

Hermit: Designing Succinct Secondary Indexing Mechanism by Exploiting Column Correlations

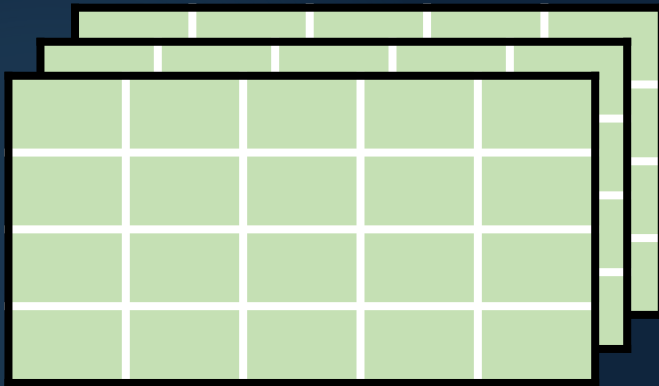
Yingjun Wu, **Jia Yu**, Yuanyuan Tian, Richard Sidle, Ronald Barber



HERMIT: MOTIVATION

1

- ▶ An index is a copy of selected columns of data from a table

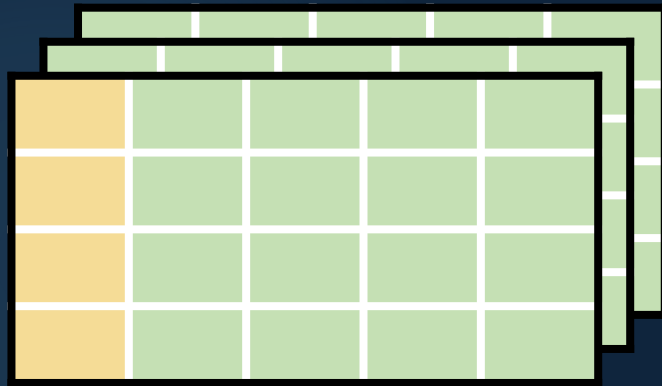


Base table

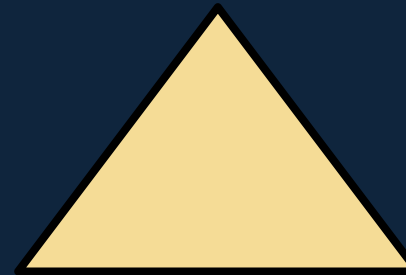
HERMIT: MOTIVATION

2

- ▶ An index is a copy of selected columns of data from a table



Base table

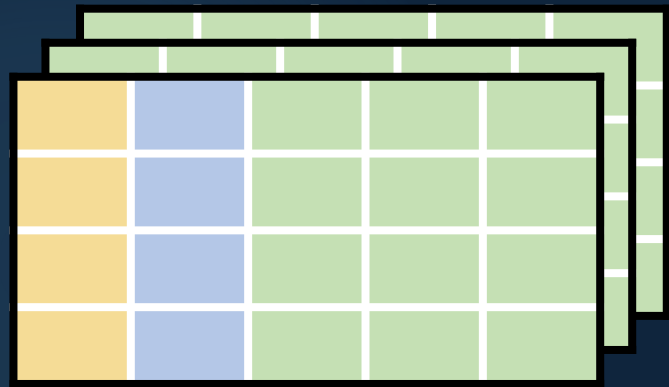


Primary index

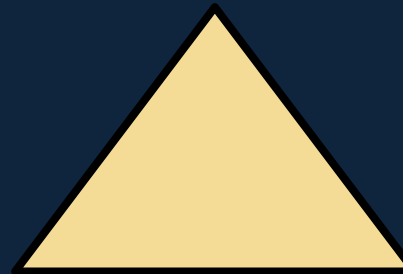
HERMIT: MOTIVATION

3

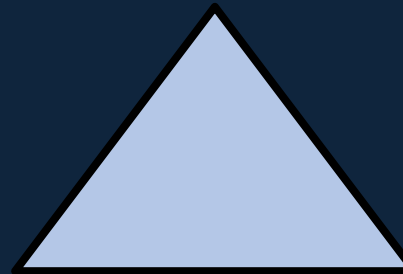
- ▶ An index is a copy of selected columns of data from a table



Base table



Primary index

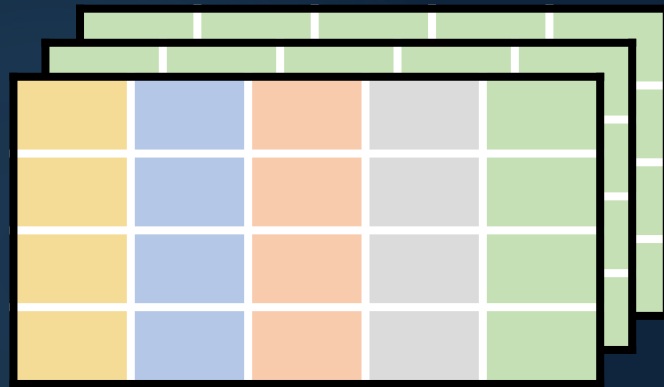


Secondary index

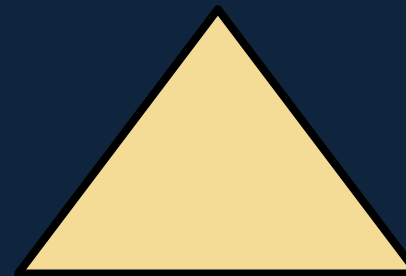
HERMIT: MOTIVATION

4

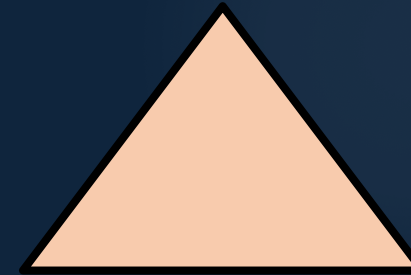
- ▶ An index is a copy of selected columns of data from a table



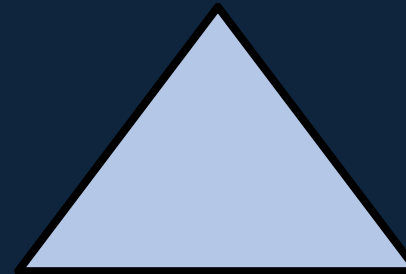
Base table



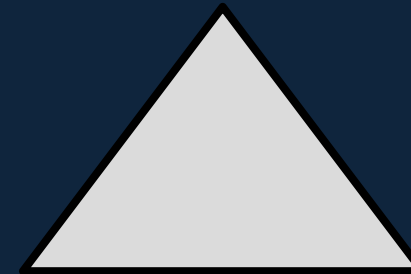
Primary index



Secondary index



Secondary index



Secondary index

HERMIT: MOTIVATION

5

- ▶ An index is a copy of selected columns of data from a table



OUT OF MEMORY

HERMIT: MOTIVATION

6

Indexes consume large amounts of space

B+ Tree on TPC-H	Index size	% space
2 GB	0.25 GB	
20 GB	2.51 GB	12.5%
200 GB	25 GB	



HERMIT: MOTIVATION

7

► Existing solutions

- Sparse index, index compression...
 - Reduce space usage
 - At the expense of reduced lookup performance
 - Previous work Hippo index (VLDB 2016)
 - 50X smaller than B-Tree
 - Only comparable to B-Tree at 0.1% selectivity
- Smart index selection
 - Low performance on unindexed columns

Disk Page Range

1 - 10

11 - 25

26 - 30

Summary

Synopsis, histogram...

Synopsis, histogram...

Synopsis, histogram...



HERMIT: OBJECTIVE

8

Consume much less space
Achieve good enough performance



HERMIT: OBSERVATION

9

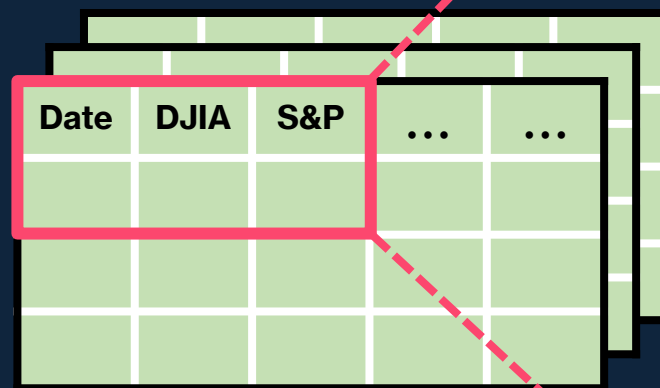
Column correlations are prevalent



HERMIT: OBSERVATION

10

Column correlations are prevalent



The diagram shows a stack of green tables representing a 'STOCK table'. A red dashed line originates from a specific cell in the first table of the stack and points to a corresponding cell in the detailed table on the right. The first table has columns labeled 'Date', 'DJIA', 'S&P', and two columns with ellipses ('...').

STOCK table

DJIA: Dow Jones Industrial Average

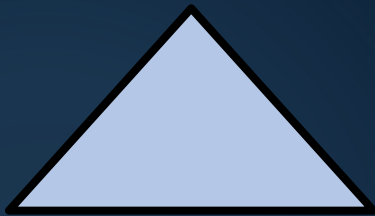
Date	DJIA	S&P
1/2/91	2736.39	343.93
2/1/91	2882.18	367.07
3/1/91	2913.86	375.22
4/1/91	2887.87	375.34
5/1/91	3027.50	389.83
6/3/91	2906.75	371.16
7/1/91	3024.82	387.81
8/1/91	3043.60	395.43
9/3/91	3016.77	387.86
10/1/91	3069.10	392.45
11/1/91	2894.68	375.22
12/2/91	3168.83	417.09
1/2/92	3223.40	408.78
2/3/92	3267.70	412.70
3/2/92	3235.50	403.69
4/1/92	3359.10	414.95
5/1/92	3396.90	415.35
6/1/92	3318.50	408.14
7/1/92	3393.80	424.21
8/3/92	3257.40	414.03
9/1/92	3271.70	417.80
10/1/92	3226.30	418.68
11/2/92	3305.20	431.35
12/1/92	3301.11	435.71

HERMIT: OBSERVATION

11

Column correlations are prevalent

QUERY 1: **SELECT** * **FROM** stock **WHERE**
(date **BETWEEN** ? **AND** ?) **AND** (djia **BETWEEN** ? **AND** ?)



Secondary index for Date+DJIA

Date	DJIA	S&P	...	...

STOCK table

Date	DJIA	S&P
1/2/91	2736.39	343.93
2/1/91	2882.18	367.07
3/1/91	2913.86	375.22
4/1/91	2887.87	375.34
5/1/91	3027.50	389.83
6/3/91	2906.75	371.16
7/1/91	3024.82	387.81
8/1/91	3043.60	395.43
9/3/91	3016.77	387.86
10/1/91	3069.10	392.45
11/1/91	2894.68	375.22
12/2/91	3168.83	417.09
1/2/92	3223.40	408.78
2/3/92	3267.70	412.70
3/2/92	3235.50	403.69
4/1/92	3359.10	414.95
5/1/92	3396.90	415.35
6/1/92	3318.50	408.14
7/1/92	3393.80	424.21
8/3/92	3257.40	414.03
9/1/92	3271.70	417.80
10/1/92	3226.30	418.68
11/2/92	3305.20	431.35
12/1/92	3301.11	435.71

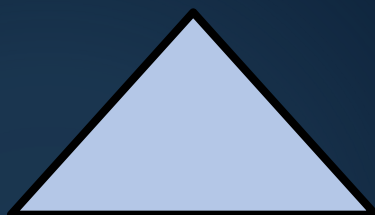
HERMIT: OBSERVATION

12

Column correlations are prevalent

QUERY 1: SELECT * FROM stock WHERE
(date BETWEEN ? AND ?) AND (djia BETWEEN ? AND ?)

QUERY 2: SELECT * FROM stock WHERE
(date BETWEEN ? AND ?) AND (S&P BETWEEN ? AND ?)



Secondary index for Date+DJIA

Date	DJIA	S&P	...	...

STOCK table

Date	DJIA	S&P
1/2/91	2736.39	343.93
2/1/91	2882.18	367.07
3/1/91	2913.86	375.22
4/1/91	2887.87	375.34
5/1/91	3027.50	389.83
6/3/91	2906.75	371.16
7/1/91	3024.82	387.81
8/1/91	3043.60	395.43
9/3/91	3016.77	387.86
10/1/91	3069.10	392.45
11/1/91	2894.68	375.22
12/2/91	3168.83	417.09
1/2/92	3223.40	408.78
2/3/92	3267.70	412.70
3/2/92	3235.50	403.69
4/1/92	3359.10	414.95
5/1/92	3396.90	415.35
6/1/92	3318.50	408.14
7/1/92	3393.80	424.21
8/3/92	3257.40	414.03
9/1/92	3271.70	417.80
10/1/92	3226.30	418.68
11/2/92	3305.20	431.35
12/1/92	3301.11	435.71

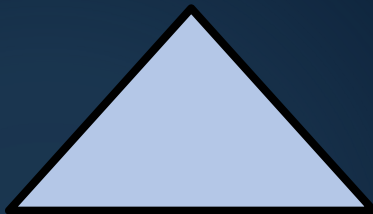
HERMIT: OBSERVATION

13

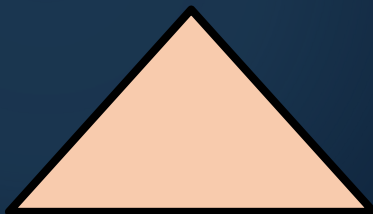
Column correlations are prevalent

QUERY 1: SELECT * FROM stock WHERE
(date BETWEEN ? AND ?) AND (djia BETWEEN ? AND ?)

QUERY 2: SELECT * FROM stock WHERE
(date BETWEEN ? AND ?) AND (S&P BETWEEN ? AND ?)



Secondary index for Date+DJIA



Secondary index for Date+S&P

Date	DJIA	S&P	...	...

STOCK table

Date	DJIA	S&P
1/2/91	2736.39	343.93
2/1/91	2882.18	367.07
3/1/91	2913.86	375.22
4/1/91	2887.87	375.34
5/1/91	3027.50	389.83
6/3/91	2906.75	371.16
7/1/91	3024.82	387.81
8/1/91	3043.60	395.43
9/3/91	3016.77	387.86
10/1/91	3069.10	392.45
11/1/91	2894.68	375.22
12/2/91	3168.83	417.09
1/2/92	3223.40	408.78
2/3/92	3267.70	412.70
3/2/92	3235.50	403.69
4/1/92	3359.10	414.95
5/1/92	3396.90	415.35
6/1/92	3318.50	408.14
7/1/92	3393.80	424.21
8/3/92	3257.40	414.03
9/1/92	3271.70	417.80
10/1/92	3226.30	418.68
11/2/92	3305.20	431.35
12/1/92	3301.11	435.71

HERMIT: OBSERVATION

14

Column correlations are prevalent

QUERY 1: SELECT * FROM stock WHERE
(date BETWEEN ? AND ?) AND (djia BETWEEN ? AND ?)

QUERY 2: SELECT * FROM stock WHERE
(date BETWEEN ? AND ?) AND (S&P BETWEEN ? AND ?)

Secondary index for Date+DJIA

The diagram shows a 3D representation of the STOCK table. A single row is highlighted in yellow, with its columns labeled Date, DJIA, S&P, and two ellipses (...). A red dashed line originates from the top-right corner of the highlighted row and extends towards the top-right corner of the table shown in the adjacent data table.

STOCK table

Date	DJIA	S&P
1/2/91	2736.39	343.93
2/1/91	2882.18	367.07
3/1/91	2913.86	375.22
4/1/91	2887.87	375.34
5/1/91	3027.50	389.83
6/3/91	2906.75	371.16
7/1/91	3024.82	387.81
8/1/91	3043.60	395.43
9/3/91	3016.77	387.86
10/1/91	3069.10	392.45
11/1/91	2894.68	375.22
12/2/91	3168.83	417.09
1/2/92	3223.40	408.78
2/3/92	3267.70	412.70
3/2/92	3235.50	403.69
4/1/92	3359.10	414.95
5/1/92	3396.90	415.35
6/1/92	3318.50	408.14
7/1/92	3393.80	424.21
8/3/92	3257.40	414.03
9/1/92	3271.70	417.80
10/1/92	3226.30	418.68
11/2/92	3305.20	431.35
12/1/92	3301.11	435.71

Can we avoid building
another complete index?

HERMIT: OBSERVATION

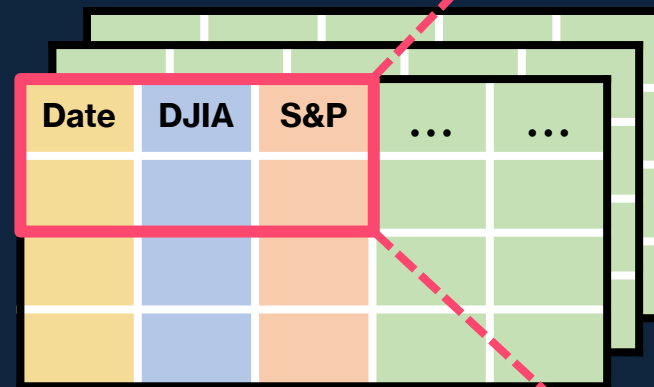
15

Column correlations are prevalent

QUERY 1: SELECT * FROM stock WHERE
(date BETWEEN ? AND ?) AND (djia BETWEEN ? AND ?)

QUERY 2: SELECT * FROM stock WHERE
(date BETWEEN ? AND ?) AND (S&P BETWEEN ? AND ?)

YES!!!



The diagram shows a stack of three tables. The top table has columns labeled 'Date', 'DJIA', 'S&P', and two columns with ellipses. A red dashed line highlights the first row of the top table, which contains the values '1/2/91', '2736.39', and '343.93'. A red arrow points from this row to the 'Date' column of the table on the right.

STOCK table

Date	DJIA	S&P
1/2/91	2736.39	343.93
2/1/91	2882.18	367.07
3/1/91	2913.86	375.22
4/1/91	2887.87	375.34
5/1/91	3027.50	389.83
6/3/91	2906.75	371.16
7/1/91	3024.82	387.81
8/1/91	3043.60	395.43
9/3/91	3016.77	387.86
10/1/91	3069.10	392.45
11/1/91	2894.68	375.22
12/2/91	3168.83	417.09
1/2/92	3223.40	408.78
2/3/92	3267.70	412.70
3/2/92	3235.50	403.69
4/1/92	3359.10	414.95
5/1/92	3396.90	415.35
6/1/92	3318.50	408.14
7/1/92	3393.80	424.21
8/3/92	3257.40	414.03
9/1/92	3271.70	417.80
10/1/92	3226.30	418.68
11/2/92	3305.20	431.35
12/1/92	3301.11	435.71

Can we avoid building
another complete index?

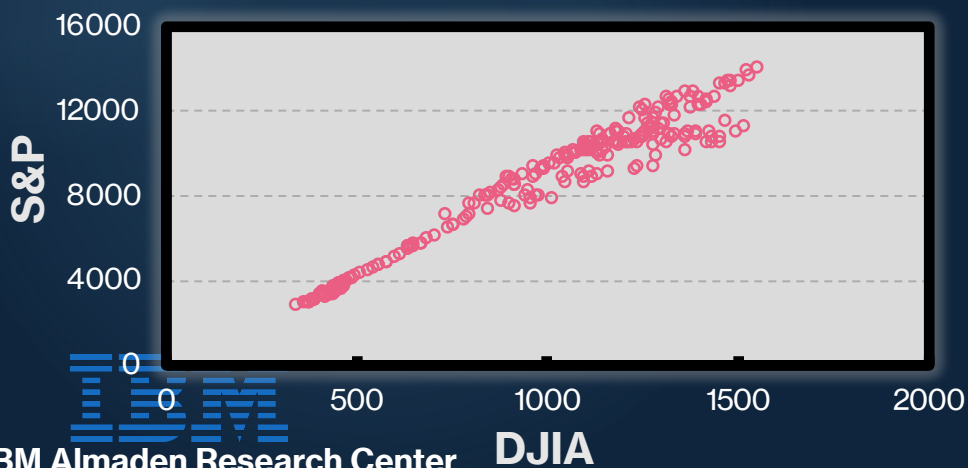


Secondary index for Date+S&P

HERMIT: OBSERVATION

16

DJIA and S&P are *highly correlated*



Date	DJIA	S&P	...	...

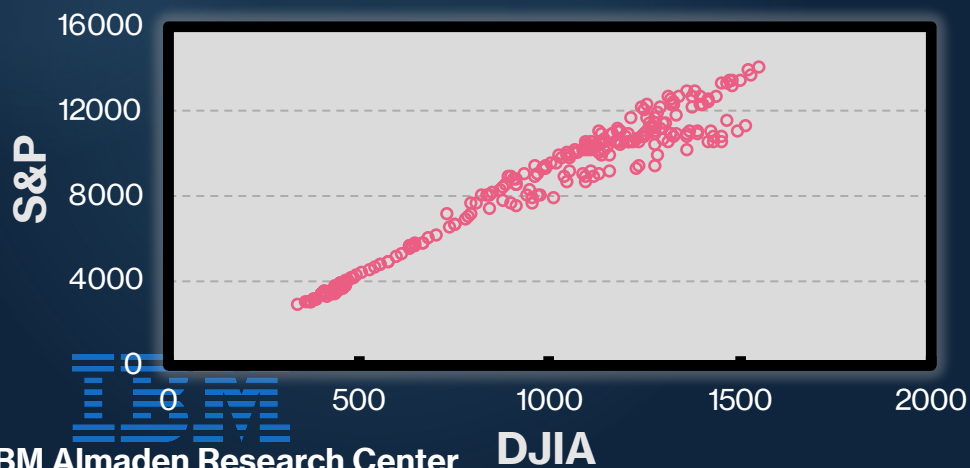
STOCK table

Date	DJIA	S&P
1/2/91	2736.39	343.93
2/1/91	2882.18	367.07
3/1/91	2913.86	375.22
4/1/91	2887.87	375.34
5/1/91	3027.50	389.83
6/3/91	2906.75	371.16
7/1/91	3024.82	387.81
8/1/91	3043.60	395.43
9/3/91	3016.77	387.86
10/1/91	3069.10	392.45
11/1/91	2894.68	375.22
12/2/91	3168.83	417.09
1/2/92	3223.40	408.78
2/3/92	3267.70	412.70
3/2/92	3235.50	403.69
4/1/92	3359.10	414.95
5/1/92	3396.90	415.35
6/1/92	3318.50	408.14
7/1/92	3393.80	424.21
8/3/92	3257.40	414.03
9/1/92	3271.70	417.80
10/1/92	3226.30	418.68
11/2/92	3305.20	431.35
12/1/92	3301.11	435.71

HERMIT: IDEA

17

Model the **correlation** between **DJIA** and **S&P**



A diagram showing a 3D stack of data tables. The top table is highlighted with a pink border and contains the following data:

Date	DJIA	S&P	...	...

STOCK table

Date	DJIA	S&P
1/2/91	2736.39	343.93
2/1/91	2882.18	367.07
3/1/91	2913.86	375.22
4/1/91	2887.87	375.34
5/1/91	3027.50	389.83
6/3/91	2906.75	371.16
7/1/91	3024.82	387.81
8/1/91	3043.60	395.43
9/3/91	3016.77	387.86
10/1/91	3069.10	392.45
11/1/91	2894.68	375.22
12/2/91	3168.83	417.09
1/2/92	3223.40	408.78
2/3/92	3267.70	412.70
3/2/92	3235.50	403.69
4/1/92	3359.10	414.95
5/1/92	3396.90	415.35
6/1/92	3318.50	408.14
7/1/92	3393.80	424.21
8/3/92	3257.40	414.03
9/1/92	3271.70	417.80
10/1/92	3226.30	418.68
11/2/92	3305.20	431.35
12/1/92	3301.11	435.71

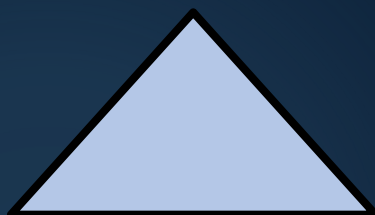
HERMIT: IDEA

18

Model the **correlation** between **DJIA** and **S&P**

QUERY 1: `SELECT * FROM stock WHERE
(date BETWEEN ? AND ?) AND (djia BETWEEN ? AND ?)`

QUERY 2: `SELECT * FROM stock WHERE
(date BETWEEN ? AND ?) AND (S&P BETWEEN ? AND ?)`



Secondary index for Date+DJIA



Secondary index for Date+S&P

Date	DJIA	S&P	...	...

STOCK table

Date	DJIA	S&P
1/2/91	2736.39	343.93
2/1/91	2882.18	367.07
3/1/91	2913.86	375.22
4/1/91	2887.87	375.34
5/1/91	3027.50	389.83
6/3/91	2906.75	371.16
7/1/91	3024.82	387.81
8/1/91	3043.60	395.43
9/3/91	3016.77	387.86
10/1/91	3069.10	392.45
11/1/91	2894.68	375.22
12/2/91	3168.83	417.09
1/2/92	3223.40	408.78
2/3/92	3267.70	412.70
3/2/92	3235.50	403.69
4/1/92	3359.10	414.95
5/1/92	3396.90	415.35
6/1/92	3318.50	408.14
7/1/92	3393.80	424.21
8/3/92	3257.40	414.03
9/1/92	3271.70	417.80
10/1/92	3226.30	418.68
11/2/92	3305.20	431.35
12/1/92	3301.11	435.71

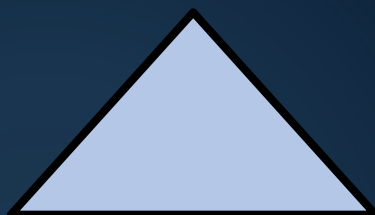
HERMIT: IDEA

19

Model the **correlation** between **DJIA** and **S&P**

QUERY 1: `SELECT * FROM stock WHERE
(date BETWEEN ? AND ?) AND (djia BETWEEN ? AND ?)`

QUERY 2: `SELECT * FROM stock WHERE
(date BETWEEN ? AND ?) AND (S&P BETWEEN ? AND ?)`



Secondary index for Date+DJIA



succinct index for S&P -> **model correlations**

Date	DJIA	S&P	...	...

STOCK table

Date	DJIA	S&P
1/2/91	2736.39	343.93
2/1/91	2882.18	367.07
3/1/91	2913.86	375.22
4/1/91	2887.87	375.34
5/1/91	3027.50	389.83
6/3/91	2906.75	371.16
7/1/91	3024.82	387.81
8/1/91	3043.60	395.43
9/3/91	3016.77	387.86
10/1/91	3069.10	392.45
11/1/91	2894.68	375.22
12/2/91	3168.83	417.09
1/2/92	3223.40	408.78
2/3/92	3267.70	412.70
3/2/92	3235.50	403.69
4/1/92	3359.10	414.95
5/1/92	3396.90	415.35
6/1/92	3318.50	408.14
7/1/92	3393.80	424.21
8/3/92	3257.40	414.03
9/1/92	3271.70	417.80
10/1/92	3226.30	418.68
11/2/92	3305.20	431.35
12/1/92	3301.11	435.71

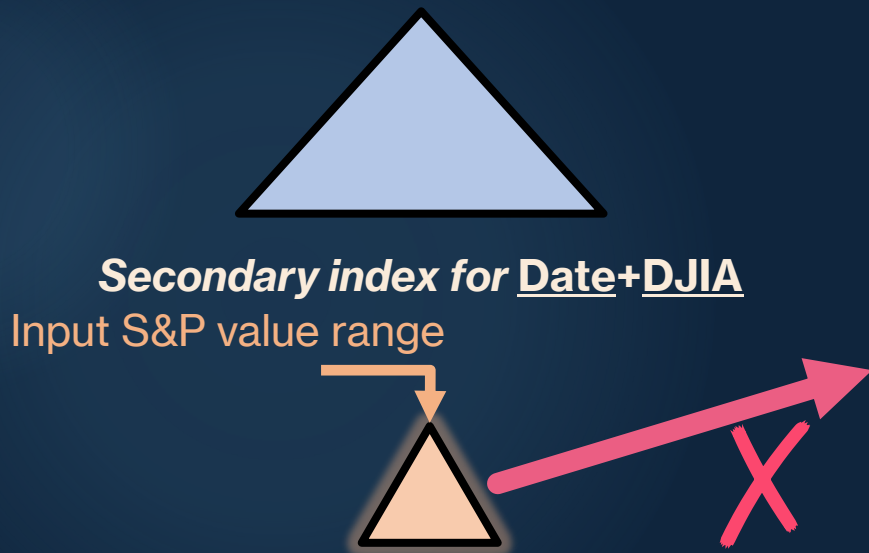
HERMIT: IDEA

20

Model the **correlation** between **DJIA** and **S&P**

QUERY 1: `SELECT * FROM stock WHERE
(date BETWEEN ? AND ?) AND (djia BETWEEN ? AND ?)`

QUERY 2: `SELECT * FROM stock WHERE
(date BETWEEN ? AND ?) AND (S&P BETWEEN ? AND ?)`



Date	DJIA	S&P	...	...

Date	DJIA	S&P
1/2/91	2736.39	343.93
2/1/91	2882.18	367.07
3/1/91	2913.86	375.22
4/1/91	2887.87	375.34
5/1/91	3027.50	389.83
6/3/91	2906.75	371.16
7/1/91	3024.82	387.81
8/1/91	3043.60	395.43
9/3/91	3016.77	387.86
10/1/91	3069.10	392.45
11/1/91	2894.68	375.22
12/2/91	3168.83	417.09
1/2/92	3223.40	408.78
2/3/92	3267.70	412.70
3/2/92	3235.50	403.69
4/1/92	3359.10	414.95
5/1/92	3396.90	415.35
6/1/92	3318.50	408.14
7/1/92	3393.80	424.21
8/3/92	3257.40	414.03
9/1/92	3271.70	417.80
10/1/92	3226.30	418.68
11/2/92	3305.20	431.35
12/1/92	3301.11	435.71

succinct index for S&P -> model correlations

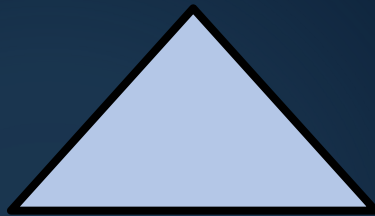
HERMIT: IDEA

21

Model the **correlation** between **DJIA** and **S&P**

QUERY 1: `SELECT * FROM stock WHERE
(date BETWEEN ? AND ?) AND (djia BETWEEN ? AND ?)`

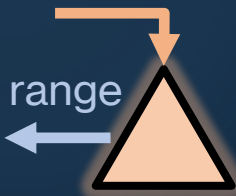
QUERY 2: `SELECT * FROM stock WHERE
(date BETWEEN ? AND ?) AND (S&P BETWEEN ? AND ?)`



Secondary index for Date+DJIA

Input S&P value range

Output DJIA value range



Date	DJIA	S&P	...	...

STOCK table

Date	DJIA	S&P
1/2/91	2736.39	343.93
2/1/91	2882.18	367.07
3/1/91	2913.86	375.22
4/1/91	2887.87	375.34
5/1/91	3027.50	389.83
6/3/91	2906.75	371.16
7/1/91	3024.82	387.81
8/1/91	3043.60	395.43
9/3/91	3016.77	387.86
10/1/91	3069.10	392.45
11/1/91	2894.68	375.22
12/2/91	3168.83	417.09
1/2/92	3223.40	408.78
2/3/92	3267.70	412.70
3/2/92	3235.50	403.69
4/1/92	3359.10	414.95
5/1/92	3396.90	415.35
6/1/92	3318.50	408.14
7/1/92	3393.80	424.21
8/3/92	3257.40	414.03
9/1/92	3271.70	417.80
10/1/92	3226.30	418.68
11/2/92	3305.20	431.35
12/1/92	3301.11	435.71

succinct index for S&P -> **model correlations**



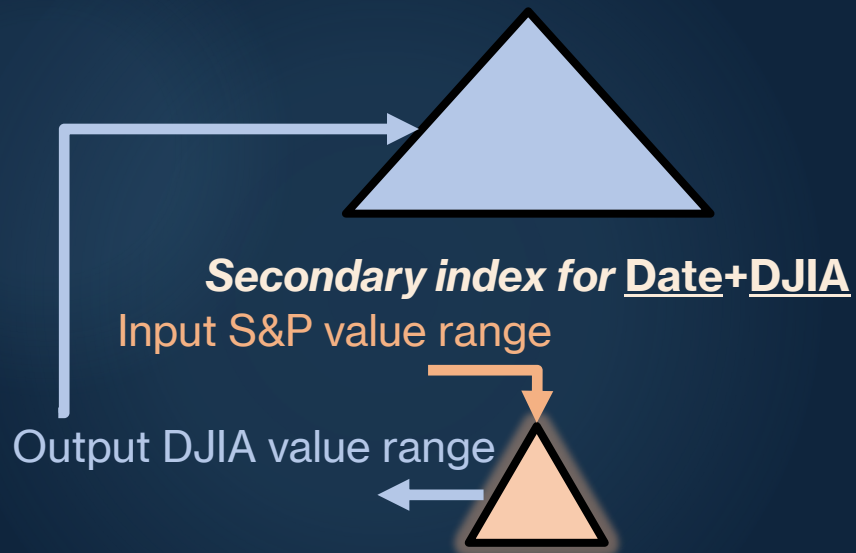
HERMIT: IDEA

22

Model the correlation between DJIA and S&P

QUERY 1: SELECT * FROM stock WHERE
(date BETWEEN ? AND ?) AND (djia BETWEEN ? AND ?)

QUERY 2: SELECT * FROM stock WHERE
(date BETWEEN ? AND ?) AND (S&P BETWEEN ? AND ?)



The diagram shows a stack of tables representing the STOCK table. The top table has columns: Date, DJIA, S&P, ..., ... A red dashed line connects the S&P column of the top table to the S&P column of the table on the right.

STOCK table

Date	DJIA	S&P
1/2/91	2736.39	343.93
2/1/91	2882.18	367.07
3/1/91	2913.86	375.22
4/1/91	2887.87	375.34
5/1/91	3027.50	389.83
6/3/91	2906.75	371.16
7/1/91	3024.82	387.81
8/1/91	3043.60	395.43
9/3/91	3016.77	387.86
10/1/91	3069.10	392.45
11/1/91	2894.68	375.22
12/2/91	3168.83	417.09
1/2/92	3223.40	408.78
2/3/92	3267.70	412.70
3/2/92	3235.50	403.69
4/1/92	3359.10	414.95
5/1/92	3396.90	415.35
6/1/92	3318.50	408.14
7/1/92	3393.80	424.21
8/3/92	3257.40	414.03
9/1/92	3271.70	417.80
10/1/92	3226.30	418.68
11/2/92	3305.20	431.35
12/1/92	3301.11	435.71

succinct index for S&P -> model correlations



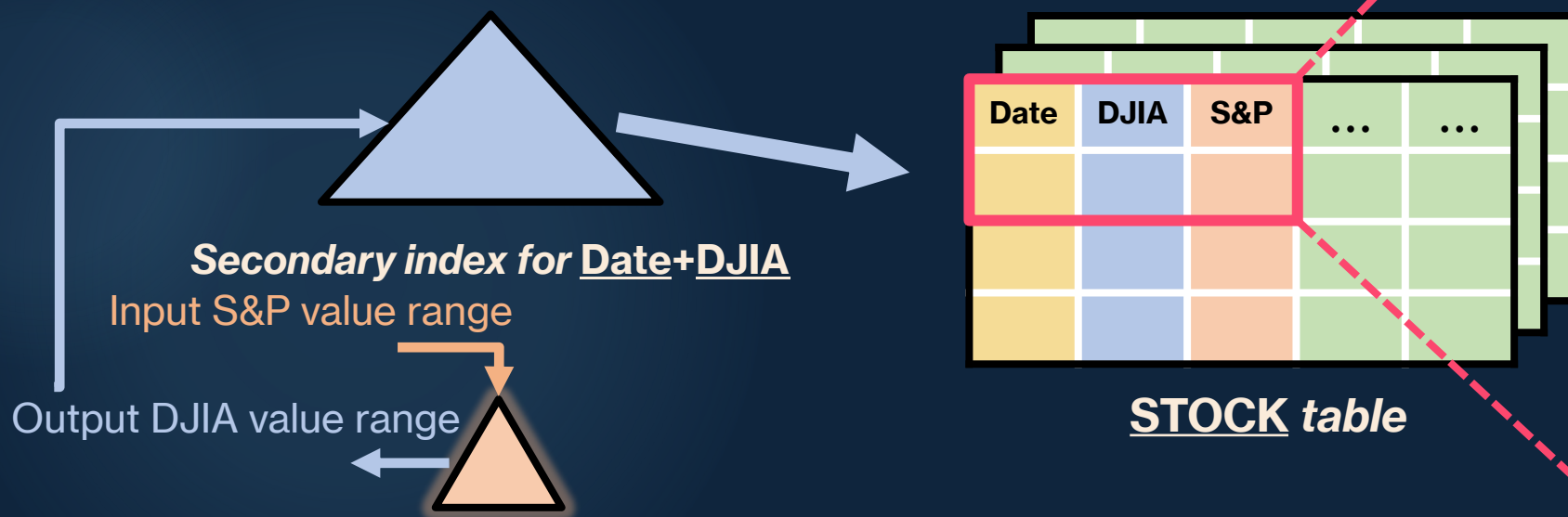
HERMIT: IDEA

23

Model the correlation between DJIA and S&P

QUERY 1: SELECT * FROM stock WHERE
(date BETWEEN ? AND ?) AND (djia BETWEEN ? AND ?)

QUERY 2: SELECT * FROM stock WHERE
(date BETWEEN ? AND ?) AND (S&P BETWEEN ? AND ?)



succinct index for S&P -> model correlations



HERMIT: IDEA

24



HERMIT

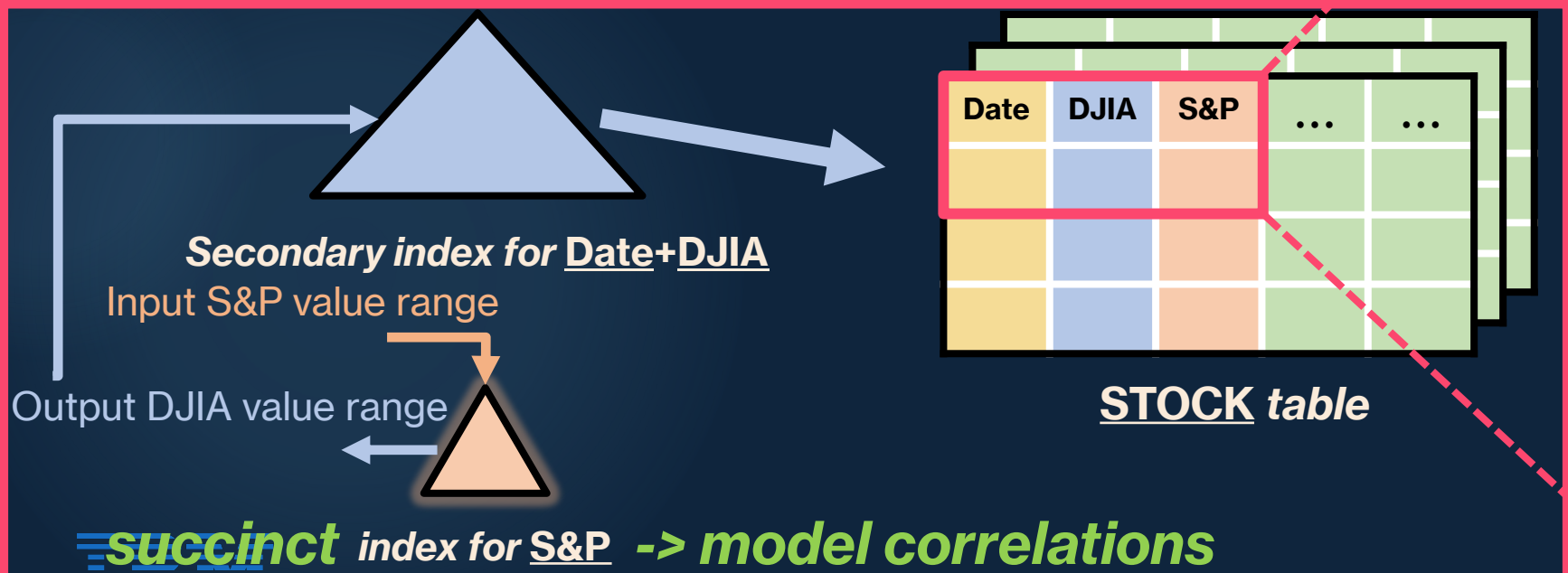
Model the correlation between DJIA and S&P

QUERY 1: SELECT * FROM STOCK WHERE
(date BETWEEN ? AND ?) AND (S&P BETWEEN ? AND ?)

QUERY 2: SELECT * FROM STOCK WHERE
(date BETWEEN ? AND ?) AND (S&P BETWEEN ? AND ?)

Date	DJIA	S&P
1/2/91	2736.39	343.93
2/1/91	2882.18	367.07
3/1/91	2913.86	375.22
4/1/91	2887.87	375.34
5/1/91	3027.50	389.83
6/3/91	2906.75	371.16
7/1/91	3024.82	387.81

8/1/91	3043.60	395.43
9/3/91	3016.77	387.86
10/1/91	3069.10	392.45
11/1/91	2894.68	375.22
12/2/91	3168.83	417.09
1/2/92	3223.40	408.78
2/3/92	3267.70	412.70
3/2/92	3235.50	403.69
4/1/92	3359.10	414.95
5/1/92	3396.90	415.35
6/1/92	3318.50	408.14
7/1/92	3393.80	424.21
8/3/92	3257.40	414.03
9/1/92	3271.70	417.80
10/1/92	3226.30	418.68
11/2/92	3305.20	431.35
12/1/92	3301.11	435.71



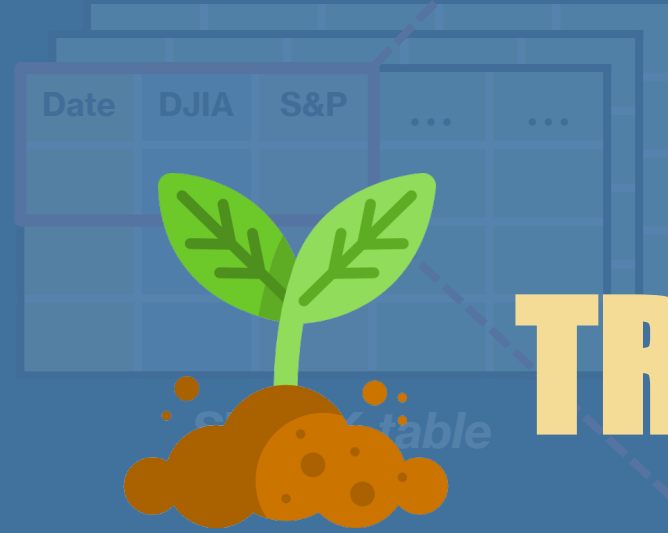
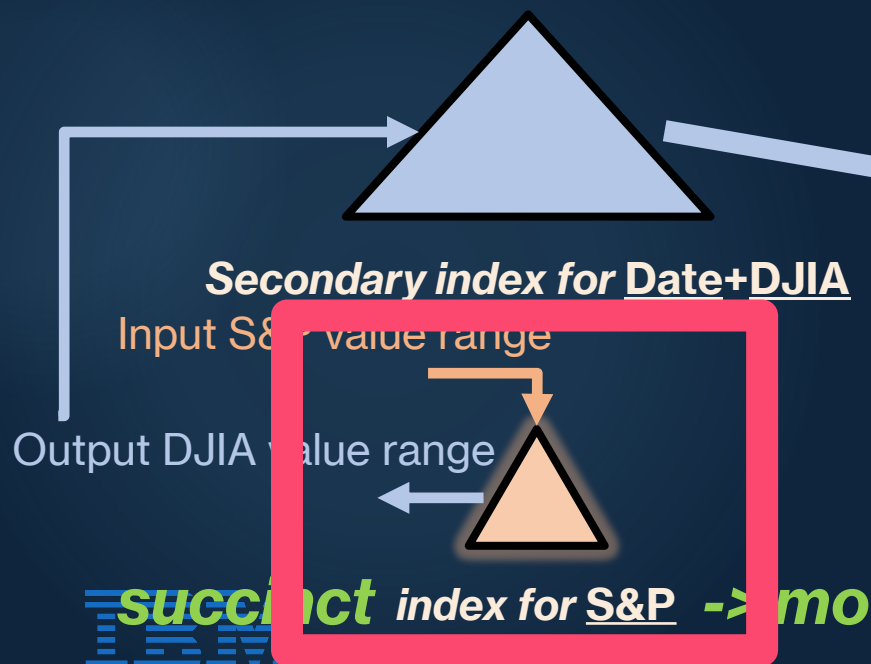
HERMIT: IDEA

25



HERMIT

Date	DJIA	S&P
1/2/91	2736.39	343.93
2/1/91	2882.18	367.07
3/1/91	2913.86	375.22
4/1/91	2887.87	375.34
5/1/91	3027.50	389.83
6/3/91	2906.75	371.16
7/1/91	3024.82	387.81
8/1/91	3043.60	395.43
9/3/91	3016.77	387.86
10/1/91	3069.10	392.45
11/1/91	2894.68	375.22
12/2/91	3168.83	417.09
1/2/92	3223.40	408.78
2/3/92	3267.70	412.70
3/2/92	3235.50	403.69
4/1/92	3259.10	414.95
5/1/92	3259.10	415.35
6/1/92	3259.10	408.14
7/1/92	3259.10	424.21
8/3/92	3257.40	414.03
9/1/92	3271.70	417.80
10/1/92	3226.30	418.68
11/2/92	3305.20	431.35
12/1/92	3301.11	435.71



TRS-TREE

HERMIT: TRS-TREE DESIGN

26

- ▶ Tiered Regression Search Tree (TRS-Tree)



HERMIT: TRS-TREE DESIGN

27

- ▶ Tiered Regression Search Tree (TRS-Tree)
 - ▶ A succinct, *ML-enhanced*, tree-structured index

Feature #1



HERMIT: TRS-TREE DESIGN

28

- ▶ Tiered Regression Search Tree (TRS-Tree)
 - ▶ A succinct, **ML-enhanced**, tree-structured index
 - ▶ Capture **correlation** while handling **outliers**

Feature #1



ML

Tree

Feature #2



Correlation

Outlier



HERMIT: TRS-TREE DESIGN

29

- ▶ Tiered Regression Search Tree (TRS-Tree)
 - ▶ A succinct, **ML-enhanced**, tree-structured index
 - ▶ Capture **correlation** while handling **outliers**
 - ▶ **Adaptively** and **dynamically** construct and reorganize internal structures

Feature #1



ML

Tree

Feature #2



Correlation

Outlier

Feature #3



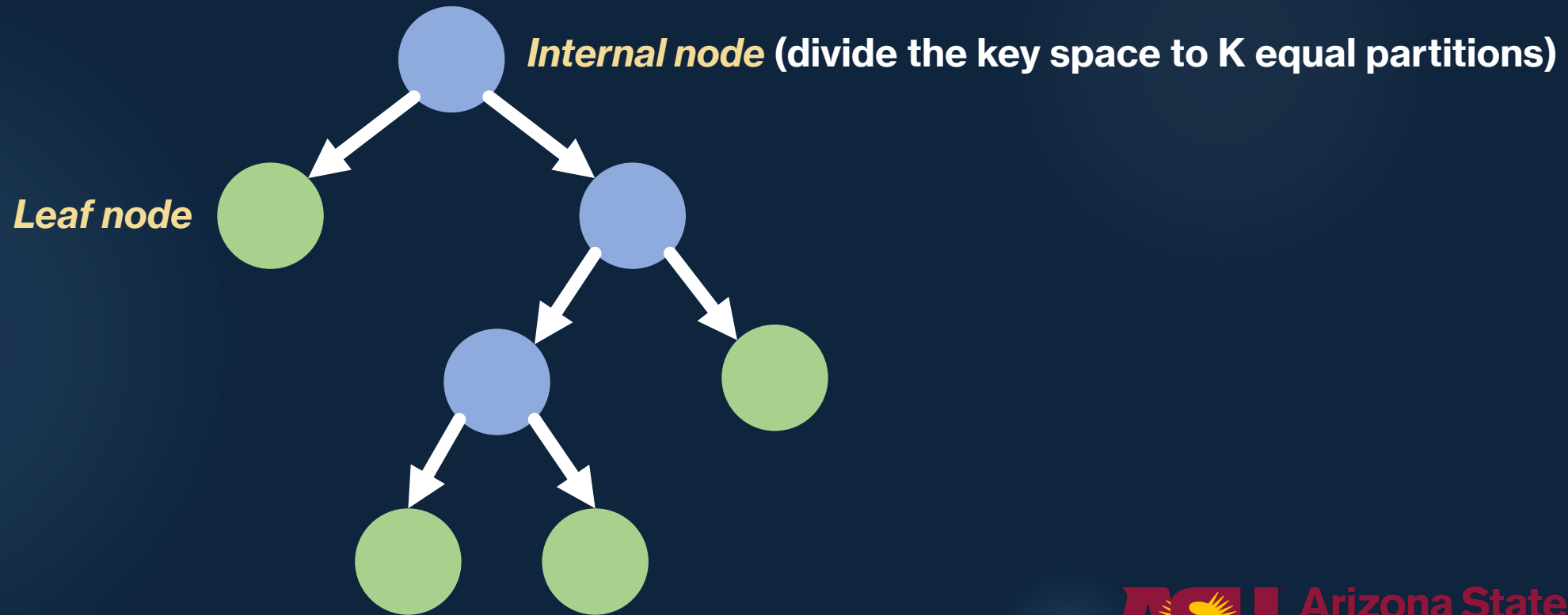
Adaptive

Dynamic



HERMIT: TRS-TREE DESIGN

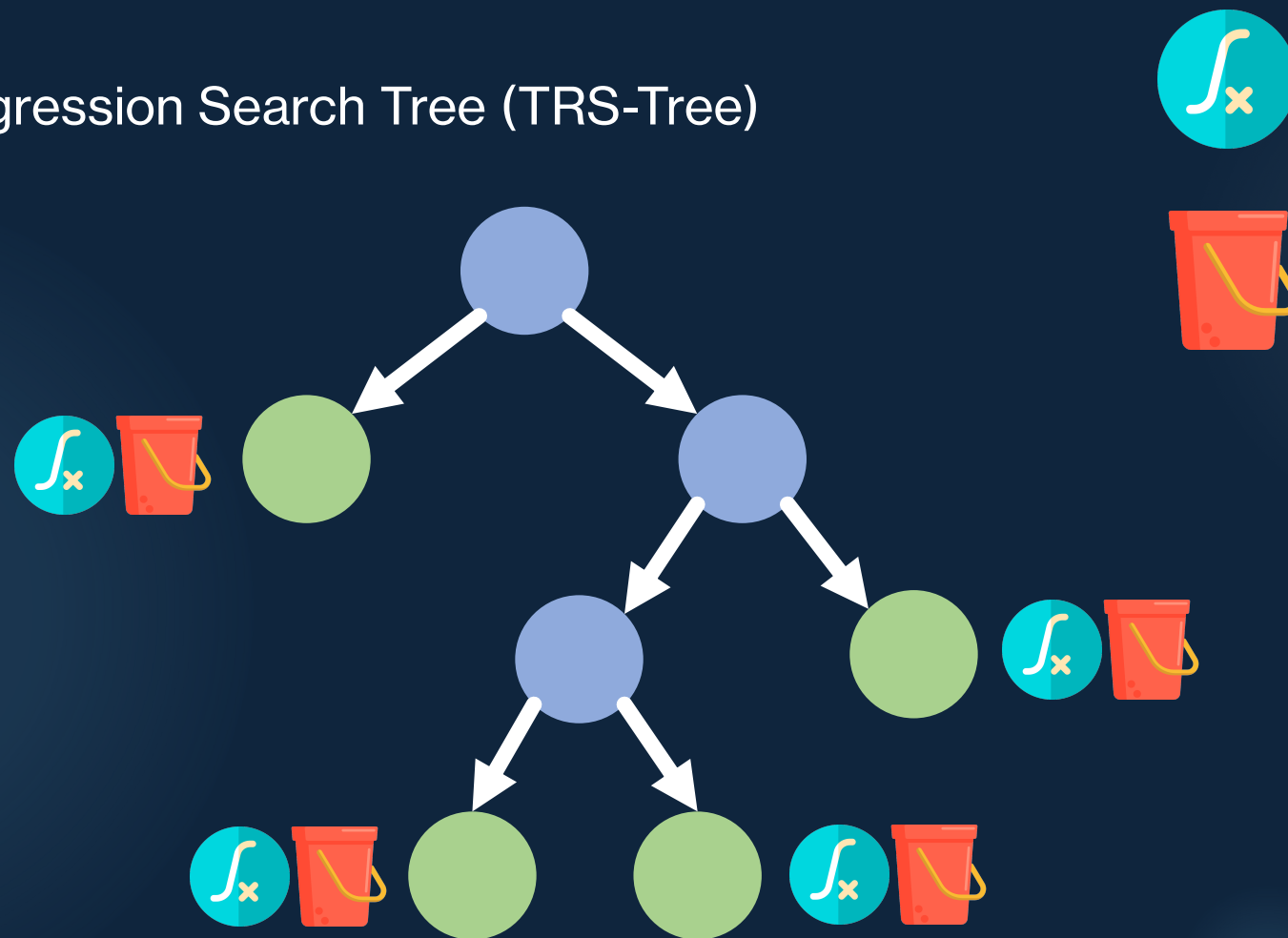
- ## ► Tiered Regression Search Tree (TRS-Tree)



HERMIT: TRS-TREE DESIGN

31

► Tiered Regression Search Tree (TRS-Tree)



Correlation function
 $y = \beta x + \alpha \pm \epsilon$
(simple linear regression)

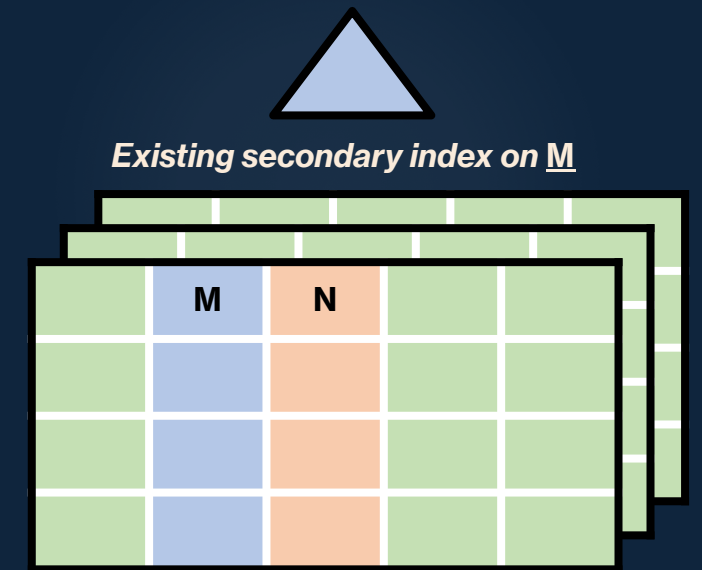


Outlier buffer

HERMIT: TRS-TREE DESIGN

32

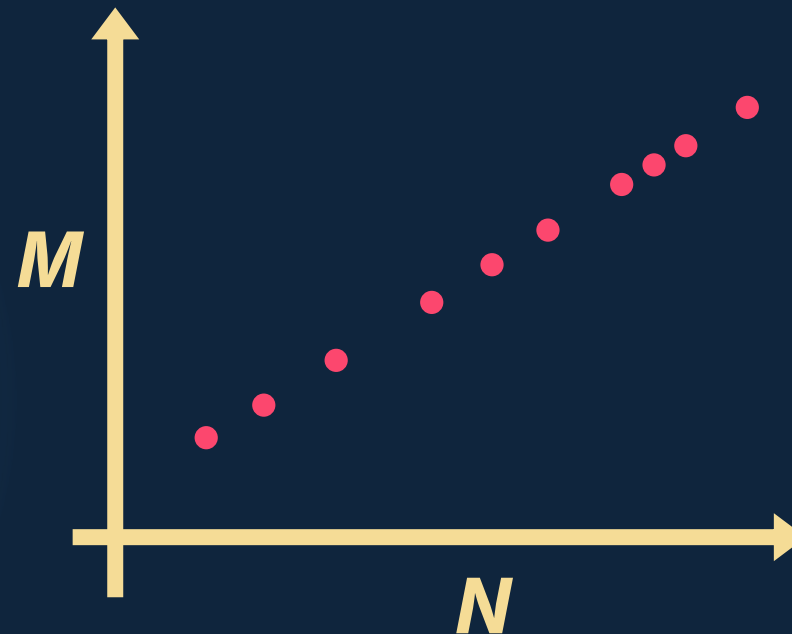
- ▶ TRS-Tree: model correlation between columns **M** and **N**



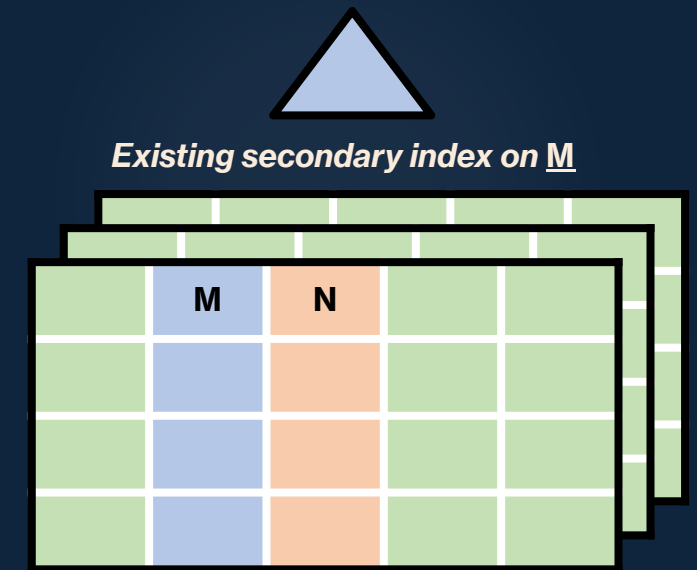
HERMIT: TRS-TREE DESIGN

33

- ▶ TRS-Tree: model correlation between columns **M** and **N**



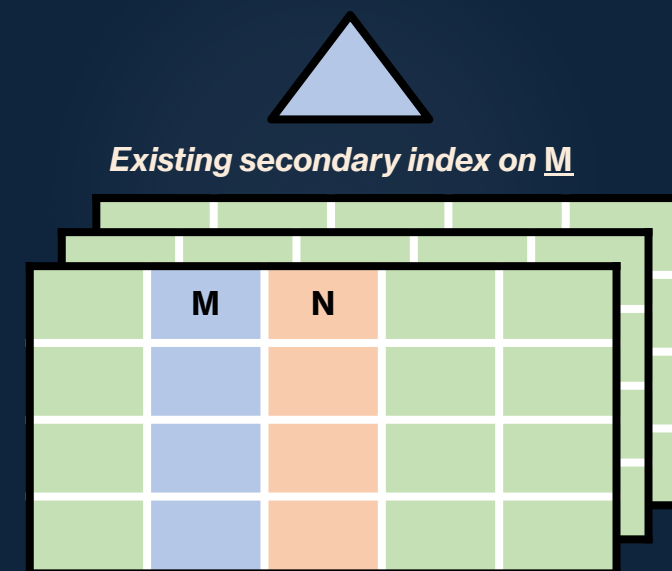
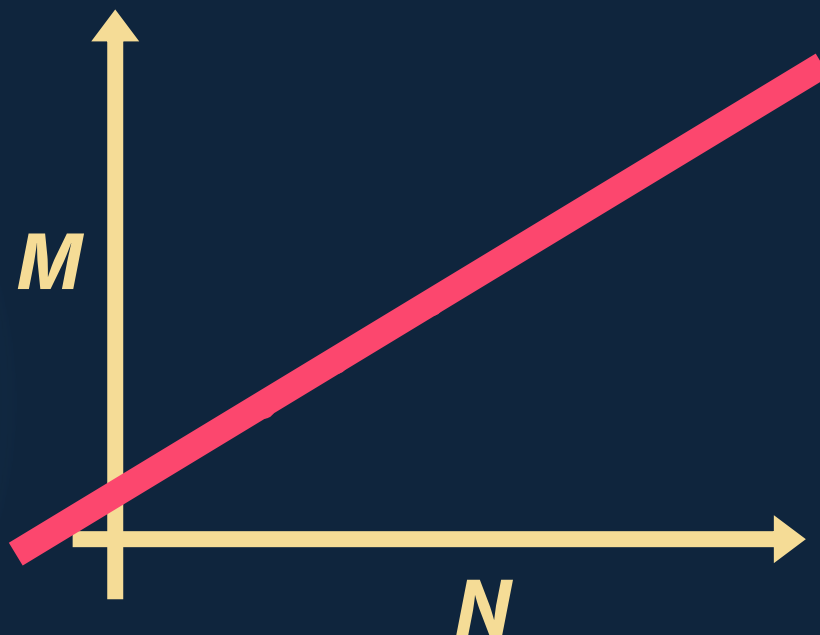
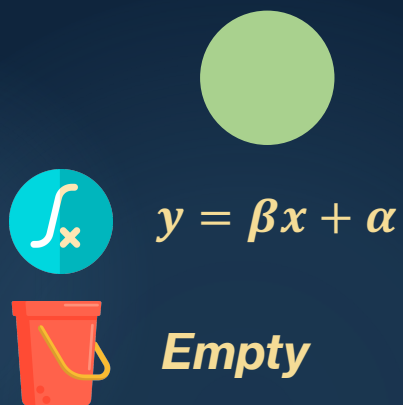
TRS-Tree as a mapping from N to M



HERMIT: TRS-TREE DESIGN

34

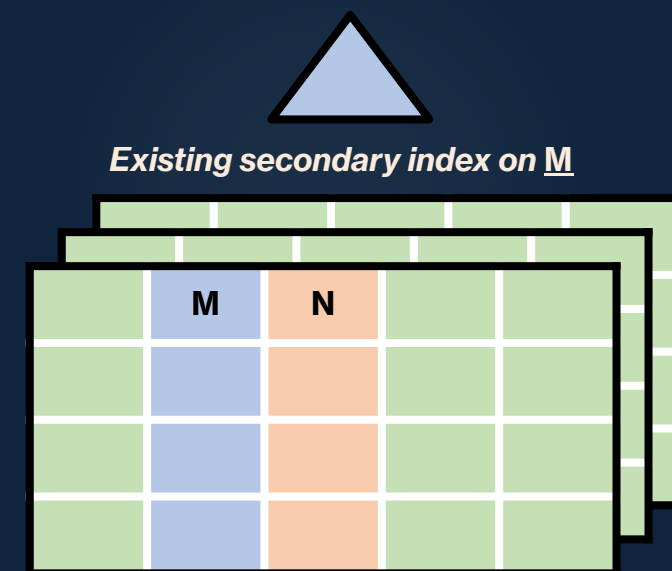
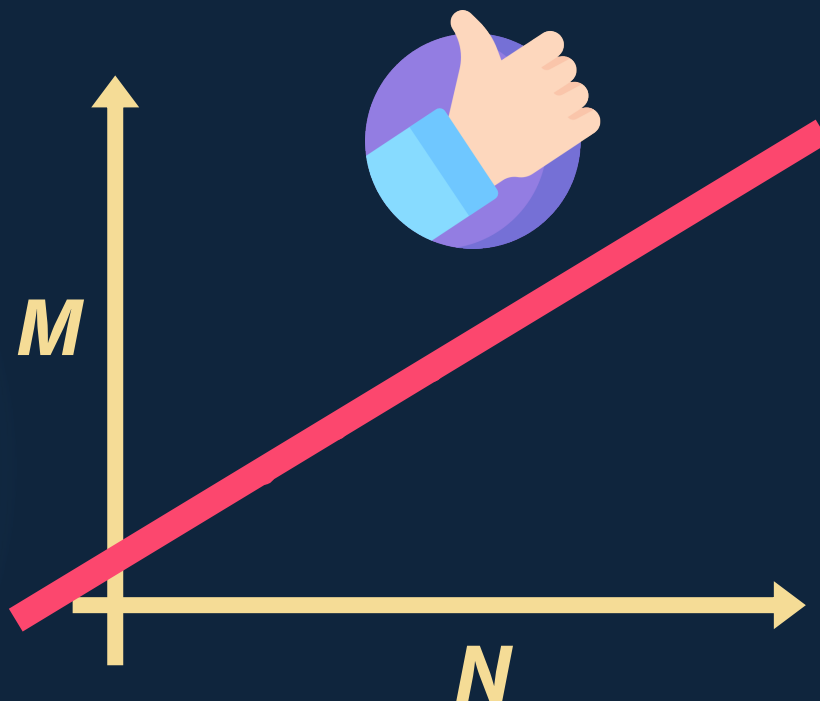
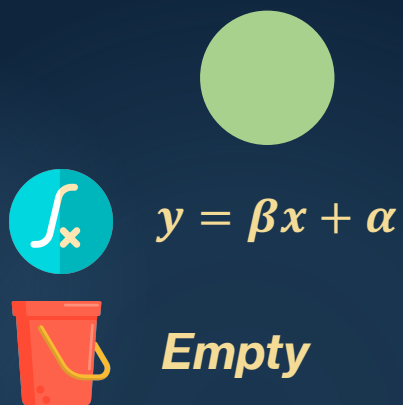
- ▶ TRS-Tree: model correlation between columns **M** and **N**



HERMIT: TRS-TREE DESIGN

35

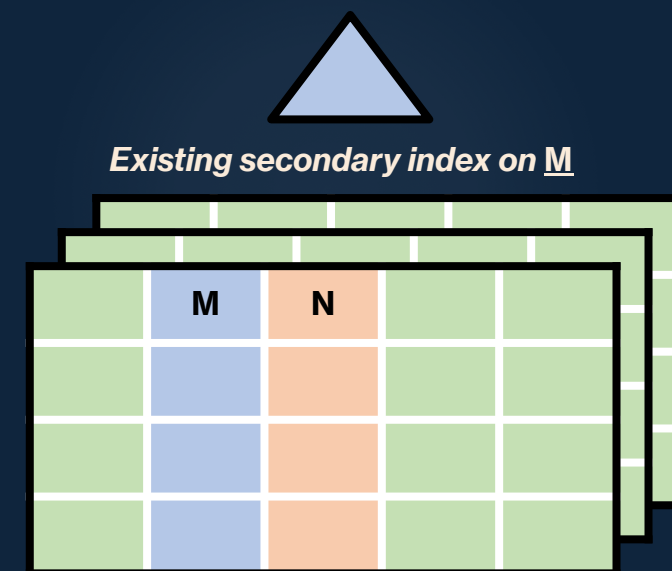
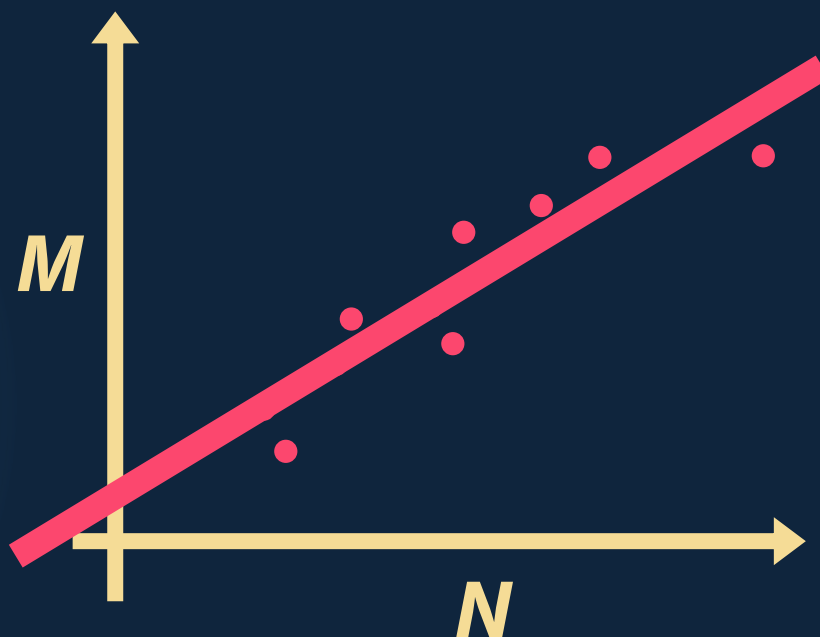
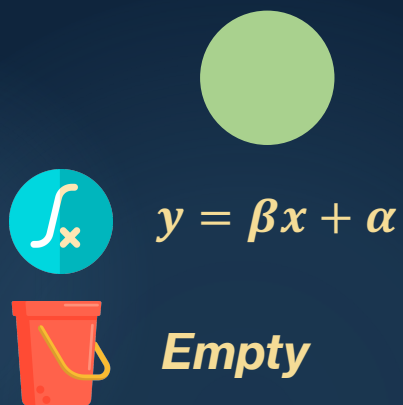
- ▶ TRS-Tree: model correlation between columns **M** and **N**



HERMIT: TRS-TREE DESIGN

36

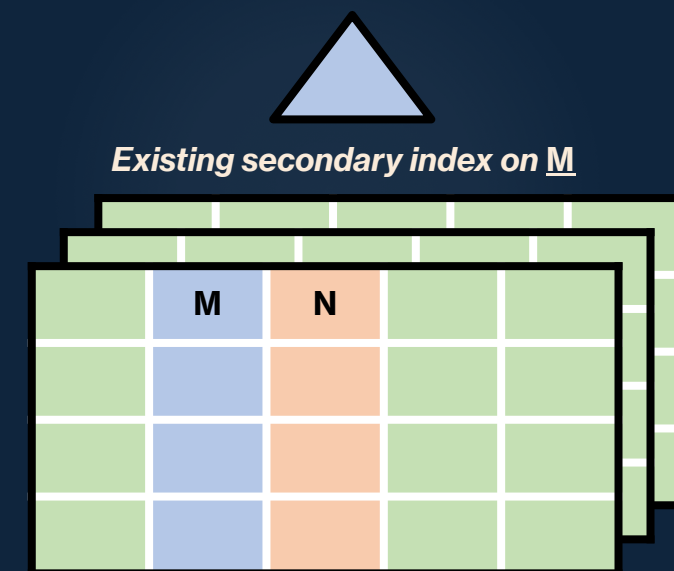
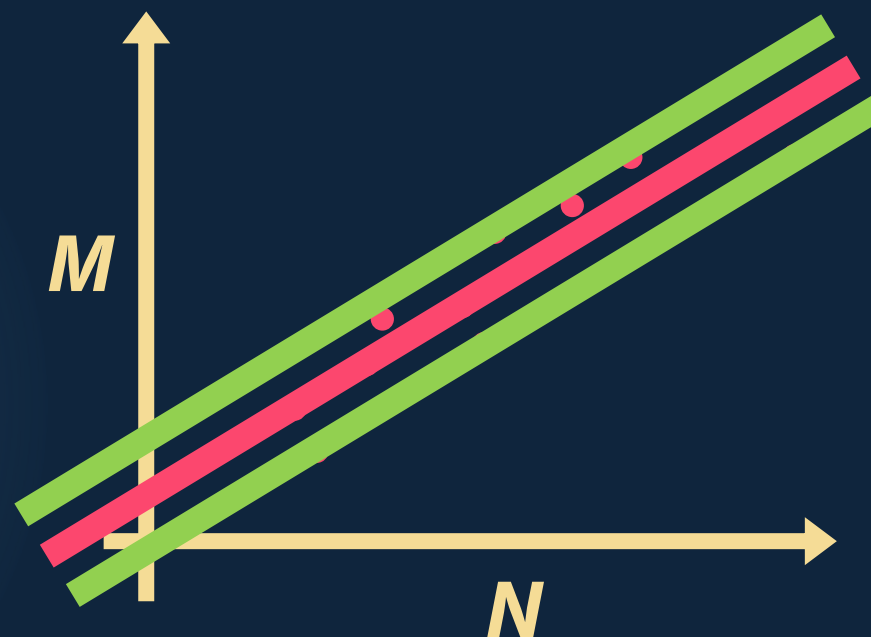
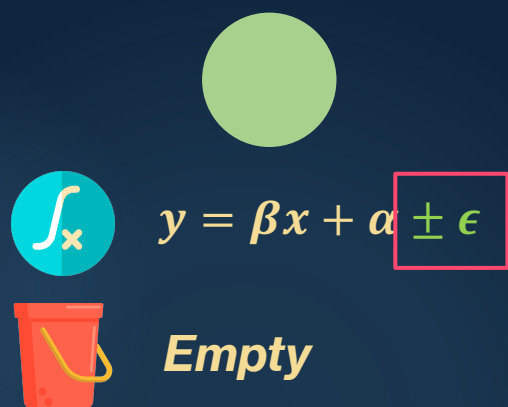
- ▶ TRS-Tree: model correlation between columns **M** and **N**



HERMIT: TRS-TREE DESIGN

37

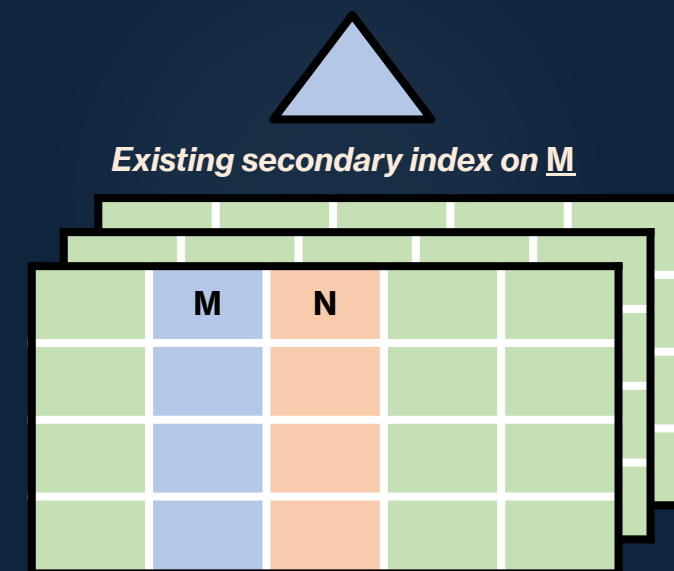
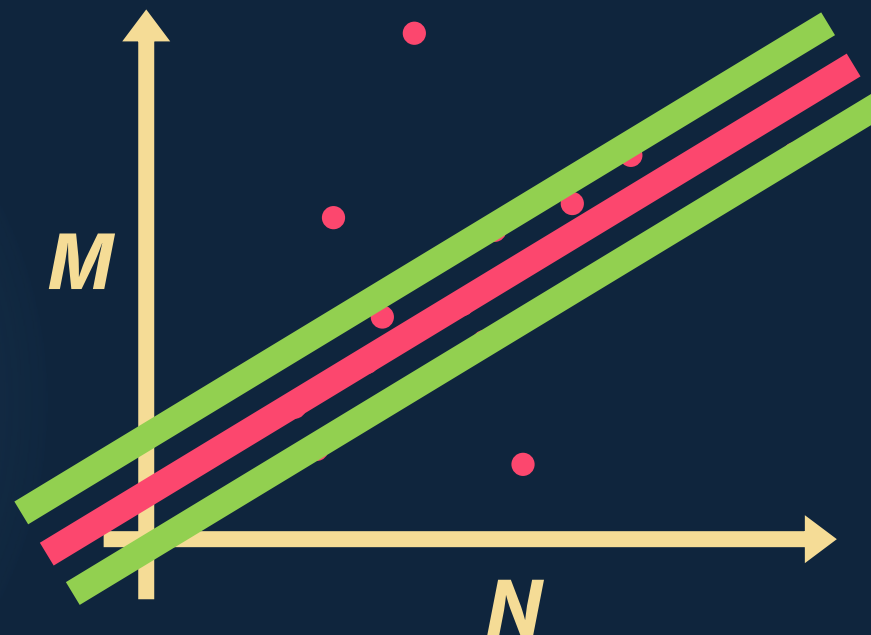
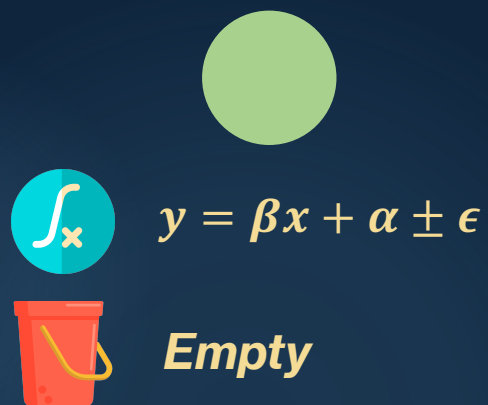
- TRS-Tree: model correlation between columns **M** and **N**



HERMIT: TRS-TREE DESIGN

38

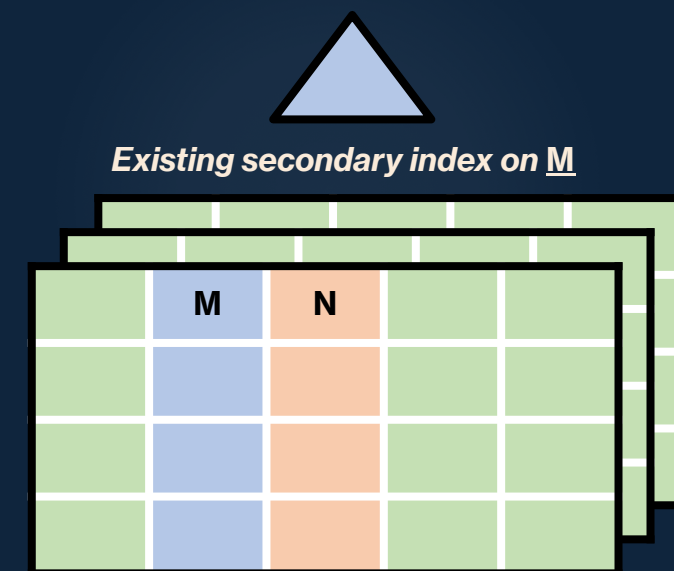
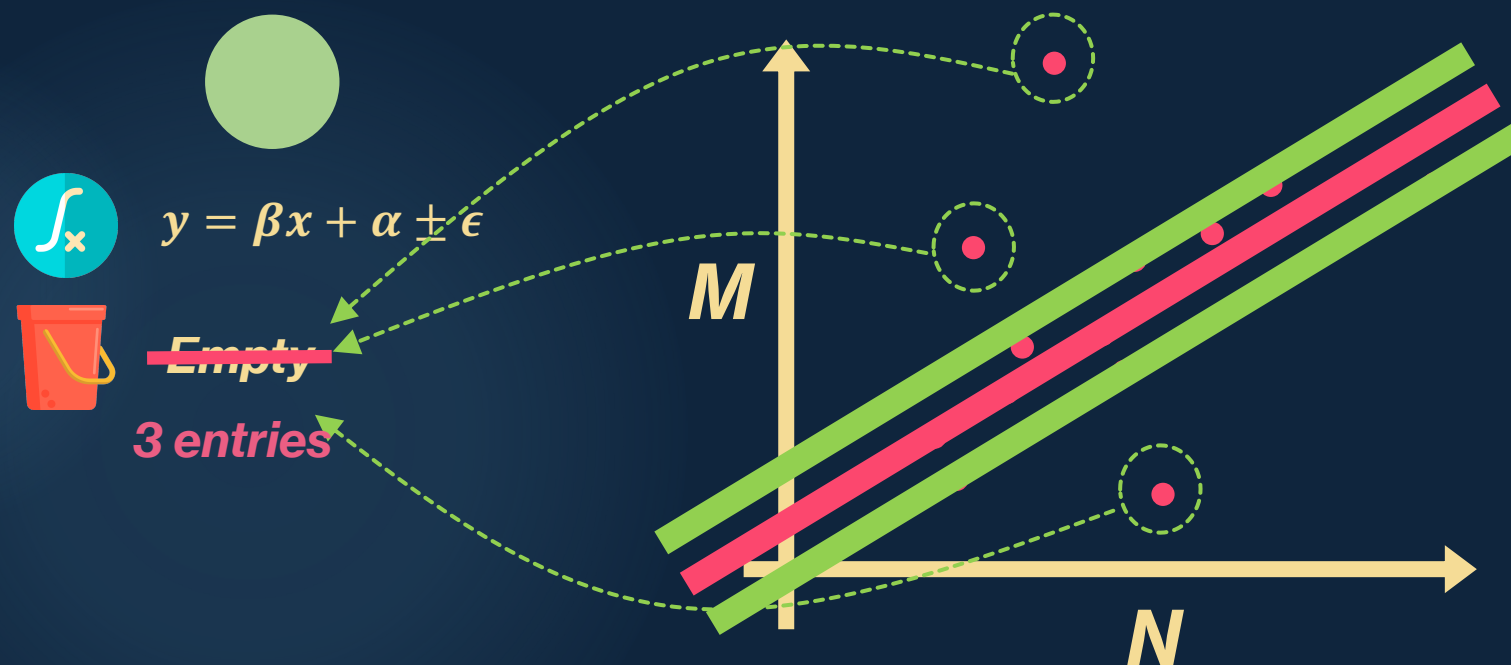
- ▶ TRS-Tree: model correlation between columns **M** and **N**



HERMIT: TRS-TREE DESIGN

39

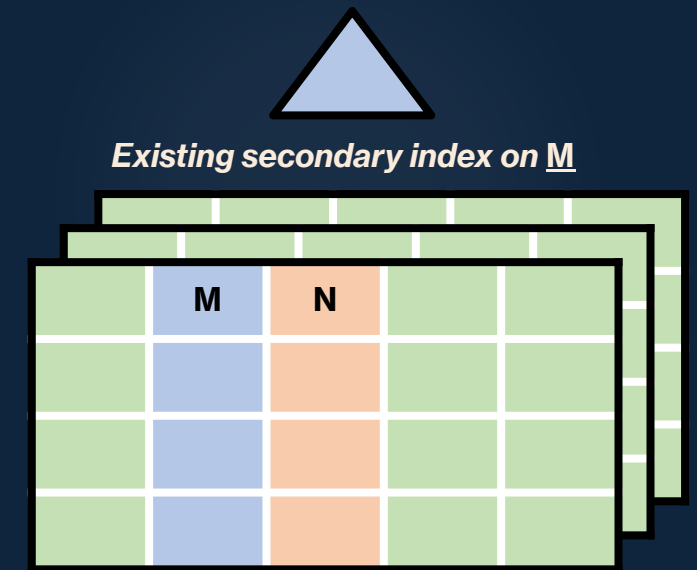
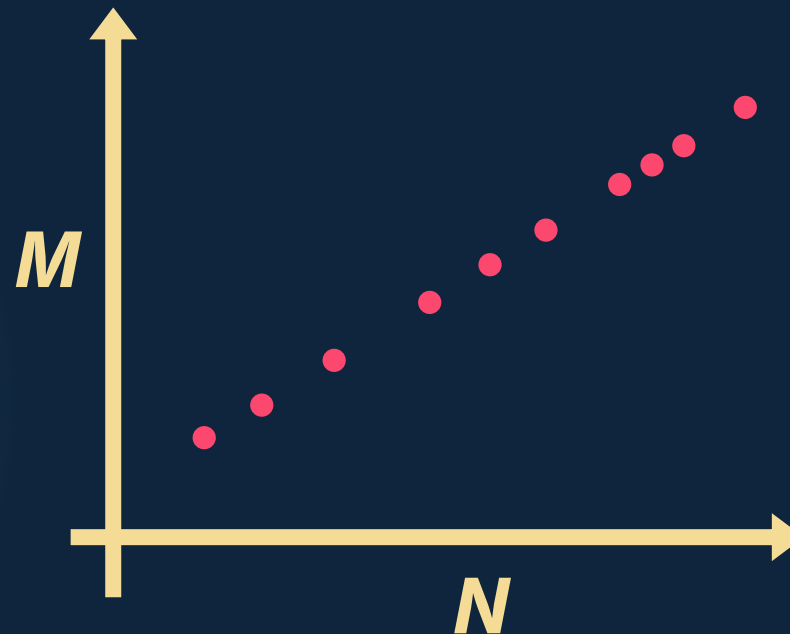
- TRS-Tree: model correlation between columns **M** and **N**



HERMIT: TRS-TREE DESIGN

40

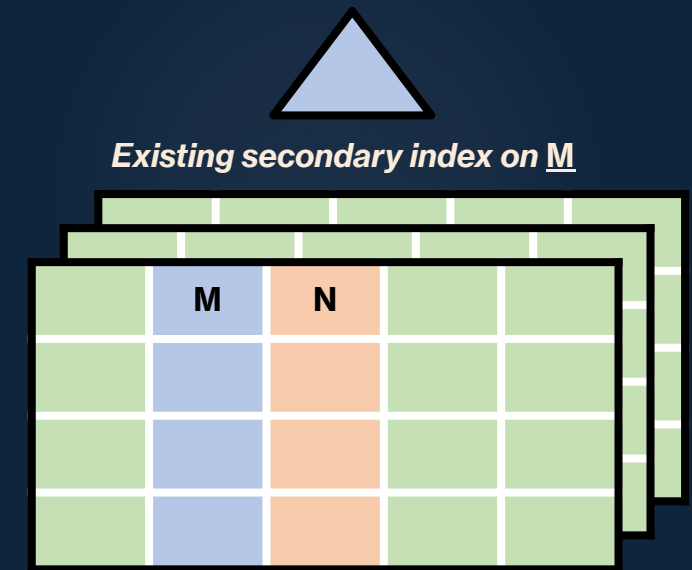
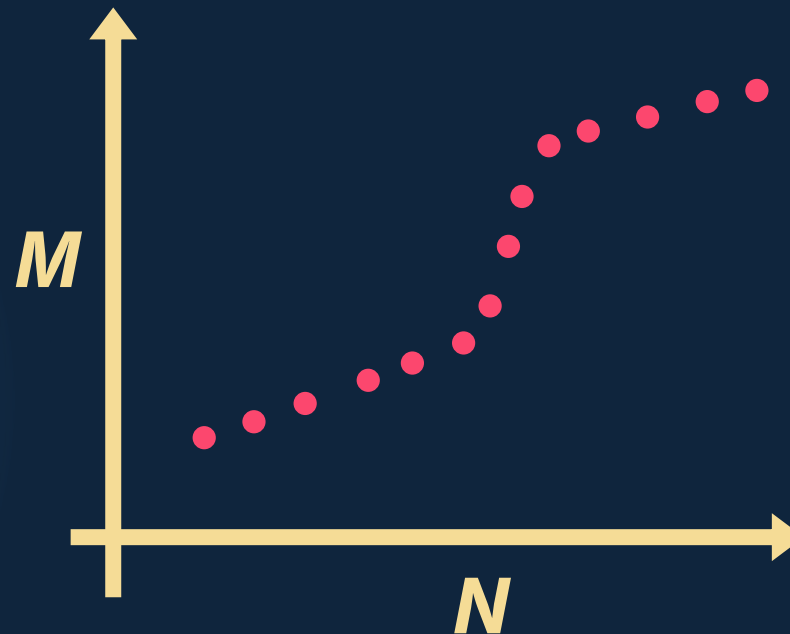
- ▶ TRS-Tree: model correlation between columns **M** and **N**



HERMIT: TRS-TREE DESIGN

41

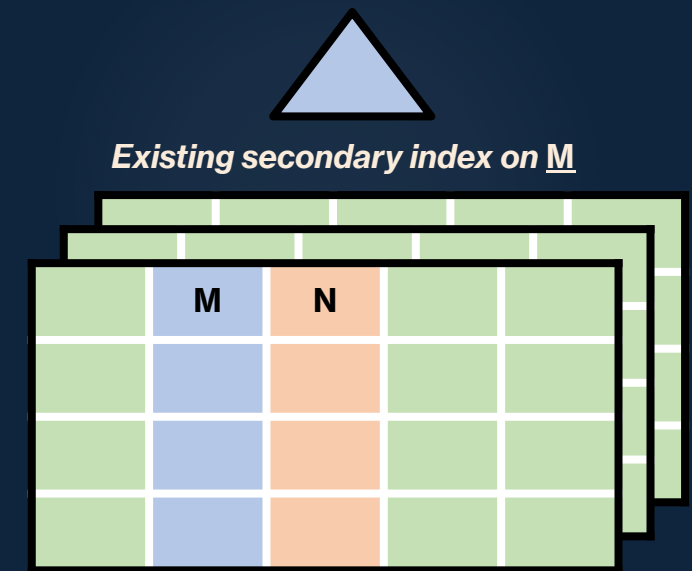
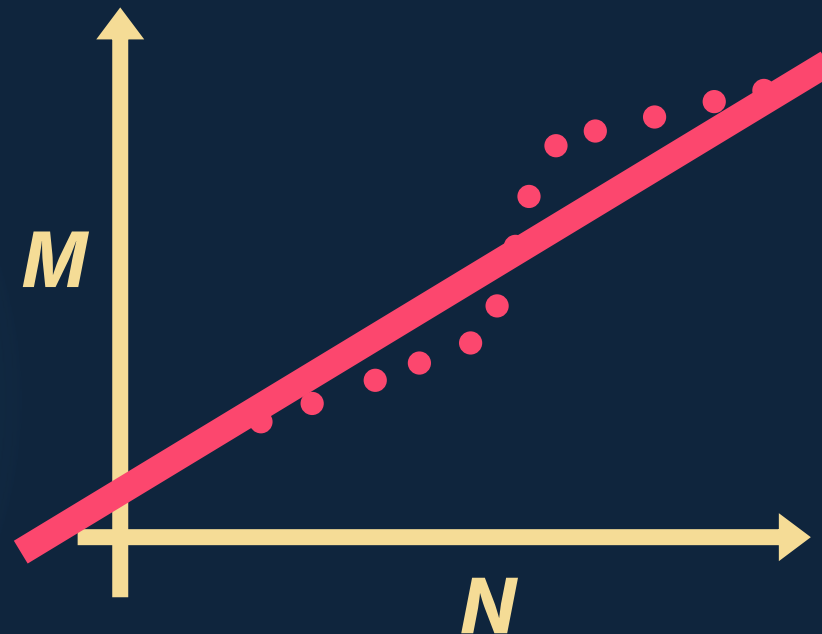
- ▶ TRS-Tree: model correlation between columns **M** and **N**



HERMIT: TRS-TREE DESIGN

42

- ▶ TRS-Tree: model correlation between columns **M** and **N**

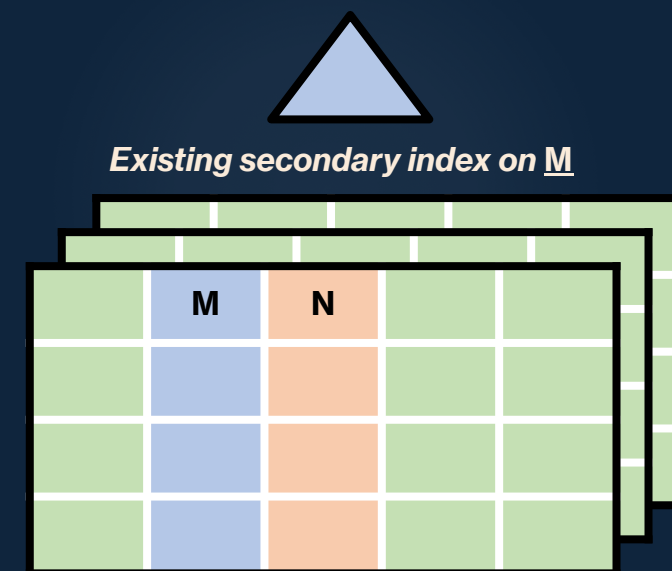
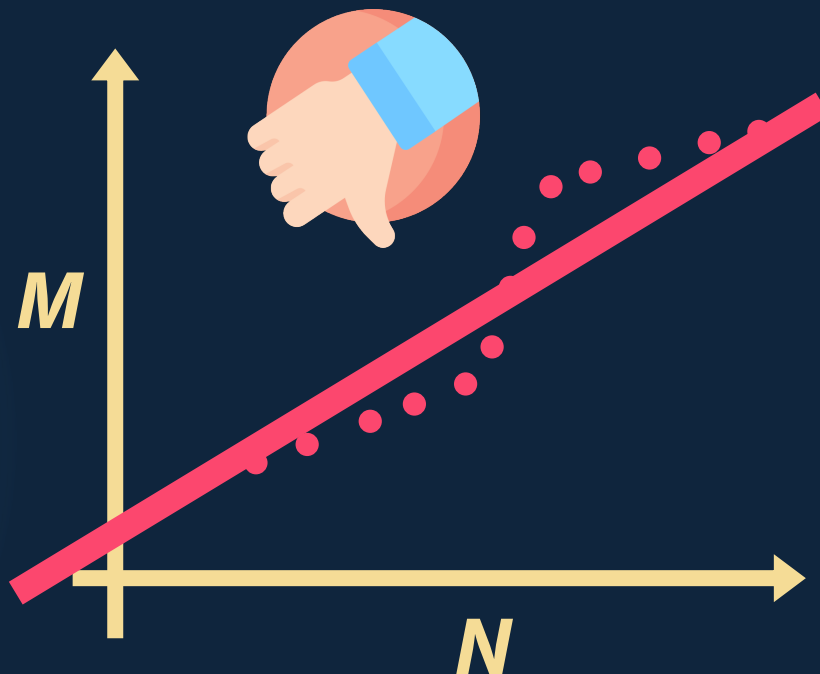


HERMIT: TRS-TREE DESIGN

43

- ▶ TRS-Tree: model correlation between columns **M** and **N**


 $y = \beta x + \alpha \pm \epsilon$
 *Lots of entries!!*

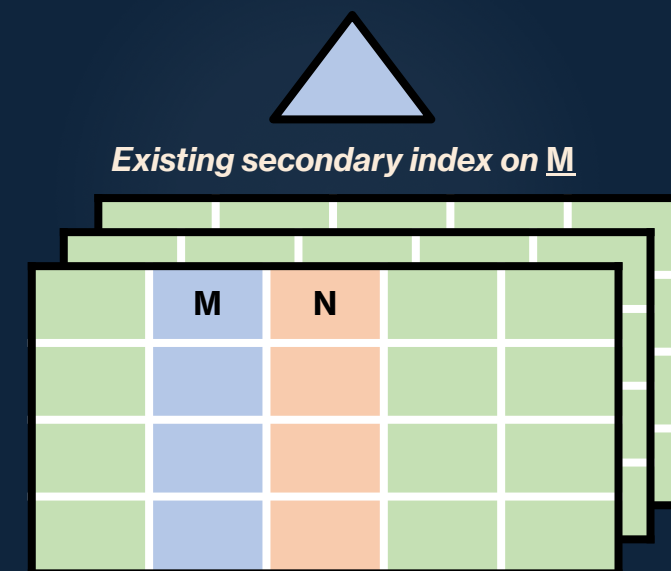
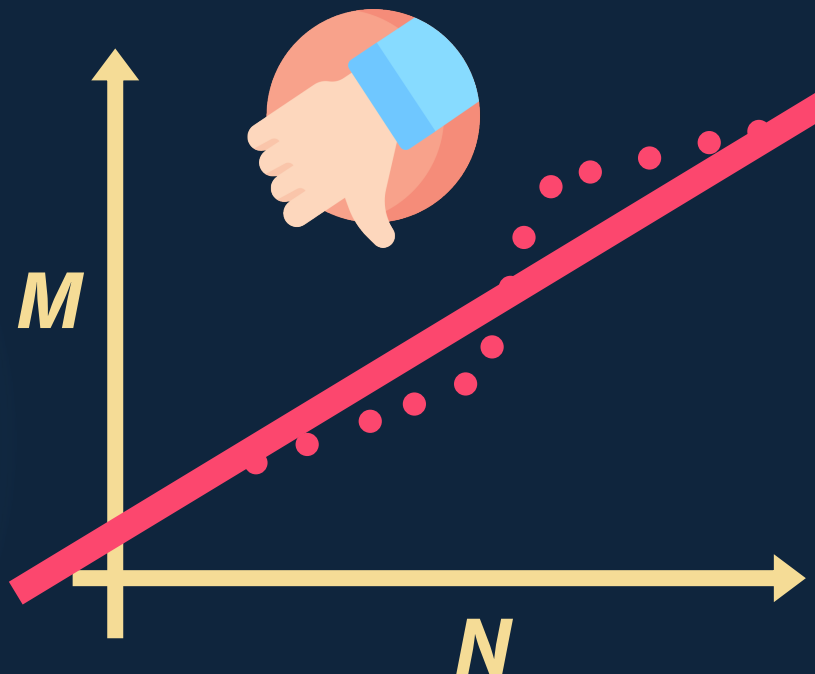
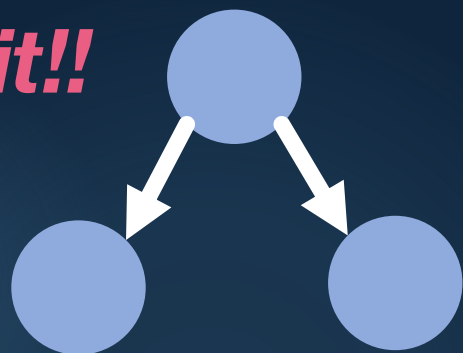


HERMIT: TRS-TREE DESIGN

44

- TRS-Tree: model correlation between columns **M** and **N**

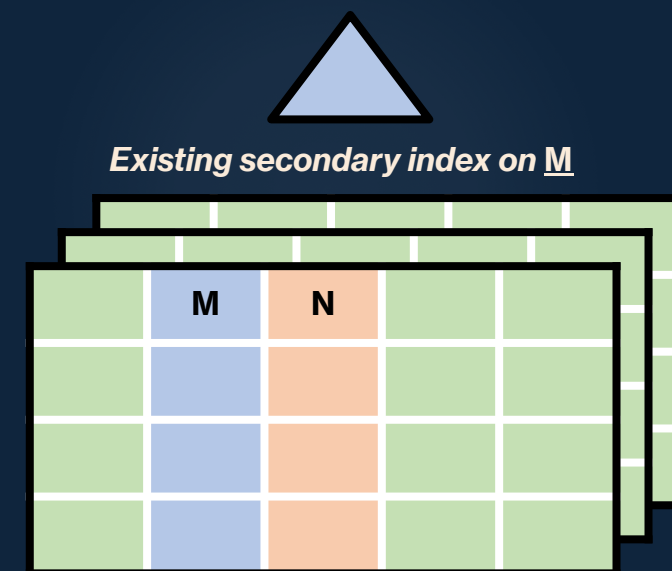
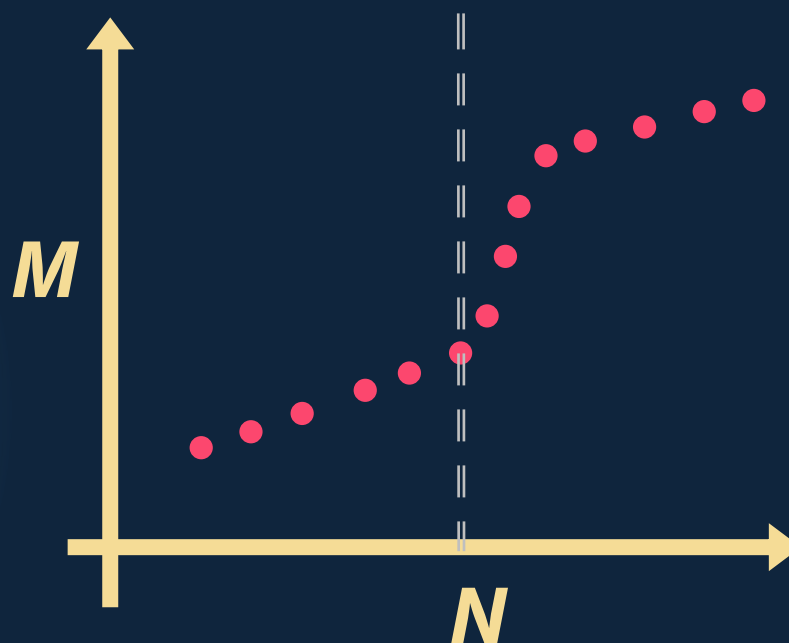
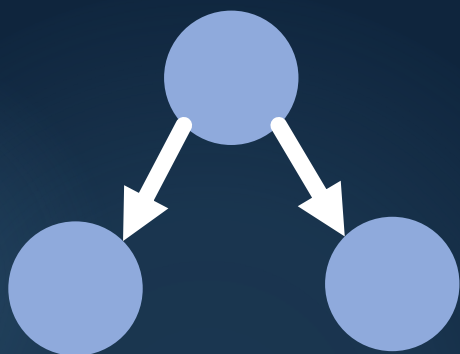
Split!!



HERMIT: TRS-TREE DESIGN

45

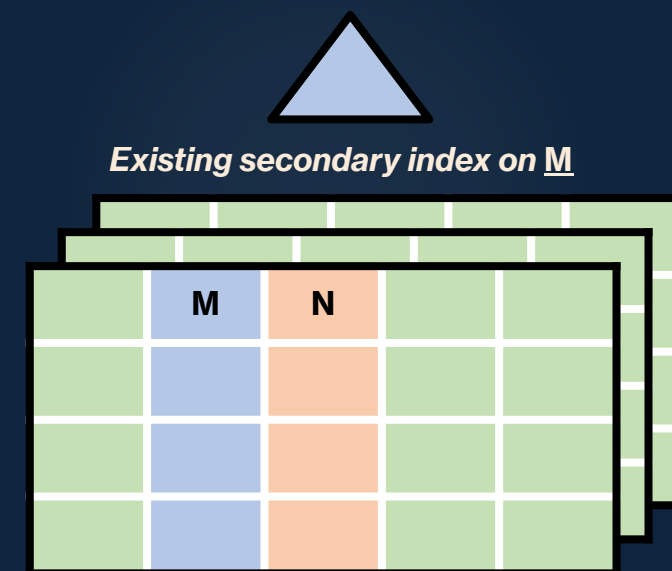
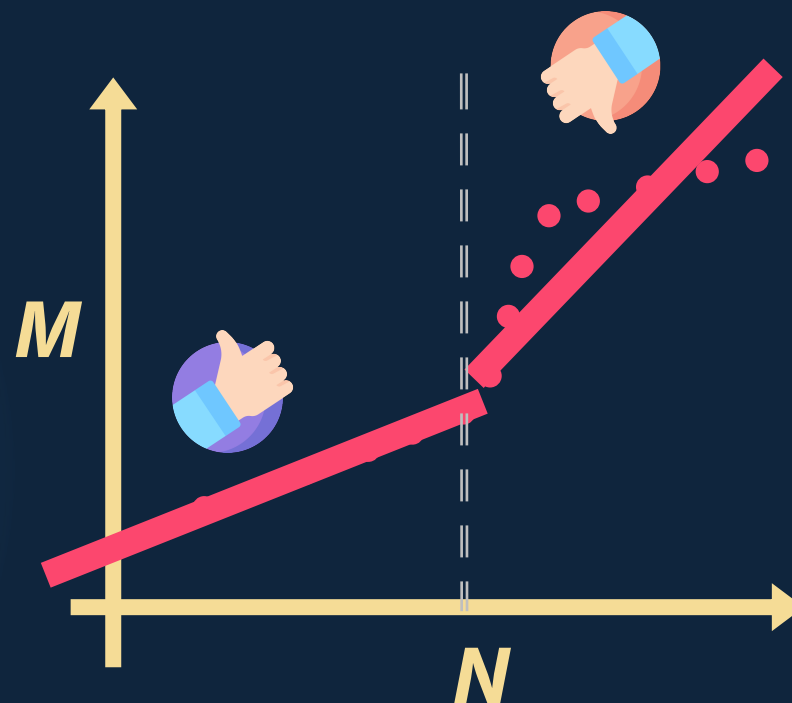
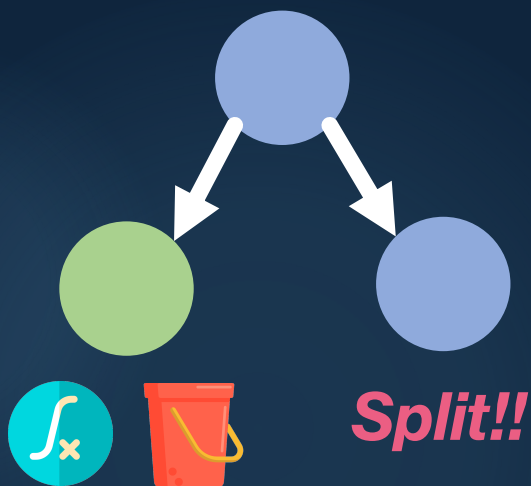
- ▶ TRS-Tree: model correlation between columns **M** and **N**



HERMIT: TRS-TREE DESIGN

46

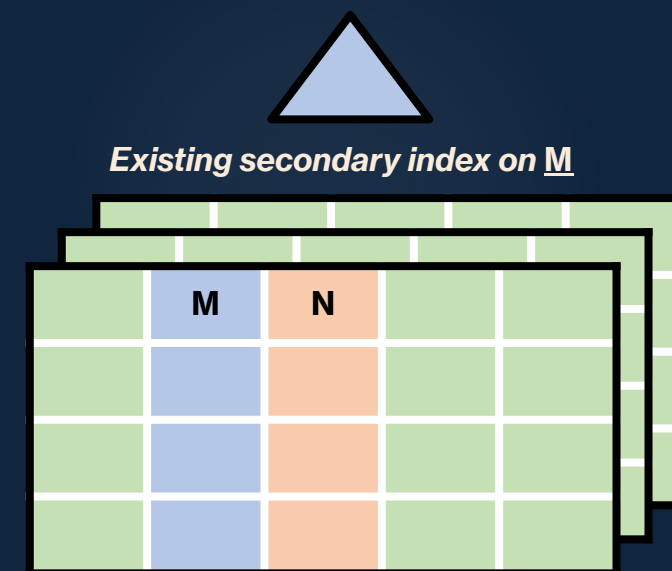
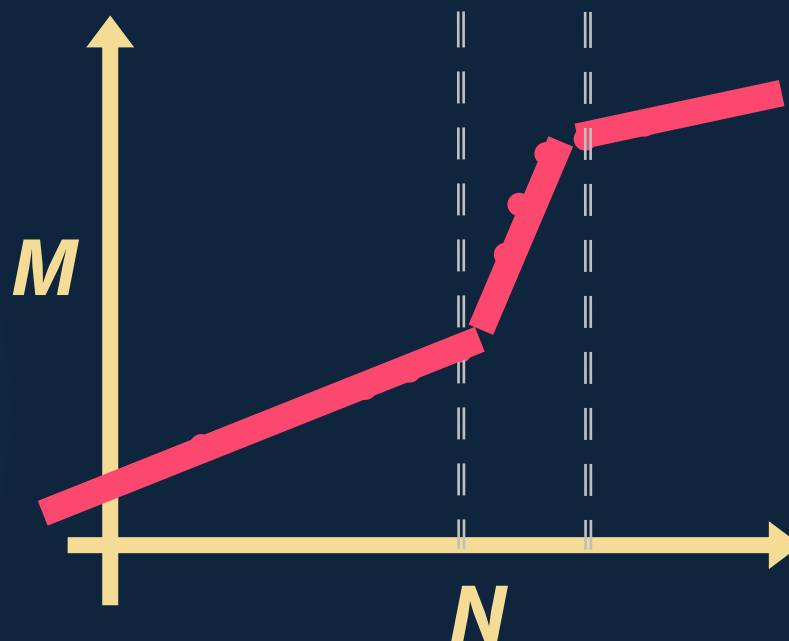
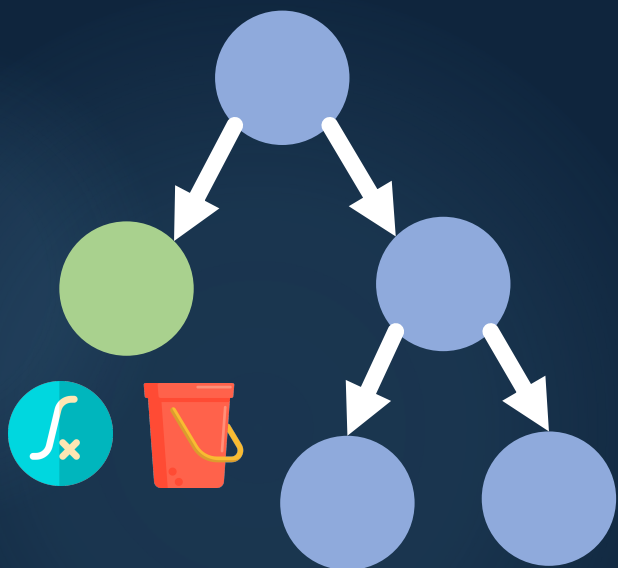
- TRS-Tree: model correlation between columns **M** and **N**



HERMIT: TRS-TREE DESIGN

47

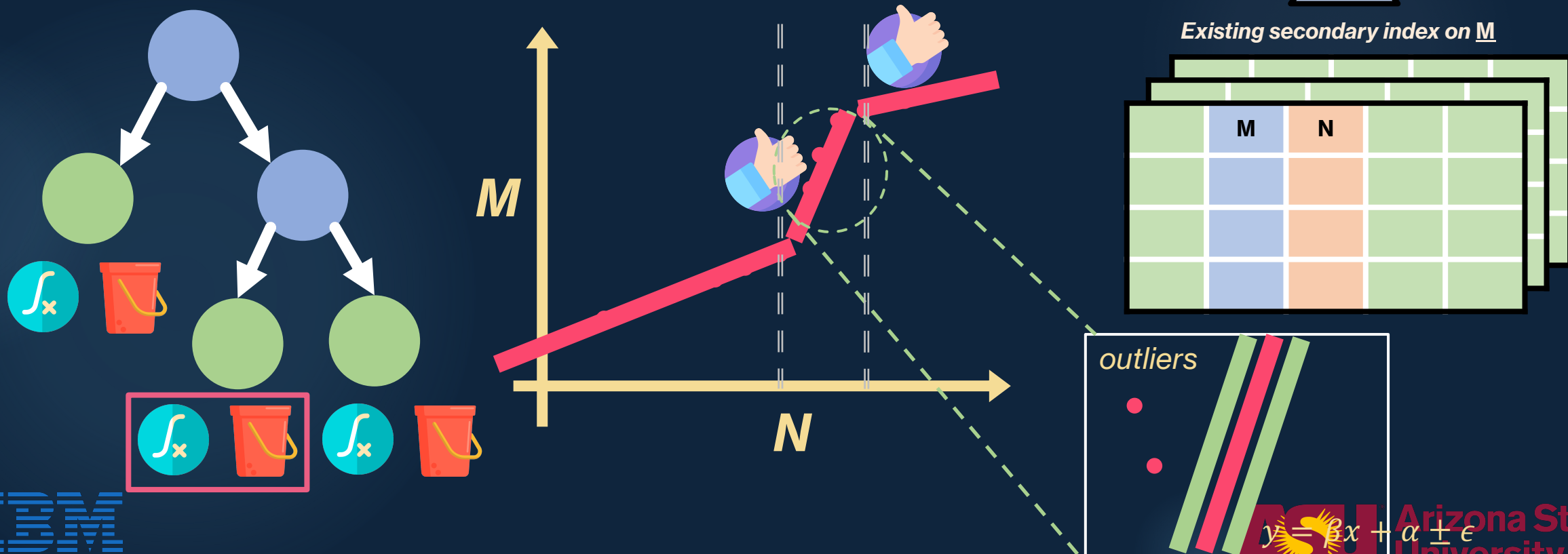
- TRS-Tree: model correlation between columns **M** and **N**



HERMIT: TRS-TREE DESIGN

48

- TRS-Tree: model correlation between columns **M** and **N**

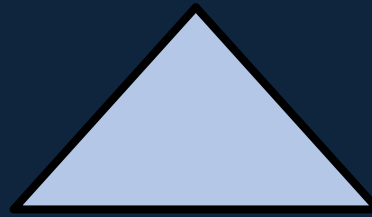


HERMIT: ACCESS METHOD

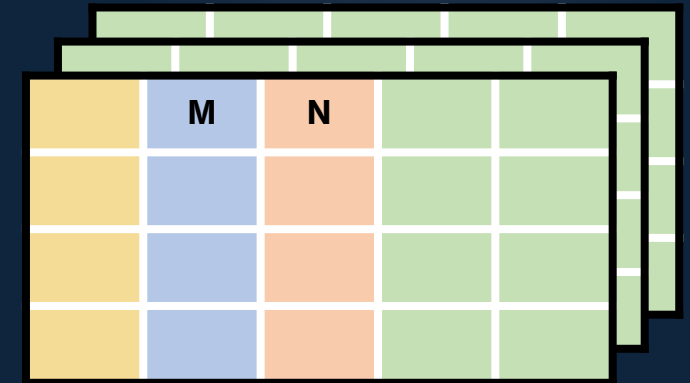
49



TRS-Tree Index on N



Existing Secondary Index on M

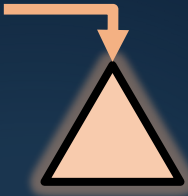


Base table

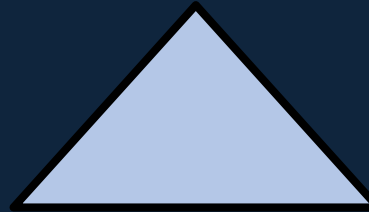
HERMIT: ACCESS METHOD

50

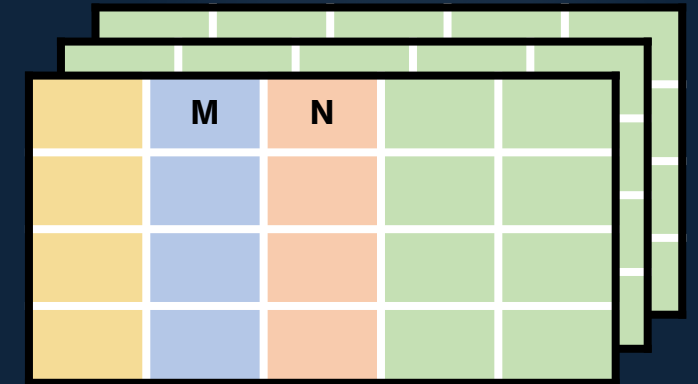
Input value range on N



TRS-Tree Index on N



Existing Secondary Index on M

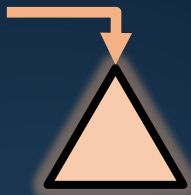


Base table

HERMIT: ACCESS METHOD

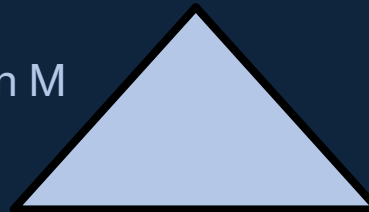
51

Input value range on N

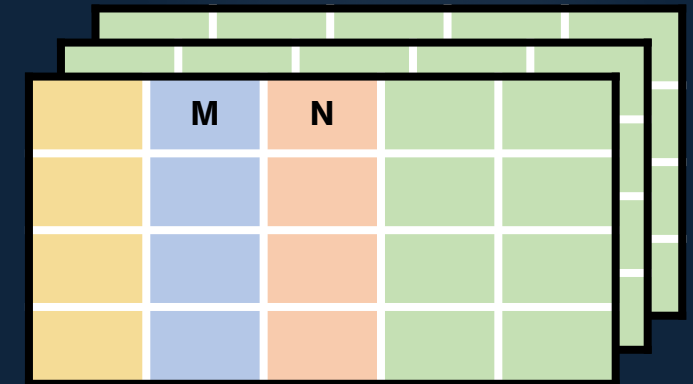


TRS-Tree Index on N

Output value range on M



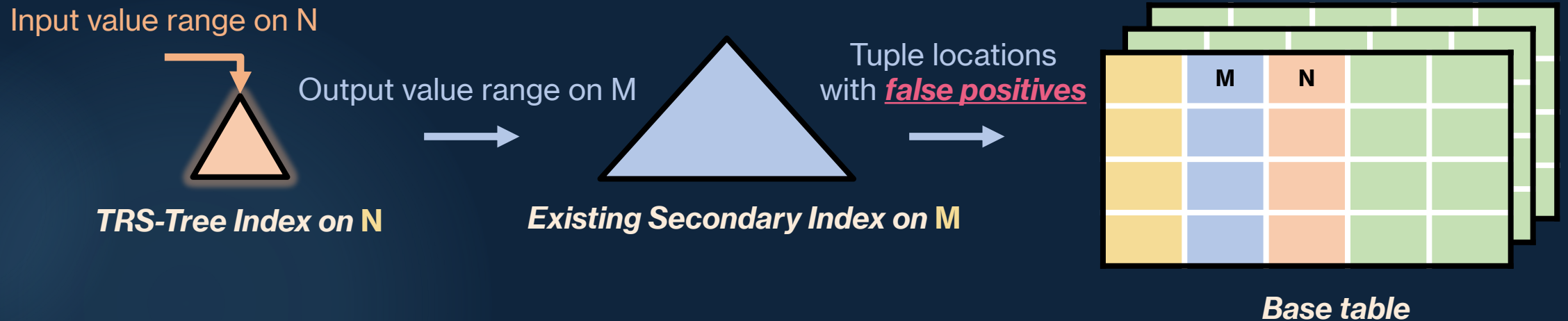
Existing Secondary Index on M



Base table

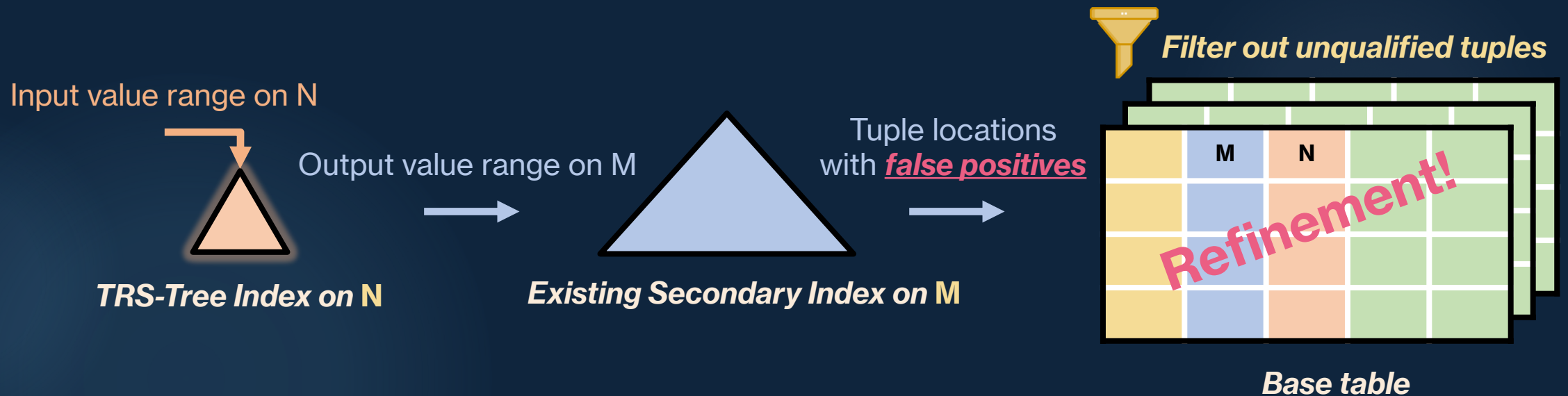
HERMIT: ACCESS METHOD

52



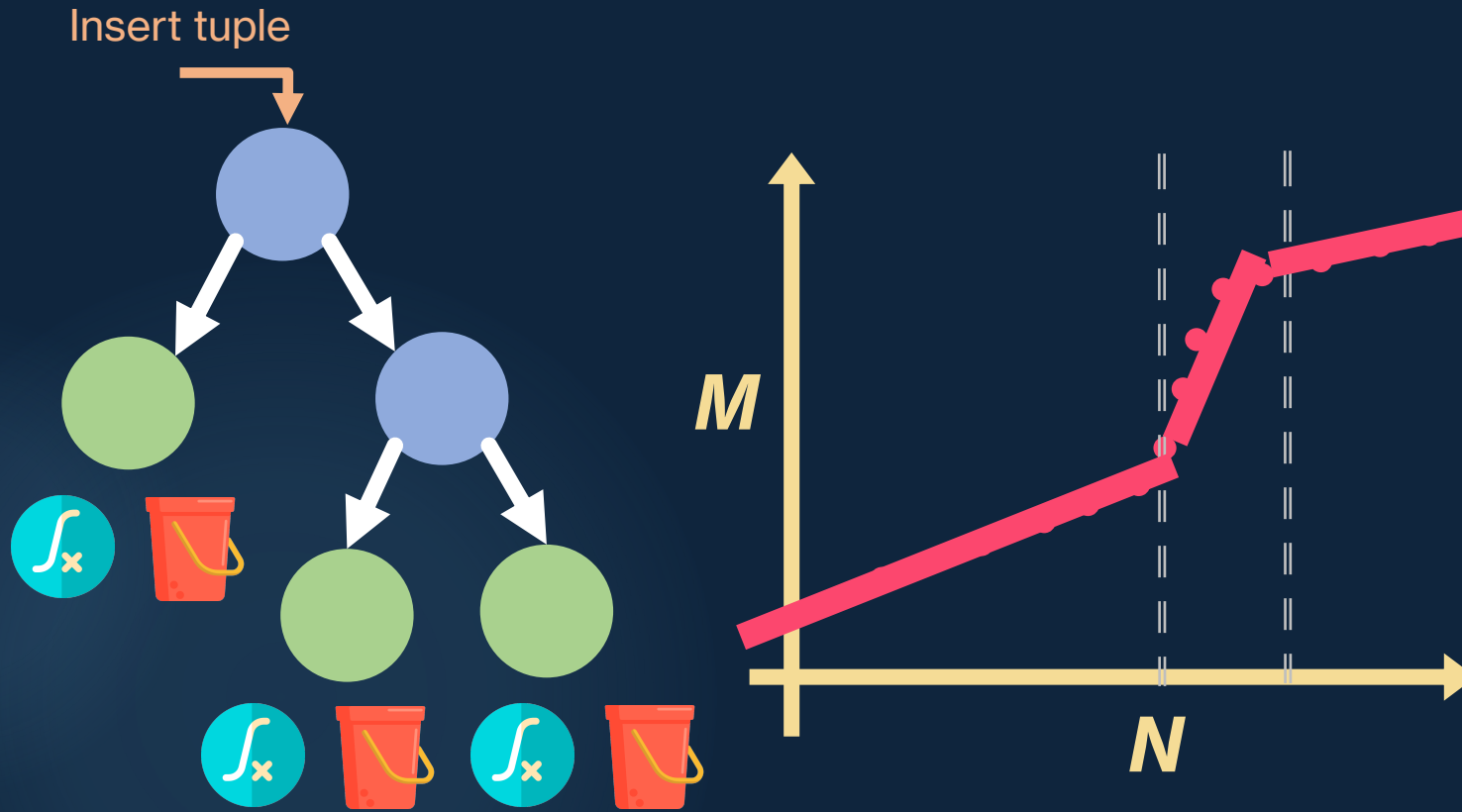
HERMIT: ACCESS METHOD

53



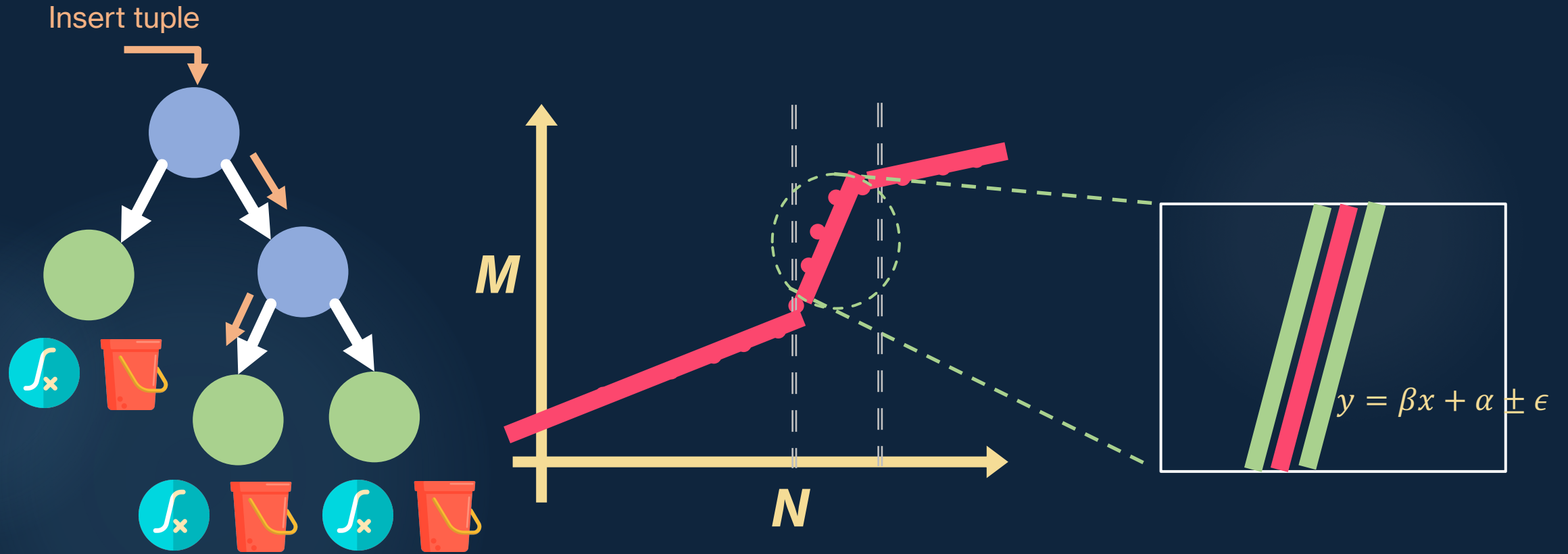
HERMIT: INSERT

54



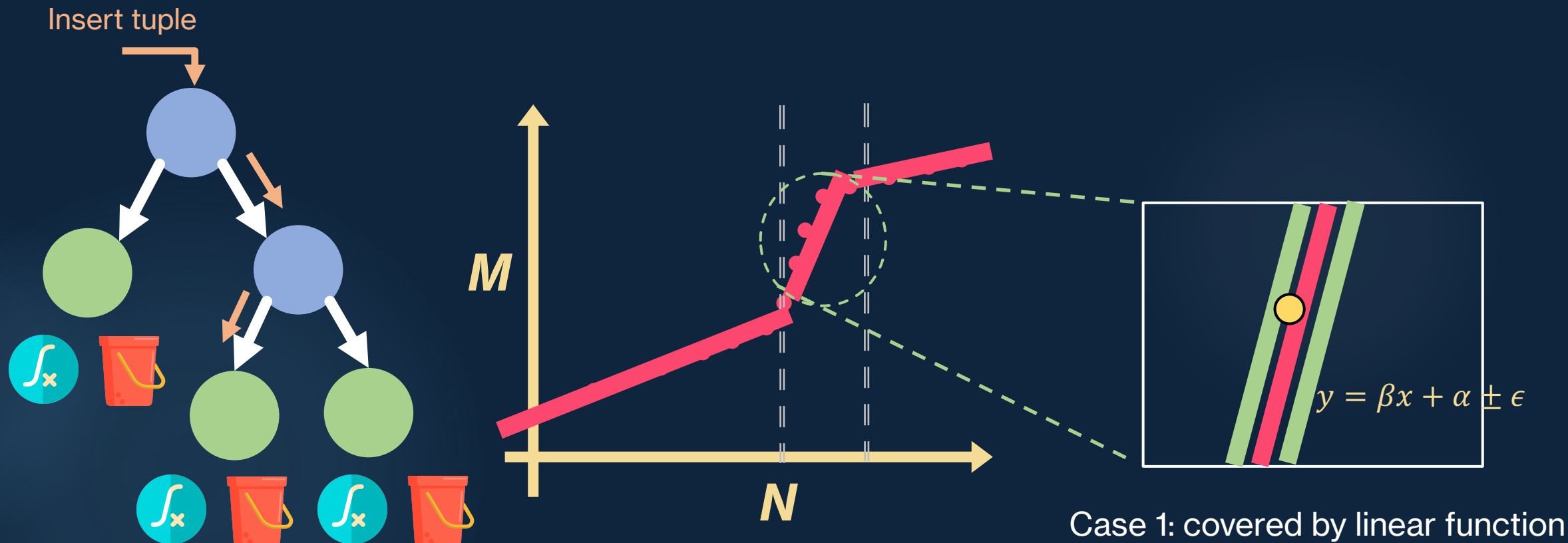
HERMIT: INSERT

55



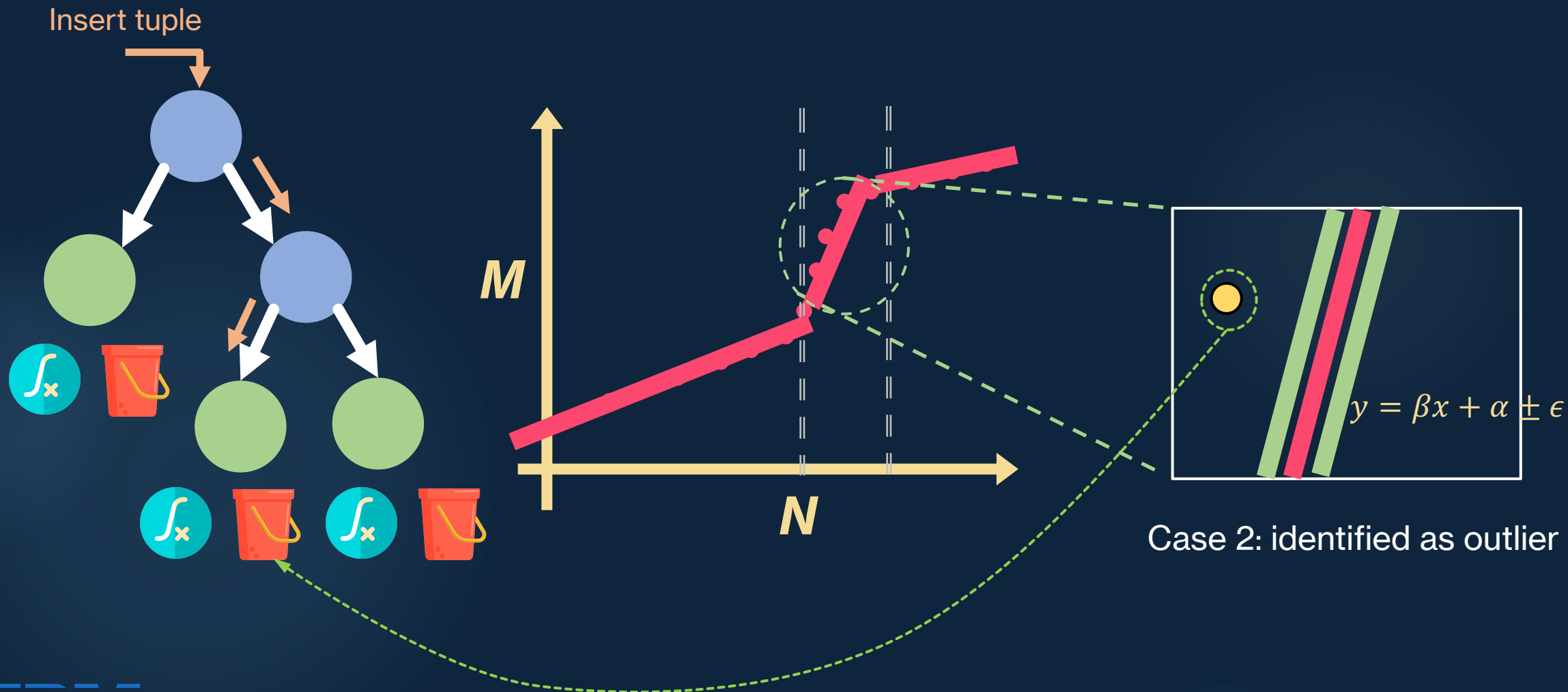
HERMIT: INSERT

56



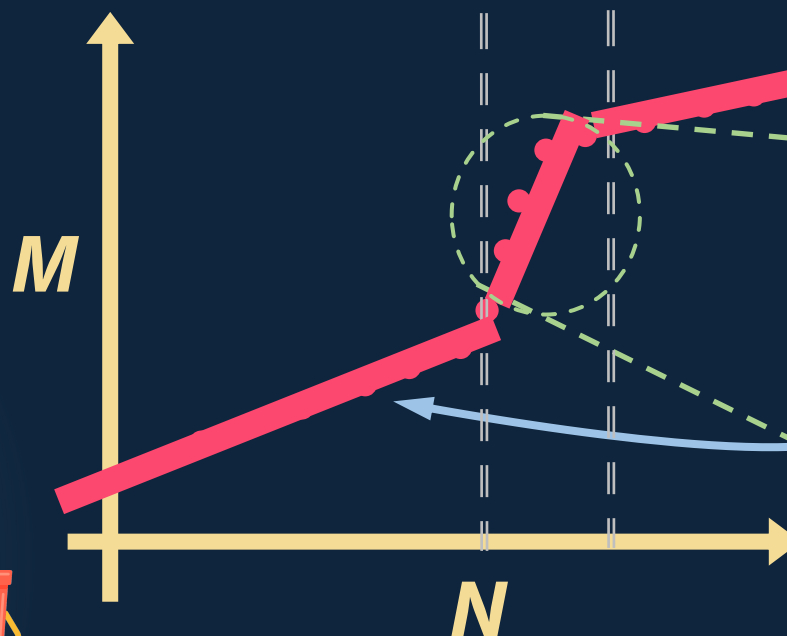
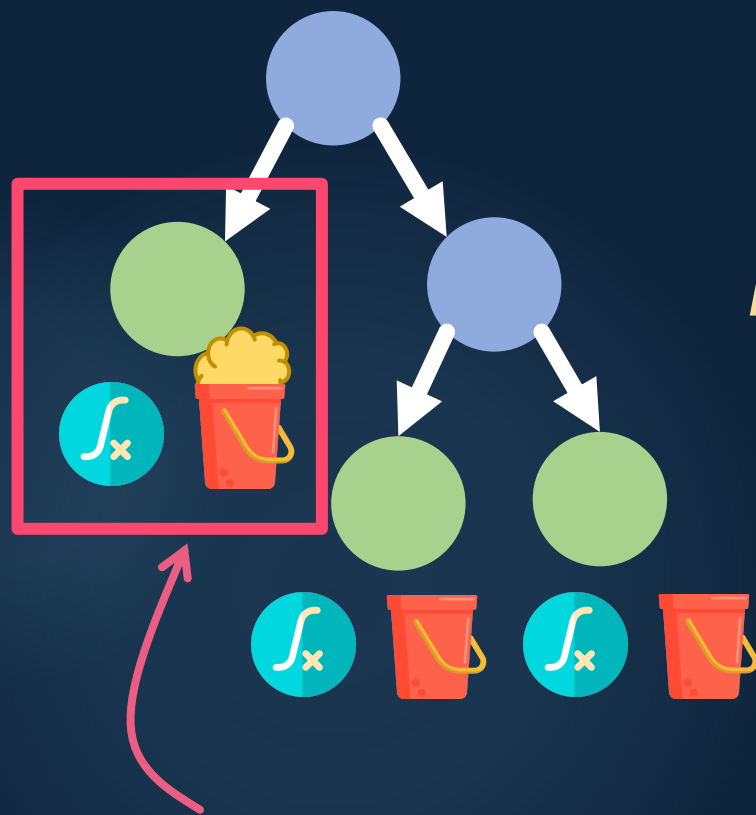
HERMIT: INSERT

57



HERMIT: MAINTENANCE

58



Background thread reorganizes internal nodes on demand



Once outlier bucket becomes full, then reorganization is required.



HERMIT: EVALUATION

59

- ▶ Benchmark
 - ▶ Stock: 100 stocks with highest and lowest prices
 - ▶ Sensor: 16 gas concentration sensors in different locations
 - ▶ Synthetic: Linear function, Sigmoid function
- ▶ Environment
 - ▶ In-memory performance
 - ▶ System: Main-memory database prototype
 - ▶ Baseline: B+-tree index



HERMIT: EVALUATION

60

- ▶ Benchmark
 - ▶ Stock: 100 stocks with highest and lowest prices
 - ▶ Sensor: 16 gas concentration sensors in different locations
 - ▶ ***Synthetic: Linear function, Sigmoid function***
- ▶ Environment
 - ▶ ***In-memory performance***
 - ▶ ***System: Main-memory database prototype***
 - ▶ ***Baseline: B+-tree index***



HERMIT: EVALUATION

61

Query: “select all the tuples with N’s value falling between x and y”

- ▶ Benchmark
 - ▶ Stock: 100 stocks with highest and lowest prices
 - ▶ Sensor: 16 gas concentration sensors in different locations
 - ▶ **Synthetic: Linear function, Sigmoid function**
 $M = \beta N + \alpha$ $M = \text{Sigmoid}(N)$
- ▶ Environment
 - ▶ In-memory performance
 - ▶ System: Main-memory database prototype
 - ▶ Baseline: B+-tree index



Key	M	N
1	100	115
2	200	204
3	150	165
4	400	410
...	...	...



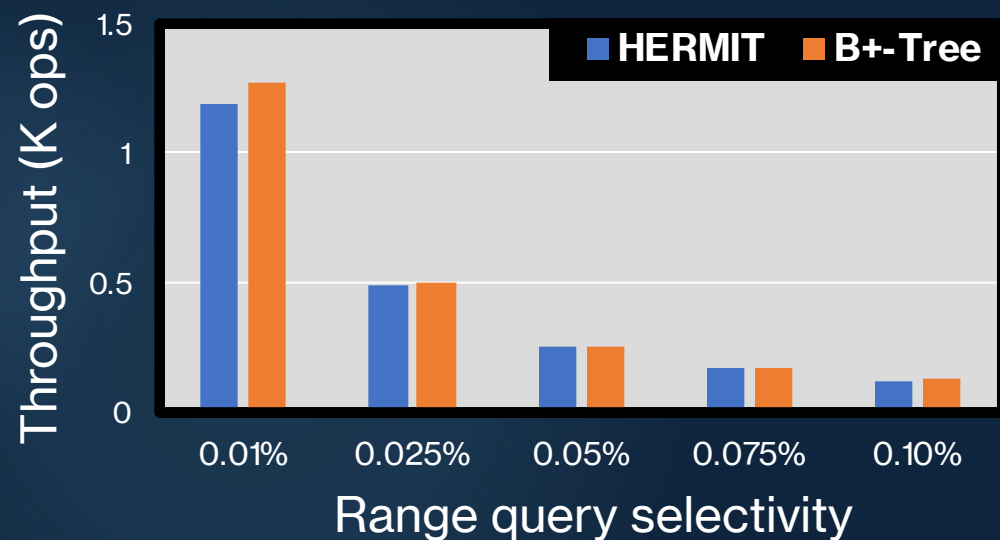
Hermit B+-Tree



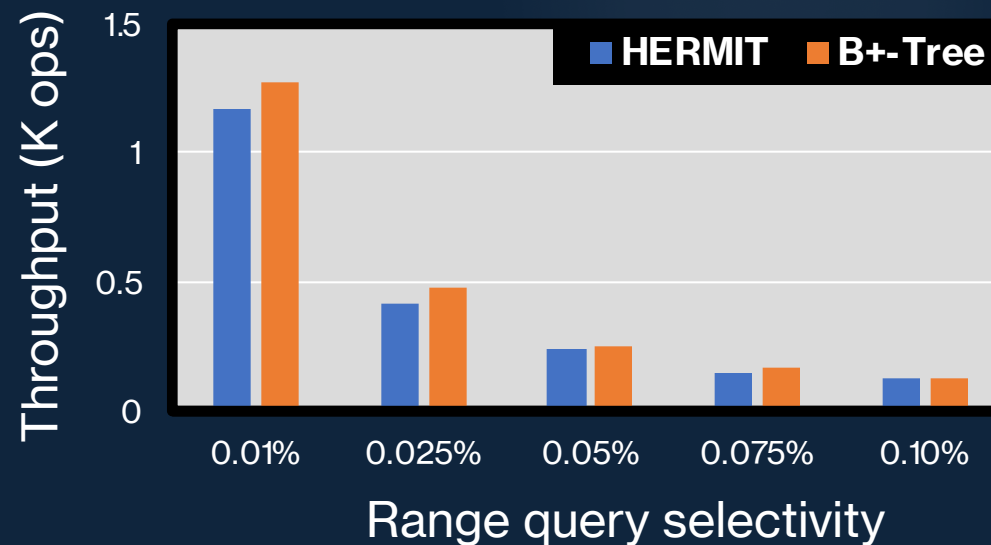
HERMIT: EVALUATION

62

- Synthetic benchmark (range lookup throughput)



Linear function



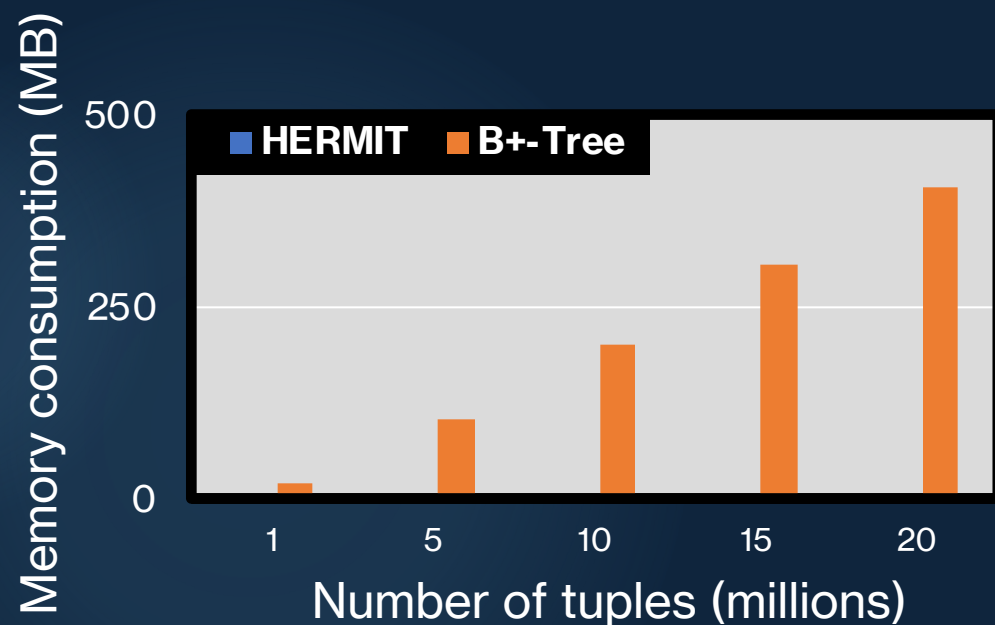
Sigmoid function



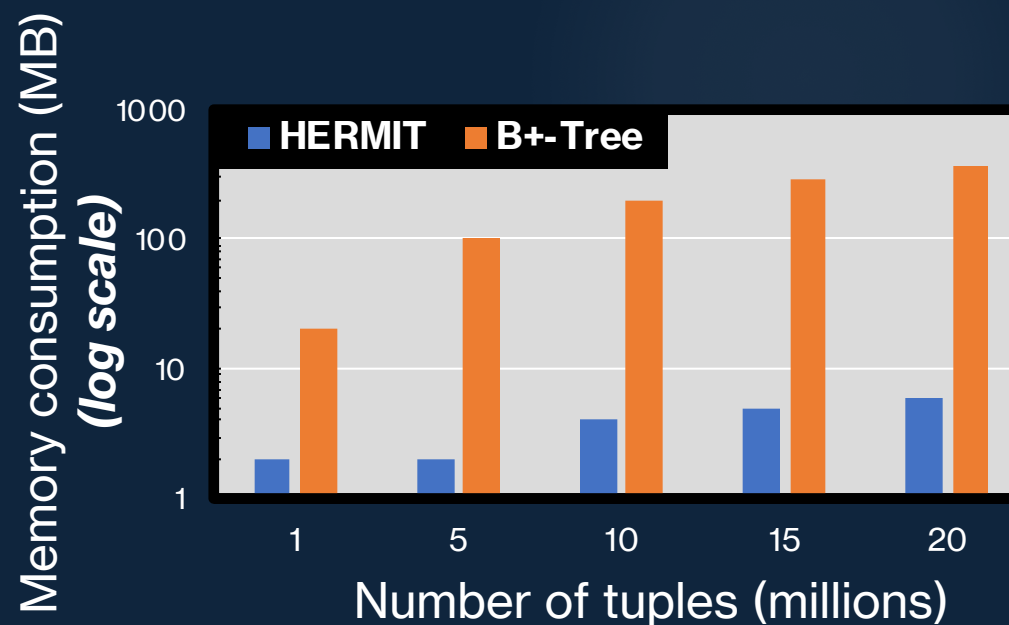
HERMIT: EVALUATION

63

- ▶ Synthetic benchmark (index memory consumption)



Linear function



Sigmoid function



HERMIT: CONCLUSION

64

- ▶ HERMIT is a succinct secondary indexing mechanism
 - ▶ Use ML-enhanced TRS tree to model correlation and capture outliers
 - ▶ Support all kinds of operations
 - ▶ Show good lookup / insertion performance and significantly reduce index size



HERMIT: EVALUATION

65

- ▶ Benchmark
 - ▶ Stock: 100 stocks with highest and lowest prices
 - ▶ Sensor: 16 gas concentration sensors in different locations
 - ▶ ***Synthetic: Linear function, Sigmoid function***
- ▶ Environment
 - ▶ ***In-memory performance***
 - ▶ ***System: Main-memory database prototype***
 - ▶ ***Baseline: B+-tree index***



HERMIT: EVALUATION

66

Query: “during which time periods do Stock X’s highest price fall between Y and Z?”

- ▶ Benchmark
 - ▶ **Stock: 100 stocks with highest and lowest prices**
 - ▶ Sensor: 16 gas concentration sensors in different locations
 - ▶ Synthetic: Linear function, Sigmoid function
- ▶ Environment
 - ▶ In-memory performance
 - ▶ System: Main-memory database prototype
 - ▶ Baseline: B+-tree index



Date	IBM (low)	IBM (high)	MSFT (low)	MSFT (high)	...	...
90/1/1	200	220	110	132		
90/1/2	180	190	120	180		
90/1/3	190	201	124	130		
90/1/4	195	207	116	125		
...	...	...	...	...		

HERMIT: EVALUATION

67

Query: “during which time periods do Stock X’s highest price fall between Y and Z?”

- ▶ Benchmark
 - ▶ **Stock: 100 stocks with highest and lowest prices**
 - ▶ Sensor: 16 gas concentration sensors in different locations
 - ▶ Synthetic: Linear function, Sigmoid function
- ▶ Environment
 - ▶ In-memory performance
 - ▶ System: Main-memory database prototype
 - ▶ Baseline: B+-tree index



Date	IBM (low)	IBM (high)	MSFT (low)	MSFT (high)	...	...
90/1/1	200	220	110	132		
90/1/2	180	190	120	180		
90/1/3	190	201	124	130		
90/1/4	195	207	116	125		
...	...	...	...	...		

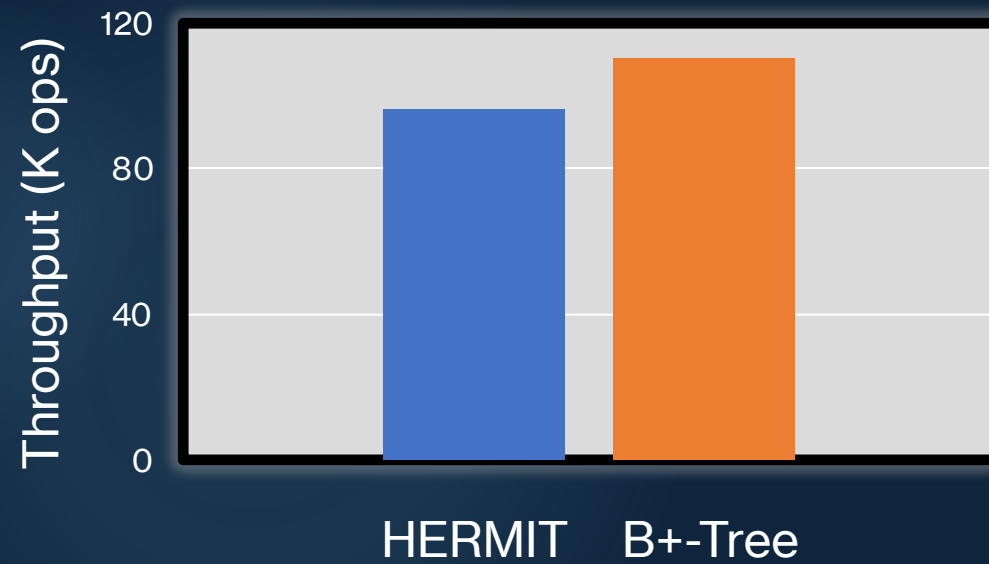
v.s. v.s.

Hermit B+-Tree

HERMIT: EVALUATION

68

► Stock benchmark



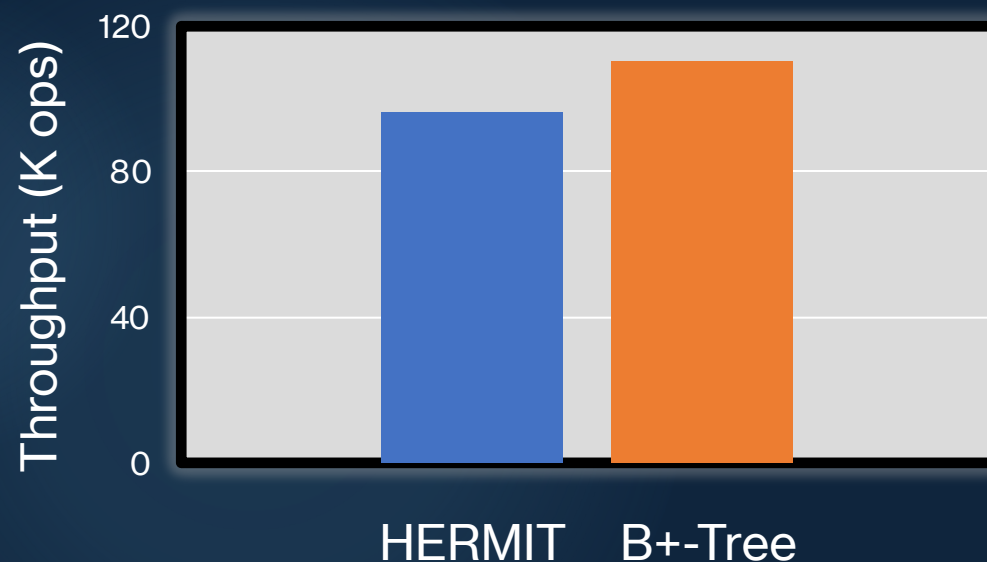
Range lookup throughput



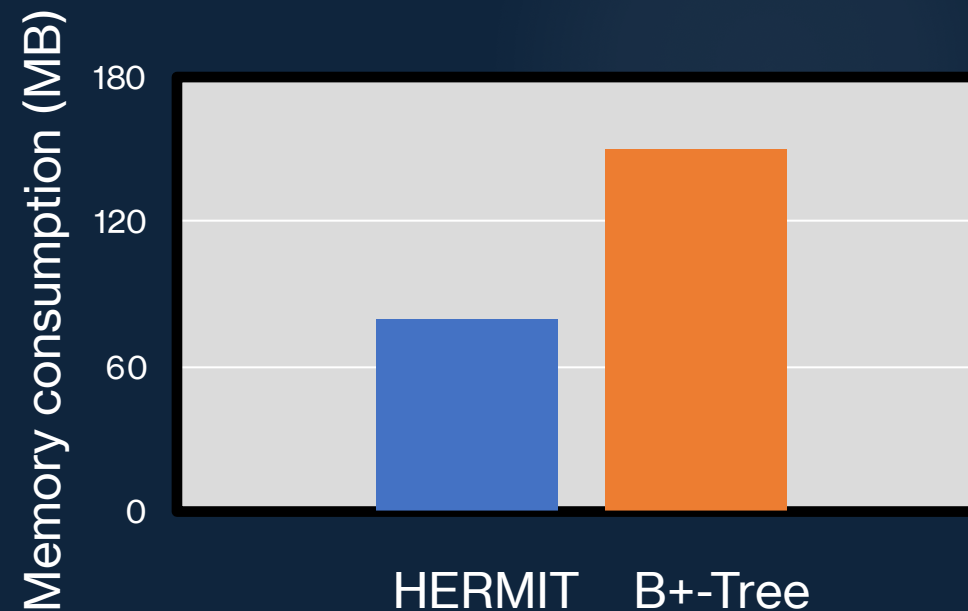
HERMIT: EVALUATION

69

► Stock benchmark



Range lookup throughput



Overall memory consumption



HERMIT: COMPLEX MODEL

70

Training time for different ML models (scikit-learn)

Number of tuples	1 K	10 K	100 K
Linear regression	0.42 ms	0.81 ms	3.2 ms
SVR (RBF)	0.09 s	4.5 s	> 120 s
SVR (linear)	0.28 s	29 s	> 120 s
SVR (polynomial)	0.29 s	24 s	> 120 s

