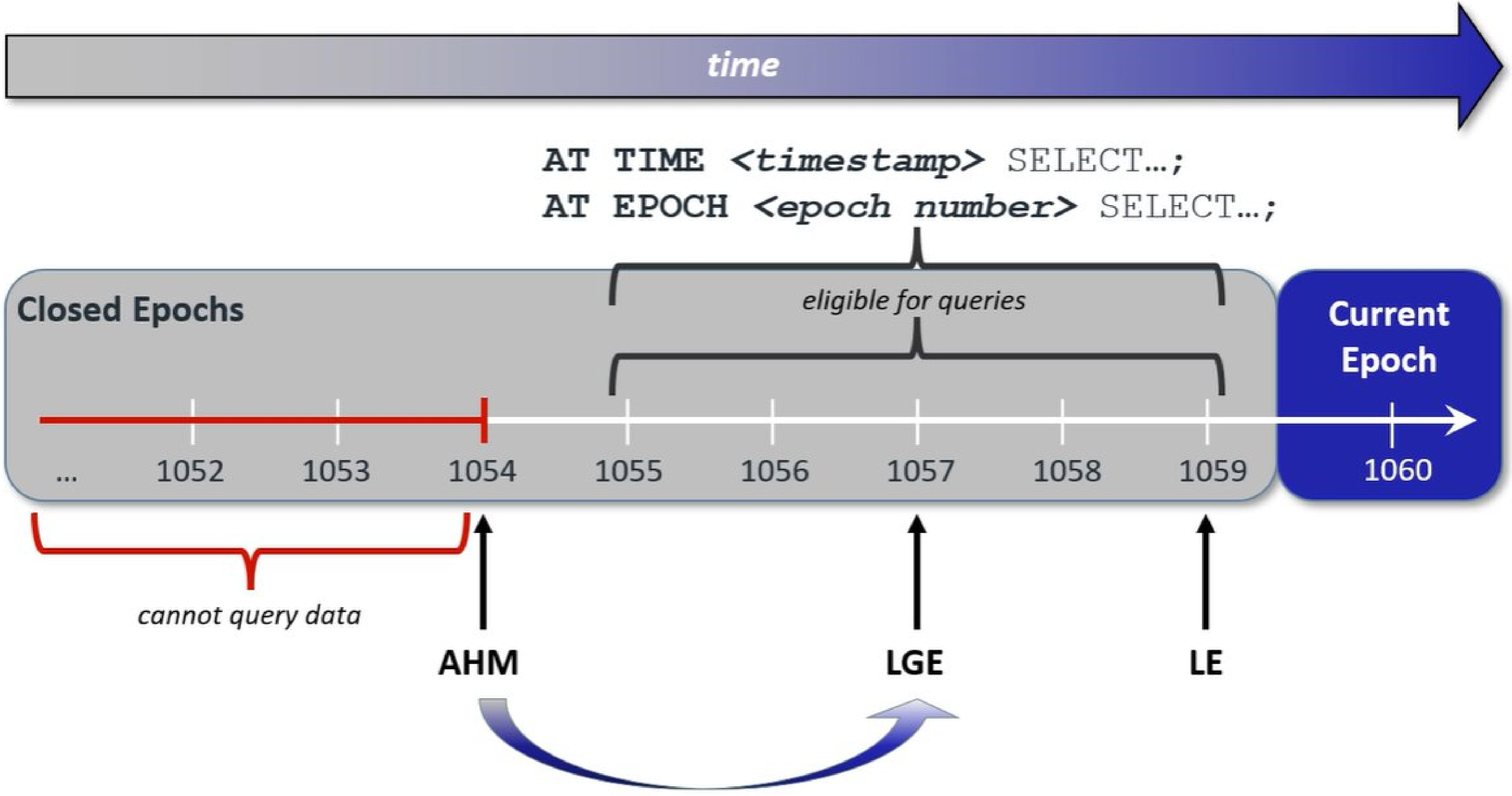
Removing Data

Epoch model



AT TIME ***<timestamp>*** SELECT…;

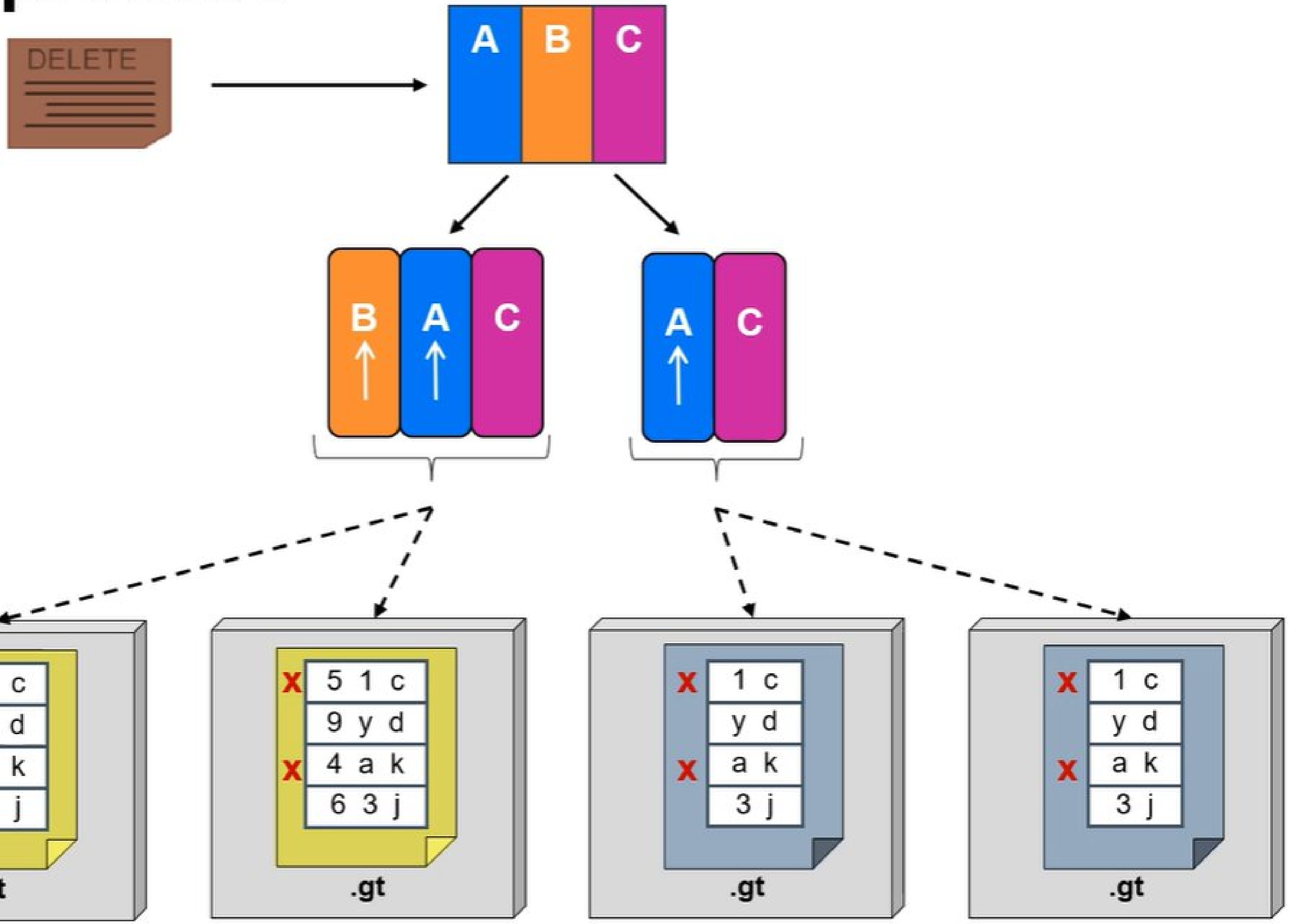
AT EPOCH ***<epoch number>*** SELECT…;

SELECT **MAKE_AHM_NOW();**

Ways of removing data

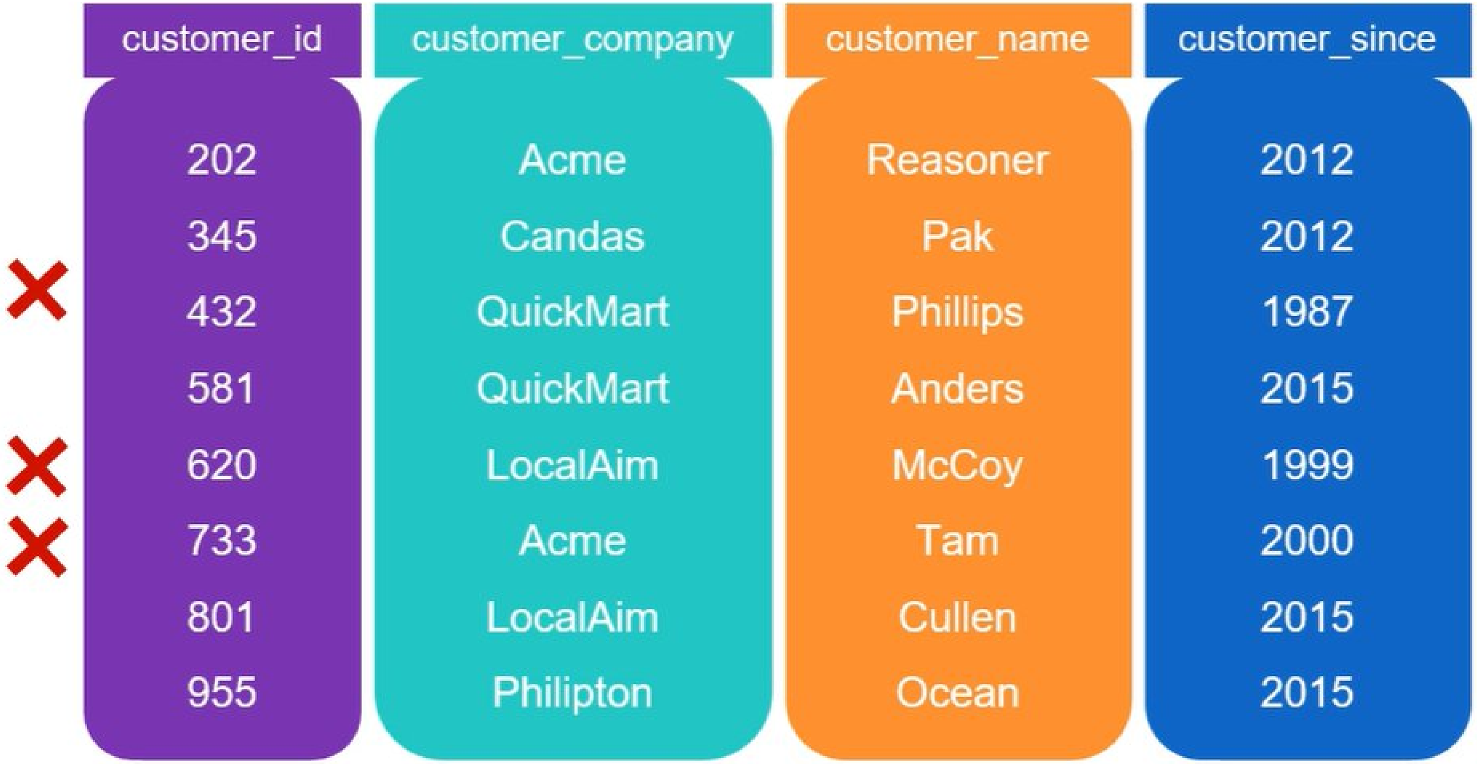
Process	**Consequence**
DELETE FROM *table*	Marks rows with delete vectors and stores them so data can be rolled back to a previous epoch. The data must eventually be purged before the database can reclaim disk space.
TRUNCATE TABLE *table*	Permanently removes all storage and history associated with a table. The table structure is preserved for future use. Disk space is recovered immediately.
DROP TABLE *table*	Permanently removes a table, its definition, the associated projections, and all contained data. Disk space is recovered immediately.
DROP PARTITION	Permanently removes one partition from a partitioned table. Partitioned data can be dropped efficiently and provides query performance benefits. Disk space is recovered immediately.

Data deletion process



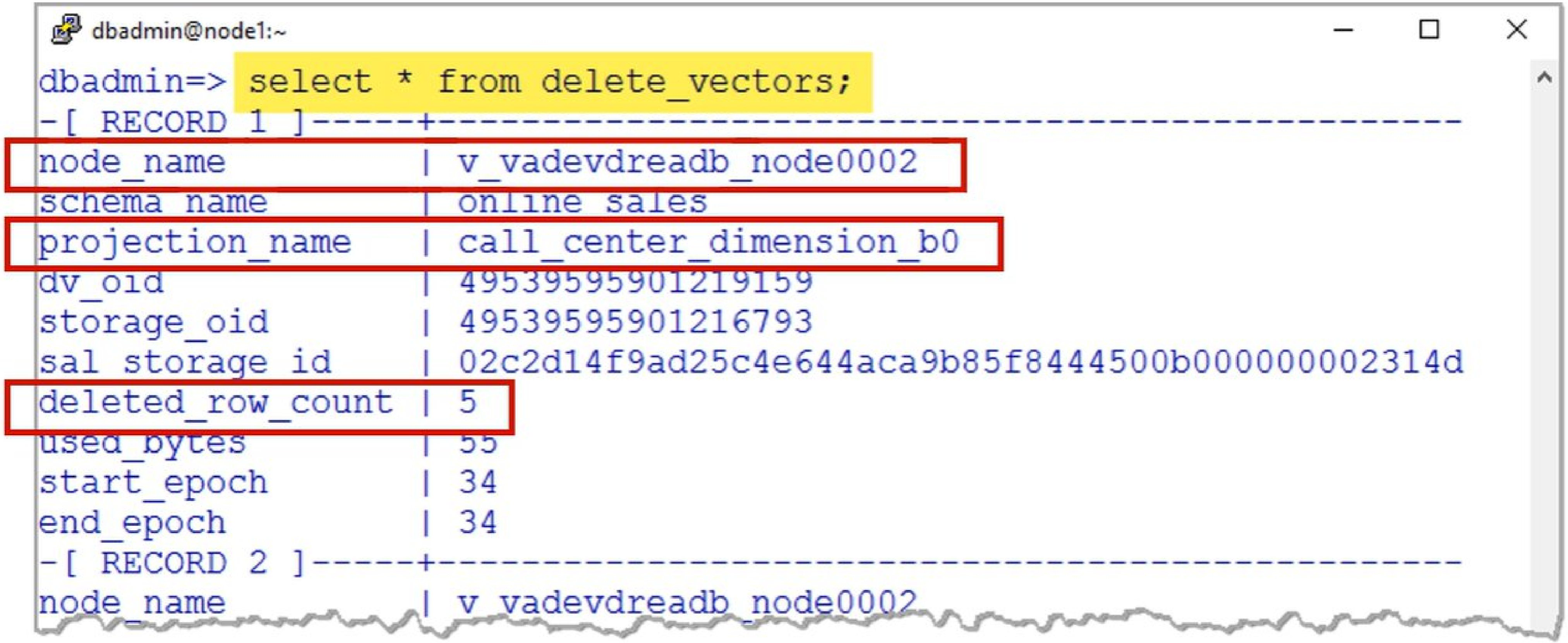
Delete vectors in queries

```
SELECT customer_id, customer_name FROM CUSTOMERS
WHERE is_current = 't';
```



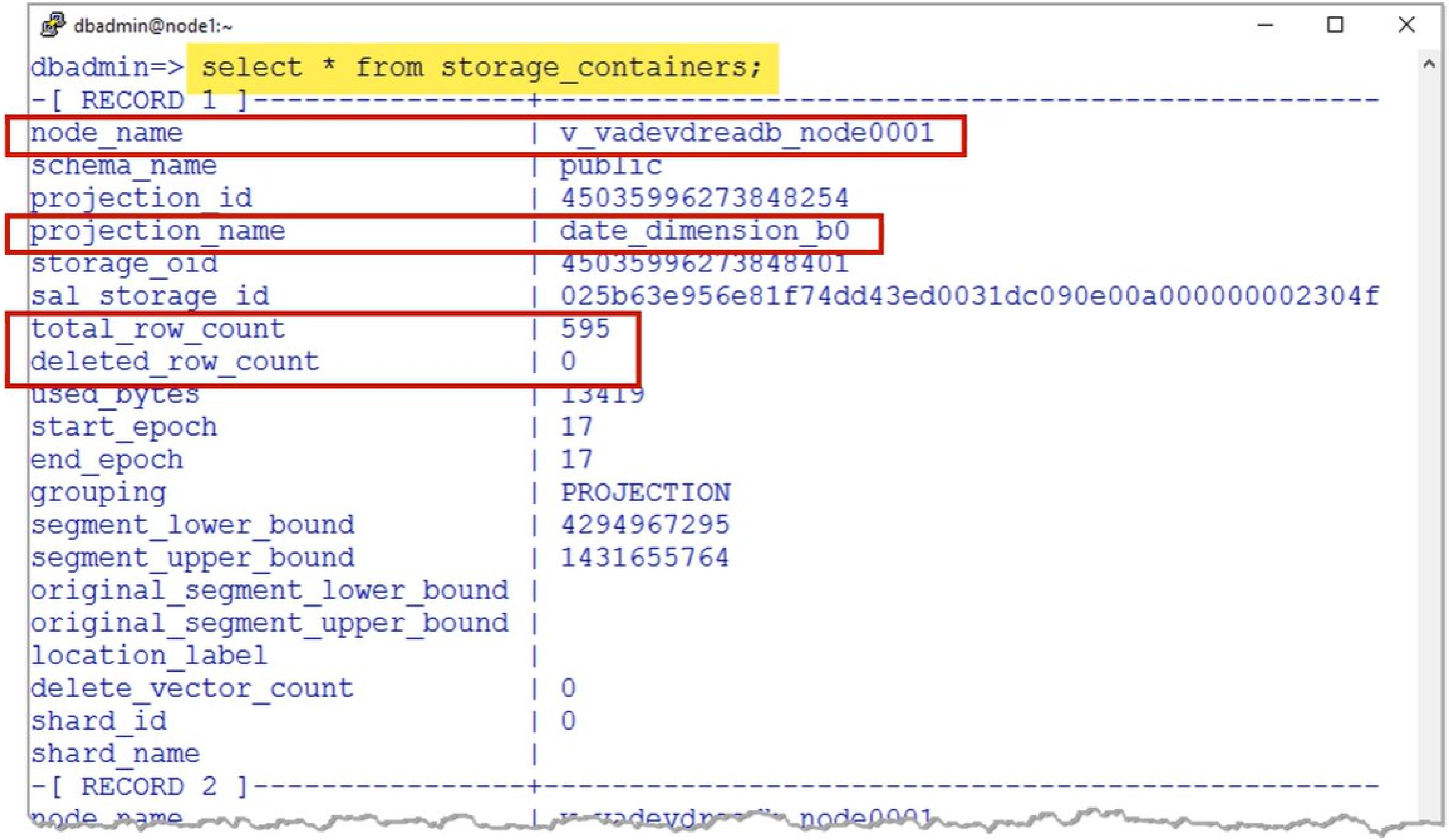
	customer_id	customer_company	customer_name	customer_since
	202	Acme	Reasoner	2012
	345	Candas	Pak	2012
✗	432	QuickMart	Phillips	1987
	581	QuickMart	Anders	2015
✗	620	LocalAim	McCoy	1999
✗	733	Acme	Tam	2000
	801	LocalAim	Cullen	2015
	955	Philipton	Ocean	2015

System table: DELETE_VECTORS



```
dbadmin=> select * from delete_vectors;
-[ RECORD 1 ]-----+-------------------------------------------------
node_name         | v_vadevdreadb_node0002
schema_name       | online_sales
projection_name   | call_center_dimension_b0
dv_oid            | 49539595901219159
storage_oid       | 49539595901216793
sal_storage_id    | 02c2d14f9ad25c4e644aca9b85f8444500b000000002314d
deleted_row_count | 5
used_bytes        | 55
start_epoch       | 34
end_epoch         | 34
-[ RECORD 2 ]-----+-------------------------------------------------
node_name         | v_vadevdreadb_node0002
```

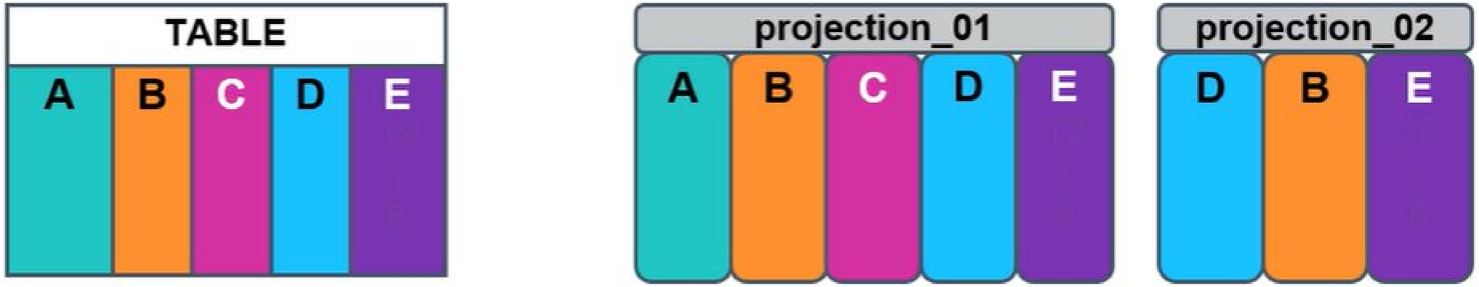
System table: STORAGE_CONTAINERS



```
dbadmin=> select * from storage_containers;
-[ RECORD 1 ]----------------+-------------------------------------------------
node_name                    | v_vadevdreadb_node0001
schema_name                  | public
projection_id                | 45035996273848254
projection_name              | date_dimension_b0
storage_oid                  | 45035996273848401
sal_storage_id               | 025b63e956e81f74dd43ed0031dc090e00a000000002304f
total_row_count              | 595
deleted_row_count            | 0
used_bytes                   | 13419
start_epoch                  | 17
end_epoch                    | 17
grouping                     | PROJECTION
segment_lower_bound          | 4294967295
segment_upper_bound          | 1431655764
original_segment_lower_bound |
original_segment_upper_bound |
location_label               |
delete_vector_count          | 0
shard_id                     | 0
shard_name                   |
-[ RECORD 2 ]----------------+-------------------------------------------------
node_name                    | v_vadevdreadb_node0001
```

Dropping vs. truncating tables

```
TRUNCATE TABLE tablename
```



TABLE

A	B	C	D	E

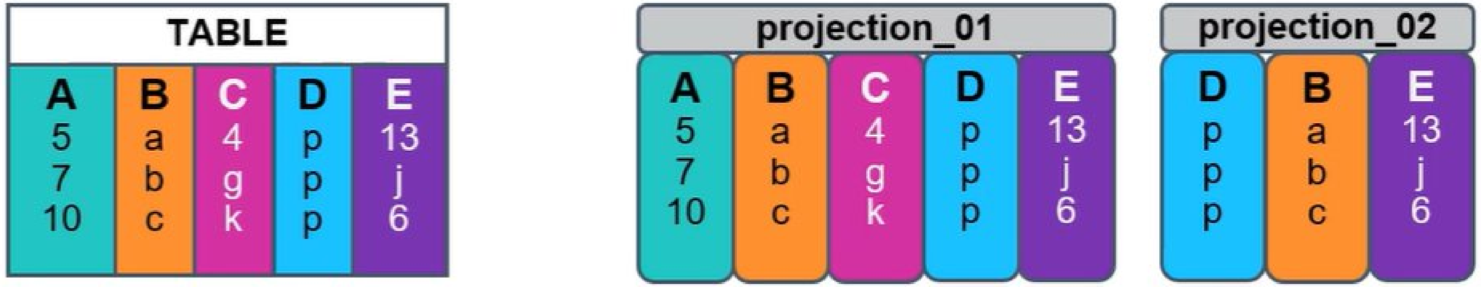
projection_01

A	B	C	D	E

projection_02

D	B	E

```
DROP TABLE tablename
```



TABLE

A	B	C	D	E
5	a	4	p	13
7	b	g	p	j
10	c	k	p	6

projection_01

A	B	C	D	E
5	a	4	p	13
7	b	g	p	j
10	c	k	p	6

projection_02

D	B	E
p	a	13
p	b	j
p	c	6

opentext™

Purging delete vectors

1. Move the AHM to the LGE: `SELECT MAKE_AHM_NOW();`

2. Purge the delete vectors

From all projections on all tables	`SELECT PURGE();`
From all projections on a specific table	`SELECT PURGE_TABLE('table_name');`
From a specific projection	`SELECT PURGE_PROJECTION('projection_name');`
From a partition	`SELECT PURGE_PARTITION('table_name','partition_key');`

Thank you