ECS 165: Database Systems

Project’s Second Milestone

Winter 2025

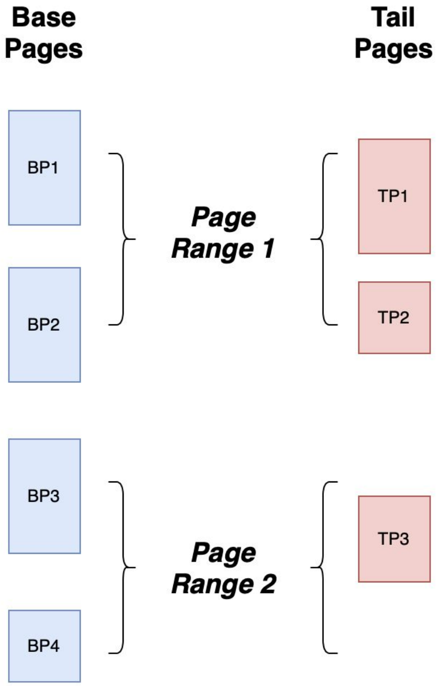
Note: There are typos in the diagrams, all ‘Indirectory’s should be corrected to ‘Indirection’s

UCDAVIS
UNIVERSITY OF CALIFORNIA

Page Range

- Records are virtually partitioned into disjoint page ranges.
- Each page range consists of a set of base pages.
- Each page range also consists of a set of tail pages.
- Tail pages are the granularity of the page range, not base pages.

Page Range



- Every page range consists of 2 base pages.
- Tail records of PageRange1 are written to TP2 since TP1 is full.
- Tail records of PageRange2 are written to TP3 until TP3 is full.
- Then, we allocate a new tail page for PageRange2 once TP3 is full.

Durability

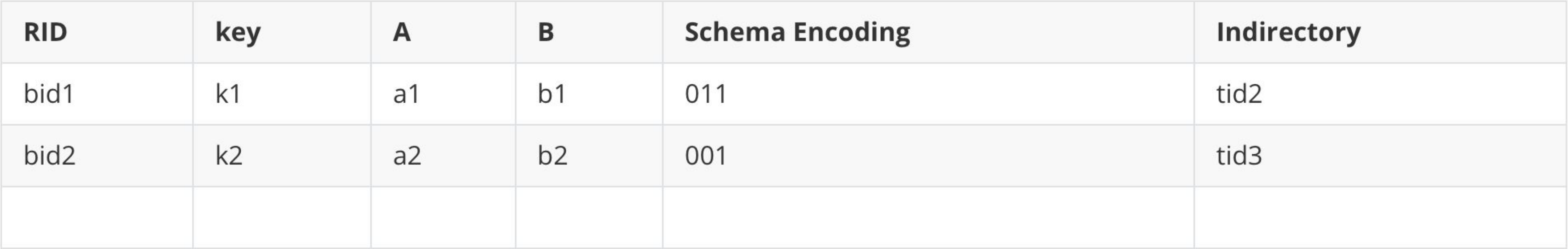
- Store data to the disk
  - Record Data
  - Table Information (name, column number, primary key column, what columns have indexes)
  - Indexes (Optional)
- Load data from the disk
- Generate Index while loading data
  - Load the Index file: fast but more complicated for programmers
  - Generate the Index by loading all the data: slow but easier for programmers

Durability

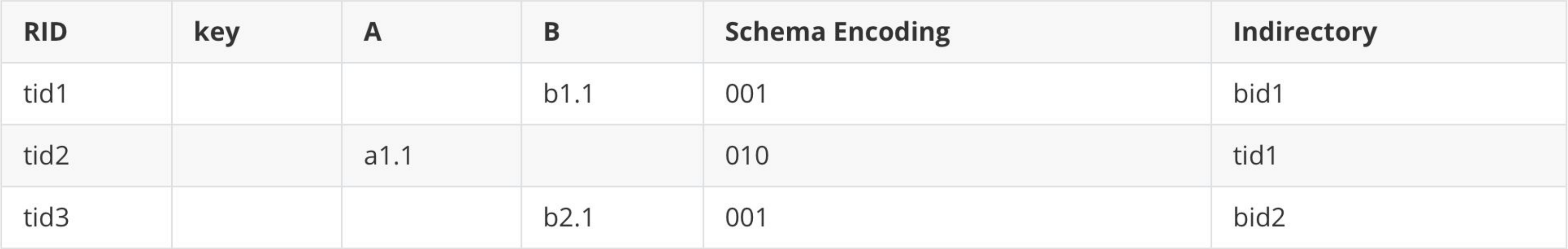
- Page Directory
  - What if the page is in memory?
  - What if the page is on disk?
- Page Replacement
  - Least Recently Used (LRU)

Merge

What is inefficient?



RID	key	A	B	Schema Encoding	Indirectory
bid1	k1	a1	b1	011	tid2
bid2	k2	a2	b2	001	tid3

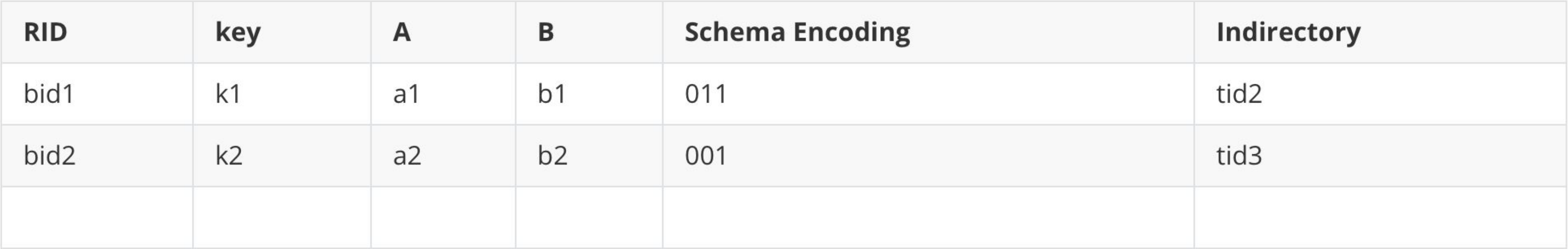


RID	key	A	B	Schema Encoding	Indirectory
tid1			b1.1	001	bid1
tid2		a1.1		010	tid1
tid3			b2.1	001	bid2

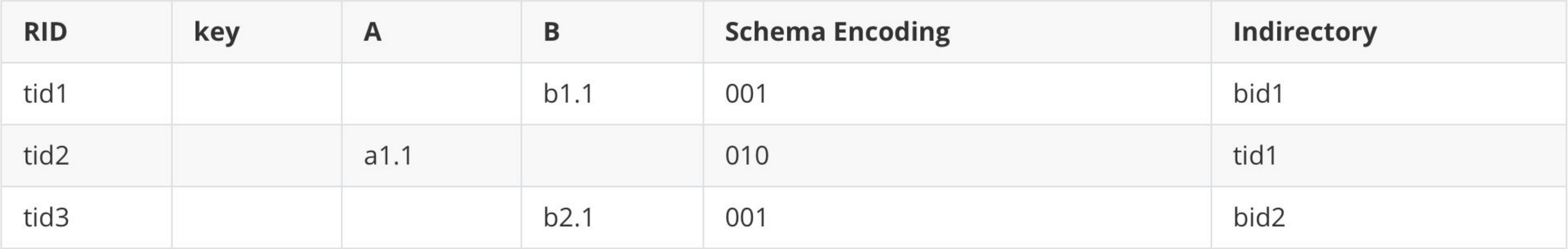
Merge

What is inefficient?

Update-friendly but not Read-friendly



RID	key	A	B	Schema Encoding	Indirectory
bid1	k1	a1	b1	011	tid2
bid2	k2	a2	b2	001	tid3



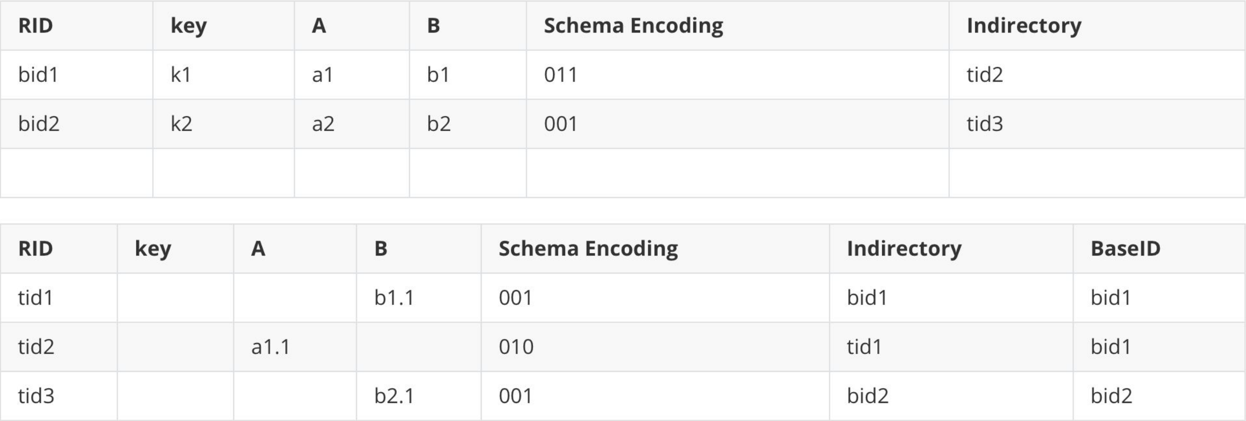
RID	key	A	B	Schema Encoding	Indirectory
tid1			b1.1	001	bid1
tid2		a1.1		010	tid1
tid3			b2.1	001	bid2

Merge

- To alleviate the performance degradation, the tail pages (or tail records) are periodically merged with their corresponding base pages (or base records) in order to bring base pages “almost” up-to-date.
- 5-steps:
  - Step 1: Identify committed tail records in tail pages
  - Step 2: Load the corresponding outdated base pages
  - Step 3: Consolidate the base and tail pages
  - Step 4: Update the page directory
  - Step 5: De-allocate the outdated base pages

Merge

- Step 1: Identify committed tail records in tail pages
- Step 2: Load the corresponding outdated base pages
  - BaseID: tracks the RID of the original base record for each tail record.



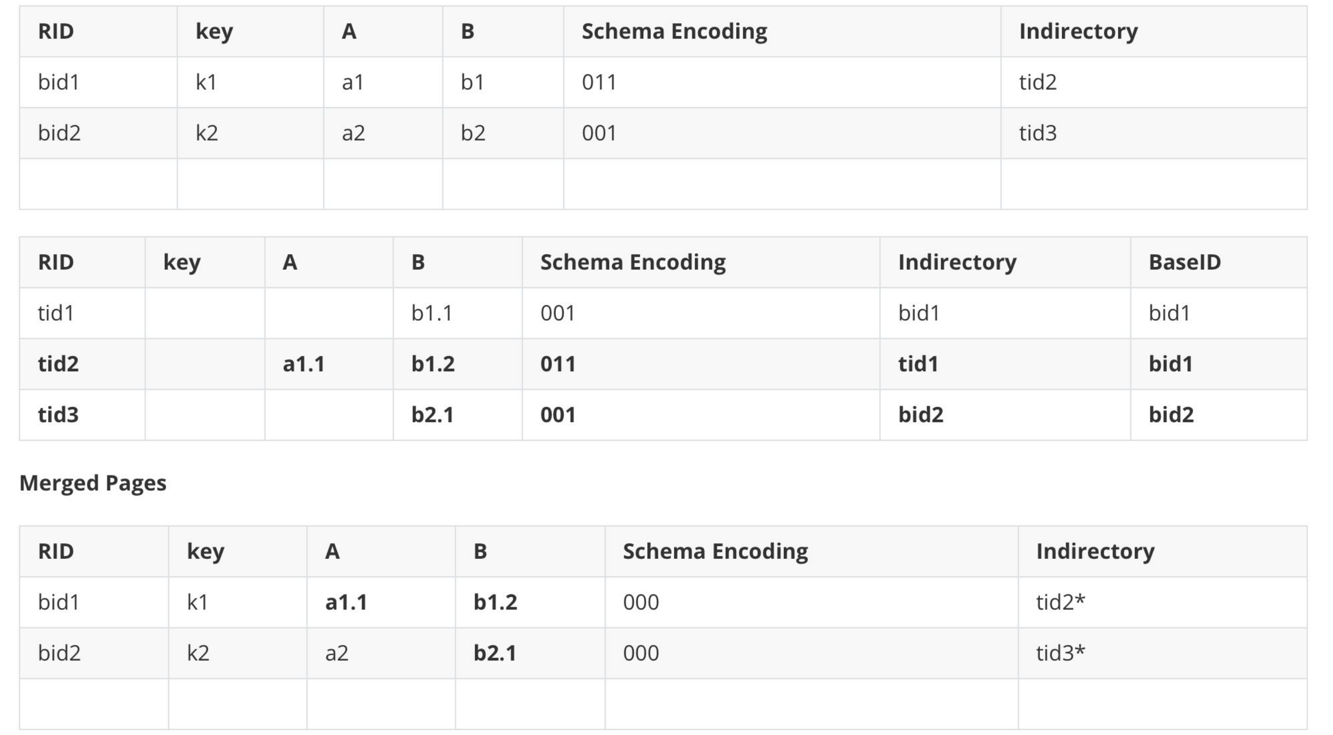
RID	key	A	B	Schema Encoding	Indirectory
bid1	k1	a1	b1	011	tid2
bid2	k2	a2	b2	001	tid3

RID	key	A	B	Schema Encoding	Indirectory	BaseID
tid1			b1.1	001	bid1	bid1
tid2		a1.1		010	tid1	bid1
tid3			b2.1	001	bid2	bid2

Merge

- Step 3: Consolidate the base and tail pages
  - For every updated column, first the BaseID column of the committed tail pages (from Step 1) are scanned in reverse order to find the list of the latest version of every updated record since the last merge.
  - Second, apply the latest tail records in a reverse order to the base records until the list is exhausted.

Merge



RID	key	A	B	Schema Encoding	Indirectory
bid1	k1	a1	b1	011	tid2
bid2	k2	a2	b2	001	tid3

RID	key	A	B	Schema Encoding	Indirectory	BaseID
tid1			b1.1	001	bid1	bid1
tid2		**a1.1**	**b1.2**	**011**	**tid1**	**bid1**
tid3			**b2.1**	**001**	**bid2**	**bid2**

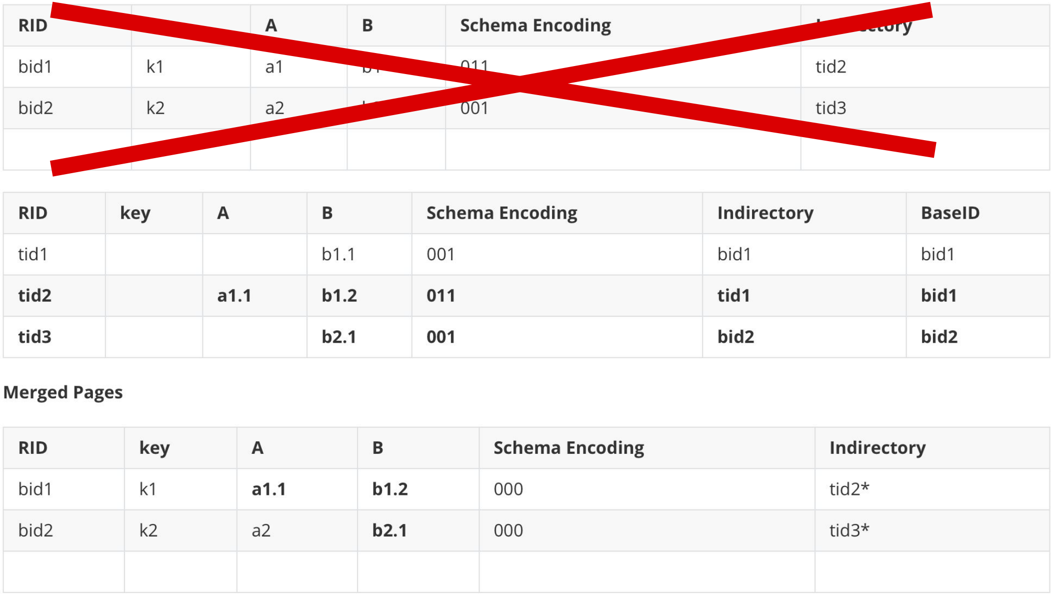
Merged Pages

RID	key	A	B	Schema Encoding	Indirectory
bid1	k1	**a1.1**	**b1.2**	000	tid2*
bid2	k2	a2	**b2.1**	000	tid3*

Merge

- Step 4: Update the page directory
- Step 5: De-allocate the outdated base pages

What is the problem?



~~RID~~		~~A~~	~~B~~	~~Schema Encoding~~	~~Indirectory~~
~~bid1~~	~~k1~~	~~a1~~	~~b1~~	~~011~~	~~tid2~~
~~bid2~~	~~k2~~	~~a2~~	~~b2~~	~~001~~	~~tid3~~

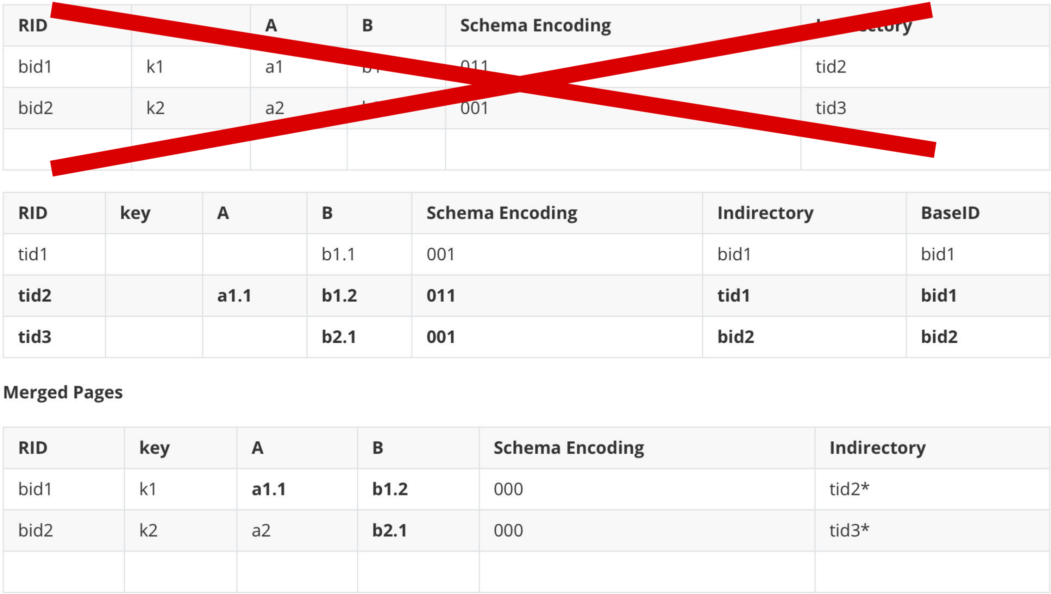
RID	key	A	B	Schema Encoding	Indirectory	BaseID
tid1			b1.1	001	bid1	bid1
tid2		**a1.1**	**b1.2**	**011**	**tid1**	**bid1**
tid3			**b2.1**	**001**	**bid2**	**bid2**

Merged Pages

RID	key	A	B	Schema Encoding	Indirectory
bid1	k1	**a1.1**	**b1.2**	000	tid2*
bid2	k2	a2	**b2.1**	000	tid3*

Merge

- Step 4: Update the page directory
- Step 5: De-allocate the outdated base pages



RID		A	B	Schema Encoding	Indirectory
bid1	k1	a1	b1	011	tid2
bid2	k2	a2	b2	001	tid3

RID	key	A	B	Schema Encoding	Indirectory	BaseID
tid1			b1.1	001	bid1	bid1
tid2		**a1.1**	**b1.2**	**011**	**tid1**	**bid1**
tid3			**b2.1**	**001**	**bid2**	**bid2**

Merged Pages

RID	key	A	B	Schema Encoding	Indirectory
bid1	k1	**a1.1**	**b1.2**	000	tid2*
bid2	k2	a2	**b2.1**	000	tid3*

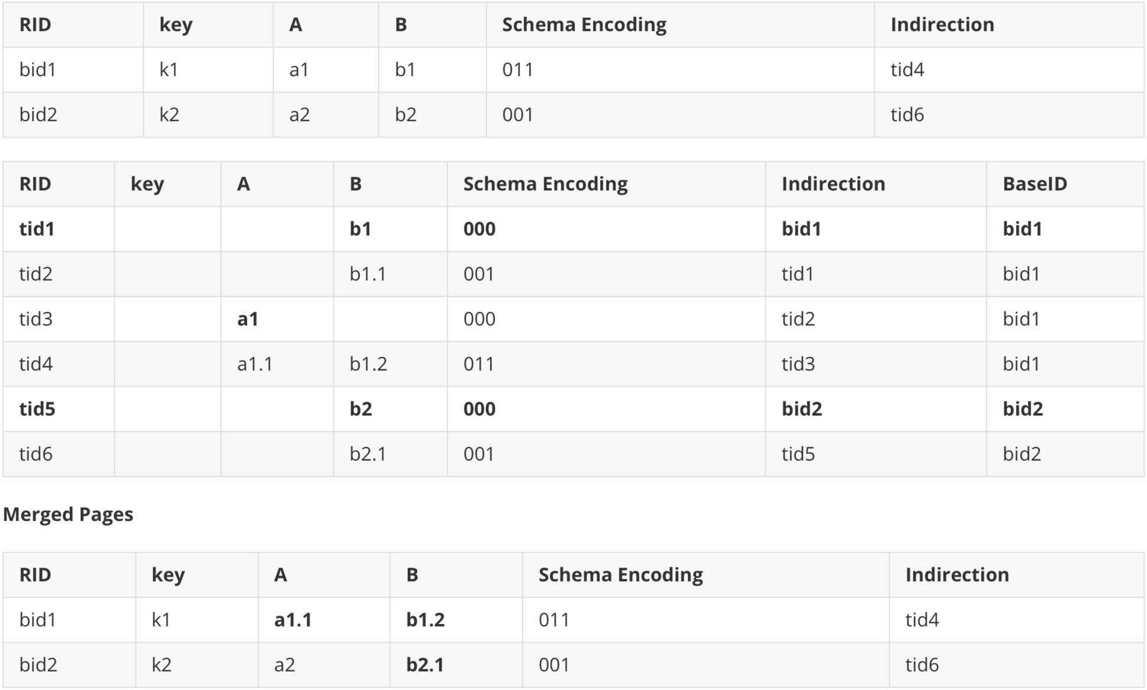
What is the problem?

We will lose the original version of updated columns.

UCDAVIS

Merge

- Make a copy of the original value when updating a column for the first time



RID	key	A	B	Schema Encoding	Indirection
bid1	k1	a1	b1	011	tid4
bid2	k2	a2	b2	001	tid6

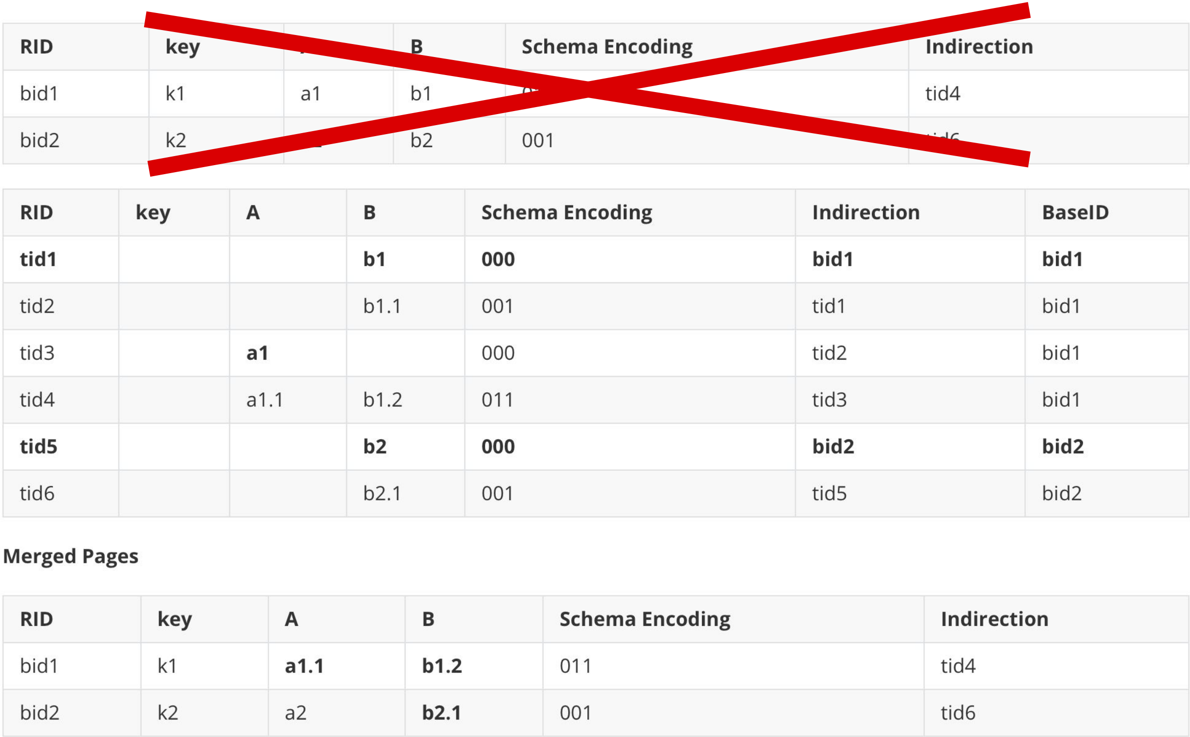
RID	key	A	B	Schema Encoding	Indirection	BaseID
tid1			**b1**	**000**	**bid1**	**bid1**
tid2			b1.1	001	tid1	bid1
tid3		**a1**		000	tid2	bid1
tid4		a1.1	b1.2	011	tid3	bid1
tid5			**b2**	**000**	**bid2**	**bid2**
tid6			b2.1	001	tid5	bid2

Merged Pages

RID	key	A	B	Schema Encoding	Indirection
bid1	k1	**a1.1**	**b1.2**	011	tid4
bid2	k2	a2	**b2.1**	001	tid6

Merge

- Make a copy of the original value when updating a column for the first time



RID	key	A	B	Schema Encoding	Indirection
~~bid1~~	~~k1~~	~~a1~~	~~b1~~	~~[illegible]~~	~~tid4~~
~~bid2~~	~~k2~~	~~a2~~	~~b2~~	~~001~~	~~tid6~~

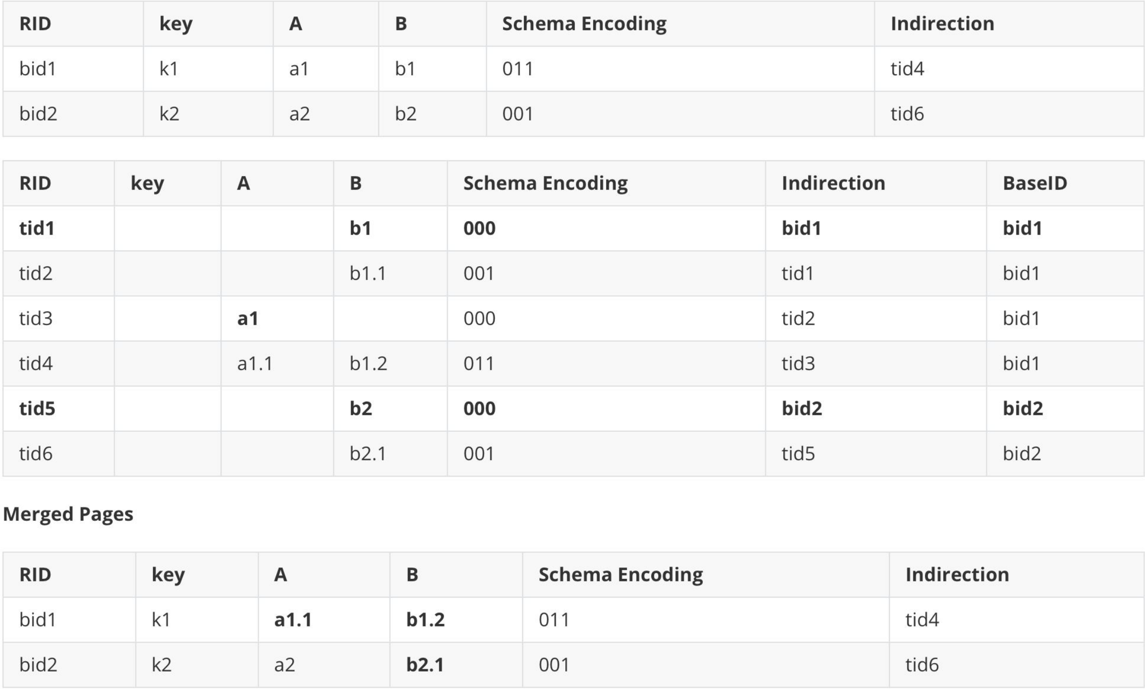
RID	key	A	B	Schema Encoding	Indirection	BaseID
tid1			**b1**	**000**	**bid1**	**bid1**
tid2			b1.1	001	tid1	bid1
tid3		**a1**		000	tid2	bid1
tid4		a1.1	b1.2	011	tid3	bid1
tid5			**b2**	**000**	**bid2**	**bid2**
tid6			b2.1	001	tid5	bid2

Merged Pages

RID	key	A	B	Schema Encoding	Indirection
bid1	k1	**a1.1**	**b1.2**	011	tid4
bid2	k2	a2	**b2.1**	001	tid6

Tail-Page Sequence Number (TPS)

What is the problem?



RID	key	A	B	Schema Encoding	Indirection
bid1	k1	a1	b1	011	tid4
bid2	k2	a2	b2	001	tid6

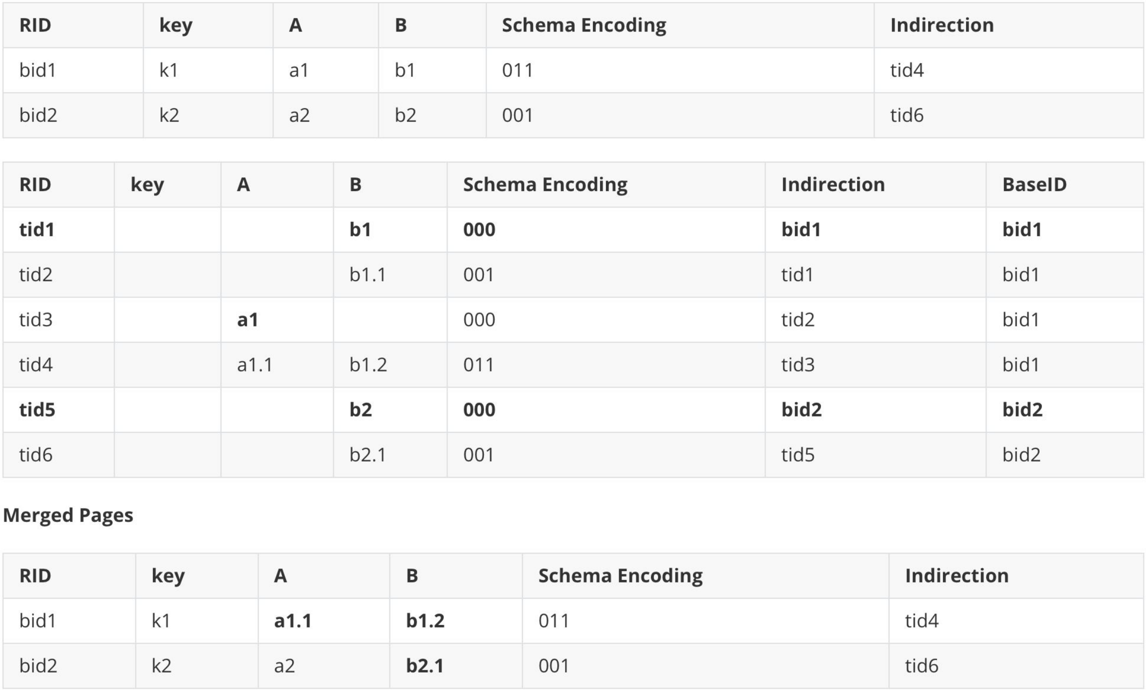
RID	key	A	B	Schema Encoding	Indirection	BaseID
tid1			**b1**	**000**	**bid1**	**bid1**
tid2			b1.1	001	tid1	bid1
tid3		**a1**		000	tid2	bid1
tid4		a1.1	b1.2	011	tid3	bid1
tid5			**b2**	**000**	**bid2**	**bid2**
tid6			b2.1	001	tid5	bid2

Merged Pages

RID	key	A	B	Schema Encoding	Indirection
bid1	k1	**a1.1**	**b1.2**	011	tid4
bid2	k2	a2	**b2.1**	001	tid6

Tail-Page Sequence Number (TPS)

What is the problem? We do not know if the the base records have the latest values



RID	key	A	B	Schema Encoding	Indirection
bid1	k1	a1	b1	011	tid4
bid2	k2	a2	b2	001	tid6

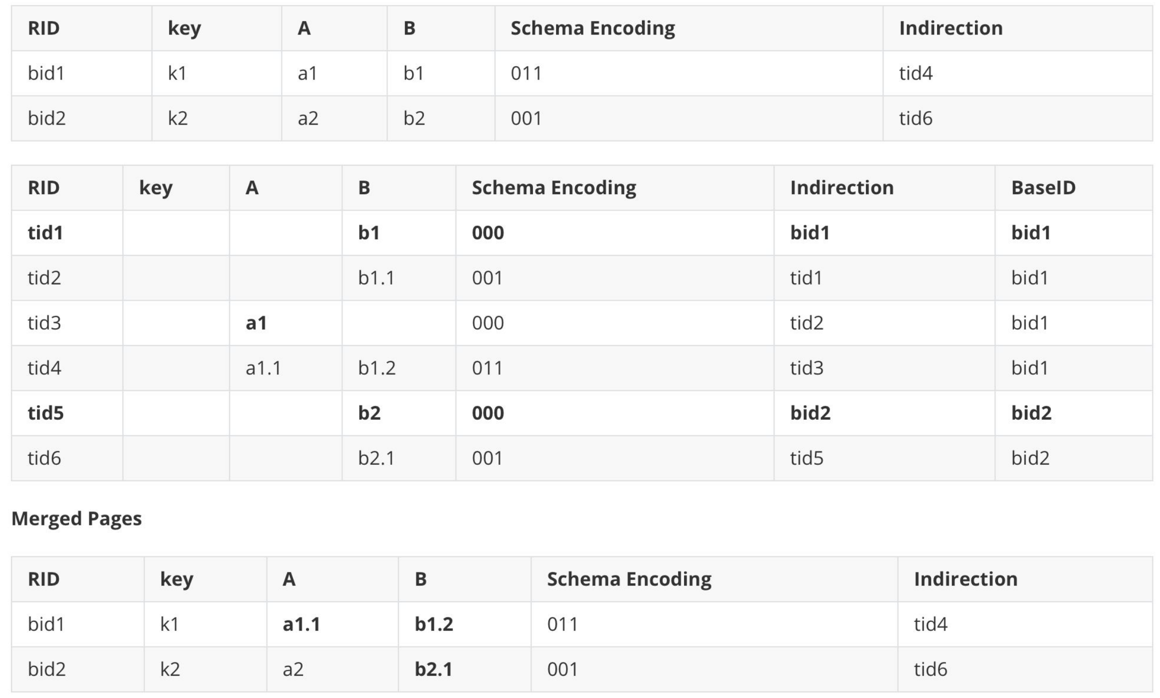
RID	key	A	B	Schema Encoding	Indirection	BaseID
tid1			**b1**	**000**	**bid1**	**bid1**
tid2			b1.1	001	tid1	bid1
tid3		**a1**		000	tid2	bid1
tid4		a1.1	b1.2	011	tid3	bid1
tid5			**b2**	**000**	**bid2**	**bid2**
tid6			b2.1	001	tid5	bid2

Merged Pages

RID	key	A	B	Schema Encoding	Indirection
bid1	k1	**a1.1**	**b1.2**	011	tid4
bid2	k2	a2	**b2.1**	001	tid6

Tail-Page Sequence Number (TPS)

TPS tracks the RID of the last tail record (TID) that has been applied (merged) to a base page.



RID	key	A	B	Schema Encoding	Indirection
bid1	k1	a1	b1	011	tid4
bid2	k2	a2	b2	001	tid6

RID	key	A	B	Schema Encoding	Indirection	BaseID
tid1			**b1**	**000**	**bid1**	**bid1**
tid2			b1.1	001	tid1	bid1
tid3		**a1**		000	tid2	bid1
tid4		a1.1	b1.2	011	tid3	bid1
tid5			**b2**	**000**	**bid2**	**bid2**
tid6			b2.1	001	tid5	bid2

Merged Pages

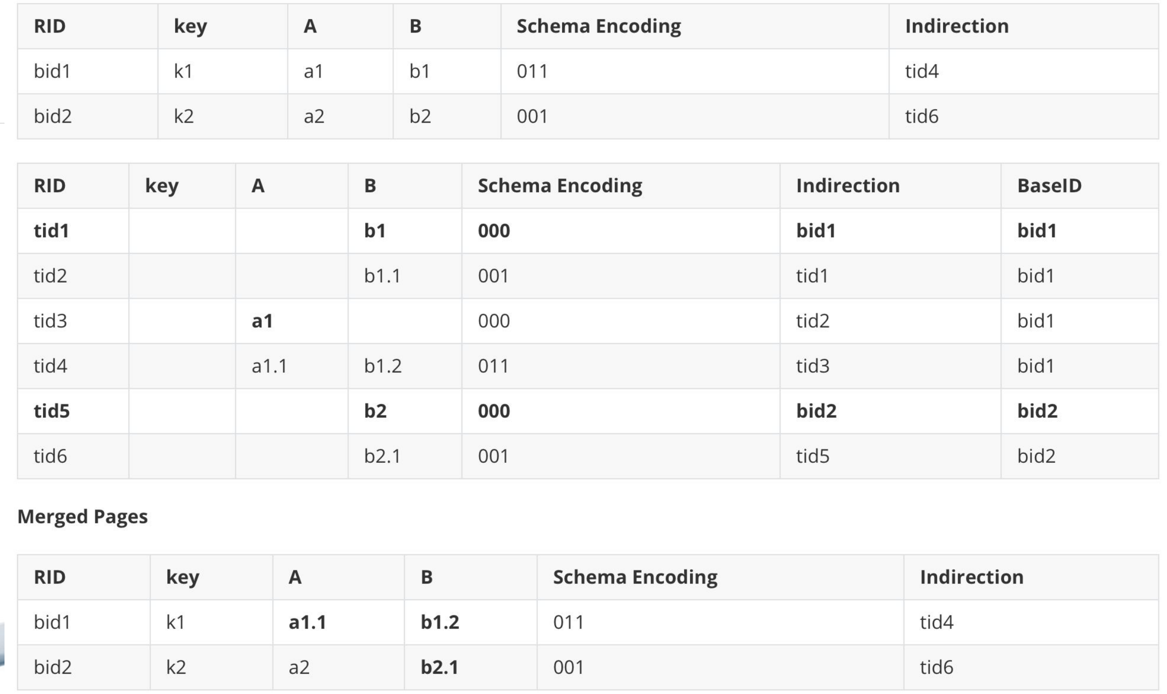
RID	key	A	B	Schema Encoding	Indirection
bid1	k1	**a1.1**	**b1.2**	011	tid4
bid2	k2	a2	**b2.1**	001	tid6

In this example, for the merged page, TPS = tid6

If Indirection > tid6, then we know the base record does not have the latest values.

Tail-Page Sequence Number (TPS)

What will happen if we have a new update while merging? Should we update the **Indirection** column of the Merged Pages?



RID	key	A	B	Schema Encoding	Indirection
bid1	k1	a1	b1	011	tid4
bid2	k2	a2	b2	001	tid6

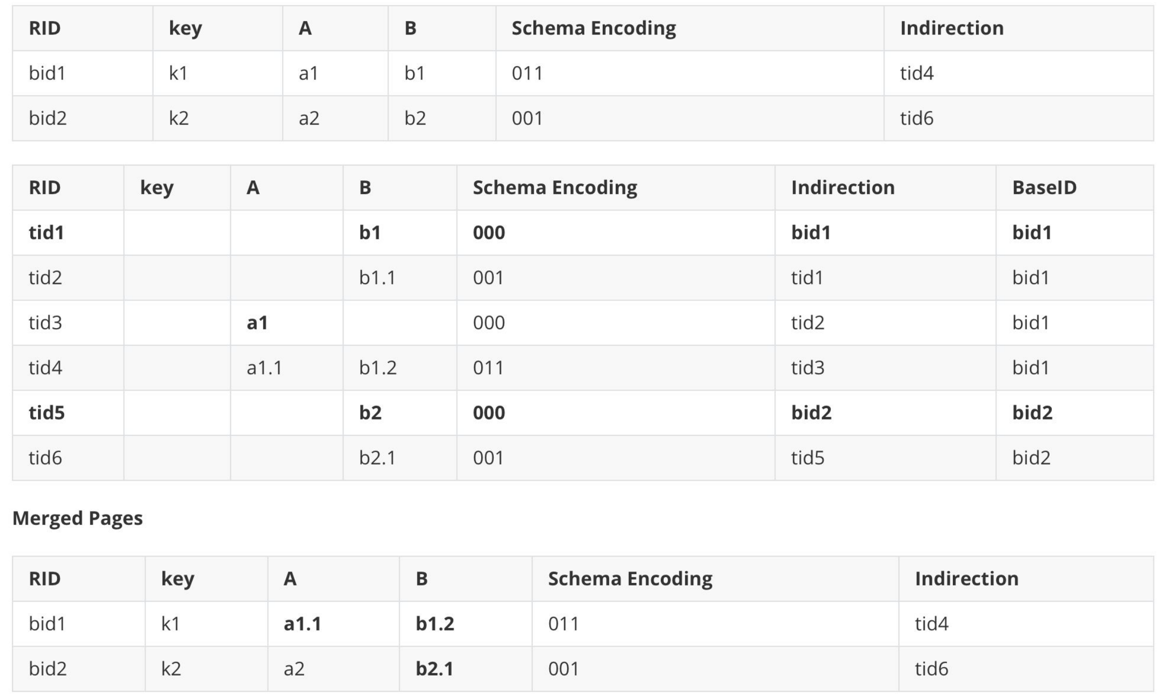
RID	key	A	B	Schema Encoding	Indirection	BaseID
tid1			**b1**	**000**	**bid1**	**bid1**
tid2			b1.1	001	tid1	bid1
tid3		**a1**		000	tid2	bid1
tid4		a1.1	b1.2	011	tid3	bid1
tid5			**b2**	**000**	**bid2**	**bid2**
tid6			b2.1	001	tid5	bid2

Merged Pages

RID	key	A	B	Schema Encoding	Indirection
bid1	k1	**a1.1**	**b1.2**	011	tid4
bid2	k2	a2	**b2.1**	001	tid6

Tail-Page Sequence Number (TPS)

What will happen if we have a new update while merging? Should we update the **Indirection** column of the Merged Pages? Incorrect Indirection / Contention



RID	key	A	B	Schema Encoding	Indirection
bid1	k1	a1	b1	011	tid4
bid2	k2	a2	b2	001	tid6

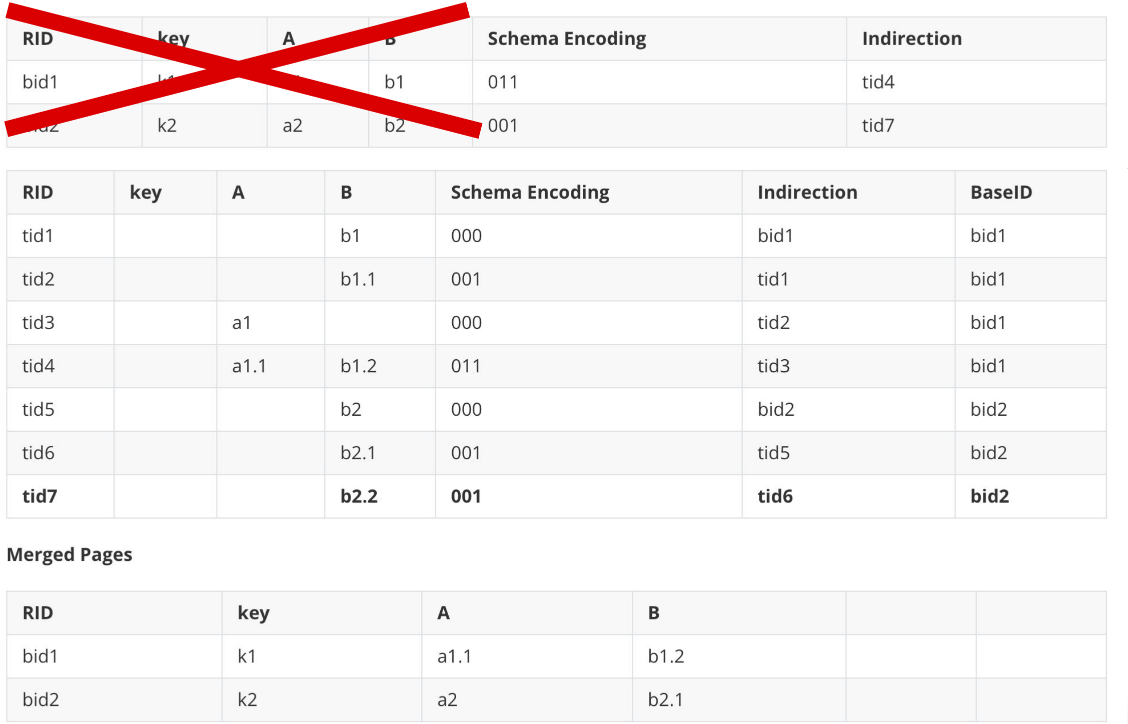
RID	key	A	B	Schema Encoding	Indirection	BaseID
tid1			**b1**	**000**	**bid1**	**bid1**
tid2			b1.1	001	tid1	bid1
tid3		**a1**		000	tid2	bid1
tid4		a1.1	b1.2	011	tid3	bid1
tid5			**b2**	**000**	**bid2**	**bid2**
tid6			b2.1	001	tid5	bid2

Merged Pages

RID	key	A	B	Schema Encoding	Indirection
bid1	k1	**a1.1**	**b1.2**	011	tid4
bid2	k2	a2	**b2.1**	001	tid6

Tail-Page Sequence Number (TPS)

Solution: Don't merge the metadata columns!



~~RID~~	~~key~~	~~A~~	~~B~~	Schema Encoding	Indirection
~~bid1~~	~~k1~~	[illegible]	~~b1~~	011	tid4
~~bid2~~	~~k2~~	~~a2~~	~~b2~~	001	tid7

RID	key	A	B	Schema Encoding	Indirection	BaseID
tid1			b1	000	bid1	bid1
tid2			b1.1	001	tid1	bid1
tid3		a1		000	tid2	bid1
tid4		a1.1	b1.2	011	tid3	bid1
tid5			b2	000	bid2	bid2
tid6			b2.1	001	tid5	bid2
tid7			**b2.2**	**001**	**tid6**	**bid2**

Merged Pages

RID	key	A	B		
bid1	k1	a1.1	b1.2		
bid2	k2	a2	b2.1		

You will get 10% mark off if you merge the metadata columns

Pinning/Unpinning Pages

- Anytime a page in the bufferpool is accessed, the page will be **pinned**.
- Once the transaction no longer needs the page, the page will be **unpinned**.
- **Pin value** records the number of transactions accessing the page.
- The page cannot be replaced if the **pin value** is not zero.

Pinning/Unpinning Pages

- Anytime a page in the bufferpool is accessed, the page will be **pinned**.
- Once the transaction no longer needs the page, the page will be **unpinned**.
- **Pin value** records the number of transactions accessing the page.
- The page cannot be replaced if the **pin value** is not zero.
- Important for the single-thread L-store in M2 since merge is done in a background thread.