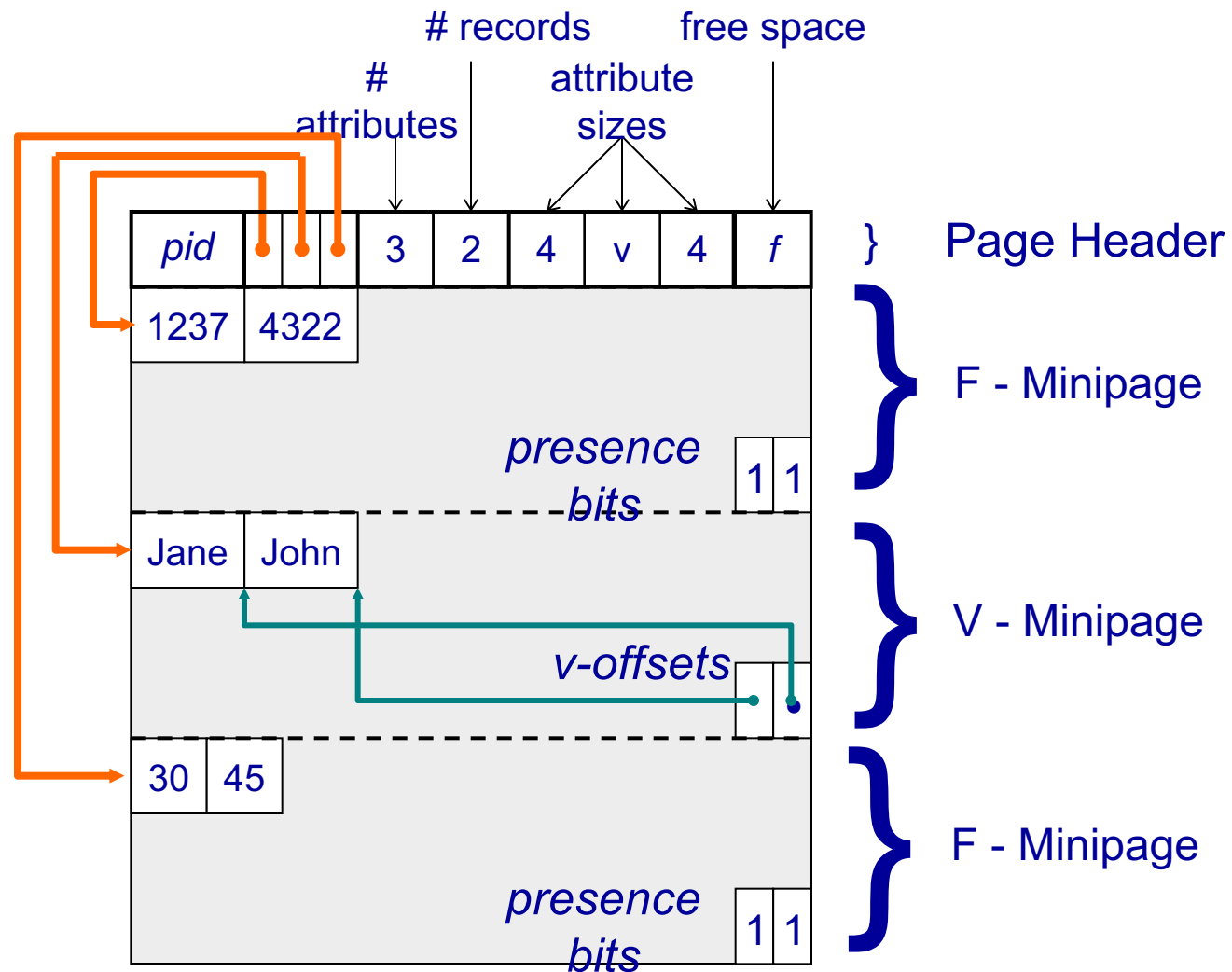
Fewer cache misses, low reconstruction cost

A Real NSM Record



NSM: All fields of record stored together + slots

PAX: Detailed Design



PAX: Group fields + amortizes record headers

PAX - Summary

- Improves processor cache locality
- Does not affect I/O behavior
 - Same disk accesses for NSM or PAX storage
 - No need to change the buffer manager
- Column stores:
 - Store each attribute in a different file

From Row to Column Storage (Modern Designs)

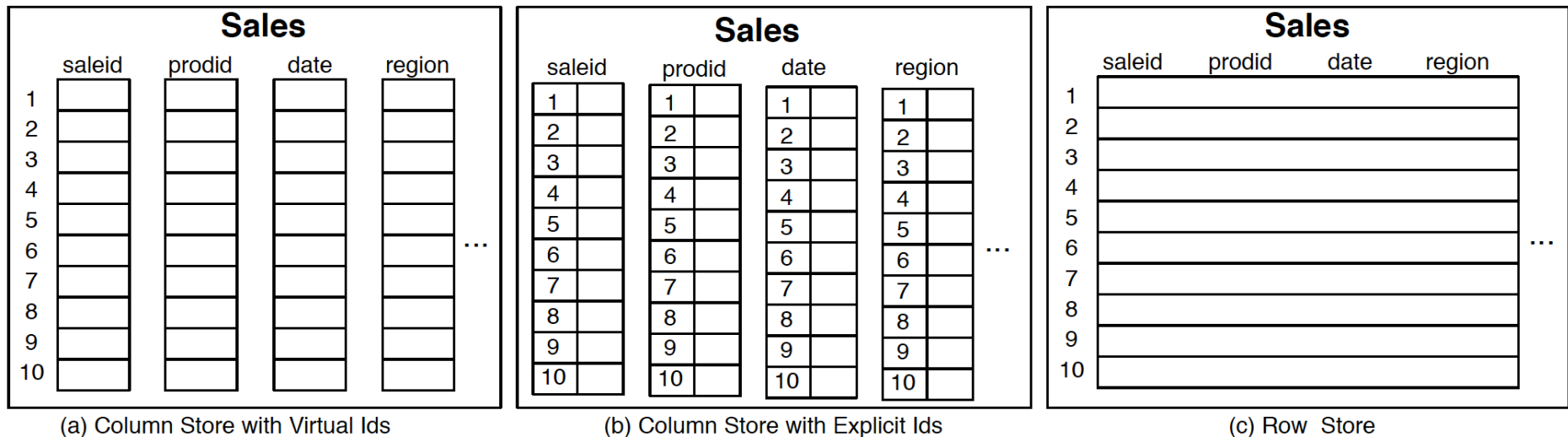


Figure 1.1: Physical layout of column-oriented vs row-oriented databases.

Basic tradeoffs:

- Reading all attributes of one records, v.s.
- Reading some attributes of many records